Institutional Repository - Research Portal
Dépôt Institutionnel - Portail de la Recherche
researchportal.unamur.be
University of Namur

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Élaboration d'un logiciel de gestion de Budget pour personnes handicapées mentales

Culot, Jean-Marc

Award date:
1992

Awarding institution:
Universite de Namur

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 02. May. 2026

Année académique 1991-1992

Elaboration d'un logiciel de gestion de Budget pour personnes handicapées mentales

Jean-Marc CULOT

Promoteur: **Madame Monique Noirhomme-Fraiture**

Co-promoteur: **Monsieur Michel Mercier,** professeur au Département de Psychologie de la faculté de Médecine, F.U.N.D.P.

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître
en Informatique.

Remerciements

Nous voudrions témoigner toute notre reconnaissance à Madame Monique Noirhomme-Fraîture qui a assuré la direction et le suivi de ce mémoire, ainsi que le soutien qu'elle n'a jamais manqué de nous apporter. Nous remercions également Monsieur Michel Mercier pour l'attention qu'il nous a prêtée. Nous désirons remercier tout particulièrement Madame Anne Faton pour sa collaboration active et pour les conseils judicieux qu'elle n'a cessé de nous prodiguer.

Un grand merci à Gilles Le Cardinal pour son accueil à Compiègne et pour le temps qu'il nous a consacré.

Nous remercions également Martine Galand pour sa collaboration et pour nous avoir consacré du temps.

Nous tenons également à remercier les éducateurs des institutions avec lesquelles nous nous sommes entretenus et plus particulièrement Amparo, Bernard, Joëlle, Martine et Pascal pour leur aimable collaboration.

Nous remercions François, Jean-François, Noël et Zyed pour leur gentillesse et le temps passé ensemble lors de notre stage à Compiègne.

Un grand merci enfin à toutes les personnes qui, d'une façon ou d'une autre, ont contribué à la réalisation de ce mémoire.

Abstract

Ce travail présente un projet de logiciel de gestion de budget pour personnes handicapées mentales. Le développement du programme suit une approche pluridisciplinaire composée par une collaboration entre psychologues, éducateurs et informaticiens.

This work presents a project of budget management software for mentally disabled persons. The program's development follows a multi-disciplinary approach composed of a collaboration between psychologists, educators and computer specialists.

TABLE DES MATIERES

INTRODUCTION 1

CHAPITRE I

Handicap mental & Connaissance

I. INTRODUCTION 4

II. AU NIVEAU DE L'ENFANT 5

1. **Les différents stades de développement présentés par Piaget** 5

1.1. Le stade sensori-moteur (de la naissance à 2 ans). 5

1.2. Le stade des opérations concrètes (de 2 ans à 11/12 ans) 6

1.2.1. ***Le niveau préopératoire (de 2 à 7/8 ans)*** 6

1.2.2. ***Le stade opératoire concret (de 7/8 ans à 11/12 ans)*** 6

1.3. Le stade des opérations formelles (à partir de 11/12 ans) 7

2. **La notion du Nombre** 7

3. **Conservation** 8

III. AU NIVEAU DE LA PERSONNE HANDICAPEE MENTALE 9

IV. CONCLUSION 10

CHAPITRE II

Informatique & Handicap mental

I. INTRODUCTION 11
II. PRESENTATION DES INSTITUTIONS 11
1. Le "Carat" 11
2. La "Clairière" 12
3. La "Marlagne" 12
4. La "Passerelle" 13
III. ANALYSE DE L'EXISTANT 13
1. Le "Carat" 13
2. La "Clairière" 15
3. La "Marlagne" 17
4. La "Passerelle" 17
IV. ANALYSE D'OPPORTUNITE 17
V. DIFFICULTES RENCONTREES PAR LES INSTITUTIONS EN CE QUI CONCERNE L'INFORMATISATION 19
VI. EVALUATION DE REALISATION DE PROGRAMMES POUR PERSONNES HANDICAPEES MENTALES 20
VII. CONCLUSION 21

CHAPITRE III

Elaboration du cahier des charges

I. INTRODUCTION 22
II. NAISSANCE DU PROJET 22
III. LES PERSONNES CONCERNEES 23
IV. OBJECTIFS DU LOGICIEL 23

V. CAHIER DES CHARGES INITIAL 24
VI. MODIFICATIONS APPORTEES AU CAHIER DES CHARGES 26
1. Un bilan 26
2. Un Test 26
3. Sortie sur imprimante 26
VII. LE PROGRAMME DE PARAMETRISATION ASSOCIE AU LOGICIEL 27
VIII. FONCTIONNALITES DU LOGICIEL BUDGET 28
IX. LES TRACES D'UTILISATION 33
X. CONCLUSION. 34

CHAPITRE IV

Interface du logiciel de Gestion de Budget

I. GENERALITES 35
1. L'homogénéité des écrans 35
2. Les commandes 37
3. La jauge 39
4. Les icônes 40
5. Les titres 41
6. Les confirmations et les messages 41
7. Les couleurs 42
8. Les icônes des mois 42
9. Les postes de dépenses 43
II. REPRESENTATION GRAPHIQUE 43
1. La représentation graphique constitue une aide 43
2. Le problème du pourcentage 44
3. Le problème des échelles 45

III. APPORTS 46
1. Un bilan 46
2. L'aide 46
3. Sortie sur imprimante 46
4. Visualisation des prévisions 47
IV. PROTOTYPAGE DES ECRANS 47
1. La saisie du prénom (1-2) 73
1.1. La saisie du prénom au moyen du clavier (1) 73
1.2. La saisie du prénom à l'aide de la souris (2) 73
2. La saisie des recettes principales (3-4-5) 74
2.1. Saisie des recettes principales à l'aide du clavier (3) 75
2.2. Saisie des recettes à l'aide de la calculette (4) 76
2.3. Saisie des recettes principales à l'aide des billets (5) 76
3. La prévision des dépenses (6) 77
3.1. Création d'une dépense 78
3.1.1. ***Saisie du poste (7)*** 78
3.1.2. ***Saisie du montant prévu de la dépense (8a-8b-8c)*** 78
a. *Saisie à l'aide du clavier (8a)* 79
b. *Saisie au moyen de la calculette (8b)* 79
3.1.3. ***Saisie de la périodicité de la dépense (9-10)*** 79
a. *Saisie Régulière - Occasionnelle (9)* 80
b. *Saisie de la fréquence (10) (dépense régulière)* 80
3.2. Le fait d'effectuer une dépense 81
3.3. L'option "jeter" 82
3.4. Visualisation des prévisions de dépenses (11) 82
4. Le menu principal (13) 83
5. Effectuer une recette (7 - 8 - 14) 83
6. Effectuer une dépense (7 - 8 - 14) 84
7. Les économies (28 et suivants) 84
7.1. Les achats à crédit (29) 85
7.2. Economies spécifiques (ES) (30) 85
7.3. Transferts (15 - 16) 86

8. La prévision des dépenses (6 et suivants) 87
9. La prévision des recettes 87
10. Test (24) 87
11. Le bilan (20-21-22-23) 88
12. L'état des finances (17-18-19) 89
13. L'aide (12) 90

CONCLUSIONS 91

BIBLIOGRAPHIE 93

★ ★ ★
★ ★

INTRODUCTION

Depuis une dizaine d'années, l'informatique n'a cessé d'étendre ses horizons: elle n'est plus réservée aux informaticiens; elle a pénétré la plupart des milieux professionnels et a même conquis nos foyers.

Cette ouverture a considérablement élargi le champ des utilisateurs, et ceci a entraîné de nouveaux domaines à explorer en informatique. En effet, s'il est toujours crucial de concevoir des programmes qui répondent aux spécifications et qui réalisent correctement leurs fonctionnalités, ils faut également qu'ils soient conviviaux et adaptés aux utilisateurs. C'est l'ordinateur qui doit le mieux s'adapter à l'homme et non l'inverse.

Bien sûr, il reste toujours une certaine adaptation à la machine, mais l'idée est de faire faire le plus gros effort à cette dernière. Cette adaptation est prise en charge par l'interface homme-machine.

Nous avons dit que le champ des utilisateurs s'est considérablement élargi dans les dernières années. Ainsi dans la vie professionnelle, toute personne, quel que soit son niveau de qualification, peut être appelée à utiliser l'informatique à son poste de travail. De plus, dans la vie privée, on trouve des utilisateurs de tous les âges.

En quelques années, nous nous sommes habitués à cette nouvelle situation, par contre, il n'est pas fréquent de destiner des produits informatiques à des utilisateurs handicapés. Pour ces utilisateurs, le problème de l'interface homme-machine est vital.

Dans le cas d'un handicap physique, les adaptations seront plus de type matériel, tandis que dans le cas d'un handicap mental les solutions seront de type logiciel. En tout cas, quel que soit le handicap de l'utilisateur,un gros effort devra être fourni au niveau de l'interface homme-machine.

Le problème de l'interface homme-machine nécessite déjà une approche pluridisciplinaire, dès lors avec des utilisateurs handicapés cette approche est une condition sine qua non a tout travail sérieux. En effet, les connaissances mises en oeuvre relèvent de l'art graphique et sonore, des techniques d'affichage de données, des techniques d'interaction, de la compréhension des capacités motrices, perceptuelles et cognitives de l'être humain face à l'ordinateur. Il est donc nécessaire de faire collaborer des personnes travaillant en informatique, en ergonomie et en psychologie. Et malgré tout, rien ne garantit un résultat correct au premier essai. C'est pourquoi le développement d'un projet est un processus itératif; chaque sous-produit réalisé est évalué afin d'en découvrir les lacunes éventuelles et d'en spécifier les modifications.

Lorsque les utilisateurs sont handicapés, le problème de l'interface homme-machine est encore plus complexe que pour les applications interactives classiques. Toutefois, il faut aussi spécifier les fonctionnalités de l'application. Dans une application interactive classique, les futurs utilisateurs peuvent participer à la définition de leurs besoins, tandis que dans le cas d'utilisateurs ayant un handicap mental, cette situation ne se produit pas; un utilisateur handicapé n'exprime pas directement ses besoins. Ceux-ci émanent des psychologues, des éducateurs, ... qui deviennent, alors, acteurs intermédiaires entre informaticiens et utilisateurs finals.

Le sujet de ce mémoire est le développement d'une application informatique pour personnes handicapées mentales: un logiciel de gestion de budget. Les phases de

spécification de l'interface et de spécification des fonctionnalités, très importantes, ont été longuement explicitées. Voici le plan que nous avons suivi au cours de notre travail:

Dans le **chapitre I**, nous faisons une approche théorique du niveau intellectuel du handicap mental; peu de travaux ont été entrepris dans ce domaine; par contre, quantité de travaux concerne les enfants. Nous tentons, dès lors, de faire le rapprochement entre personnes handicapées mentales et les enfants sur le plan intellectuel et nous faisons le lien avec le logiciel budget.

Le **chapitre II** présente les différentes institutions pour personnes handicapées mentales ayant collaboré à la mise au point du logiciel et analyse de l'existant: comment géraient-elles leur budget sans ordinateur. Les attentes et les inconvénients de l'informatique seront mentionnés.

Le **chapitre III** expose le contexte dans lequel le projet est né, en passant en revue les personnes concernées, les objectifs du logiciel, le cahier des charges, les différents outils, et les fonctionnalités du logiciel budget.

Le **chapitre IV** présente l'interface du logiciel budget. Tout au long de ce chapitre, des généralités concernant l'interface vont être proposées ainsi que des photos des écrans implémentés et des prototypes des écrans, non encore implémentés.

Enfin dans le **chapitre V**, nous tirons quelques conclusions sur l'ensemble de ce travail.

★ ★ ★
★ ★

CHAPITRE I

Handicap mental & Connaissance

I. INTRODUCTION

Depuis longtemps, on s'interroge sur ce qu'est l'intelligence et sur sa quantification. Dès le début de ce siècle, la notion de Quotient Intellectuel a été introduite. Elle permet de réduire une différence d'intelligence à une différence d'âge. Une personne handicapée mentale est dès lors assimilée à une personne dont l'âge chronologique est supérieur à l'âge mental[1].

Quel est l'âge mental des personnes handicapées mentales. Très peu de travaux ont été faits en ce qui concerne les personnes handicapées mentales. Par contre, une multitude de travaux s'intéressent au développement intellectuel de l'enfant.

Une des pistes pour comprendre le développement intellectuel des personnes handicapées mentales est de s'interroger sur le développement intellectuel de l'enfant.

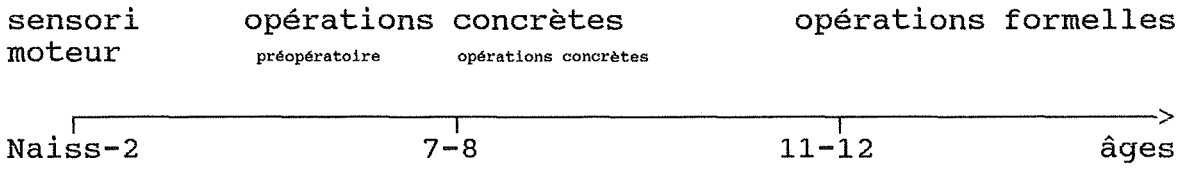
[1]L'âge mental est défini lors du test fait pour calculer le Quotient Intellectuel.

II. AU NIVEAU DE L'ENFANT

1. Les différents stades de développement présentés par Piaget

Jean Piaget a été un des premiers à publier quantité d'ouvrages sur le développement cognitif de l'enfant. Ses premières publications datent de cinquante ans, mais bon nombre de travaux ultérieurs s'en inspirent. Il définit des stades de développement communs à tous les individus. Il montre que les acquisitions qu'un individu peut avoir se font toujours dans le même ordre de succession. Chaque stade se caractérise par une structure d'ensemble.

Le développement cognitif de l'enfant est un processus continu. Chaque stade est plus complexe que le(s) précédent(s) et intègre celui(ceux)-ci:



1.1. Le stade sensori-moteur (de la naissance à 2 ans).

A ce stade, l'enfant commence par avoir ses premiers réflexes. Il recherche un moyen approprié pour atteindre un but différencié. Tout ce qu'acquiert l'enfant, a pour point de départ une action.

1.2. Le stade des opérations concrètes (de 2 ans à 11/12 ans)

Ce stade est subdivisé en 2 sous-stades:

1.2.1. *Le niveau préopératoire (de 2 à 7/8 ans)*

A ce niveau, l'enfant devient capable d'effectuer des opérations simples (Classification[2] et sériation[3], groupement[4], dénombrement, opérations sur des nombres peu complexes, ...). De plus, il désigne des objets et des situations par des systèmes symboliques et commence à avoir une représentation. A la fin de ce stade, l'enfant est en mesure d'acquérir la notion du nombre.

1.2.2. *Le stade opératoire concret (de 7/8 ans à 11/12 ans)*

L'enfant arrive à faire face à des opérations plus complexes. Il acquiert des notions telles que espace, temps, vitesse, ... et des notions de conservation (de substance, de poids, de volume). Une opération est ce qui transforme un état A en un état B en laissant au moins une

---

[2]La *classification* regroupe des objets selon leurs équivalences. L'enfant est capable de bien les arranger, les mettre en ordre, mettre ensemble ceux qui vont ensemble, ou mettre ensemble ceux qui se ressemblent le plus ...

[3]La *sériation* regroupe les objets selon leurs différences ordonnées. Par exemple, pour faire une sériation sur des réglettes de longueurs croissantes, une des technique consiste à faire un escalier en mettant à un bout la réglette la plus petite et en plaçant les autres de plus en plus grandes jusqu'à la plus grande de toutes, à l'autre bout.

[4]Un *groupement* est un système tel que 1) au moyen d'une opération donnée il engendre de proche en proche de nouveaux éléments du système: par exemple, réunir deux classes en une troisième qui les contienne; 2) cette opération peut être inversée; 3) le produit de l'opération et de son inverse est l'opération identique. Les classifications et les sériations sont les deux plus importants des groupements se constituant vers 7/8 ans.

propriété invariante au cours de la transformation. Or aux niveaux préopératoires, l'enfant conçoit la transformation comme modifiant toutes les données à la fois, sans aucune conservation, rendant impossible le retour au point de départ.

1.3. Le stade des opérations formelles (à partir de 11/12 ans)

L'enfant manipule, à ce stade, des opérations formelles (dénuées de contenu se référant à des objets physiques). Il commence à raisonner abstraitement et à déduire opératoirement à partir de simples hypothèses énoncées verbalement. *Si ... alors ... sinon.*

2. La notion du Nombre

Pour qu'un enfant puisse dénombrer les éléments d'un ensemble, il ne doit pas se soucier des propriétés particulières de chaque élément qui compose cet ensemble; l'enfant devrait en faire abstraction; ainsi ces éléments deviendraient équivalents.

Vers 7/8 ans, l'enfant est en mesure d'acquérir la notion du nombre: (I) < (I) + (I) < (I) + (I) + (I) ... que l'enfant interprète de la manière suivante: « la classe contenant un objet est incluse dans la classe qui contient un et un objet, celle-ci est incluse dans celle qui contient un et un et un objet, ... Le nombre apparaît alors, comme une synthèse opératoire des articulations par ordination ou inclusions successives. C'est en fait, une nouvelle forme de groupement dont l'opération est une opération d'inclusion et auquel une classification et une sériation ont été appliquées.

3. Conservation

Une fois qu'une connaissance est acquise, il faudrait pouvoir la conserver. Pour cela, il lui faudrait une certaine forme d'invariance afin qu'elle aboutisse à la formation de concepts, de propositions générales, d'une connaissance approfondie. Le jeune enfant construit, au cours du stade sensori-moteur, deux invariants cognitifs qu'il enrichira par la suite pour passer par un maximum d'objectivité vers 9/10 ans: il s'agit de la constance perceptive et la permanence des objets.

Les constances perceptives consistent à modifier ce qui est perçu par des éléments de la connaissance afin de conserver les caractéristiques propres de l'objet indépendamment des facteurs locaux ou temporels. Donc, le sujet, ayant développé des constances perceptives, tend à maintenir invariantes certaines propriétés perceptives malgré des conditions locales de perceptions. La permanence de l'objet ne concerne plus la perception d'un objet mais l'objet lui-même.

Au cours du stade des opérations concrètes, l'enfant construit des invariants concernant les Grandeurs Physiques:
La Longueur (6/8 ans), par exemple, on déforme un fil de fer et l'enfant devient capable de voir que la longueur du fil de fer est la même.
La Surface (7/8 ans), deux surfaces égales restent égales si on leur retire à chacune une partie égale mais disposée différemment.
Le Volume (10/12 ans), une même quantité de liquide est transvasée dans des récipients de formes variables.

Au niveau des opérations formelles, le principe de l'invariance s'applique aux propositions:

Si ... Alors sous-entend Si (toutes choses étant égales par ailleurs) ... Alors

III. AU NIVEAU DE LA PERSONNE HANDICAPEE MENTALE

Au cours de ce chapitre, nous allons tenter de faire le lien entre l'approche piagétienne et la déficience mentale. Comme nous l'avons déjà dit, très peu d'ouvrages ont été publiés dans ce domaine. Nous allons nous baser sur un article, écrit par Paour, qui reprend les conclusions de la plupart des rares travaux entrepris sur le développement de la déficience mentale, et sur un ouvrage de Barbël Inhelder, collaboratrice de Piaget et première à s'être penchée sur l'approche piagétienne de la déficience mentale.

Beaucoup de recherches ont confirmé une des assertions de Inhelder, à savoir qu'il y a un parallélisme du processus de construction et d'intégration des notions de conservation chez l'enfant "normal" et "anormal". En d'autres mots, les "débiles" et les "normaux" ne se distinguent que par leur vitesse de développement et le niveau terminal de la construction des opérations intellectuelles.

D'après Inhelder, « *l'âge de 11 ans constitue la limite chronologique à partir de laquelle un sujet ne peut plus accéder à l'opérativité concrète* ». Une personne handicapée mentale adulte n'est donc pas susceptible d'acquérir des notions de conservations.

Par la suite, des méthodes d'apprentissage des structures logiques ont été construites, afin de faire acquérir aux personnes handicapées mentales (enfants et adultes) des notions de conservation (au niveau opératoire concret).
Voici une de ces méthodes:

"Dans un premier temps, à l'aide d'une première application d'épreuves opératoires (pré-test), on retient des sujets qui ne possèdent pas la (ou les) notion(s) qui fait (font) l'objet de l'apprentissage. On répartit ensuite les sujets ainsi sélectionnés en deux

groupes. L'un de ces groupes bénéficie d'un apprentissage de notions opératoires concrètes (groupe expérimental), l'autre non (groupe témoin). Une deuxième présentation des épreuves (post-test) fait suite à la période d'apprentissage; elle doit permettre d'évaluer les effets sur la (les) notion(s) considérée(s)."

Les quelques études faites montrent qu'environ 60 à 70 % des sujets handicapés mentaux bénéficiant d'un apprentissage ont acquis des notions de conservation du stade préopératoire et environ 40% pour le stade opératoire concret.

Une des conclusions qu'on peut dégager de l'ensemble de ces travaux est qu'il est possible d'accélérer l'apparition de comportements opératoires concrets chez la personne handicapée mentale.

IV. CONCLUSION

Quels sont les rapports avec le logiciel budget (voir dans les chapitres suivants) ? Tout d'abord, les utilisateurs potentiels du logiciel sont des personnes handicapées mentales dont l'âge mental correspond à celui des enfants se situant dans le stade des opérations concrètes. Le logiciel tient compte des remarques formulées ci-dessus: rangement dans l'ordre croissant (sériation), quand l'argent n'est pas visible, cela ne veut pas dire qu'il n'est pas là (permanence de l'objet), représentation de l'argent, utilisation de chiffres (notion de nombre), déformation des représentations (constances perceptives), classification des dépenses, recettes, économies, ...

Pour pouvoir utiliser ce logiciel, l'âge mental se situant dans le stade des opérations concrètes constitue un minimum. Il est évident que des personnes ayant un âge mental supérieur à 7/8 ans auront plus de facilités à utiliser ce logiciel.

★ ★ ★
★ ★

CHAPITRE II

Informatique & Handicap mental

I. INTRODUCTION

Durant le semestre consacré exclusivement à l'élaboration du logiciel de gestion de budget, nous avons visité quatre institutions pour personnes handicapées mentales. Les objectifs de nos visites étaient de collecter des informations sur la façon dont les personnes gèrent leur argent sans ordinateur, de connaître les attentes des responsables de ces institutions vis-à-vis d'un programme d'ordinateur qui aiderait les pensionnaires à gérer leur argent et de savoir comment les personnes handicapées mentales réagissent face à un ordinateur.

II. PRESENTATION DES INSTITUTIONS

1. Le "Carat"

Le Carat se situe à Woluwé Saint-Lambert et a pour but de rendre effective la participation des personnes handicapées à l'ensemble de la société. Pour réaliser cela, il offre différents services dont les plus importants sont: un Service d'Accompagnement (le Carat propose un soutien individuel et sur mesure à des jeunes et

à des adultes mentalement déficients désireux d'accéder à une vie normale), un Service d'Information et de Documentation (grâce à ce service le Carat apporte une réponse aussi complète que possible à toute demande d'information ayant trait à la participation sociale et professionnelle des personnes handicapées) et un Service de Recherche (particulièrement spécialisé dans tous les domaines touchant à la participation des personnes handicapées). De plus, il contribue aussi à améliorer l'image même des personnes handicapées au sein de la société.

2. La "Clairière"

La Clairière accueille plus de 450 handicapés mentaux modérés et sévères, enfants, adolescents, adultes. Elle comprend une école gardienne, primaire, secondaire, un Institut Médico-Pédagogique avec semi-internat, un centre de Réadaptation Fonctionnelle, cinq maisons communautaires, un home et un Centre de Jour avec des ateliers de cuisine, couture, bois, modelage, tissage, vannerie, imprimerie, ... Ces ateliers approvisionnent la "Boutique" et des stands de vente occasionnels. La Clairière se situe, pour la majeure partie dans la région Bruxelloise, mais elle a d'autres bâtiments dispersés en Belgique. De plus, elle s'occupe d'Appartements Supervisés (des personnes handicapées mentales vivant dans des appartements).

3. La "Marlagne"

La Marlagne, à Namur, est une institution réservée à 10 femmes handicapées mentales, non travailleuses.

4. La "Passerelle"

La Passerelle est un home pouvant accueillir 11 personnes handicapées mentales adultes et travailleuses, localisé à Spa.

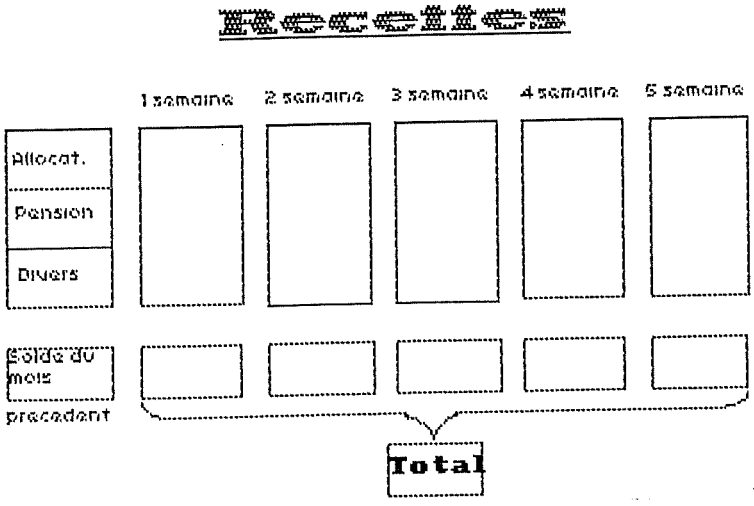
III. ANALYSE DE L'EXISTANT

Comment les personnes handicapées mentales gèrent-elles leur argent sans ordinateur ? Dans les différentes institutions, les personnes handicapées mentales gèrent leur budget avec l'aide des éducateurs.

1. Le "Carat"

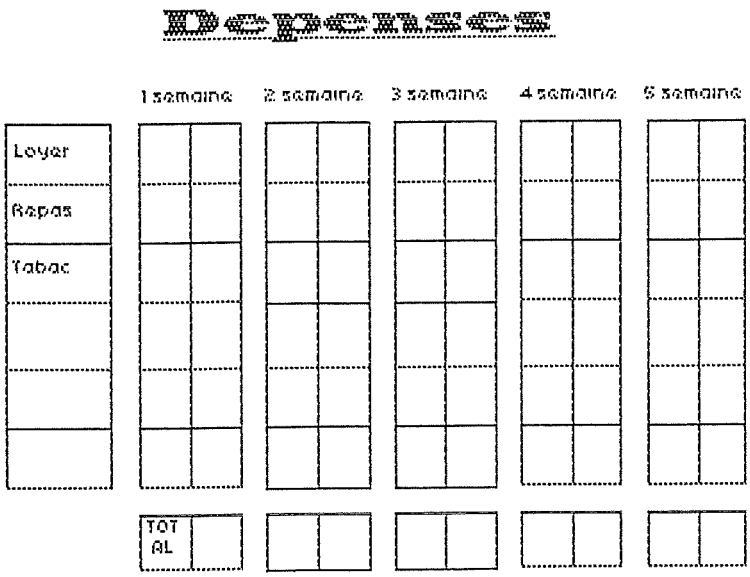
La période de gestion est d'un mois. Ce mois est subdivisé en semaines. Les personnes handicapées dites ***stagiaires*** font une mise au point hebdomadaire de leur budget au Carat.

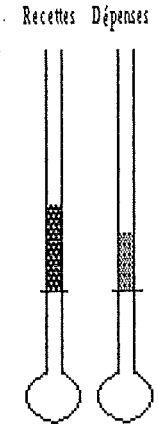
Les salariés gèrent des sommes de plus ou moins 30.000 FB par mois. Chaque stagiaire dispose d'une "feuille budget" divisée en quatre parties: les recettes - les dépenses - la répartition - les thermomètres.



Sur ce tableau figure la liste des différentes recettes spécifiques au stagiaire. Chaque semaine, le stagiaire y inscrit ses recettes. Les recettes prévues mais non acquises n'entrent donc pas en jeu.

Le tableau "dépenses" présente une liste des postes de dépenses spécifiques au stagiaire.
Chaque semaine, il y note les dépenses prévues. Les postes sont classés suivant les dépenses qui seront payées au comptant et celles payées par virement.

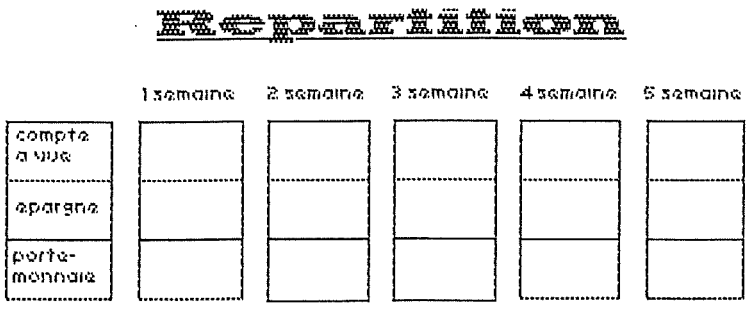




Il y a deux thermomètres, celui des recettes et celui des dépenses. Ils sont gradués. Après avoir noté une recette et les dépenses prévues, le stagiaire utilise les thermomètres pour visualiser la circulation de l'argent et la quantité encore disponible.

Un tableau de répartition permet de visualiser comment l'argent est répartit:

- ★ compte à vue
- ★ compte épargne
- ★ bons de caisse
- ★ argent liquide . . .

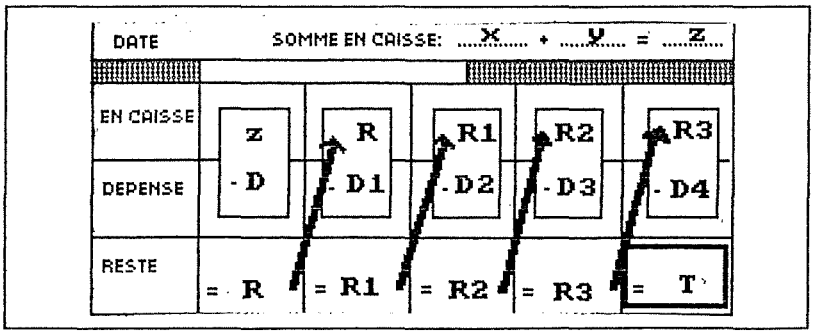


2. La "Clairière"

La gestion se fait différemment suivant que l'on se trouve dans le home ou dans les appartements supervisés. Dans le cas du home, les personnes handicapées mentales reçoivent de l'argent au début de chaque semaine qui leur servira d'argent de poche. Cette gestion n'est pas prévisionnelle; donc, il s'agit plus d'une comptabilité.

En ce qui concerne les Appartements Supervisés, la gestion est mensuelle et les personnes peuvent prévoir leurs dépenses. Certains types de dépenses sont prévus par semaine (par exemple, l'alimentation), d'autres, par mois par exemple, le loyer, les assurances ...).

a) Les prévisions par semaine: les personnes tiennent à jour une fiche. A chaque type de dépense correspond une fiche:

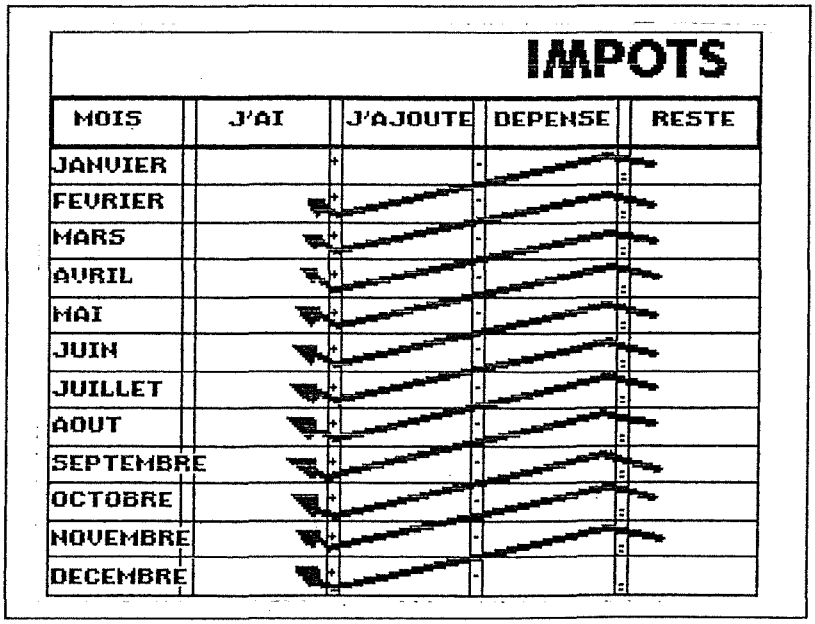


Dans le cartouche est indiqué: "Le solde de la semaine passée" (x), "La somme d'argent prévue pour cette dépense et pour la semaine"(y) et la somme des deux, c'est-à-dire ("la somme totale disponible pour effectuer cette dépense" (z)).

La fiche est subdivisée en cinq colonnes représentant les cinq premiers jours de la semaine (Le week-end, les personnes retournent, en général, chez elles). Chaque colonne est divisée en trois. La première ligne contient l'argent disponible, la deuxième la somme dépensée et la troisième, la différence entre les deux.

Au début de chaque semaine, une certaine somme (y) est retirée du compte à vue et est répartie dans des enveloppes. En fin de semaine, la personne dresse un bilan avec l'éducateur (T).

b) Les prévisions par mois: les personnes tiennent à jour, un carnet de dépôts. Il y a autant de carnets de dépôts qu'il y a de types de dépenses. Chaque carnet couvre une année complète et est subdivisé en 12 mois:



Le fonctionnement d'un carnet de dépôt est le suivant: chaque mois, au solde du mois passé, on ajoute ce qu'on prévoit et on retire ce qu'on a dépensé. A la fin du mois, la personne dresse un bilan en compagnie de l'éducateur.

3. La "Marlagne"

La majorité des pensionnaires de la Marlagne gère leur argent hebdomadairement. Pour elles, cette gestion est une comptabilité; il ne s'agit pas de prévoir ses dépenses: au début de chaque semaine, elles reçoivent une certaine somme; celle-ci sera leur argent de poche. En cours de semaine, elles notent dans un cahier leurs recettes et leurs dépenses. Et en fin de semaine, elles font le bilan avec l'aide de l'éducateur.

Quelques-unes gèrent leur argent mensuellement. Dans ce cas, elles ont la possibilité de prévoir leurs dépenses. Au début du mois, elles reçoivent une certaine somme d'argent. Elles répartissent cette somme en différentes sommes selon leurs prévisions. Elles mettent ces sommes dans des enveloppes (une enveloppe par type de dépenses). Elles notent leurs recettes et leurs dépenses. Lorsqu'une semaine est écoulée, elles en font le bilan avec l'éducateur et à la fin du mois, elles font un bilan général. De plus, à la fin de chaque mois, elles ont la possibilité soit d'économiser le surplus, soit de le reporter au mois suivant.

4. La "Passerelle"

Certains pensionnaires gèrent leurs comptes hebdomadairement. D'autres, font leur budget mensuellement.

IV. ANALYSE D'OPPORTUNITE

Dans le cas d'une gestion budgétaire, l'élaboration d'un programme d'ordinateur pouvant aider les personnes handicapées mentales à gérer leur argent d'une façon plus ou moins autonome, séduit les éducateurs. Leurs attentes quant à un tel programme

convergent vers des mêmes idées :

- Gérer son argent n'est pas marrant tous les jours! Le fait que cela se fasse à l'aide d'un ordinateur peut les motiver. Cela peut aussi servir à démystifier la croyance selon laquelle l'ordinateur ne serait pas à la portée d'une personne handicapée mentale. De ce fait, elles auraient un autre statut et seraient considérées comme plus importantes.

- Une attente serait de conserver des traces par écrit de toute utilisation du logiciel, en vue par exemple, de les consulter dans le futur pour pouvoir mieux répartir l'argent. Cela sous-entend l'emploi d'une imprimante.

- Nous pensons qu'il est important de simuler une dépense, de "faire comme si" on effectuait réellement une dépense. La personne pourrait voir les impacts et, éventuellement, y remédier.

- En plus d'une aide à la gestion, l'ordinateur devrait aussi apprendre aux personnes handicapées mentales à se familiariser avec des termes ou concepts tels que recette, dépense, économie, état des finances, bilan, prévision,...

De manière plus générale, l'ordinateur présente a priori maints avantages pour les personnes handicapées mentales:

- La possibilité d'envoyer de nombreux stimuli audio-visuels. Comme l'interface est particulièrement importante pour des applications informatiques avec des personnes handicapées, l'ordinateur est très sollicité pour cet avantage. Le logiciel Budget exploite grandement les possibilités graphiques et sonores de l'ordinateur sur lequel il tourne.

- La possibilité d'offrir des interfaces adaptées à de nombreux cas particuliers; une interface matérielle (hardware) sera, généralement employée comme solution à un handicap physique et une interface de type logiciel (software) pour personnes handicapées mentales (écrans du logiciel budget).
- La possibilité d'enregistrer des résultats pour de plus amples analyses;
- La possibilité de réagir autrement qu'une personne, c'est-à-dire avec:
  - une patience illimitée
  - une mémoire illimitée
  - une absence de jugement

- La possibilité de suivre les progrès pédagogiques et technologiques (matériels et logiciels)
- Certains logiciels éducatifs arrivent à mettre à jour bien des capacités jusque là restées dans l'ombre ou totalement méconnues.
- L'informatique favorise la créativité de certaines personnes handicapées mentales. Bien que la technique offre des possibilités de plus en plus sophistiquées, elle ne sera jamais qu'un outil; la valeur artistique d'une production sera toujours liée à la sensibilité de celui qui la réalise et de celui qui la regarde.

V. DIFFICULTES RENCONTREES PAR LES INSTITUTIONS EN CE QUI CONCERNE L'INFORMATISATION

Deux difficultés majeures sont rencontrées par la plupart des institutions. A celles-là, viennent s'ajouter quantité d'autres problèmes propres à chaque institution.

La difficulté principale est le manque d'argent de la part de l'institution pour acquérir un ordinateur. De plus, pour une institution diposant d'argent, pour obtenir les crédits nécessaires à l'acquisition d'un ordinateur, il faudrait que l'on puisse montrer les bienfaits qui lui sont dispensés. Un logiciel adéquat pour personnes handicapées

mentales est plus difficile à trouver qu'un logiciel destiné à tout autre personne.

Un autre problème important est le suivant: une grande majorité des éducateurs n'a jamais été en contact direct avec un ordinateur. Il faudrait donc les y sensibiliser tout en étant à l'écoute des réticences.

A côté de ces problèmes généraux, il existe une quantité de problèmes locaux. Par exemple, un problème de place des ordinateurs, bouleversement des horaires, ...

VI. EVALUATION DE REALISATION DE PROGRAMMES POUR PERSONNES HANDICAPEES MENTALES

Durant le mois d'octobre 1990, nous avons fait un court stage dans une institution pour personnes handicapées mentales: Le Bercail, à Liège. Notre objectif était de faire connaissance avec la population handicapée mentale et de voir comment elle réagissait face à un ordinateur. Pour ce faire, nous avions à notre disposition deux logiciels pour personnes handicapées mentales réalisés dans le cadre de mémoires: le logiciel Compte et le logiciel Connaissance du corps.

Il est à noter que l'institution avait, en prêt, un ordinateur Amiga et les deux logiciels cités ci-dessus. De ce fait, certaines personnes étaient, plus ou moins, familiarisées avec ces logiciels. Nous avons constaté un certain intérêt à utiliser l'ordinateur de la part de quelques personnes handicapées mentales, ainsi que de la fierté.

Le logiciel Compte est le premier sous-système utile du logiciel de gestion de budget. Il consiste à rentrer les recettes et les dépenses ayant été effectuées, de les comparer avec son argent de poche et de voir si on est ou non en équilibre.

Quant au logiciel de Connaissance du Corps, il s'agit d'un logiciel éducatif destiné aux personnes handicapées mentales. Il comprend deux types d'exercices: les puzzles et les exercices de désignation. Le puzzle consiste à recréer un personnage à partir de ses membres. L'exercice de désignation a pour objet de désigner un membre, qui a été demandé par l'ordinateur, sur un personnage. Un niveau de difficultés est associé à chaque exercice.

VII. CONCLUSION

Beaucoup d'institutions ont déjà mis au point un système élaboré de gestion de l'argent des personnes handicapées mentales. L'informatique permettrait d'améliorer ce système. L'aspect attrayant de l'informatique n'est pas à négliger et semble être important.

★ ★ ★
★ ★

CHAPITRE III

Elaboration du cahier des charges

I. INTRODUCTION

Le développement d'un logiciel nécessite un cahier des charges. Ce cahier des charges est l'énoncé du logiciel. Il correspond aux attentes des demandeurs. Son but est de spécifier ce que l'on veut que le logiciel fasse (les fonctionnalités) et l'interface.

Un cahier des charges a initialement été élaboré, puis a été affiné par après. Dans ce chapitre, les cahiers des charges initial et final seront présentés. L'interface sera traitée dans le chapitre IV.

II. NAISSANCE DU PROJET

Le Centre PSINHA (PSychologie - INformatique - HAndicap), un service de recherche, de documentation et d'information qui traite de problèmes liés à l'informatique et au handicap, et qui dépend du Département de Psychologie de la Faculté de Médecine de Namur.

Des groupes de travail y sont organisés à la demande des institutions. C'est ainsi qu'un groupe s'est constitué pour développer un logiciel de gestion de budget. L'objectif premier de ce logiciel était d'aider des personnes handicapées mentales à gérer leur budget hebdomadaire ou mensuel de façon plus ou moins autonome.

Une première version de ce logiciel a été réalisée sur PC en Turbo Pascal. Cette version n'a pas été jugée satisfaisante, et dès lors, la décision a été prise de le développer sur Commodore Amiga. La raison principale de cette décision était que cette machine offrait de nombreuses capacités sonores et graphiques fort intéressantes pour une application interactive avec des personnes handicapées mentales. De plus, son prix était et est encore très abordable pour les institutions susceptibles de l'acquérir.

III. LES PERSONNES CONCERNEES

Le logiciel de gestion de budget s'adresse à des personnes adultes handicapées mentales, ayant un handicap léger ou modéré, travailleuses et non travailleuses et vivant seules ou en institution. Certaines d'entre elles ne savent pas lire, n'ont aucune notion de la valeur relative des nombres et n'ont pas la notion du temps.

IV. OBJECTIFS DU LOGICIEL

L'objectif principal du logiciel Budget est d'aider la personne handicapée mentale à gérer son budget de la façon la plus autonome possible. Par gestion, nous entendons la prévision et la tenue des comptes. Un autre objectif serait de la familiariser avec des notions telles que recettes, dépenses, économies, ... Nous pensons qu'il serait intéressant de conserver des traces par écrit de toute utilisation du logiciel, en vue par exemple, de les consulter dans le futur. Et enfin, le logiciel sera capable de faire des simulations de dépenses. La période à gérer peut s'étendre sur une semaine ou sur un mois.

V. CAHIER DES CHARGES INITIAL

Détail des tâches à pouvoir exécuter:

La période de gestion est subdivisée en 3 étapes:

1° En début de gestion

La personne handicapée mentale doit avoir la possibilité d'entrer ses coordonnées, c'est-à-dire son nom (chaque utilisateur possède sa propre disquette), prendre connaissance du solde (liquide et/ou compte à vue, d'après les paramètres), se situer dans le temps (date...), entrer la "recette principale" (son montant, son origine, son moment) et la visualiser, prévoir ses dépenses ("régulières" et "occasionnelles") et les visualiser (les dépenses régulières spécifiques à la personne handicapée doivent être mémorisées), prendre connaissance des "économies" (réparties en différents postes) et les visualiser et enfin avoir accès à un "état des finances".

2° En cours de gestion

Elle doit pouvoir se situer dans le temps (date...), entrer une dépense (son montant, son origine, son moment) et la visualiser; cette dépense devant être située parmi les dépenses prévues (régulières ou occasionnelles) ou non prévues, annuler une prévision de dépense, entrer une recette (son montant, son origine, le moment où elle a été faite) et la visualiser, accéder aux différents postes des "économies". Là, elle peut soit noter une dépense ou une recette dans "économies générales", soit créer, alimenter, un "achat à crédit" ou soit créer, alimenter, supprimer une "économie spécifique". Elle pourra aussi tester ses dépenses et ses recettes et avoir accès à un état des finances.

3° En fin de gestion

La personne utilisatrice pourra se situer dans le temps (date), elle aura aussi la possibilité de choisir ce qu'elle va faire de l'argent qui lui reste de sa recette principale par choix de poste (par exemple, elle peut le transférer à la période de gestion suivante, dépenser immédiatement le surplus ou le transférer aux "économies") et enfin elle peut avoir accès à un "état des finances".

Lexique:

Recette principale: Somme d'argent que l'utilisateur reçoit et qui sert à gérer son budget pendant la période de gestion.

Dépense régulière: Dépense qui se fait régulièrement, toutes les X semaines; suivant une fréquence fixée.

Dépense occasionnelle: Dépense qui se fait exceptionnellement.

Dépense prévue: Dépense que l'utilisateur a l'intention de faire ou doit faire mais qu'il n'a pas encore faite. Les dépenses régulières sont toujours des dépenses prévues.

Economie spécifique: Economie destinée à un achat précis. L'utilisateur peut annuler cette économie, s'il le veut et ainsi, récupérer son argent.

Achat à crédit: Contrairement aux économies spécifiques, ce compte ne peut pas être annulé.

Economie générale: L'économie générale comprend les éventuels comptes à vue ou d'épargne.

Somme à gérer: Somme totale d'argent disponible pour la gestion du budget. Cette somme à gérer comprend :

- la recette principale
- les économies
- économies spécifiques
- achats à crédit
- économies générales

Etat des finances: Visualisation de la somme à gérer.

VI. MODIFICATIONS APPORTEES AU CAHIER DES CHARGES

Lors de notre stage à Compiègne[5], et suite aux entrevues que nous avons eues avec les éducateurs, le cahier des charges s'est quelque peu complété. Voici ces apports:

1. Un bilan

A la fin de chaque période de gestion, qu'il s'agisse d'un mois ou d'une semaine, nous avons trouvé qu'il serait intéressant que l'ordinateur fasse un résumé des dépenses effectuées durant cette période. Il serait aussi intéressant de faire un bilan sur une année entière.

On aurait trois sortes de bilan:

- bilan d'une semaine
- bilan d'un mois
- bilan d'une année.

2. Un Test

Il serait très intéressant de pouvoir simuler une action (recette, dépense, économie...). Donc, avant de faire quelque chose qui peut être irréversible, la personne peut en mesurer les impacts.

3. Sortie sur imprimante

Pour certaines institutions, il serait souhaitable d'avoir une trace écrite. Une

[5] Stage réalisé à l'Université de Technologie sous la direction de Gilles Le Cardinal.

des solutions serait de pouvoir faire une impression de tous les écrans, éventuellement, d'après des paramètres.

VII. LE PROGRAMME DE PARAMETRISATION ASSOCIE AU LOGICIEL

La population des personnes handicapées mentales est fort hétérogène et le logiciel veut s'adresser à une large population. Pour réaliser cet objectif, le logiciel est conçu de manière à proposer parfois des éléments de dialogue différents pour la même fonctionnalité. Un programme de paramétrisation associé au logiciel Budget permet, pour chaque individu, de sélectionner les éléments de dialogue adéquats et de n'utiliser que des fonctionnalités qui conviennent à la personne. Ce programme doit donc être utilisé par l'éducateur avant l'expérimentation pour paramétriser le logiciel selon l'individu.

Voici ces paramètres:

- ★ Utilise-t-on une imprimante ? Si on désire ne pas l'employer, l'icône-print (voir au chapitre 4) sera absente de tous les écrans.

- ★ Doit-on entendre les titres des écrans ? Si on répond négativement à cette question, les titres des écrans ne seront pas énoncés oralement et l'icône-oreille (voir au chapitre 4) sera absente de tous les écrans.

- ★ La saisie du nom se fait-elle au clavier ou à l'aide de la souris ? (voir le chapitre 4 pour ces deux types de saisie).

- ★ La saisie des recettes principales se fait-elle à l'aide du clavier, de la calculette ou des billets ? (voir le chapitre 4 pour ces trois types de saisie).

★ La saisie d'un montant d'argent se fait-elle à l'aide du clavier, de la calculette ou des billets ? (voir le chapitre 4 pour ces trois types de saisie).

★ Faut-il indiquer une ligne du temps ?

★ Quelle est la période de gestion, une semaine ou un mois ?

★ Si la période de gestion est la semaine, quel est le jour de début de gestion ?

VIII. FONCTIONNALITES DU LOGICIEL BUDGET

Voici la liste des fonctionnalités offertes par le logiciel de gestion de budget:

1° Saisie du prénom: Suivant un paramètre prédéfini par l'éducateur, le prénom peut être saisi soit au clavier, soit au moyen de la souris.
- *Saisie au clavier*: Le prénom entré doit être identique à celui de la disquette de données.
- *Saisie à l'aide de la souris*: L'utilisateur entre son nom lettre par lettre en cliquant sur le clavier dessiné à l'écran.
- *Effacer une lettre*: Une fois que le prénom est écrit à l'aide du clavier, il est possible de le corriger en effaçant la dernière lettre.

2° Saisie des recettes principales: Le choix du type de saisie a dû être prédéfini par l'éducateur.
- *Entrer une recette*: Un montant doit être communiqué soit en utilisant le clavier, soit en utilisant avec la souris, une calculette simulée, soit en indiquant le nombre de billets et/ou pièces de chaque type qu'on a reçu.
- *Additionner les recettes*: Chaque recette entrée doit de plus s'additionner au montant total précedent des recettes et du solde éventuel.
- *Visualisation du solde et des recettes*: Une fois entrée, chaque recette figure en vert dans la jauge et s'inscrit à l'écran.

- *Effacer le dernier chiffre entré*: Dans le cas d'une entrée au moyen du clavier ou de la souris, il est possible de le corriger en effaçant la dernière lettre.
- *Effacer le dernier billet entré*: Dans le cas d'une entrée au moyen des billets et/ou pièces, il est possible de le corriger en effaçant la dernière lettre.
- *Situation dans le temps*: Suivant un paramètre prédéfini, un calendrier du mois ou de la semaine est présent à l'écran.

3° Prévisions de dépenses

- *Situation dans le temps*: Suivant un paramètre prédéfini, un calendrier du mois ou de la semaine est présent à l'écran.
- *Visualisation des prévisions de dépenses*: Les différentes prévisions apparaissent dans l'ordre croissant de leurs montants et sous forme de jauges.
- *Ajout d'une nouvelle prévision de dépenses*: Une prévision de dépenses peut être ajoutée aux autres. Pour ce faire, une série d'actions sont à faire:
  - **Choix du poste de la prévision de dépense**: 40 postes de dépenses sont affichés à l'écran et l'utilisateur doit en choisir un.
  - **Saisie du montant de la prévision**: Suivant un paramètre prédéfini, cette saisie s'effectue soit au clavier, soit a l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran.
  - **Saisie de la périodicité**: Une prévision de dépenses peut être soit régulière soit occasionnelle. L'utilisateur sélectionne l'une ou l'autre.
  - **Saisie de la fréquence**: Si la prévision est régulière, elle peut se faire: toutes les semaines, tous les 1,2,3,6 mois ou tous les ans.
  - **Saisie du mois de départ**: Si la prévision est régulière, à partir de quand sera-t-elle faite ? (par défaut, c'est le mois courant)
- *Annuler une prévision de dépenses*: On peut décider de ne plus faire une dépense, soit une fois, soit pour toujours, si elle est régulière. Dans ce cas, il faut l'éliminer des fichiers des prévisions de dépenses. Le montant d'argent disponible augmente et la jauge s'ajuste.
- *Effectuer une dépense prévue*: Une dépense prévue peut devenir réelle. Le montant d'argent disponible et la jauge s'ajustent.

4° Menu principal

- *Saisie de l'activité*: 8 activités sont proposées: Recettes - Dépenses - Economies - Prévisions de dépenses - Prévisions de recettes - Test - Bilan - Etat des finances. L'utilisateur en sélectionne une parmi: Recettes - Dépenses -

Prévisions des dépenses - Test.

- *Saisie de la fin du programme*: L'utilisateur peut quitter le programme. Les traces sont automatiquement sauvegardées sur la disquette de données.

5° Recettes: Il s'agit d'entrer une recette autre que les recettes principales.

- *Choix du poste de la recette*: 40 postes de recettes sont affichés à l'écran et l'utilisateur doit en choisir un.
- *Entrer la recette*: Suivant un paramètre prédéfini, cette saisie s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en vert dans la jauge.
- *Additionner la recette*: Le montant de la recette est automatiquement additionné au montant d'argent liquide.
- *Visualisation du liquide et de la recette*: Les montants du liquide et de la recette sont affichés dans la jauge.
- *Effacer la dernière entrée*: Afin de corriger la recette, il est possible d'effacer le dernier chiffre (dans le cas d'une saisie au clavier ou à l'aide de la souris) et le dernier billet entré (dans le cas d'une saisie à l'aide des billets/pièces).
- *Saisie du moment de la recette*: Les calendriers du mois passé et du mois actuel sont affichés à l'écran. Le jour actuel est souligné et l'utilisateur doit sélectionner un jour déjà écoulé ou le jour actuel, qui représentera le moment de la recette.

6° Dépenses: Il s'agit d'entrer une dépense qui n'a pas été prévue.

- *Choix du poste de la dépense*: 40 postes de dépenses sont affichés à l'écran et l'utilisateur doit en choisir un.
- *Entrer la dépense*: Suivant un paramètre prédéfini, cette saisie s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en rouge dans la jauge.
- *Soustraire la dépense*: Le montant de la dépense est automatiquement déduit du montant d'argent liquide.
- *Visualisation du liquide et de la dépense*: Les montants du liquide et de la recette sont affichés dans la jauge.
- *Effacer la dernière entrée*: Afin de corriger la dépense, il est possible d'effacer le dernier chiffre (dans le cas d'une saisie au clavier ou à l'aide de la souris) et le dernier billet entré (dans le cas d'une saisie à l'aide des billets/pièces).
- *Saisie du moment de la dépense*: Les calendriers du mois passé et du mois actuel

sont affichés à l'écran. Le jour actuel est souligné et l'utilisateur doit sélectionner un jour déjà écoulé ou le jour actuel, qui représentera le moment de la dépense.

7° Economies: Trois postes sont présentés à l'écran et l'utilisateur doit en choisir un.

- *Ajout d'un achat à crédit*: 40 postes sont affichés à l'écran et l'utilisateur doit en choisir un.
- *Annuler un achat à crédit*: Il est possible d'annuler un achat à crédit. L'argent liquide augmente. La jauge s'adapte.
- *Alimentation d'un achat à crédit*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en jaune dans la jauge. Le montant de l'achat à crédit augmente et le liquide diminue.
- *Ajout d'une économie spécifique*: 40 postes sont affichés à l'écran et l'utilisateur doit en choisir un.
- *Annuler une économie spécifique*: Il est possible d'annuler une économie spécifique. L'argent liquide augmente. La jauge s'adapte.
- *Alimenter une économie spécifique*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en jaune dans la jauge. Le montant de l'économie spécifique augmente et le liquide diminue.
- *Décrémenter une économie spécifique*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en jaune dans la jauge. Le montant de l'économie spécifique diminue et le liquide augmente.
- *Saisie du compte origine pour le transfert*: Différents comptes sont affichés à l'écran. L'utilisateur doit en sélectionner un comme origine du transfert.
- *Saisie du compte destination*: Différents comptes sont affichés à l'écran. L'utilisateur doit en sélectionner un comme destination du transfert.
- *Saisie du montant à transférer*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres à l'écran et est représenté en jaune dans la jauge.

8° Test: Deux questions sont inscrites à l'écran (voir chapitre IV, IV.10). L'utilisateur en choisit une.

- *Saisie du montant de la dépense à simuler*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres et est représenté en orange dans la jauge.
- *Saisie du montant à épargner par mois*: Il s'agit d'une saisie qui s'effectue soit au clavier, soit à l'aide de la souris, soit à l'aide des billets et/ou pièces. Ce montant apparaît en chiffres et est représenté en orange dans la jauge.
- *Saisie du mois*: Les 15 mois suivants le mois actuel sont affichés et l'utilisateur doit en choisir un.
- *Visualisation de la réponse*: la réponse est visualisée sous forme de jauge. Elle équivaut à diviser la dépense à simuler par le montant à épargner par mois (question 1) et à diviser la dépense à simuler par le nombre de mois (question 2).

9° Bilan: Les icônes semaine - mois - année sont affichées et l'utilisateur doit en choisir une.

- *Visualisation des différentes dépenses*: Les différenes dépenses effectuées durant la période choisie par l'utilisateur sont affichées à l'écran par ordre chronologique. Elles sont représentées sous forme de jauge.
- *Calcul de la somme des dépenses*: Tous les montants des dépenses sont additionnés et la somme apparaît à l'écran.

10° Etat des finances

- *Visualisation des différents comptes*: Les différents comptes que possède l'utilisateur sont affichés sous forme de jauge.
- *Visualisation des achats à crédit*: Les différents achats à crédit que possède l'utilisateur sont affichés sous forme de jauge.
- *Visualisation des économies spécifiques*: Les différentes économies spécifiques que possède l'utilisateur sont affichés sous forme de jauge.

IX. LES TRACES D'UTILISATION

"Les traces d'utilisation sont constituées par la suite chronologique des manipulations effectuées par une personne handicapée lors d'une session d'utilisation du logiciel."[6]

Après chaque séance, le programme Budget enregistre sous la forme d'un fichier de type texte le numéro de la séance, le nom de l'utilisateur, les paramètres choisis ainsi que les actions menées au cours de l'utilisation.

L'objectif essentiel des traces d'utilisation est de voir si le logiciel est adapté ou non à la personne. L'adaptation se situe à deux niveaux: au niveau de l'interface et des paramètres. Ces traces ne permettront pas de juger de l'intérêt ou de l'utilité du logiciel.

Certaines "erreurs" de manipulation peuvent apparaître dans les traces. Ainsi, lorsque l'utilisateur quitte une fonction sans la mener à son terme ou désire recommencer la dernière action effectuée, cela traduit généralement une difficulté. Dès lors, en dépouillant toutes les traces d'un utilisateur, on peut découvrir que la fréquence de certaines erreurs de manipulations est assez élevée et donc supposer qu'un écran pose problème, qu'un terme est mal compris ou ambigu, que l'icône est mal dessinée, que l'enchaînement des écrans doit être revu, ... De plus, la comparaison de ces fréquences d'erreurs de séance en séance pour un même utilisateur pourrait peut-être nous aider à mesurer des progrès réalisés par la personne handicapée.

[6]France Cheron et Olivier Berger, 1990

X. CONCLUSION.

Il est important de rappeler que le logiciel Budget constitue une aide à la personne handicapée mentale pour gérer son budget. Il n'a pas été conçu pour gérer le budget à sa place. Toutes les fonctions sont induites par la personne handicapée mentale.

★ ★ ★
★ ★

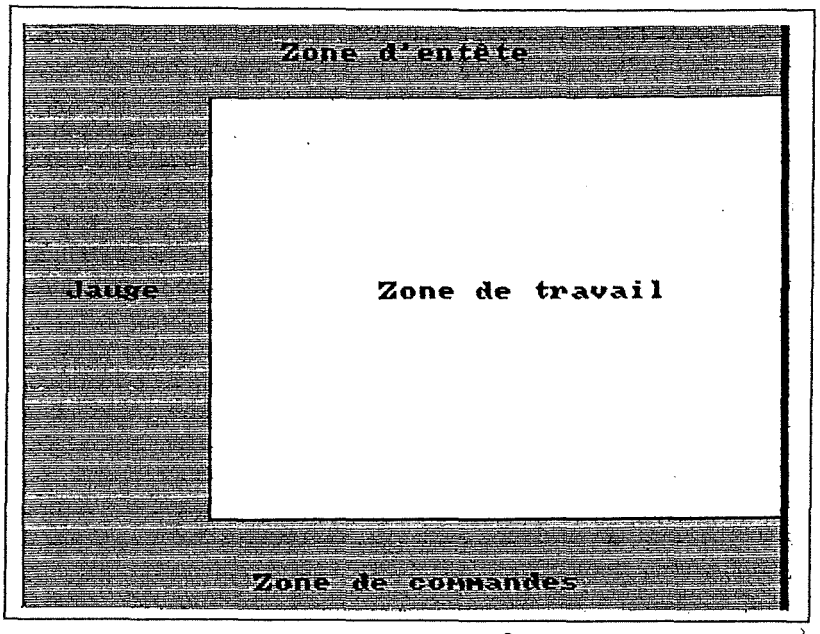
CHAPITRE IV

Interface du logiciel de Gestion de Budget

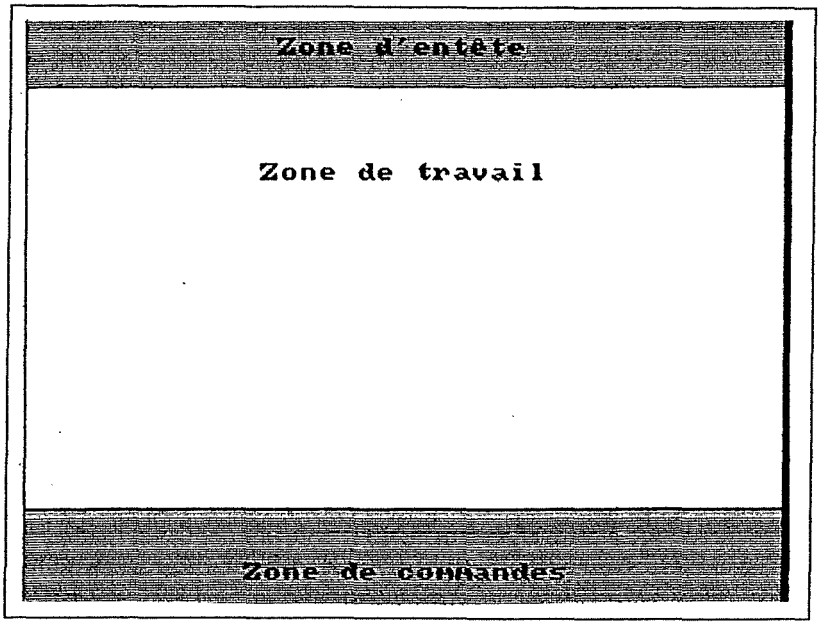
I. GENERALITES

1. L'homogénéité des écrans

Afin qu'un utilisateur, a fortiori une personne handicapée mentale, n'ait pas trop de difficulté à s'y retrouver dans la succession d'écrans (nombreux) d'un programme d'ordinateur, il nous semble que ces écrans devraient être semblables. C'est pour cette raison que nous proposons, dans le logiciel de gestion de budget, deux types d'écrans qui seront utilisés dans l'entièreté du programme.



Type d'écran N°1



Type d'écran N°2

Les deux types d'écrans diffèrent seulement du fait que le type 1 est destiné à recevoir une jauge.

La zone de travail est de couleur blanche et est destinée à recevoir des informations, spécifiques à chaque écran, sur lesquelles l'utilisateur peut agir. La zone de commandes fournit des commandes valables à plusieurs écrans. La zone d'entête contient le titre de l'écran plus sa représentation sous forme d'icône. Enfin, la jauge sert de visualisation pour l'argent.

2. Les commandes

La zone des commandes est destinée à contenir toute une série d'icônes qui ont un effet bien précis et qui, pour éviter les confusions, sera la même durant toute l'exécution du programme. Le but d'associer des icônes et des mots est d'aider des personnes qui ne savent pas lire ou qui éprouvent des difficultés de lecture. Il est également à remarquer que toutes ces icônes ont les mêmes format, taille, couleur:

La commande OK sert à passer d'un écran à un autre. Elle est représentée par une figure souriante.

L'icône OK barré sert à répondre négativement lors d'une question.

La commande écouter sert à réécouter la dernière phrase prononcée par la machine. Elle est représentée par une oreille et est présente sur chaque écran si un des paramètres l'exige.

La commande effacer est représentée par une gomme. Qu'il s'agisse d'une saisie d'une somme ou d'une saisie du prénom à l'aide de la souris, l'effet de cette commande est d'effacer le dernier caractère (ou chiffre) saisi.

L'effet de la commande recommencer est d'effacer, non seulement le dernier caractère saisi, mais aussi tous les autres. Cela revient à recommencer la saisie.

La commande jeter, représentée par une poubelle, sert à annuler une dépense prévue, un achat à crédit ou une économie spécifique.

La commande créer, représentée par un plus, sert à ajouter une dépense prévue, un achat à crédit ou une économie spécifique.

L'icône dépenser (le signe dollar) n'est présente que pour les prévisions de dépenses et a pour effet d'effectuer une dépense initialement prévue.

L'icône augmenter, à l'image d'une flèche vers le haut, a pour effet d'alimenter les économies spécifiques et les achats à crédit.

L'icône diminuer, à l'image d'une flèche vers le bas, a pour effet de réduire le montant des économies spécifiques.

L'icône SOS, représentée par une bouée de sauvetage, est destinée à fournir une aide.

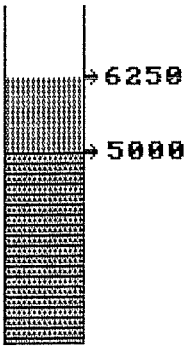
L'icône print, représentée par une imprimante, sert à imprimer l'écran en cours.

Pour terminer, la commande quitter, représentée par le mot STOP entouré d'un hexagone, sert à sortir du programme. Cette icône est présente uniquement dans l'écran du menu principal.

D'autres icônes conviendraient peut-être mieux à ces commandes. Un des reproches qu'on pourrait formuler vis-à-vis de ces icônes, cest qu'une personne handicapée mentale, ne sachant pas lire, aurait des difficultés à savoir ce que signifie un tel dessin. D'après les éducateurs, il suffit de leur dire à quoi sert ce dessin.

3. La jauge

La jauge se trouve dans l'éventuelle zone qui lui est réservée. Son contenu représente un montant d'argent:



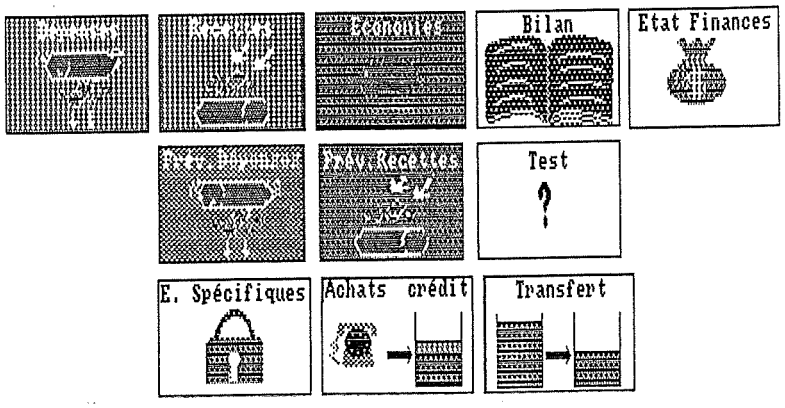
Exemple de jauge

Partout dans le programme, les jauges auront toujours la même largeur et le montant d'argent qu'elles représentent auront une couleur spécifique[7].

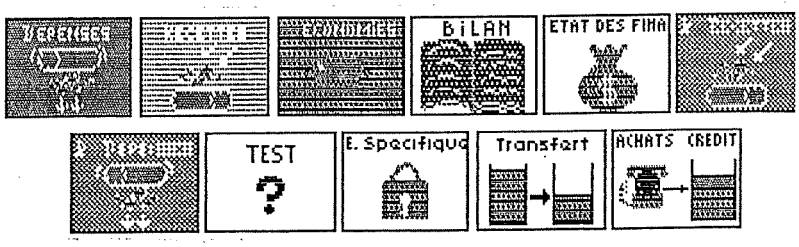
[7]voir le point 7 ci-après...

4. Les icônes

Nous traitons, ici, des icônes se référant à des fonctionnalités du logiciel:



Il s'agit des huit icônes figurant dans le menu principal: les dépenses, les recettes, les économies, le bilan, l'état des finances, les prévisions de dépense, les prévisions de recettes, et le test servent à appeler chacun de ces écrans. En plus de ces huit icônes, il y a celles du bilan: les économies spécifiques, les achats à crédit, et les transferts. Dans le coin supérieur gauche de la plupart des écrans se trouve une copie de plus petite taille de ces icônes ou d'autres si l'écran dans lequel elle se trouve ne peut être accédé directement par le menu principal.



Cette petite icône rappelle et/ou indique l'écran dans lequel on se trouve. Elle est destinée à aider les personnes qui ont des difficultés pour lire. Le nom de l'écran apparaît à côté de l'icône.

5. Les titres

Dans chaque zone d'entête, il y a un titre. Tous les titres de chaque écran sont écrits dans le même caractère, sont centrés et écrits en blanc. De plus, ils sont tous formulés à la deuxième personne du singulier, de façon à ce que la personne se sente plus concernée. Les écrans qui réclament une action de la part de l'utilisateur ont leur titre sous forme de question, les autres sont des mots.

Par exemple, toutes les saisies réclament une action de l'utilisateur et ont pour titre "Quel est ton prénom ?", "Quelles sont tes recettes ?". L'état des finances et le bilan sont une visualisation de l'argent à un moment donné et donc, ne requièrent pas d'intervention de la part de l'utilisateur. Leurs titres sont, respectivement, "Etat des finances" et "Bilan du mois:".

6. Les confirmations et les messages

Les confirmations et les messages (d'erreur) ont le même aspect: ils ont tous les deux des icônes-commandes et contiennent du texte écrit avec le même caractère que le titre. Ils ont un fond blanc, un cadre noir et vont se poser à l'avant-plan des écrans.

La différence fondamentale entre les confirmations et les messages réside dans le fait que l'utilisateur doit agir sur les confirmations (par exemple, elle doit cliquer) et ne doit pas agir sur les messages. Ils disparaissent spontanément après quelques secondes.

Le fait que les confirmations aient le même aspect que les messages peut être source de confusion. Il se pourrait que la personne attende que le "message"

disparaisse, alors que c'est une confirmation. S'il s'avère que ceci déroute la personne, alors une autre méthode sera envisagée.

Exemple de confirmation : CONF1
Exemple de message d'erreur : M1

7. Les couleurs

Pour une personne ne sachant pas lire, nous supposons que les couleurs ont beaucoup d'importance. Nous avons représenté quelques aspect de ce logiciel par des couleurs: le vert correspond aux recettes, le rouge aux dépenses, le jaune à l'argent (en général), le vert clair aux prévisions de recettes, l'orange, aux prévisions de dépenses, le bleu, au temps. Le gris clair sert d'arrière-fond. Le noir et le blanc ont été utilisés pour le reste, les cadres, l'écriture,...

Il a été très difficile de ne pas se contredire. On a tendance, par exemple, à associer du rouge à un message d'erreur. Dans ce cas, on l'aurait peut-être confondu avec une dépense. Il ne faudrait pas non plus qu'il y ait trop de couleurs, sinon les personnes risqueraient la confusion.

Ces couleurs se manifestent comme arrière-fond des icônes, dans la jauge, dans la ligne du temps, ...

8. Les icônes des mois

A chaque mois, correspond une icône et vice versa. Grâce à cela, les personnes handicapées mentales qui ont des problèmes dans la compréhension des mois devraient avoir moins de problèmes. Supposons, par exemple, qu'on dise à

une personne handicapée mentale que son anniversaire est en décembre. Elle ne comprend pas ce que signifie "décembre". Si on lui explique qu'on fera son anniversaire lorsqu'on fera le sapin de noël, il est probable que cette explication accommodera mieux la personne. Le fait d'associer une icône à un mois (Sapin de noël - Décembre) est utilisé dans le logiciel. Les 12 icônes ont un arrière-fond bleu (temps).

9. Les postes de dépenses

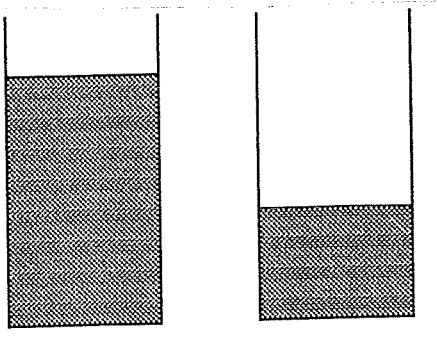
La bibliothèque des dépenses contient 40 icônes qui ont toutes la même taille. Cette bibliothèque n'est pas exhaustive (écran 7: voir ci-après).

II. REPRESENTATION GRAPHIQUE

Nous avons abondamment discuté du problème de la visualisation. Rappelons que la visualisation sert à aider les personnes qui auraient un problème de compréhension dans les calculs. Et plus particulièrement dans la compréhension des valeurs relatives des nombres.

1. La représentation graphique constitue une aide

Prenons, par exemple, une personne qui ne saisit pas la différence entre 50 et 100. Grâce à une représentation telle que:



elle aura sans doute plus facile de comprendre la différence. Cette méthode est utilisée dans le logiciel. Le contenu de la jauge représente un montant d'argent.

2. Le problème du pourcentage

Cette méthode souffre malheureusement d'un autre problème que nous n'avons pas encore réussi à résoudre.

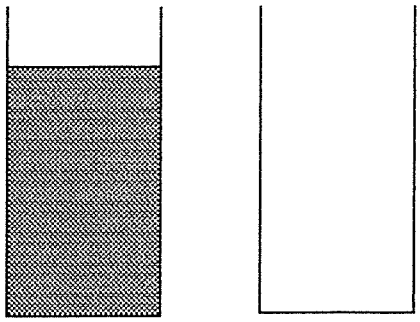
Ce problème est qu'un certain niveau dans la jauge devrait toujours représenter la même somme d'argent. Mais que se passe-t-il si la personne fait de grosses recettes ? Le récipient déborde.

Une solution serait de fixer une borne supérieure indiquant la somme maximum que la personne peut avoir. Mais quand il faut expliquer à cette personne que la somme correspond au maximum d'argent qu'elle pourrait avoir et qu'elle n'aura, sans doute, jamais, cela risquerait d'entraîner une certaine confusion. Cette façon de faire a déjà été utilisée mais n'a pas été jugée adéquate.

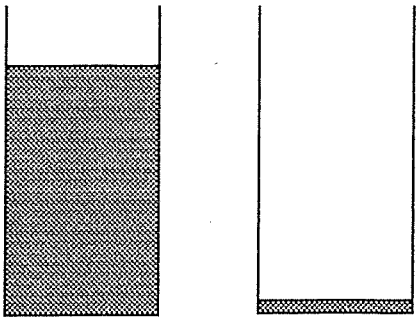
Ma solution est de combiner la méthode des pourcentages à celle de la borne supérieure. Il suffirait de fixer une borne supérieure, invisible à la personne, indiquant la somme maximum que la personne a eu la dernière fois qu'elle a utilisé l'ordinateur + un certain pourcentage (pour le taux de tolérance). Si la personne a un peu plus d'argent durant cette période, la visualisation fournira quelque chose de valable mais pour des fluctuations exceptionnelles, elle aura un peu plus de mal à comprendre la visualisation.

3. Le problème des échelles

Un autre problème engendré par la visualisation est un problème d'échelle lors de la comparaison de plusieurs sommes. Supposons que la personne a 50.000 francs dans un compte et 50 francs d'argent liquide. Si on veut comparer ces 50.000 francs avec 50 francs, on obtient quelque chose comme ceci:



On a ici, une échelle de 1/1000. On a l'impression que le deuxième récipient est vide. Mais non! Pour résoudre ce problème, j'ai proposé de "tricher". C'est-à-dire de montrer qu'on a un peu d'argent mais que par rapport au montant du compte, celui du liquide est minime:



Il existe encore de nombreux problèmes posés par la représentation et la visualisation. Le logiciel de gestion de budget n'est pas fait pour apprendre à la personne handicapée mentale à mieux percevoir les valeurs mais pour aider la personne à gérer son budget.

III. APPORTS

En cours d'année, et suite à nos stages, nous avons soulevé quelques ajouts par rapport au cahier des charges. Quatre points peuvent en ressortir: un bilan, une aide, une sortie sur imprimante et une visualisation des prévisions.

1. Un bilan

A la fin de chaque période de gestion, qu'il s'agisse d'un mois ou d'une semaine, il serait bien que l'ordinateur fasse un résumé des dépenses effectuées durant cette période. Il serait aussi intéressant de faire un bilan sur une année entière. Voir écran 20 et suivants...

2. L'aide

Il s'agit d'une commande, située dans le bas, à droite de l'écran, permettant de recevoir une explication sur la fonction des différentes commandes.

3. Sortie sur imprimante

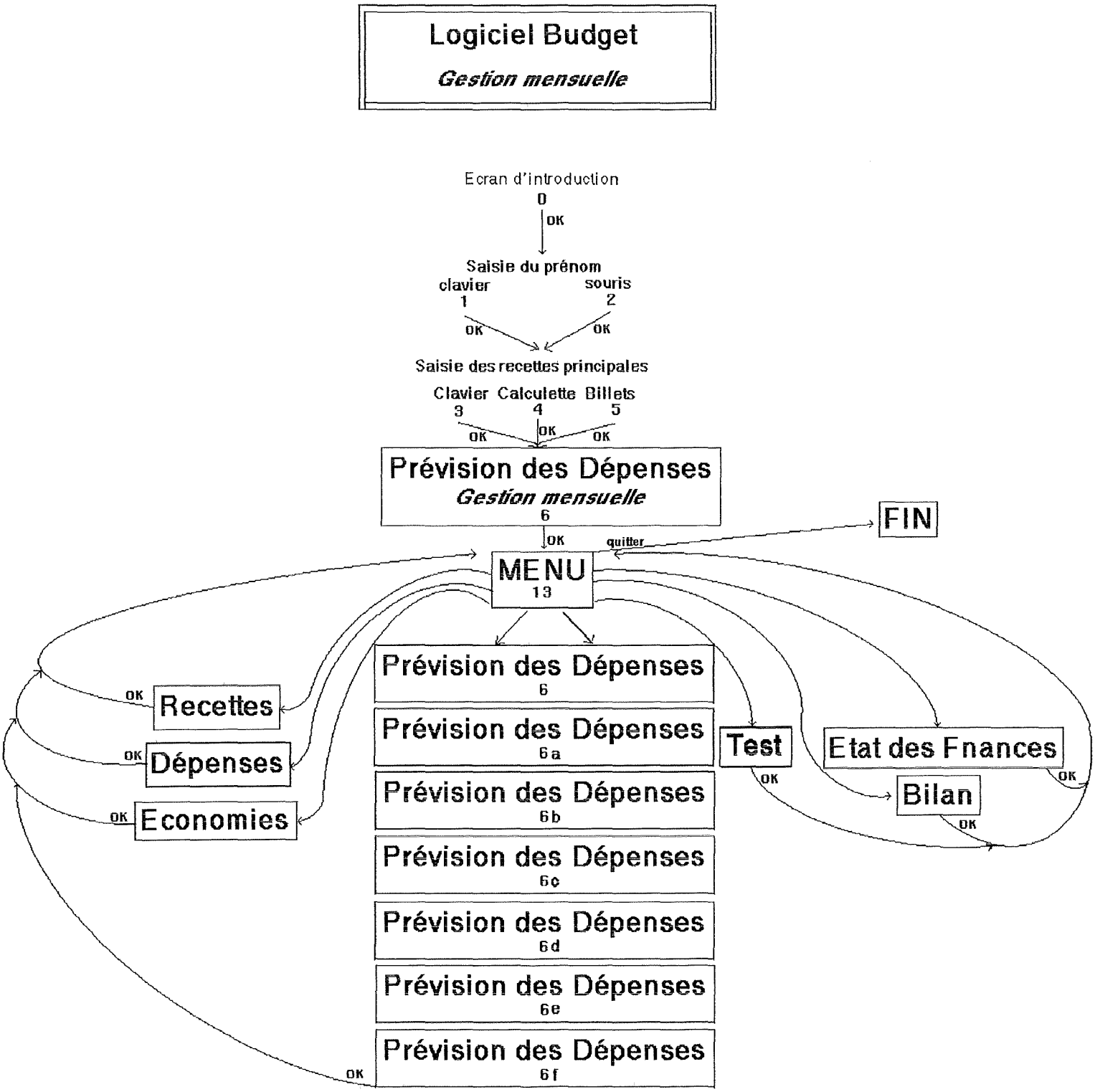
Pour certaines personnes, il serait souhaitable d'avoir une trace écrite, qui serait ***bien à elles***. Une des solutions, simple à utiliser pour la personne utilisatrice, serait de rajouter une icône-commande sous forme d'imprimante, qui ferait une impression de l'écran dans lequel elle se trouve. Cette icône serait présente dans tous les écrans, éventuellement, d'après des paramètres.

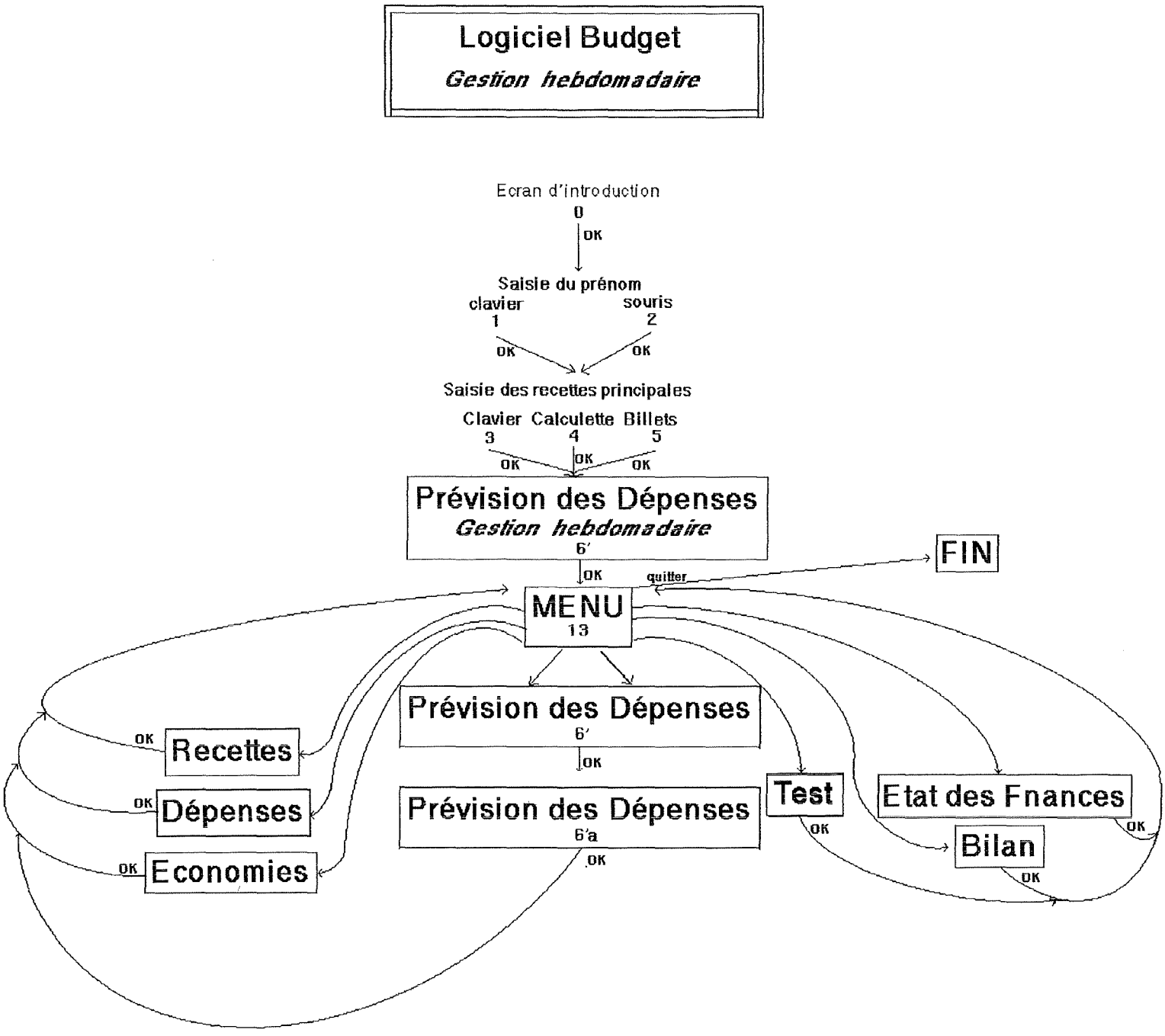
4. Visualisation des prévisions

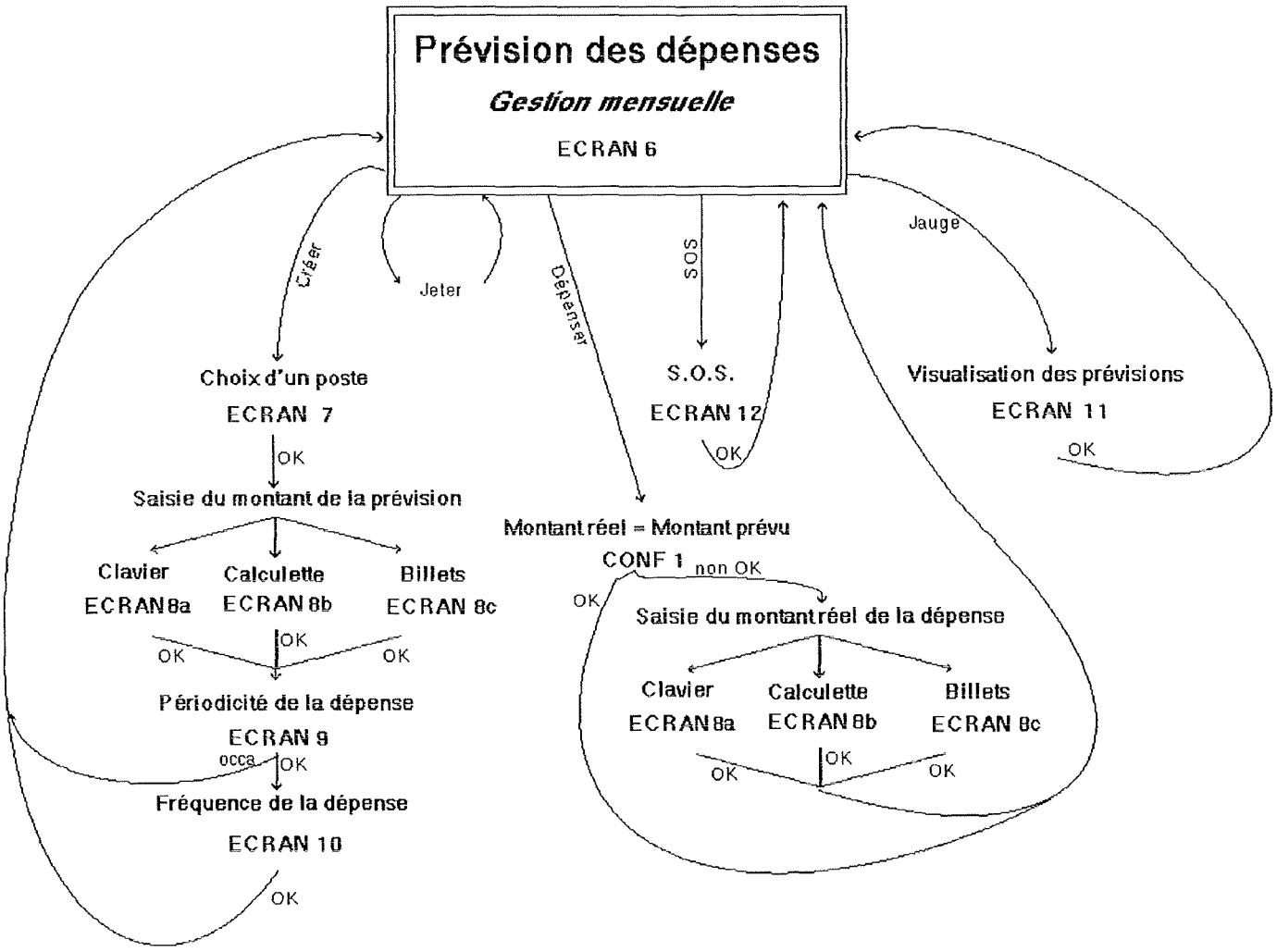
La visualisation devrait permettre à la personne handicapée mentale de mieux saisir la différence entre plusieurs montants d'argent. Lorsque l'on se trouve dans l'écran des prévisions de dépense (6), on "clique" dans la jauge et on arrive à l'écran des visualisations des prévisions (11).

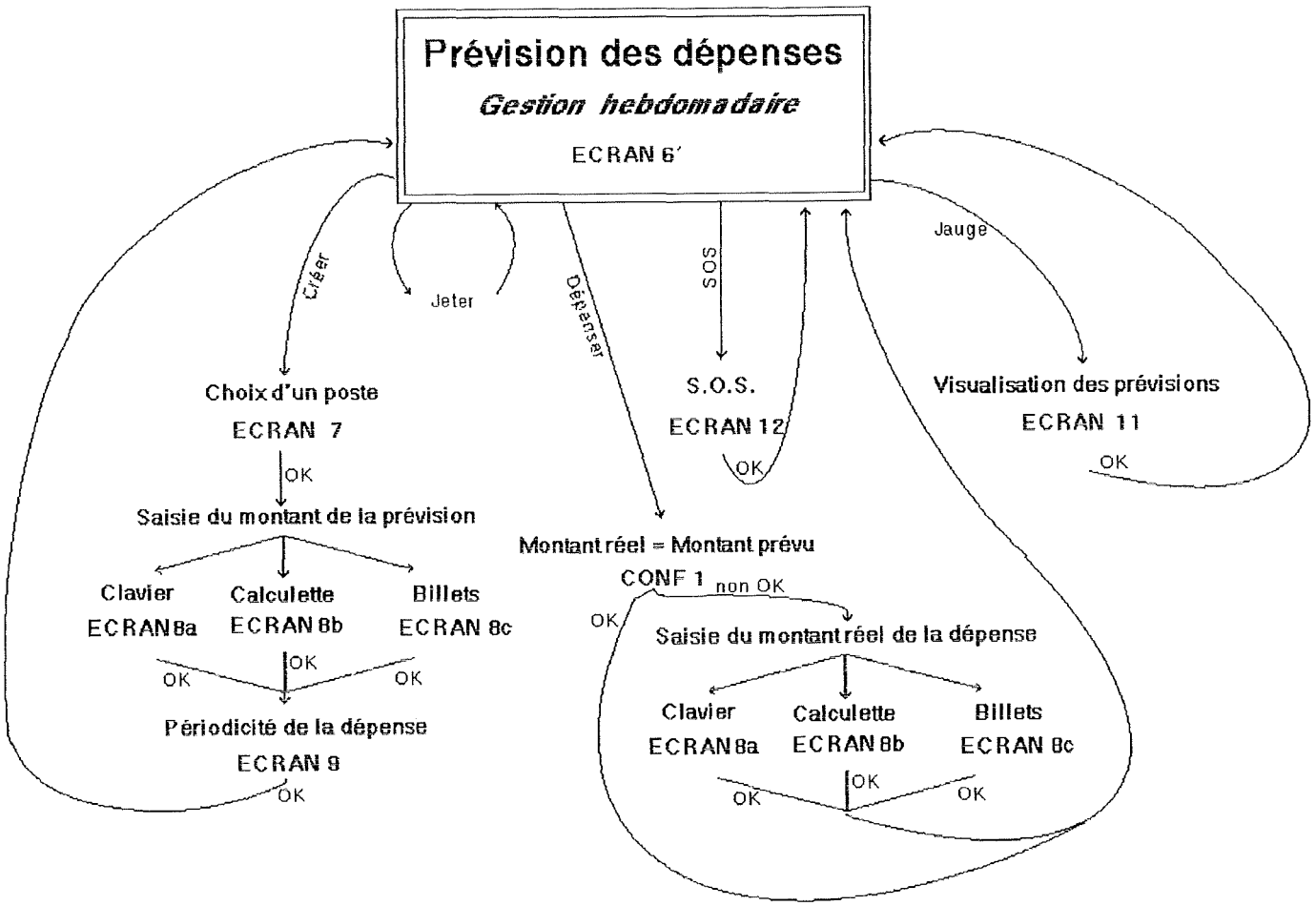
IV. PROTOTYPAGE DES ECRANS

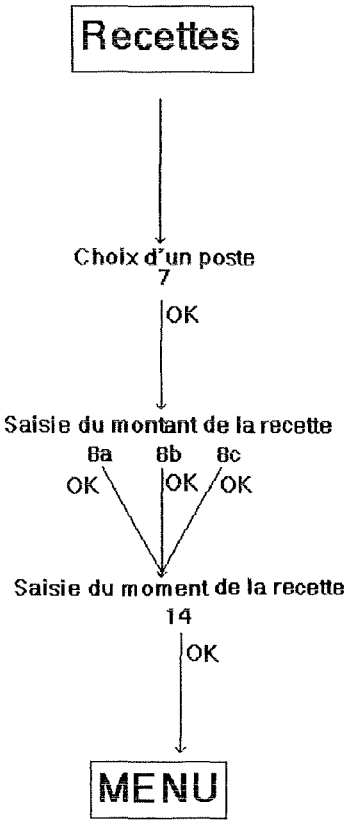
Voici la succession des écrans du logiciel Budget:

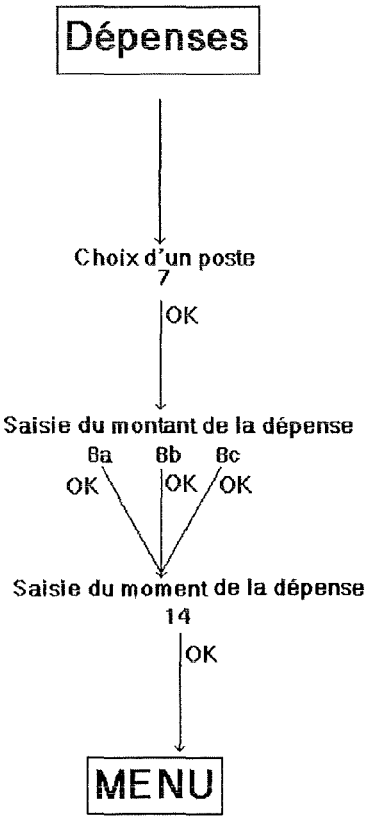

Logiciel Budget
Gestion mensuelle
Ecran d'introduction
0
OK
Saisie du prénom
clavier
1
souris
2
OK
OK
Saisie des recettes principales
Clavier
3
Calculette
4
Billets
5
OK
OK
OK
Prévision des Dépenses
Gestion mensuelle
6
OK
quitter
FIN
MENU
13
OK
Recettes
OK
Dépenses
OK
Economies
Prévision des Dépenses
6
Prévision des Dépenses
6a
Prévision des Dépenses
6b
Prévision des Dépenses
6c
Prévision des Dépenses
6d
Prévision des Dépenses
6e
Prévision des Dépenses
6f
OK
Test
OK
Etat des Fnances
OK
Bilan
OK

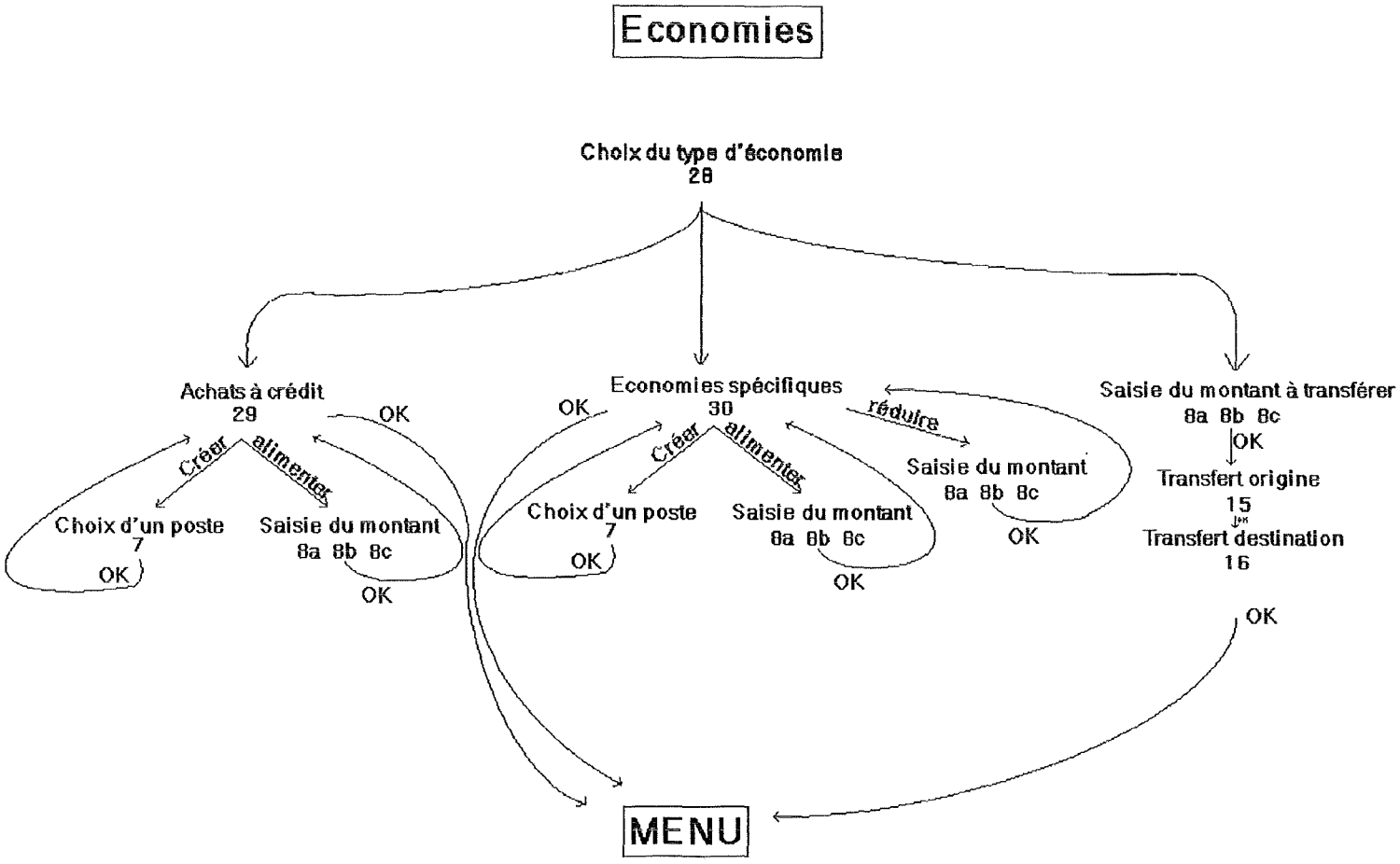

Logiciel Budget
Gestion hebdomadaire
Ecran d'introduction
0
OK
Saisie du prénom
clavier
souris
1
2
OK
OK
Saisie des recettes principales
Clavier Calculette Billets
3
4
5
OK
OK
OK
Prévision des Dépenses
Gestion hebdomadaire
6'
OK
quitter
FIN
MENU
13
Prévision des Dépenses
6'
OK
Recettes
OK
Dépenses
OK
Economies
OK
Prévision des Dépenses
6'a
OK
Test
OK
Etat des Fnances
OK
Bilan
OK

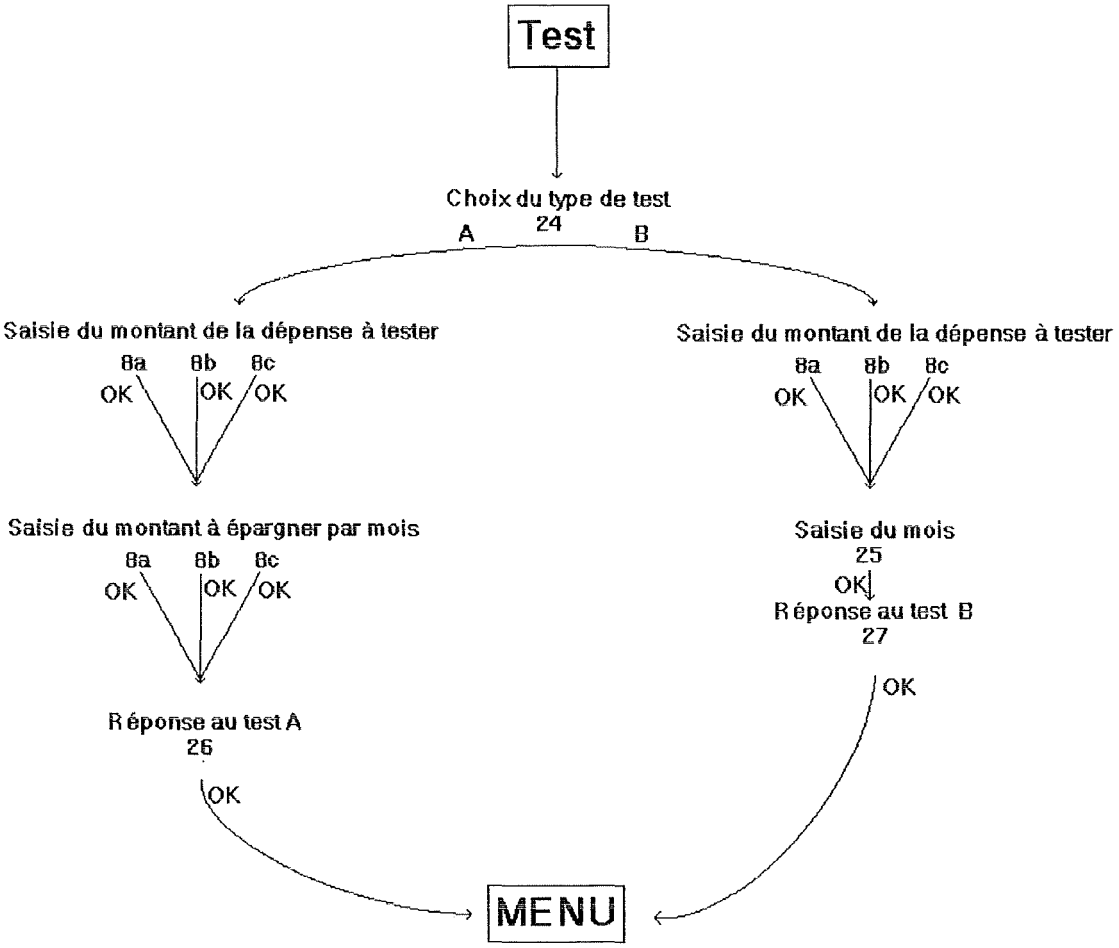

Prévision des dépenses
Gestion mensuelle
ECRAN 6
Créer
Jeter
Dépenser
SOS
Jauge
Choix d'un poste
ECRAN 7
OK
Saisie du montant de la prévision
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK
Périodicité de la dépense
ECRAN 9
occa
OK
Fréquence de la dépense
ECRAN 10
OK
S.O.S.
ECRAN 12
OK
Montant réel = Montant prévu
CONF 1
non OK
OK
Saisie du montant réel de la dépense
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK
Visualisation des prévisions
ECRAN 11
OK

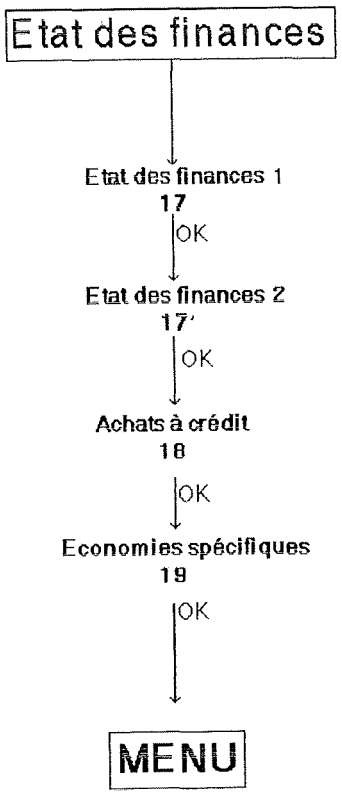

Prévision des dépenses
Gestion hebdomadaire
ECRAN 6'
Créer
Jeter
Dépenser
SOS
Jauge
Choix d'un poste
ECRAN 7
OK
Saisie du montant de la prévision
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK
Périodicité de la dépense
ECRAN 9
OK
S.O.S.
ECRAN 12
OK
Visualisation des prévisions
ECRAN 11
OK
Montant réel = Montant prévu
CONF 1
non OK
OK
Saisie du montant réel de la dépense
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK

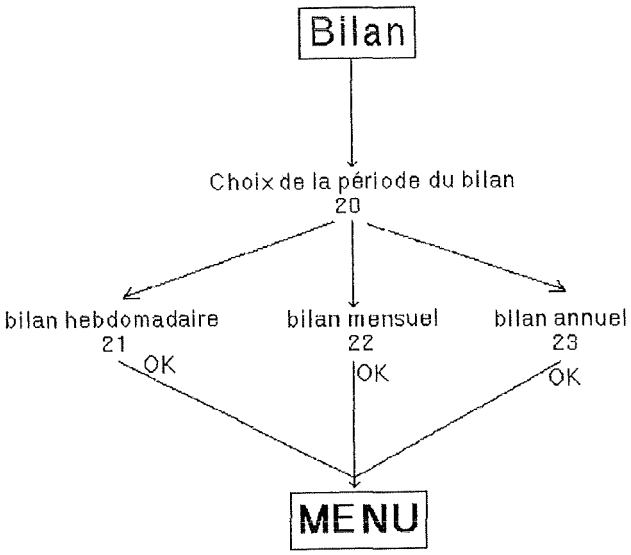

Recettes
Choix d'un poste
7
OK
Saisie du montant de la recette
8a
8b
8c
OK
OK
OK
Saisie du moment de la recette
14
OK
MENU


Dépenses
Choix d'un poste
7
OK
Saisie du montant de la dépense
8a
8b
8c
OK
OK
OK
Saisie du moment de la dépense
14
OK
MENU


Economies
Choix du type d'économie
28
Achats à crédit
29
Economies spécifiques
30
Saisie du montant à transférer
8a 8b 8c
OK
Transfert origine
15
Transfert destination
16
Créer
alimenter
Choix d'un poste
7
OK
Saisie du montant
8a 8b 8c
réduire
MENU

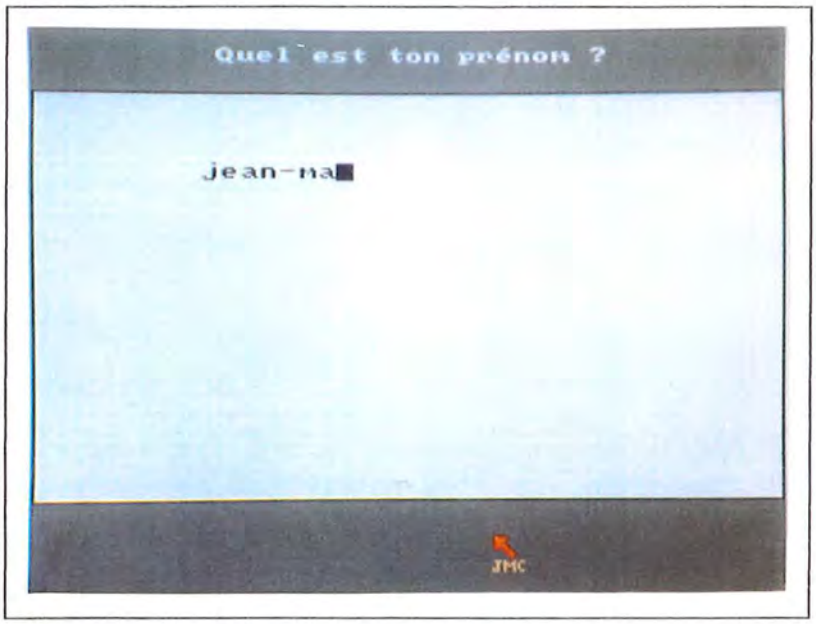

Test
Choix du type de test
A
24
B
Saisie du montant de la dépense à tester
8a
8b
8c
OK
OK
OK
Saisie du montant à épargner par mois
8a
8b
8c
OK
OK
OK
Réponse au test A
26
OK
Saisie du montant de la dépense à tester
8a
8b
8c
OK
OK
OK
Saisie du mois
25
OK
Réponse au test B
27
OK
MENU


Etat des finances
Etat des finances 1
17
OK
Etat des finances 2
17'
OK
Achats à crédit
18
OK
Economies spécifiques
19
OK
MENU

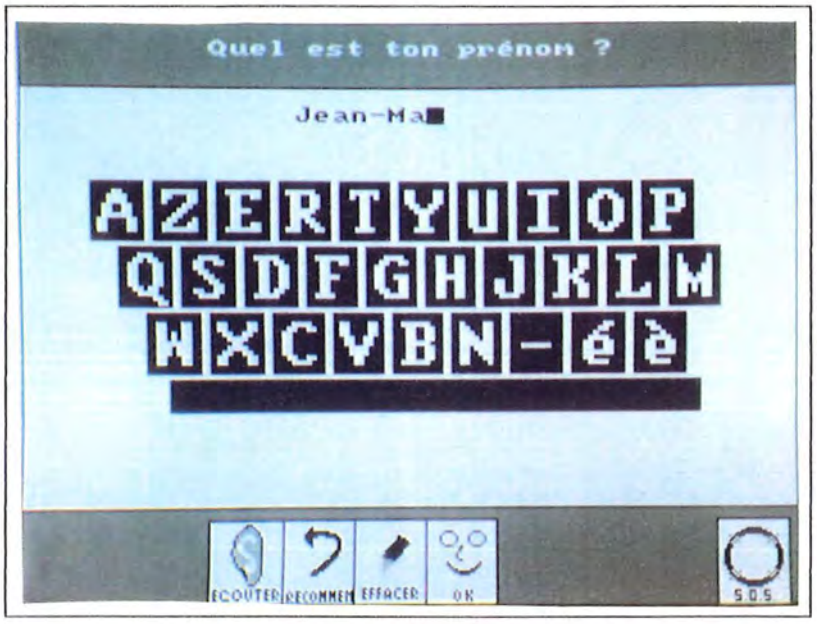

Bilan
Choix de la période du bilan
20
bilan hebdomadaire
21
OK
bilan mensuel
22
OK
bilan annuel
23
OK
MENU

Les écrans ayant été implémentés sont des photos couleurs, les autres sont des prototypes en noir et blanc.

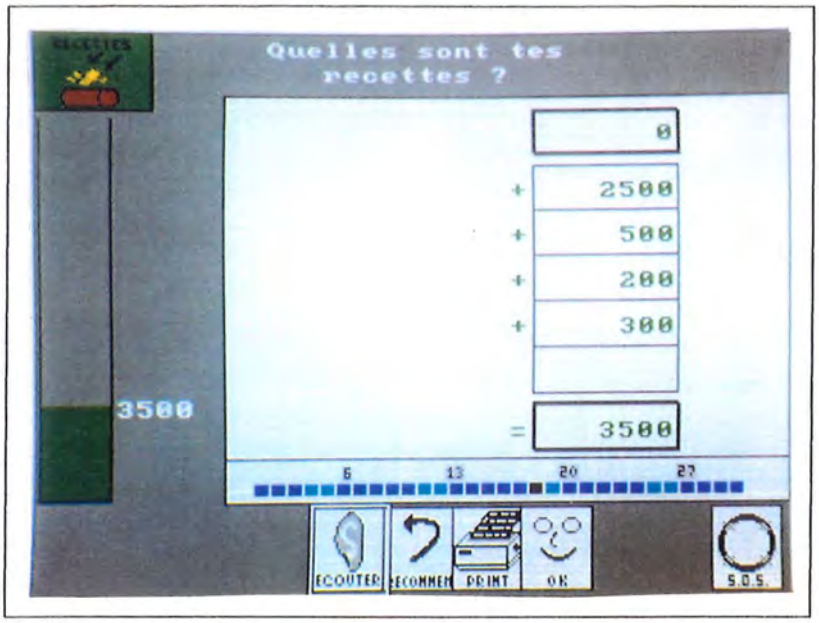
Ecran d'introduction (0)



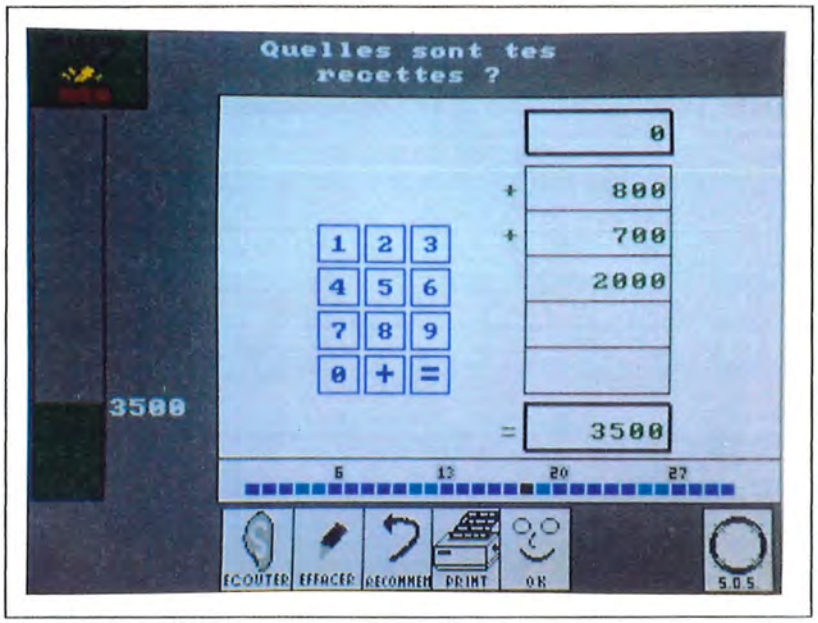
Saisie du prénom clavier (1)



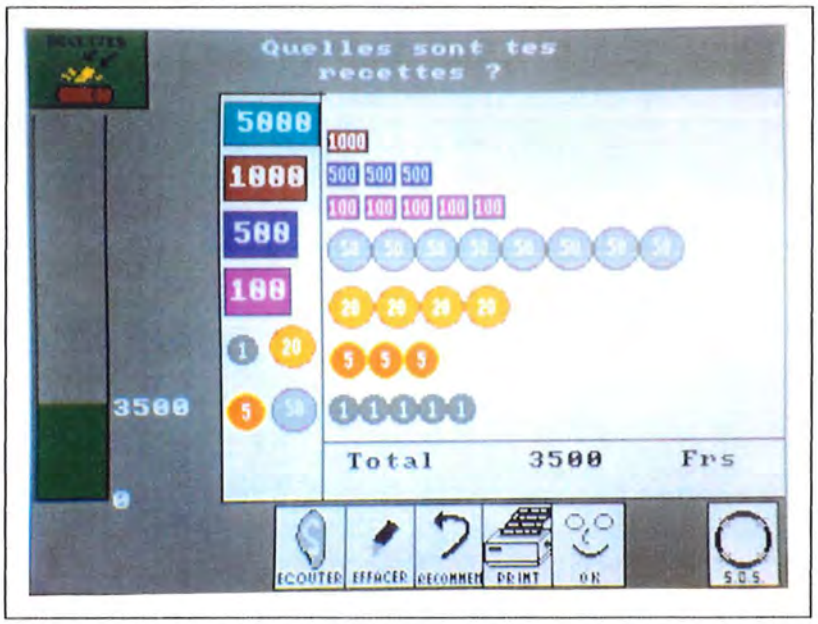
Saisie du prénom - souris (2)



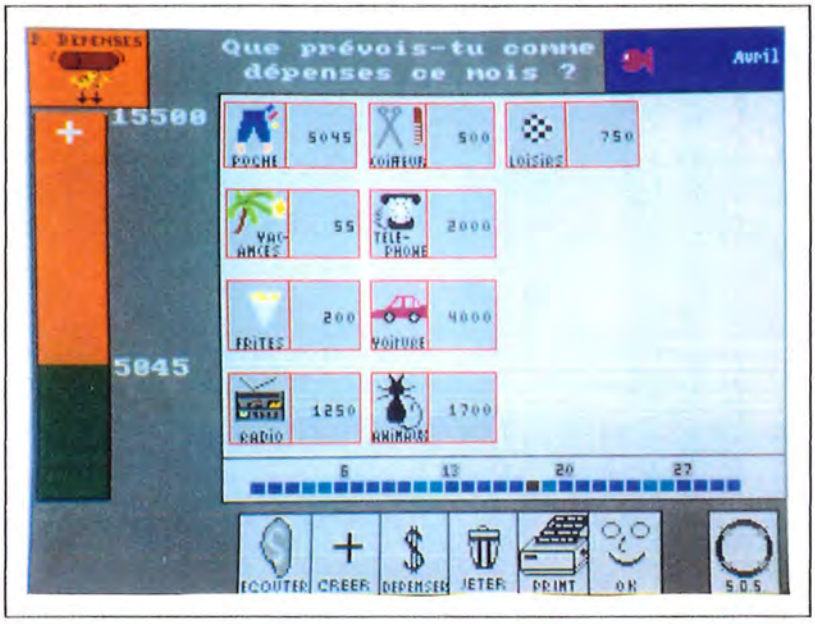
Saisie des recettes principales - clavier (3)



Saisie des recettes principales - calculette (4)

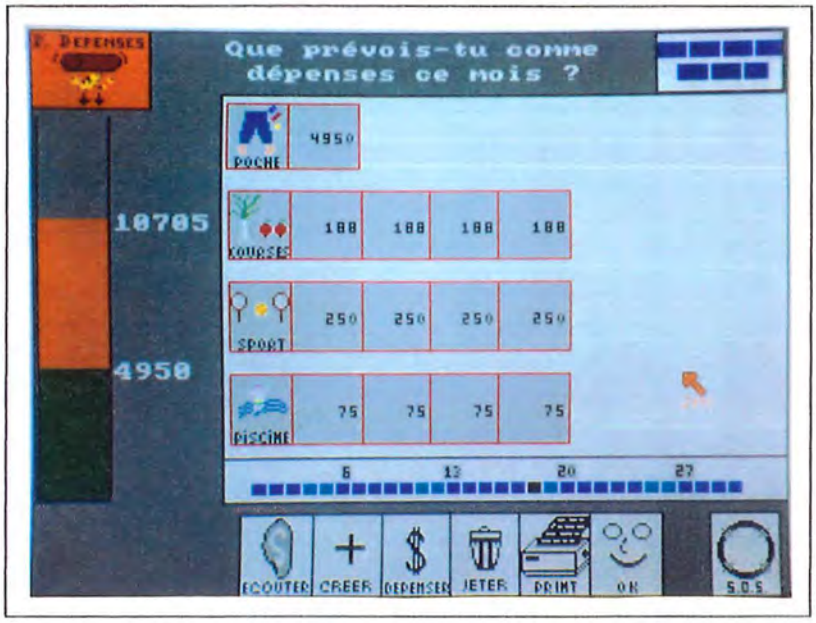


Saisie des recettes principales - billets (5)



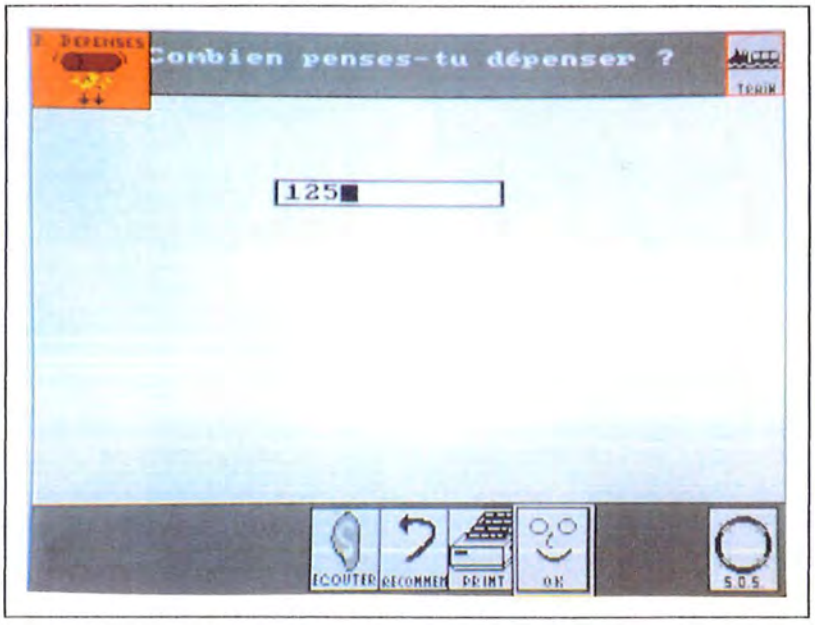
Prévision des dépenses: Gestion mensuelle (6)

Prévision des dépenses: Gestion hebdomadaire (6')

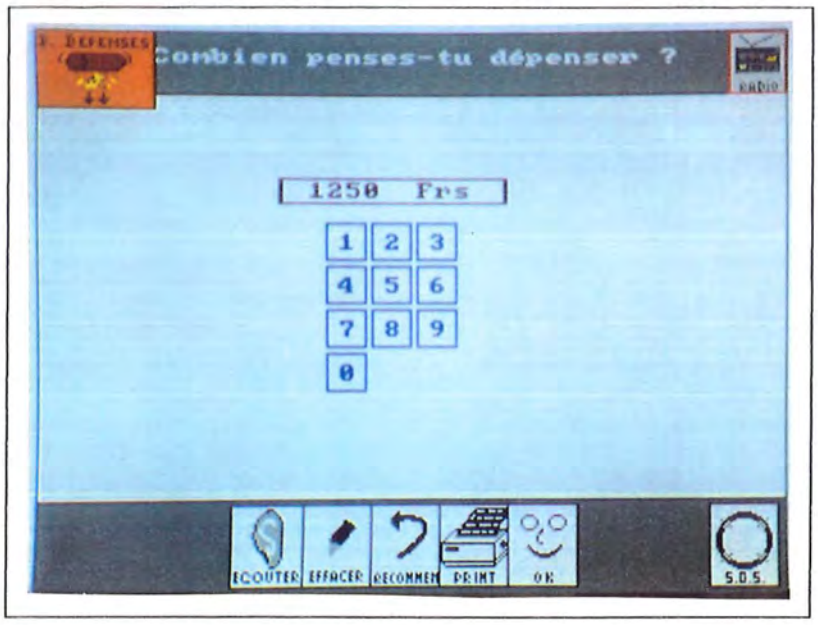


Prévision des dépenses: par semaines (6a)

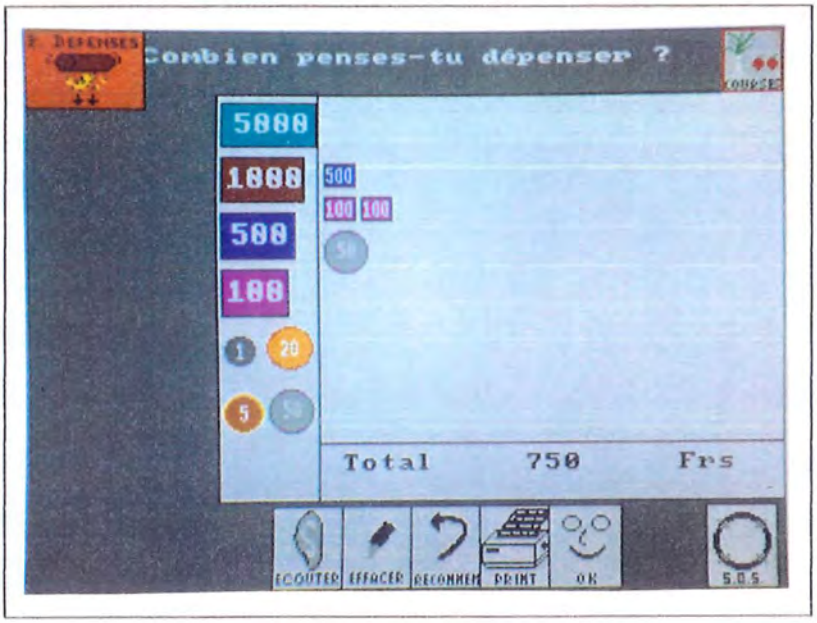
choix d'un poste (7)



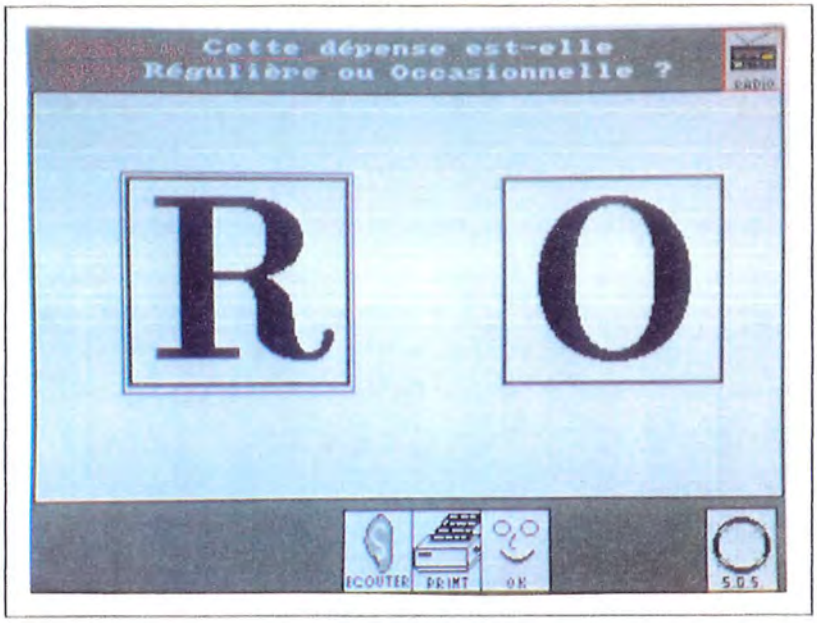
Saisie de somme - clavier (8a)



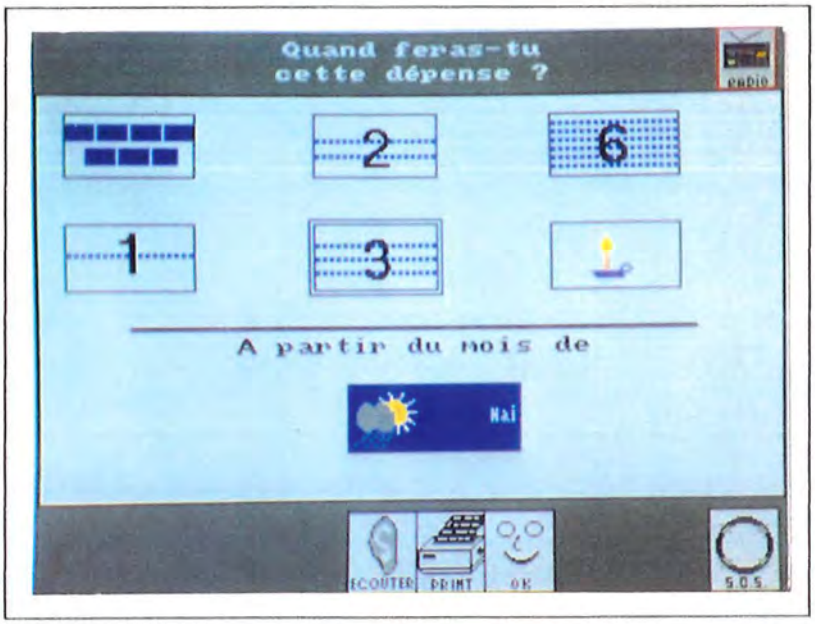
Saisie de somme - calculette (8b)



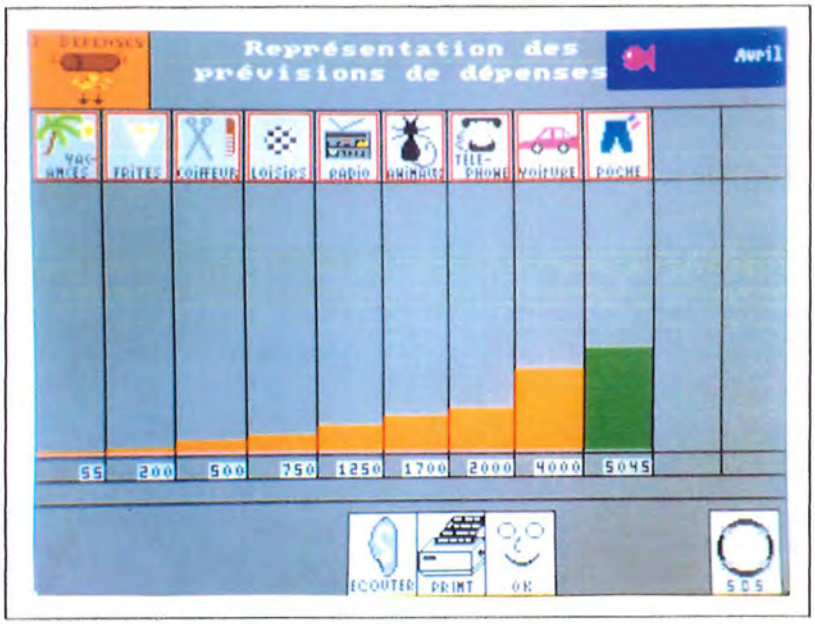
Saisie de somme - billets (8c)



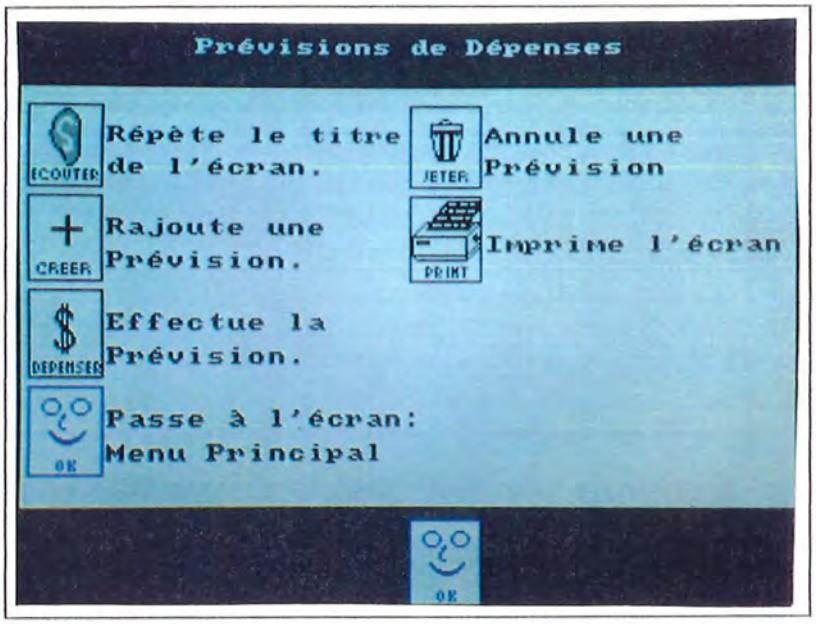
Saisie de la périodicité (d'une dépense prévue) (9)



fréquence de la dépense régulière (10)

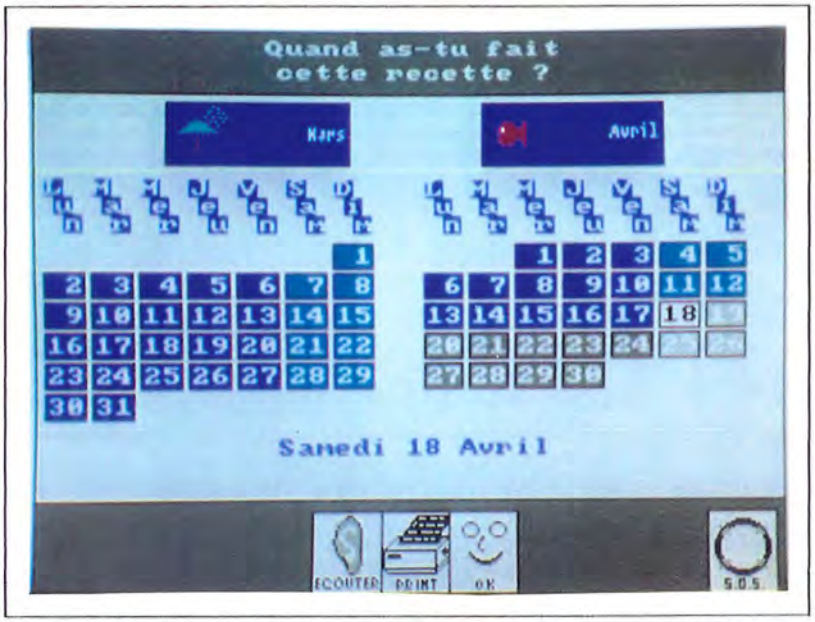


Visualisation des prévisions de dépenses (11)

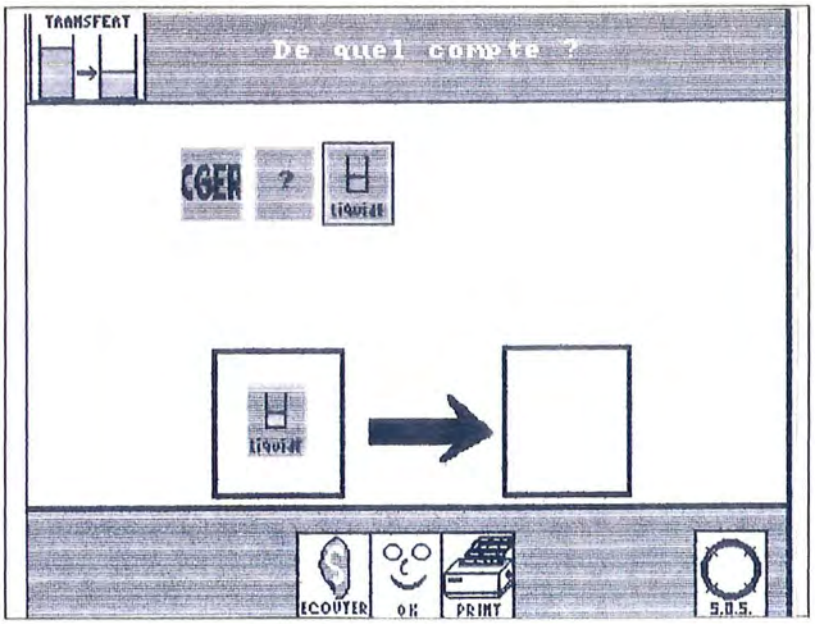


L'aide (12)

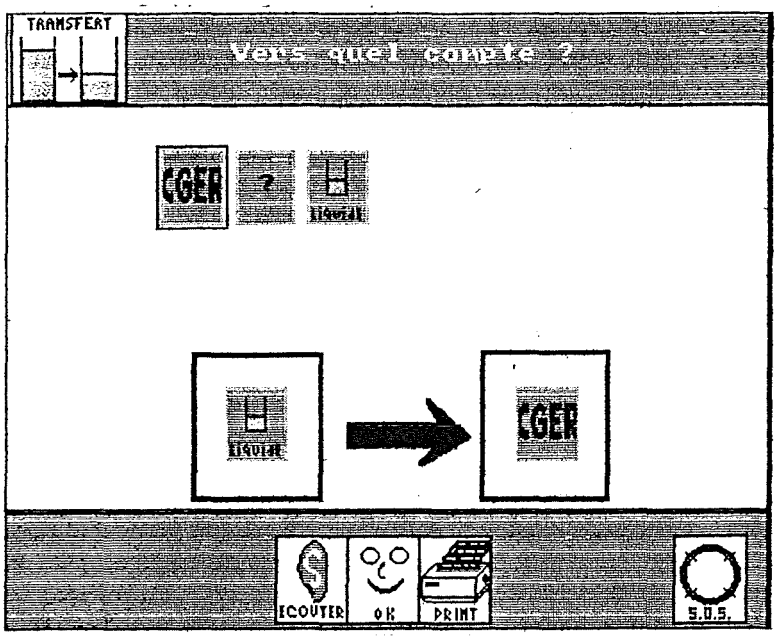
Menu principal (13)



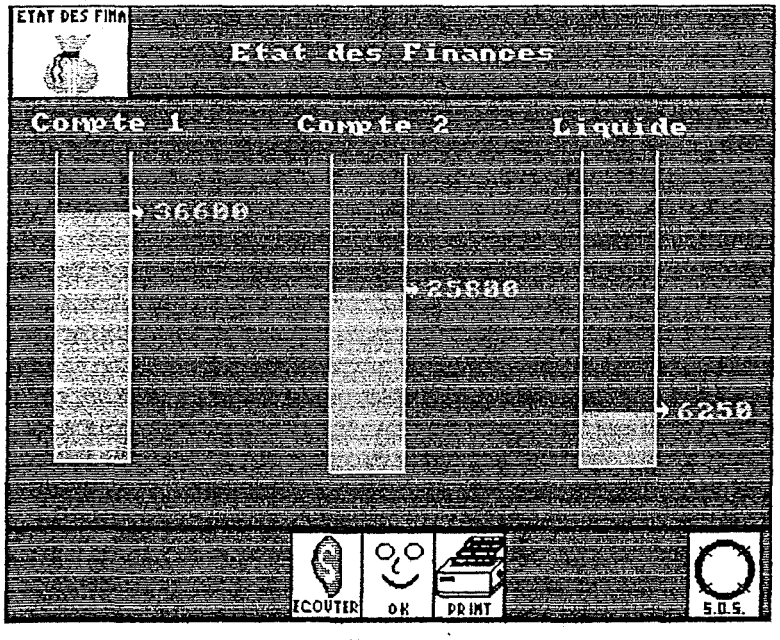
Saisie du moment d'une recette ou dépense (14)



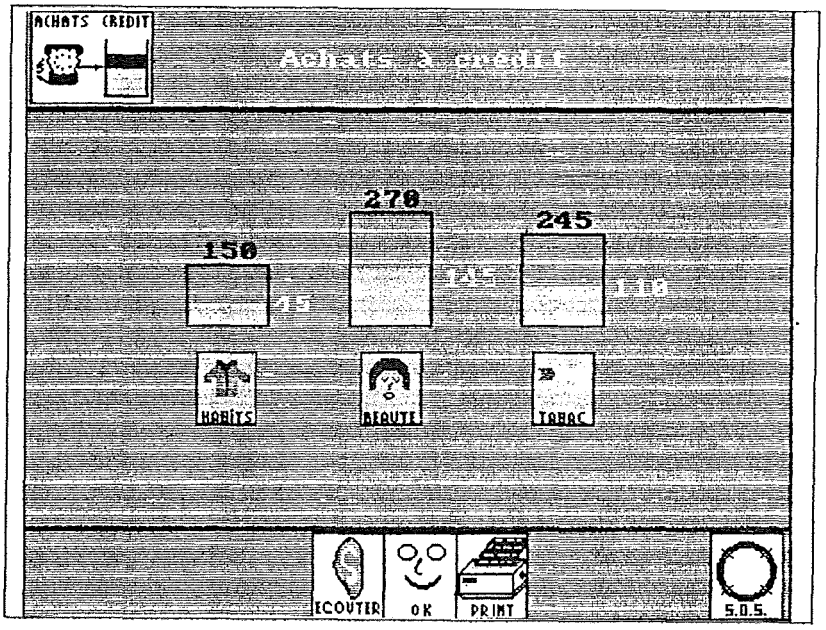
Saisie de l'origine d'un transfert (15)



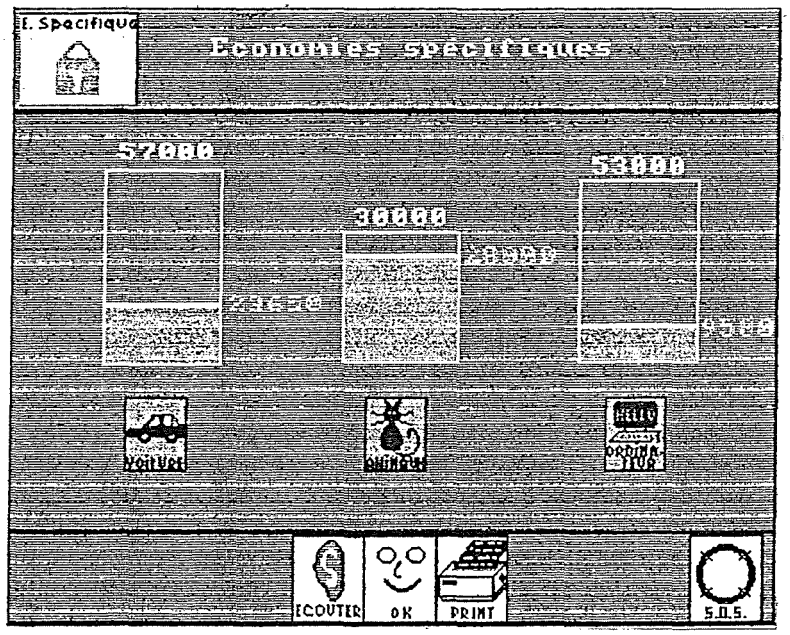
Saisie de la destination d'un transfert (16)



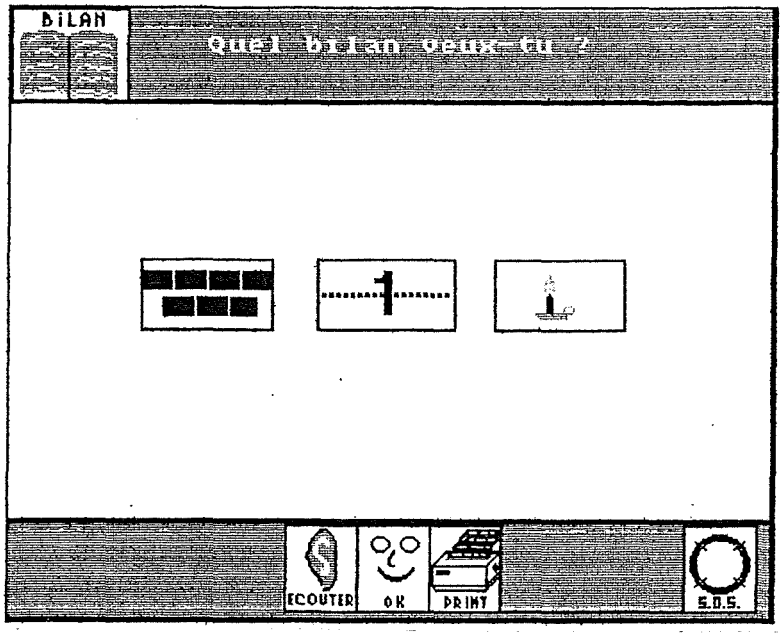
Etat des finances: comptes (17)



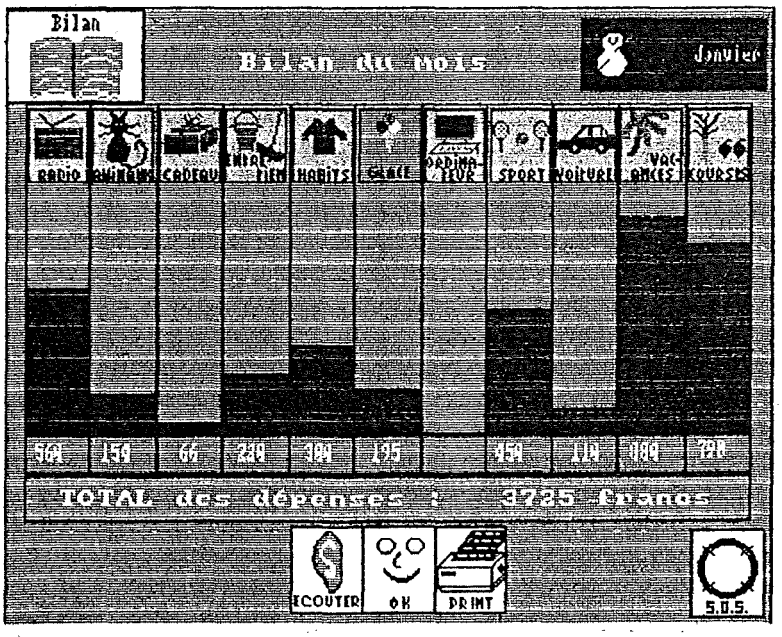
Etat des finances: achats à crédit (18)



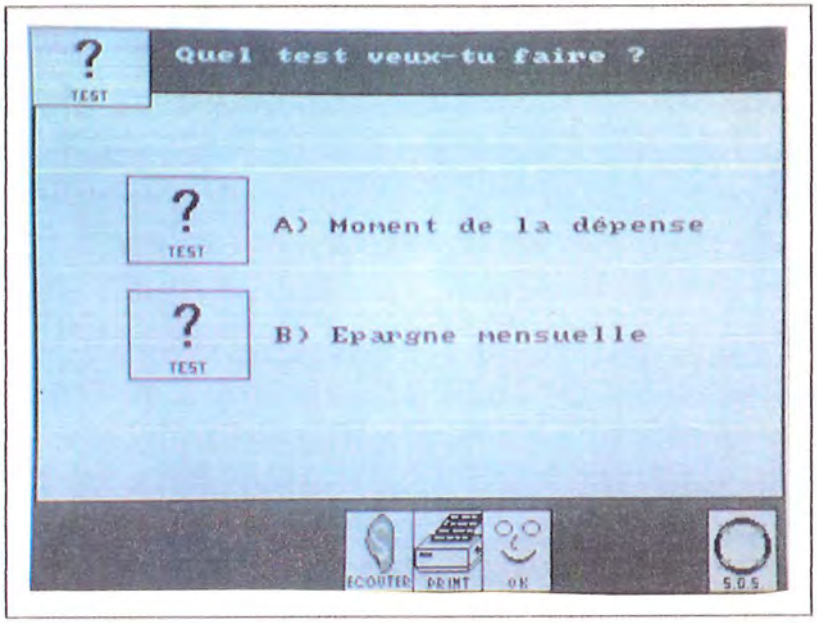
Etat des finances: économies spécifiques (19)



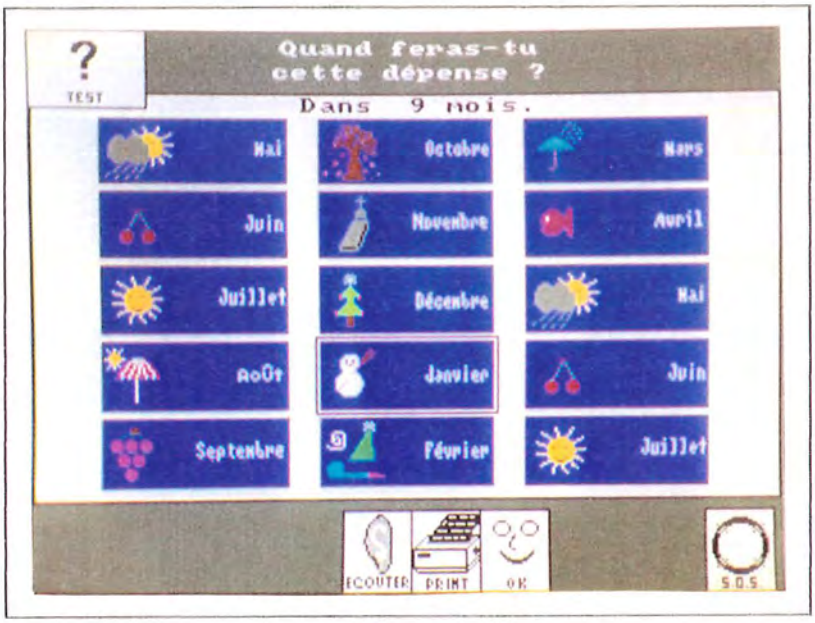
Choix de la période du bilan (20)



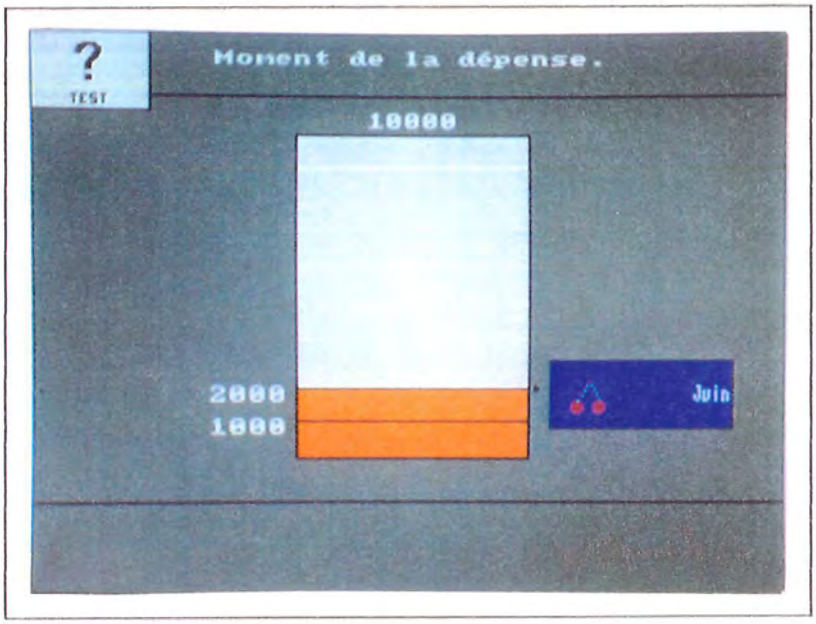
Bilan mensuel (22)



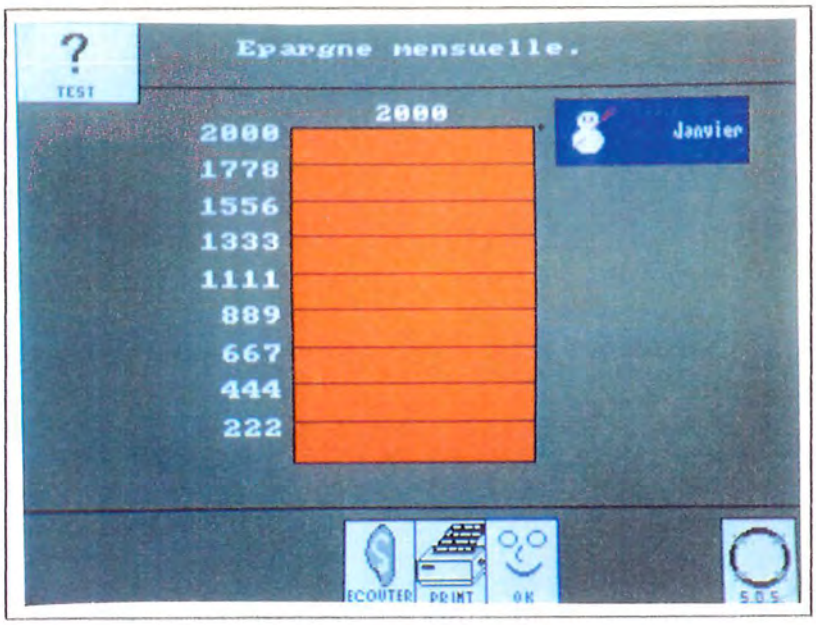
Choix du type de test (24)



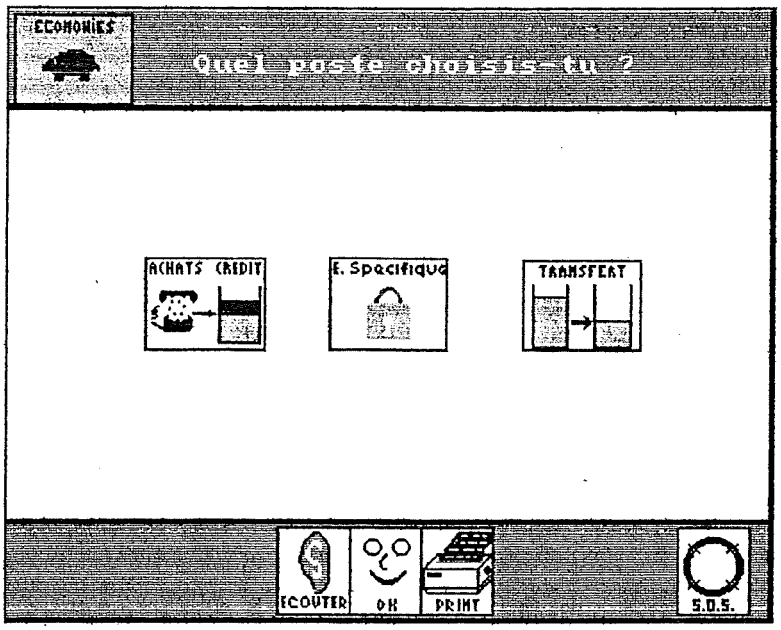
Saisie du mois (25)



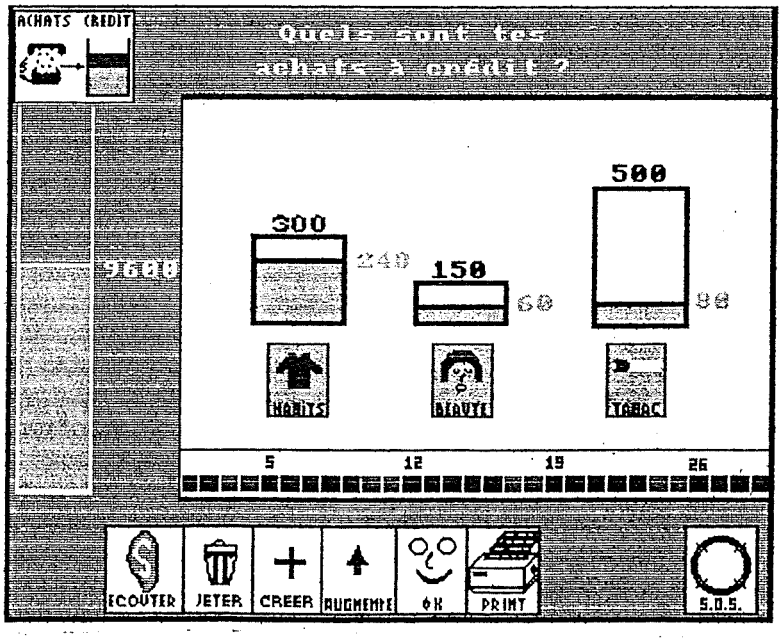
Réponse au test A (26)



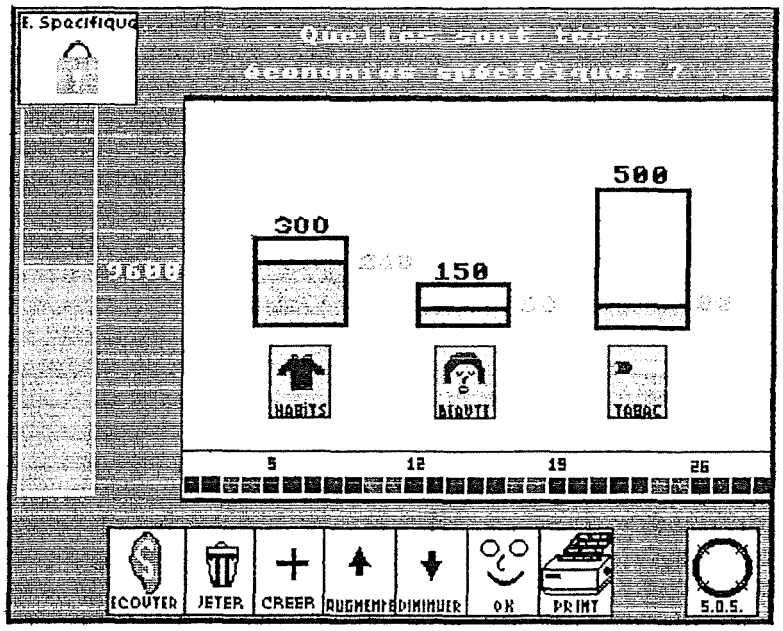
Réponse au test B (27)



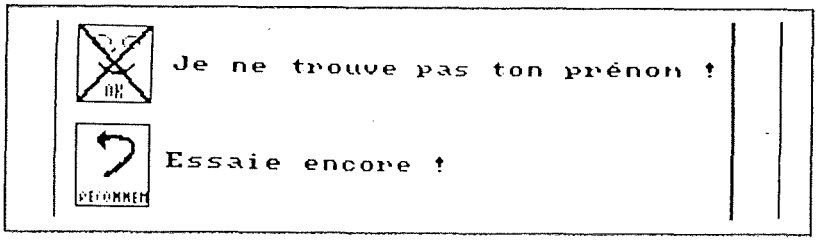
Choix du type d'économie (28)



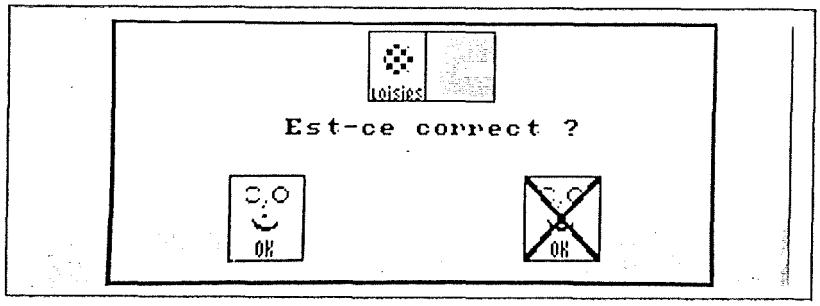
Economies: achats à crédit (29)



Economies: économies spécifiques (30)



Message d'erreur (M1)



Confirmation (CONF1)

1. La saisie du prénom (1-2)

Suivant la valeur d'un certain paramètre, l'utilisateur pourra entrer son prénom, soit à l'aide du clavier, soit à l'aide de la souris.

1.1. La saisie du prénom au moyen du clavier (1)

La question "Quel est ton prénom ?" se présente à l'écran et est prononcée oralement. L'utilisateur répond à l'aide du clavier.

Aucune demande de confirmation n'est exigée.

Lorsque la personne handicapée mentale a entré un prénom mauvais (faute d'orthographe, ...), le message (M1) vient s'afficher et disparaît spontanément après quelques secondes. La personne est de nouveau invitée à entrer son prénom grâce à la question "Quel est ton prénom ?" posée oralement.

1.2. La saisie du prénom à l'aide de la souris (2)

Cette interface a déjà été utilisée dans le logiciel Comptes mais elle a été retravaillée. Elle a été baptisée écritoire et permet à l'utilisateur d'entrer son prénom grâce à la souris.

Comme dans le cas de la saisie du prénom à l'aide du clavier, la question "Quel est ton prénom ?" est posée oralement et sur écran.

Un clavier a été dessiné (selon l'ordre AZERTY) ne reprenant que les 26

lettres de l'alphabet, le "é", le "è", le trait d'union et la barre d'espacement. Chaque caractère est représenté par une icône sur laquelle la personne "clique" pour communiquer son prénom. La touche "return" n'a pas été représentée. Elle a été remplacée par l'icône "OK". Dans le même ordre d'idées, la touche d'effacement (<-) a été remplacée par l'icône "effacer". Une icône "recommencer" est également présente. Lorsque la personne "clique" sur une lettre, celle-ci apparaît sous l'interrogation "Quel est ton prénom ?". Si elle a remarqué qu'elle avait fait une faute d'orthographe, elle peut, soit "cliquer" sur l'icône "effacer" qui effacera le dernier caractère, soit sur l'icône "recommencer" qui reprendra la saisie du prénom au début. Quand la personne a composé son prénom, elle "clique sur "OK".

Ici aussi, si la personne handicapée mentale s'est trompée en entrant son prénom, le message (M1) apparaît et disparaît spontanément après quelques secondes. La personne est de nouveau invitée à entrer son prénom grâce à la question "Quel est ton prénom ?" posée oralement.

2. La saisie des recettes principales (3-4-5)

La saisie des recettes principales est une saisie du montant des recettes que la personne utilisatrice espère recevoir. On peut sélectionner jusqu'à 5 recettes sur la période de gestion.

D'après un paramètre, la saisie peut se faire à l'aide du clavier, de la calculette ou des billets.

Quelques éléments se retrouvent dans ces trois types de dépense. Tout d'abord, la question "Quelles sont tes recettes ?" est posée oralement et sur l'écran. L'icône représentant la prévision des recettes se trouve dans le coin supérieur gauche de l'écran.

L'icône représentant le mois courant se situe dans le coin supérieur droit. La jauge est également présente. Dans la zone de commandes, les icônes "OK" et "recommencer" sont présentes. Seule la zone de travail est modifiée lorsqu'on a affaire à un autre type de saisie.

Chacun de ces trois types de saisie, est illustré par un exemple: Le solde restant de la période passée est égal à 5000 francs et la personne vient de rentrer une recette de 1250 francs.

2.1. Saisie des recettes principales à l'aide du clavier (3)

Il y a cinq cadres rectangulaires plus celui destiné à afficher le total des recettes déjà saisies. Un autre cadre peut se trouver en première position s'il existe un solde de la période précédente. Dans ce cas, ce cadre ainsi que le cadre-total contiendront la valeur du solde. L'icône "OK" est en surbrillance et pour accéder aux autres icônes de commande, la personne utilise les flèches de direction du clavier. Pour activer cette commande, elle doit appuyer sur la touche "return" lorsque l'icône est en surbrillance.

Lorsqu'elle enfonce un chiffre, celui-ci apparaît dans un des cadres. Si elle appuie sur "return", le montant du cadre est validé, il est visualisé sur la jauge, le cadre destiné à afficher la somme des recettes au temps **t** contient la valeur de ce cadre au temps **t-1** plus le montant qui vient d'être validé et enfin le cadre courant devient le suivant. Si la personne sélectionne l'icône "recommencer", tous les cadres se vident et l'utilisateur doit recommencer la saisie.

Si elle a fini ses saisies - il se peut qu'elle ne veuille saisir moins de recettes qu'il n'y a de cadres -, elle sélectionne l'icône "OK" et est confrontée à une

demande de confirmation, qui comprend l'icône "OK" et l'icône "non OK". L'utilisateur sélectionne l'icône "OK", au moyen des flèches du clavier pour passer à l'écran suivant et l'icône "non OK" pour recommencer les saisies.

2.2. Saisie des recettes à l'aide de la calculette (4)

La zone de travail comprend les cinq cadres rectangulaires destinés à recevoir le montant des recettes plus celui destiné à afficher la somme des recettes déjà saisies, plus, éventuellement, celui du solde. Lors de la saisie à l'aide de la calculette, dans la zone de travail, l'icône "effacer" est présente. A côté des cadres, se trouve la calculette: un dessin du clavier correspondant aux touches numériques et un "+" et un "=".

Chaque fois que la personne "clique" sur une des icônes numériques, le chiffre vient s'afficher dans un des cadres, à la manière d'une calculette. Pour valider un montant, la personne doit "cliquer" sur "+". Cela a le même effet que quand on appuie sur "return" lors de la saisie au moyen du clavier. Si on veut effacer le dernier chiffre entré, il suffit de "cliquer" sur l'icône "effacer". Les icônes "recommencer" et "OK" ont le même effet que lors de la saisie par clavier. Par contre, pour les sélectionner, il faut "cliquer" dessus, de même pour la demande de confirmation.

2.3. Saisie des recettes principales à l'aide des billets (5)

La partie gauche de la zone de travail contient différents dessins de pièces et de billets qui sont des objets interactifs sur lesquels l'utilisateur peut "cliquer". Ces pièces et ces billets sont représentés de façon à ce qu'ils ressemblent bien aux pièces et billets belges. Dans la zone de commandes se trouve en plus des deux

icônes "OK" "recommencer" et, l'icône "effacer". Dans la partie inférieure droite, un rectangle affichant "Total: ... francs" comptabilise la somme déjà entrée, plus, éventuellement, le solde.

Lorsque la personne handicapée mentale "clique" sur une pièce ou un billet, une trace de celui-ci apparaît dans la partie droite de l'écran et le total augmente en conséquence. Si la personne clique sur l'icône "effacer" la dernière trace du billet ou de la pièce s'efface et le total diminue de la valeur de cette trace. L'icône "recommencer" a pour effet d'effacer la totalité des traces, de réinitialiser le total à zéro et de recommencer la saisie. La commande "OK" a le même effet que dans la saisie par calculette, ainsi que la demande de confirmation.

3. La prévision des dépenses (6)

L'écran des prévisions de dépenses contient dans le coin supérieur gauche l'icône, orange, des prévisions de dépenses. Dans le cas d'une gestion mensuelle, l'icône du mois se trouve dans le coin supérieur droit. Si la gestion est hebdomadaire, le coin supérieur droit est vide. La jauge contient le total des recettes prévues et la zone des commandes, les icônes: "créer", "dépenser", "jeter" et "OK". La question: "Que prévois-tu comme dépenses ce mois ?" ou "Que prévois-tu comme dépenses cette semaine ?" (selon la période) apparaît à l'écran et est prononcée oralement.

La zone de travail contient, en bas, la ligne du temps du mois courant ou de la semaine courante. Le reste est destiné à recevoir les différentes dépenses prévues pour cette période et leur montants respectifs. Initialement, elle comprend une seule icône: celle représentant l'argent de poche (différence entre le total des recettes et la somme des dépenses prévues; donc, initialement, le montant d'argent de poche est égal au total des recettes).

3.1. Création d'une dépense

3.1.1. *Saisie du poste (7)*

Lorsque la personne désire rajouter une prévision de dépenses, elle "clique" sur l'icône "créer". Cela entraîne une saisie, de la part de la personne, du poste pour lequel cette dépense est prévue.

La question "Quel poste choisis-tu ?" est posée à l'écran et oralement. La zone de travail comprend quarante pictogrammes représentant quarante dépenses possibles. Pour sélectionner un poste, la personne doit "cliquer" sur un pictogramme. Celui-ci clignote et est prononcé oralement. Si la personne s'est trompée de pictogramme, il lui suffit de "cliquer" sur un autre. L'ancien pictogramme est désélectionné et le nouveau clignote. Une fois que le poste est saisi, la personne "clique" sur l'icône "OK" pour passer à la suite.

3.1.2. *Saisie du montant prévu de la dépense (8a-8b-8c)*

Similairement à la saisie des recettes principales plusieurs types de saisie du montant d'une dépense sont envisagés: la saisie à l'aide du clavier et au moyen d'une calculette. La saisie au moyen des billets a, volontairement, été omise car, selon des éducateurs, la personne handicapée mentale a, en général, plus facile d'entrer ses recettes au moyen des billets et ses dépenses, à l'aide de la calculette.

a. *Saisie à l'aide du clavier (8a)*

Dans le coin supérieur gauche, se trouve l'icône des prévisions de dépenses; dans le coin supérieur droit, le pictogramme représentant la dépense dont on doit saisir le montant. Entre les deux se trouve la question "Combien penses-tu dépenser pour cette dépense ?"; celle-ci étant aussi posée oralement. La personne utilise le clavier pour entrer le montant de la dépense. Quand elle a fini, elle active l'icône "OK" (cfr la saisie des recettes principales.

b. *Saisie au moyen de la calculette (8b)*

La partie supérieure de cet écran est exactement la même que celle de la saisie au moyen du clavier. La zone de commandes contient les icônes "effacer", "recommencer" et "OK". La zone de travail comprend un seul cadre, destiné à recevoir le montant de la dépense, saisi grâce à la calculette, qui se trouve en dessous de ce cadre. Le principe de fonctionnement de la calculette et des icônes-commandes est le même que lors de la saisie des recettes principales, sauf que, maintenant, la touche "+" de la calculette n'a plus d'utilité. On aurait très bien pu ne pas le mettre mais cela aurait, peut-être, apporté la confusion chez les utilisateurs.

3.1.3. *Saisie de la périodicité de la dépense (9-10)*

Après avoir saisi le poste puis le montant de la dépense, il s'agit, maintenant, de savoir si la dépense se fera régulièrement. Dans ce cas, il nous faudra connaître la fréquence à laquelle s'effectuera la dépense. Cette

fréquence sera automatiquement sauvée par l'ordinateur. De cette façon, la personne qui utilisera le logiciel Budget aura sous les yeux la liste des dépenses qu'elle devrait faire. Le fait de créer de nouvelles prévisions peut sembler lourd pour la personne mais est fait une fois pour toutes.

a. *Saisie Régulière - Occasionnelle (9)*

Dans le coin supérieur gauche de cet écran, se trouve l'icône des prévisions de dépenses; dans le coin droit, le pictogramme représentant la dépense; entre les deux, la question "Cette dépense est-elle Régulière ou occasionnelle ?" cette question est également prononcée. La zone de commandes a l'icône "OK". La zone de travail contient les deux lettres (en grand) "R" et "O" signifiant respectivement Régulière et Occasionnelle. La personne doit alors "cliquer" sur une de ces lettres. Ceci fait, la lettre sélectionnée clignote. L'icône "OK" sert à valider ce choix.

La lettre "O" signifie que la dépense se fera à l'occasion. On n'a donc pas besoin de saisir la fréquence puisqu'il n'y en a pas. Par contre, pour une dépense régulière il est nécessaire de savoir la fréquence à laquelle se fera cette dépense, ainsi que le mois de départ.

b. *Saisie de la fréquence (10) (dépense régulière)*

Les icônes hors de la zone de travail sont les mêmes que pour la saisie Régulière ou occasionnelle. La question "A quelle fréquence fais-tu cette dépense ?" est inscrite et prononcée. La partie supérieure de la zone de travail sert à saisir la fréquence et celle du bas, le mois à partir duquel cette dépense se fera.

En ce qui concerne la fréquence, la personne a le choix entre toutes les semaines, tous les mois, tous les 2,3,6 mois et tous les ans. L'icône du mois courant est présente. Pour sélectionner un autre mois de départ, la personne "clique" sur cette icône et le mois suivant s'affiche. Pour valider la fréquence et le mois de départ, une fois qu'ils ont été sélectionnés, la personne "clique" sur l'icône "OK".

3.2. Le fait d'effectuer une dépense

Pour effectuer une dépense qu'elle avait prévu, la personne doit "cliquer" sur le pictogramme représentant la dépense puis sur l'icône "dépenser". Si la dépense est régulière et s'exécute toutes les semaines, alors on l'effectue pour le montant d'une semaine, sinon le montant affiché à côté du pictogramme est pris comme la somme prévue pour la dépense. Cela a pour effet une saisie pour voir si le montant prévu de la dépense est égal au montant réel de la dépense (8a-8b-8c).

Cette saisie apparaît sous forme de confirmation (Conf 1) dans laquelle se trouvent le pictogramme de la dépense et son montant. La question "Est-ce correct ?" apparaît et est prononcée et la personne a le choix entre l'icône "OK", si la réponse est oui et l'icône "non OK", dans le cas contraire.

Si la personne choisit "non OK", elle est invitée à entrer sa somme réelle par une saisie de somme.

Dans les deux cas, la personne retourne à l'écran des prévisions de dépenses.

3.3. L'option "jeter"

Si la personne a prévu régulièrement une dépense et que celle-ci ne peut se faire pour une raison ou une autre, alors, elle a la possibilité de la "jeter", c'est-à-dire de l'annuler provisoirement. Lors d'une autre période, cette dépense sera encore prévue.

Pour ce faire, elle a la possibilité de l'annuler soit une fois durant la période, soit durant la période entière. Pour la "jeter", il suffit de la sélectionner et, ensuite, de "cliquer" sur l'icône "jeter". Pour la sélectionner, il faut "cliquer" sur la partie pictogramme de la dépense, si on veut l'annuler une seule fois, et sur la partie chiffres, pour l'annuler durant la période entière. Cela implique que le montant d'argent réservé à cette dépense sera libre et ainsi, l'argent de poche augmentera.

Une fois la dépense annulée, la personne revient à l'écran des prévisions de dépenses avec l'argent de poche ajusté et, éventuellement, une dépense en moins.

3.4. Visualisation des prévisions de dépenses (11)

Lorsque la personne désire visualiser ces prévisions de dépenses, elle "clique" dans la jauge. Elle arrive à l'écran 11 intitulé "Représentation des prévisions de dépense." Il est à remarquer que l'arrière-fond est gris. Chaque prévision de dépenses est représentée par son pictogramme. En dessous de celui-ci est indiquée la somme en chiffres et une jauge (en orange). Les prévisions sont triées en ordre croissant. L'écran comprend seulement dix prévisions plus une représentation du liquide (en vert). S'il y a plus de dix prévisions, l'utilisateur clique sur l'icône-OK et visualise les suivantes. Il clique ensuite sur l'icône-OK pour revenir à l'écran 6.

4. Le menu principal (13)

Une fois la saisie du prénom, des recettes principales et des prévisions de dépense effectuées, l'utilisateur arrive au menu principal duquel il peut choisir une des huit activités: Recettes, Dépenses, Economies, Prévisions de recettes, Prévisions de dépenses, Test, Etat des finances et Bilan.

L'icône-commande "Quitter" sert à sortir du programme. Le titre "Quelle activité choisis-tu ?" est posé par écrit et oralement. Pour choisir une activité, la personne doit "cliquer" sur l'icône représentant cette activité: on a le choix entre effectuer une recette, une dépense, accéder aux économies, aux prévissions de recettes, de dépenses, au test, au bilan et à l'état des finances.

5. Effectuer une recette (7 - 8 - 14)

Le fait d'effectuer une recette, qui n'a pas été, au préalable, prévue, se déroule en trois étapes:

- ★ La saisie du poste de la recette (7). Le même écran que lors du choix du poste d'une dépense prévue est présenté à la personne; excepté le cadre des pictogrammes qui est en vert.

- ★ La saisie du montant de la recette est une des 3 saisies de somme (8a-8b-8c): au moyen du clavier, de la calculette ou des billets. L'écran est le même sauf le titre qui devient: "A combien s'élève cette recette ?".

- ★ La saisie du moment de la recette (14). La question "Quand as-tu fait cette recette ?" apparaît à l'écran. Les représentations des mois passé et actuel sont présentes

et la personne "clique" sur le jour de la recette. Elle confirme en "cliquant" sur l'icône "OK" et revient au menu principal.

6. Effectuer une dépense (7 - 8 - 14)

L'activité "effectuer une dépense" a pour but de faire une dépense qui n'a pas encore été prévue. Elle se déroule en trois étapes:

- ★ La saisie du poste de la dépense (7). Le même écran que lors du choix du poste d'un dépense prévue est présenté à la personne; excepté le cadre des pictogrammes qui est en rouge.

- ★ La saisie du montant de la dépense est une des 3 saisies de somme (8a-8b-8c): au moyen du clavier, de la calculette ou des billets. L'écran est le même sauf le titre qui devient: "A combien s'élève cette dépense ?".

- ★ La saisie du moment de la dépense (14). La question "Quand as-tu fait cette dépense ?" apparaît à l'écran. Les représentations du mois passé et actuel sont présentes et la personne "clique" sur le jour de la dépense. Elle confirme en "cliquant" sur l'icône "OK" et revient au menu principal.

7. Les économies (28 et suivants)

Lorsqu'on choisit les économies du menu principal, on arrive à un sous menu (28) intitulé "Quelles sont tes économies ?". On a, ici, le choix entre les achats à crédit, les économies spécifiques et les transferts.

7.1. les achats à crédit (29)

Les achats à crédit (ou par correspondance) sont des achats dont on n'a pas encore payé le montant. On ne sait pas exactement quand il faudra payer mais on sait qu'il faudra payer d'ici peu. Les achats à crédit sont une sorte d'économie obligatoire. De temps en temps, on alimentera ces comptes et quand le montant voulu sera atteint, on pourra les décaisser.

Pour créer un compte, la personne "clique" sur l'icône "créer". Deux fonctionnalités sont proposées:

★ la saisie du poste de cet achat (7)
★ une saisie de somme pour connaître le montant à atteindre (8a-8b-8c)

Pour alimenter un achat, il suffit de "cliquer" sur l'icône "augmenter" après avoir sélectionné l'achat. Une saisie de somme est présentée à l'utilisateur pour connaître le montant qu'il faut additionner.

7.2. Economies spécifiques (ES) (30)

Les économies spécifiques sont des économies destinées à des achats précis. Supposons, par exemple, qu'une personne veuille acheter une chaîne Hi-Fi. Il faudra qu'elle mette chaque mois une certaine somme d'argent de côté. Contrairement aux achats à crédit, on peut ne plus mettre chaque mois un peu d'argent de côté. Si la personne ne veut plus de chaîne Hi-Fi, elle peut la retirer de ses économies spécifiques et, ainsi, récupérer son argent mis de côté.

Pour créer une économie spécifique, la personne "clique" sur l'icône "créer".

Deux fonctionnalités sont proposées: la saisie du poste de cette ES (7) et une saisie de somme pour connaître le montant à atteindre (8a-8b-8c).

Pour alimenter une économie spécifique, il suffit de "cliquer" sur l'icône "augmenter" après avoir sélectionné l'ES. Une saisie de somme est présentée à l'utilisateur pour connaître le montant qu'il faut additionner (8a-8b-8c).

Pour annuler une ES, il faut "cliquer" sur l'icône "jeter" après avoir sélectionné l'économie spécifique. Le montant d'argent réservé à cette ES est libéré.

Pour décrémenter une économie spécifique, il suffit de "cliquer" sur l'icône "diminuer" après avoir sélectionné l'ES. Une saisie de somme est présentée à l'utilisateur pour connaître le montant qu'il faut soustraire(8a-8b-8c).

7.3. Transferts (15 - 16)

On peut transférer une somme d'argent soit:

- entre le liquide et les comptes en banque,
- entre les comptes et les ES,
- entre les comptes et les achats à crédit.

(ex: pour transférer de l'argent du liquide vers le compte CGER, écrans 15 & 16)

A la question "De quel compte ?", on clique sur un des comptes listés pour sélectionner l'origine. Vient alors la seconde question "Vers quel compte ?" et on sélectionne alors un des comptes listés comme destination.

L'origine et la destination étant sélectionnées, il reste à saisir les deux montants à transmettre grâce à deux saisies de somme (8a-8b-8c).

8. La prévision des dépenses (6 et suivants)

La prévision des dépenses a déjà été expliquée précédemment (6.3.). Il y a, ici, sept écrans de prévisions de dépenses. Le premier est le même que dans le point 6.3. Les autres regroupent les prévisions de dépenses régulières par fréquence. Par exemple, dans le deuxième écran, sont rassemblées les prévisions de dépenses qui se font toutes les semaines, dans le troisième, tous les mois et ainsi de suite jusque tous les ans.

Ces écrans sont les mêmes, exceptés que ceux regroupés par fréquence, ont l'icône représentant la fréquence dans le coin supérieur droit.

Pour passer d'un écran à l'autre, l'utilisateur "clique" sur l'icône-OK. Lorsque cette icône est activée dans le dernier écran (tous les ans) on revient au menu principal.

9. La prévision des recettes

Les fonctionnalités des prévisions de recettes sont les mêmes que celles des prévisions de dépenses.

10. Test (24)

Deux questions nous paraissent fondamentales pour les personnes handicapées mentales. Le test tente d'y répondre.

A: "*Quand pourra-t-on faire une dépense de X francs, si on épargne une certaine somme chaque mois ?*"

B: "*Combien faut-il épargner par mois, si on veut faire une dépense de X francs dans Y mois ?*"

Si on choisit la première question, on doit d'abord faire une saisie du montant de la dépense à simuler (8a-8b-8c) puis, ensuite, on refait une saisie de somme (8a-8b-8c) pour connaître le montant à épargner par mois.

On obtient, par exemple, la réponse (26) ... le rectangle représentant la dépense se remplit progressivement par tranches de montant à épargner par mois. Une fois le rectangle complètement rempli, on obtient, par exemple, l'écran 26'.

Si on choisit la deuxième question, on doit d'abord faire une saisie du montant de la dépense à simuler (8a-8b-8c) puis, ensuite, on refait une saisie du mois (25).

La réponse de l'ordinateur sera par exemple, l'écran 27.

11. Le bilan (20-21-22-23)

Le bilan est un résumé des dépenses effectuées durant une semaine, un mois ou une année. Ces dépenses sont classées par postes. L'écran du choix du bilan est l'écran 20. Les trois écrans (21-22-23) ont le même aspect. Seul le titre change.

Pour un bilan du mois, le titre sera "Bilan du mois :" et l'icône du mois en cours sera présente à côté de ce titre. Les différentes dépenses effectuées durant ce mois sont regroupées en 11 postes maximum. En dessous de chaque pictogramme représentant ce poste, se trouve le montant dépensé pour ce poste, en chiffres et sous forme de jauge (rouge). Au bas de l'écran, le total des dépenses du mois est calculé.

12. L'état des finances (17-18-19)

L'état des finances a été créé pour pouvoir visualiser, à tout instant, le montant de tous ses avoirs. C'est, en quelque sorte, une photo de la répartition de l'argent dans différents postes. Trois postes ont été dégagés: les différents comptes et le liquide (17-17'), les économies spécifiques (18) et les achats à crédit (19).

Il est à remarquer que comme pour le bilan, la zone de travail a un arrière-fond gris. Cela s'explique par le fait que ces écrans sont une visualisation. Aucune intervention de la part de l'utilisateur n'est demandée.

L'écran 17 sert à visualiser les différents comptes. Le titre "Etat des finances" se trouve à la droite de l'icône représentant l'état des finances. Au centre de l'écran se trouvent les différents comptes. Chaque compte est représenté par une jauge et la valeur qu'elle représente est indiquée à sa droite.

L'écran 19 ne sert qu'à visualiser les achats à crédit et donc a le fond gris. Le titre "Achats à crédit" se trouve à la droite de l'icône représentant les achats à crédit. Au centre de l'écran se trouvent les différents achats. Le montant d'argent à atteindre est écrit au dessus et le montant mis de côté est indiqué en dessous. Pour passer à l'écran suivant, la personne "clique" sur l'icône "OK" .

L'écran 18 ne sert qu'à visualiser les économies spécifiques et donc a le fond gris. Le titre "Economies spécifiques" se trouve à la droite de l'icône représentant les ES. Au centre de l'écran se trouvent les différentes ES. Le montant d'argent à atteindre est écrit au-dessus et le montant mis de côté est indiqué en dessous dans la jauge.

13. L'aide (12)

A chaque écran, se trouve une icône-SOS qui, une fois activée, explique chaque commande de l'écran en cours.

★ ★ ★
★ ★

CONCLUSIONS

Le développement d'un projet informatique pour personnes handicapées mentales recouvre plusieurs disciplines. Comme les personnes handicapées mentales expriment difficilement leurs besoins, ceux-ci sont exprimés par des acteurs intermédiaires: des psychologues, des éducateurs, des animateurs, ... Le projet a été développé, dans notre cas, grâce à une collaboration entre ces acteurs intermédiaires et des spécialistes de l'informatique.

Ce logiciel répond à une demande des institutions. Bien qu'elles disposent déjà d'un moyen pour gérer le budget des personnes handicapées mentales, elles sont tentées par une version informatisée. Cette version devraient leur apporter des facilités ainsi qu'aux personnes concernées.

Nous nous permettons d'insister sur le rôle de l'interface dans le développement d'une application interactive avec des personnes handicapées mentales. Les représentations sous formes d'icônes nous paraissent très importantes, surtout pour des personnes ne sachant pas lire.

Dans le laps de temps nous ayant été attribué, nous n'avons implémenté une partie seulement, du logiciel de gestion de budget. Ce sous-système comprend la possibilité d'entrer ses recettes, ses dépenses, ses prévisions de dépenses, et de simuler une dépense.

Nous avons conçu l'interface des autres sous-systèmes; celle-ci a été revue et améliorée par les éducateurs et les psychologues.

Un petit travail de programmation reste donc à faire. Nous espérons que ce travail sera bientôt réalisé, notamment par des étudiants, dans le cadre d'un mémoire. Afin que le logiciel soit meilleur, il serait souhaitable et même indispensable de le faire tester par des personnes handicapées mentales et, éventuellement, de le modifier.

Plusieurs logiciels, notamment le logiciel "**Courses**"[8], ont été réalisés pour des personnes handicapées mentales. Il nous semble indispensable que les interfaces de ces logiciels soient très homogènes, pour rendre plus facile le passage d'un logiciel à l'autre. Afin d'obtenir un meilleur résultat, il vaudrait mieux, selon nous, évaluer les différents logiciels auprès de personnes handicapées mentales, et ensuite d'adapter tous les logiciels selon cette évaluation. Les logiciels, éventuellement conçus dans le futur, devraient tenir compte de cette évaluation.

★ ★ ★
★ ★

[8] Le logiciel *Courses* a été réalisé l'année dernière par Léopold Topet et Xavier Vônèche, étudiants en informatique, dans le cadre leur mémoire. Ce logiciel est destiné à des personnes handicapées mentale. La personne constitue des repas pendant une période qu'elle définit, et l'ordinateur établit une liste d'achats, qui reprend les produits alimentaires à acheter et les quantités à acheter de ces produits pour la préparation de ces repas.

BIBLIOGRAPHIE

★ Davin J., Delvin E., Le Polain De Waroux V., *Une vie à vivre avec les personnes handicapées mentales*, ed du Centurion, Paris, 1989.

★ Déplechin Marc et Strappazzon Gianni, *Un logiciel de gestion des comptes pour personnes handicapées mentales: développement et évaluation*, Mémoire présenté en vue de l'obtention du titre de Licencié et maître en informatique, F. N. D. P, Septembre 1989.

★ Droz. R., Rahmy M., *Lire Piaget*, Ed. Mardaga, coll. Psychologie et sciences humaines, Bruxelles, 1972.

★ Fraiture M., Machgeels C., *Le cahier des charges d'un logiciel pour personnes handicapées*, juillet 1990.

★ France Cheron et Olivier Berger, *Evaluation et amélioration d'un logiciel de gestion des comptes pour personnes handicapées mentales*, Mémoire présenté en vue de l'obtention du titre de Licencié et Maître en informatique, F.N.D.P., Septembre 1990.

★ Gleaves Richard, *Modula-2 for Pascal Programmers*, Springer Verlag, 1984.

★ Inhelder B, Piaget J, Oléron P, Gréco P, "*L'intelligence*", Presses Universitaires de France, 1980.

★ Kaufmann Arnold, *Le traitement mathématique de l'imprécis par la théorie des sous-ensembles flous*, Note de travail n°102, Mars 1983.

★ Lambert Jean-Luc, *Introduction à l'arriération mentale*, 2° ed, Mardaga, Bruxelles, ????

★ Le Cardinal Gilles, "*Handicap et communication*" in "*L'homme communique comme unique - modèle systémique de la communication interpersonnelle finalisée*", Compiègne, Mai 1989.

★ Le Cardinal G., Legrand C., "*Prévention du surhandicap de communication dû à une carence verbale et/ou écrite*", Compiègne, 1986.

★ Mercier M., "*Psychologie, psychologie cognitive et intelligence artificielle*", in "Journal de Réflexion sur l'Informatique", Namur, Mars 1988.

★ Mercier M., Billat E., "*Psychologie et aide informatisée au service de la personne avec un handicap*", in "Journal de Réflexion sur l'Informatique", Namur, Octobre 1987.

★ Mercier M., Delville J., "*Quand l'ordinateur est un outil d'adaptation à la vie quotidienne*", in "Journal de Réflexion sur l'Informatique", Namur, 1989.

★ Mercier M., Witdouck O., "*La créativité sur ordinateur pour la personne handicapée mentale*", in "Journal de Réflexion sur l'Informatique", Namur, Juin 1991.

★ Mercier M, Witdouck O, PSINHA, "*Les nouvelles technologies au service des personnes*

handicapées", Namur.

★ *Modula-2 Software Construction Set*, Version Amiga 01.00, Commodore Business Machines, Inc.

★ Niklaus Wirth, *Programming in Modula-2*, 2° ed, 1983.

★ Paour J.L., "*Apprentissage de notion de conservation et induction de la pensée opératoire concrète chez les débiles mentaux.*", in René Zazzo, Les débilités mentales, Paris, 1979.

★ ★ ★
★ ★

Année académique 1991-1992

Elaboration d'un logiciel de gestion de Budget pour personnes handicapées mentales

ANNEXES

Jean-Marc CULOT

Promoteur: **Madame Monique Noirhomme-Fraiture**

Co-promoteur: **Monsieur Michel Mercier,** professeur au Département de Psychologie de la faculté de Médecine, F.U.N.D.P.

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître
en Informatique.

CONTENU

★ Annexe 1: Mode d'emploi

★ Annexe 2: Quelques exemples de traces d'utilisation

★ Annexe 3: Listing des "implementation" modules

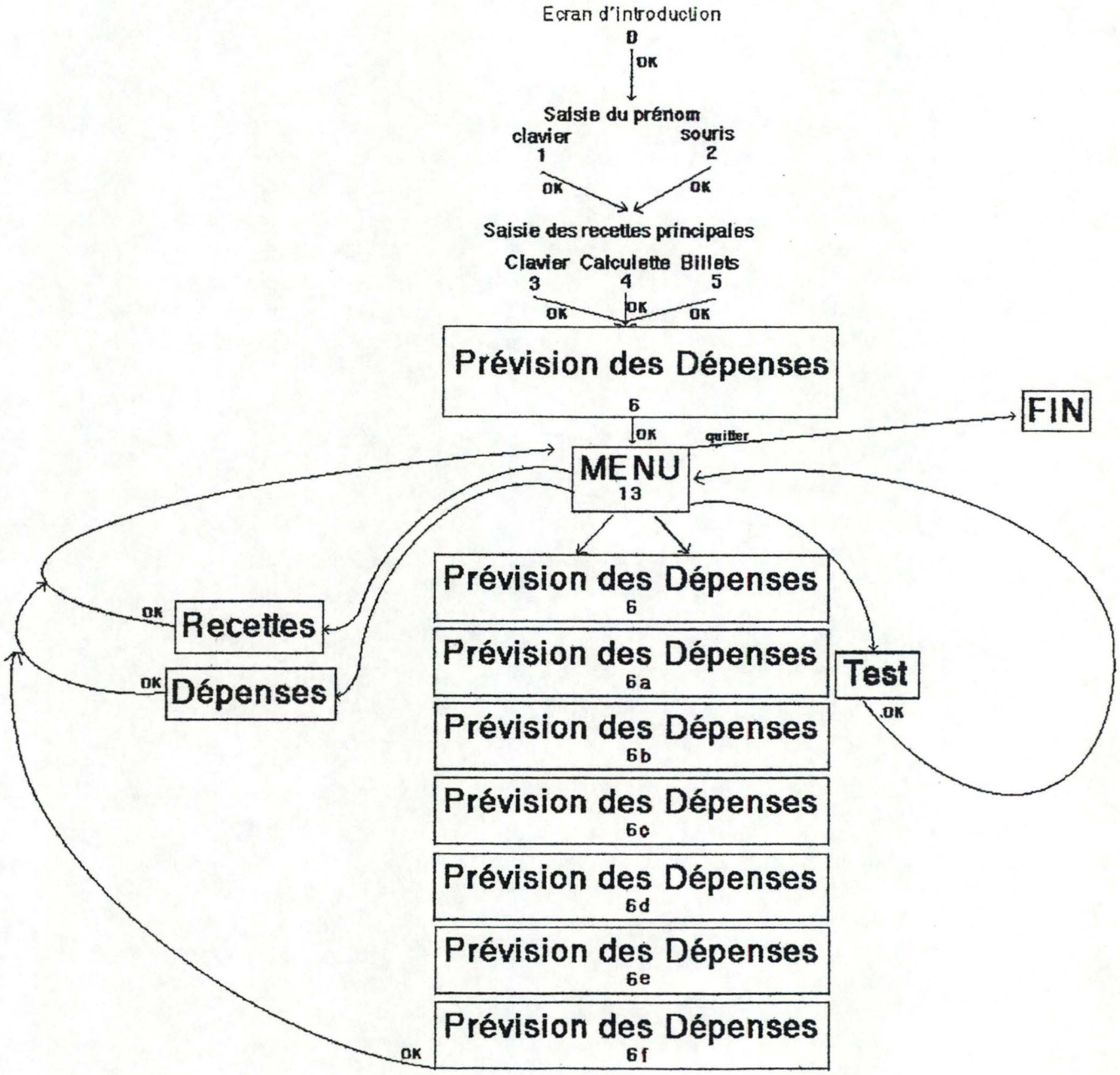
★ Annexe 4: Listing des "definition" modules

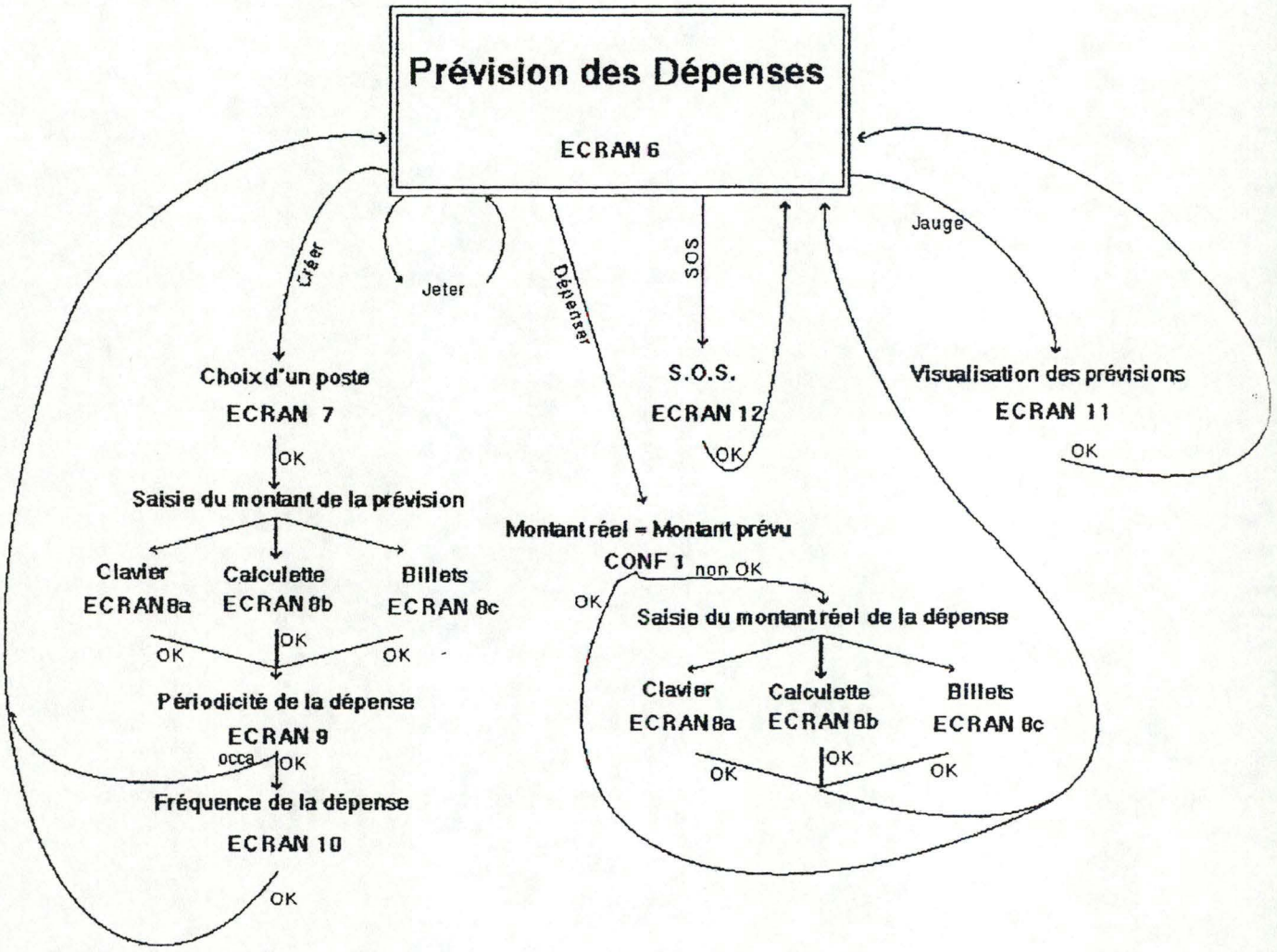
★ Annexe 5: Listing des programmes Paramètres & Traces

Mode d'emploi

Logiciel Budget

Gestion mensuelle




Prévision des Dépenses
ECRAN 6
Créer
Jeter
Dépenser
SOS
Jauge
Choix d'un poste
ECRAN 7
OK
Saisie du montant de la prévision
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK
Périodicité de la dépense
ECRAN 9
occa
OK
Fréquence de la dépense
ECRAN 10
OK
S.O.S.
ECRAN 12
OK
Montant réel = Montant prévu
CONF 1
non OK
OK
Saisie du montant réel de la dépense
Clavier
ECRAN 8a
Calculette
ECRAN 8b
Billets
ECRAN 8c
OK
OK
OK
Visualisation des prévisions
ECRAN 11
OK

Mode d'utilisation

1. Mode d'emploi "technique"

Cette partie s'adresse exclusivement aux éducateurs. Elle a pour but de leur fournir la suite des opérations qu'il est nécessaire d'effectuer avant d'utiliser le logiciel *Budget*. Les manipulations décrites ci-dessous devraient les aider à préparer les disquettes *param:* qui sont propres à chaque personne handicapée mentale et à lancer correctement le programme *Budget*.

1.1. Aux utilisateurs de l'Amiga 500

Votre ordinateur devra impérativement être muni de deux lecteurs de disquettes: un lecteur interne nommé DF0: et un lecteur externe nommé DF1:. De plus, il faut que la mémoire vive fasse au moins un MégaByte.

a) *Préparation des disquettes Param:*

Pour chaque personne handicapée mentale, il est nécessaire de préparer une disquette intitulée *Param:* avant que celle-ci n'utilise le logiciel pour la première fois. Voici la liste des opérations à effectuer pour cette préparation:

- Mettre l'ordinateur sous tension
- A la demande de l'ordinateur, introduire une disquette *Workbench* en DF0: (lecteur interne)
- Insérer une disquette vierge en DF1: (lecteur externe)
- Formater cette disquette, c'est-à-dire:

★ "cliquer gauche" une seule fois sur l'icône représentant la disquette vierge (DF1:BAD)

★ "cliquer droit" pour obtenir la barre de menu

★ Choisir dans le menu DISK l'option INITIALISE

★ une fois la disquette formatée (45 sec), il faut la renommer. Pour ce faire, il faut:

* "cliquer gauche" sur l'icône représentant la disquette vierge et formatée (EMPTY)
* "cliquer droit" pour obtenir la barre de menu
* choisir dans le menu WORKBENCH l'option RENAME
* remplacer le nom EMPTY apparaissant par PARAM

★ Retirer la disquette qui s'appelle à présent PARAM du lecteur DF1: Pour éviter tout risque de confusion, nous vous conseillons d'écrire sur l'étiquette: "Param" suivi du nom de la personne handicapée mentale concernée.

b) *Définition des paramètres*

Cette opération doit se faire impérativement avant que la personne handicapée mentale n'utilise le logiciel pour la première fois. Mais elle peut également être entreprise entre deux sessions d'utilisation du programme si l'éducateur désire modifier l'une ou l'autre des valeurs affectées précédemment aux paramètres.

Dans les deux cas, suivez la procédure suivante:

★ Introduire la disquette TOOLS en df0: et la disquette PARAM en DF1:

★ Double clic gauche sur l'icône de la disquette TOOLS

★ Double clic gauche sur l'icône du fichier PARAMETRES

★ Fixer la valeur des paramètres de travail en fonction de la personne handicapée mentale pour laquelle ils sont destinés

★ Une fois les paramètres écrits sur la disquette PARAM, retirer les 2 disquettes

c) *Lancement du programme Budget*

- Insérer les disquettes *Budget* en DF0: et *Param* en DF1:
- Allumer l'ordinateur. Après quelques secondes, l'écran de présentation s'affiche.

Lorsque l'on "clique gauche", le programme débute après quelques instants de patience.

VEILLEZ A LAISSER LA DISQUETTE "PARAM" EN DF1: JUSQU'A LA COMPLETE TERMINAISON DU PROGRAMME.

1.2. Aux utilisateurs d'un Amiga 2000 sans disque dur

Le lecteur de disquettes externe s'appelle, ici, DF2: à la place de DF1: Les trois procédures décrites pour l'Amiga 500 sont les mêmes sauf que le DF1: devient DF2:

1.3. Aux utilisateurs d'un Amiga 2000 avec disque dur

a) *Préparation du disque dur*

Cette opération doit être réalisée une fois pour toutes. Elle permet l'installation du programme *Budget* sur le disque dur.

★ allumer l'ordinateur et charger le système

★ Insérer la disquette TOOLS en DF0: et BUDGET en DF2:

★ Double clic gauche sur TOOLS

★ Double clic gauche sur DISKMASTER

Une fois entré dans le programme DiskMaster, cliquer sur FH1: (partie centrale). La liste des fichiers se trouvant dans FH1: est affichée dans la partie gauche de l'écran.

Il serait préférable de créer un répertoire pour y installer le programme *Budget* avec tous ses fichiers. Pour cela, il suffit de cliquer sur MKDIR (au centre en bas) et d'entrer le nom "Budget".

Il reste, maintenant, à copier les fichiers de *Budget* sur disque dur. Double "cliquer-gauche" sur "Budget" dans le cadre gauche de l'écran. Cliquer ensuite dans le cadre droit puis sur DF2: (partie centrale). Cliquer ensuite sur "all" (partie centrale) puis sur "copy" (partie centrale).

Après environ une minute de copiage, il faut encore copier les fichiers propres au programme sur disque dur. Cliquer sur DF0: (partie centrale), puis sur "disque dur" (à droite) ensuite sur "all" puis "copy".

Le Budget étant copié, sortir du DISKMASTER en cliquant en haut à gauche de l'écran pour fermer la fenêtre.

★ Double clic sur FH1: ou Work

★ Double clic sur Budget

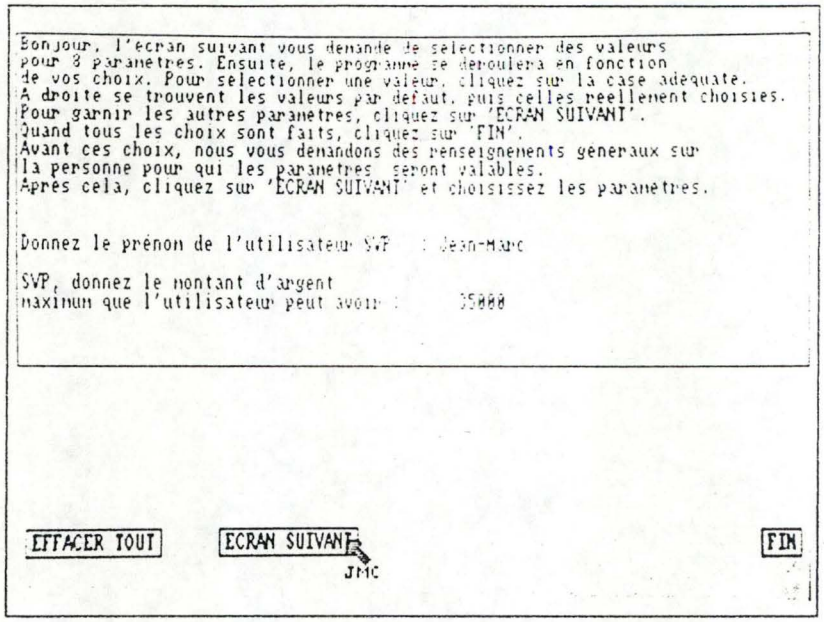
★ Tirer le fichier "Paramètres" de TOOLS dans la fenêtre Budget (à côté du programme Budget)

★ Tirer le fichier "Traces" de TOOLS dans la fenêtre Budget (à côté des programmes Budget et Paramètres)

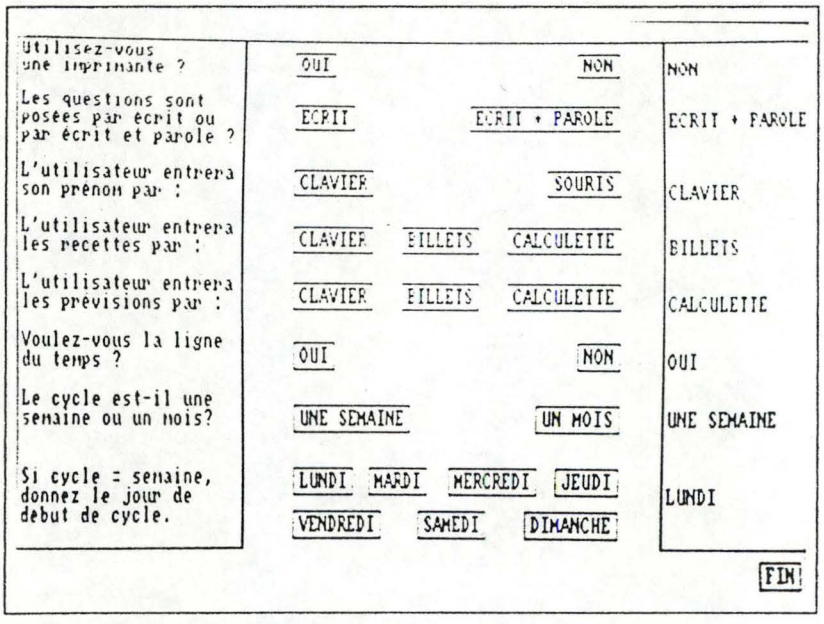
Maintenant, les disquettes *Budget* et *Tools* ne sont plus indispensables; par contre, les disquettes *Param* le sont. La procédure de préparation est la même que celle décrite ci-dessus. Pour définir les paramètres et exécuter le programme, introduire la disquette *Param* en DF0: et double cliquer, respectivement sur les icônes *Paramètres* et *Budget*

2. Le logiciel *Paramètres*

Le but principal de ce logiciel est de permettre à l'éducateur d'initialiser des paramètres pour rendre le programme de gestion de l'interaction le mieux adapté aux capacités de chaque utilisateur. Vu que le nombre des paramètres n'est pas négligeable, il est nécessaire d'aider l'éducateur dans sa tâche de préparation par un logiciel adéquat:



Logiciel Paramètres: Ecran 1

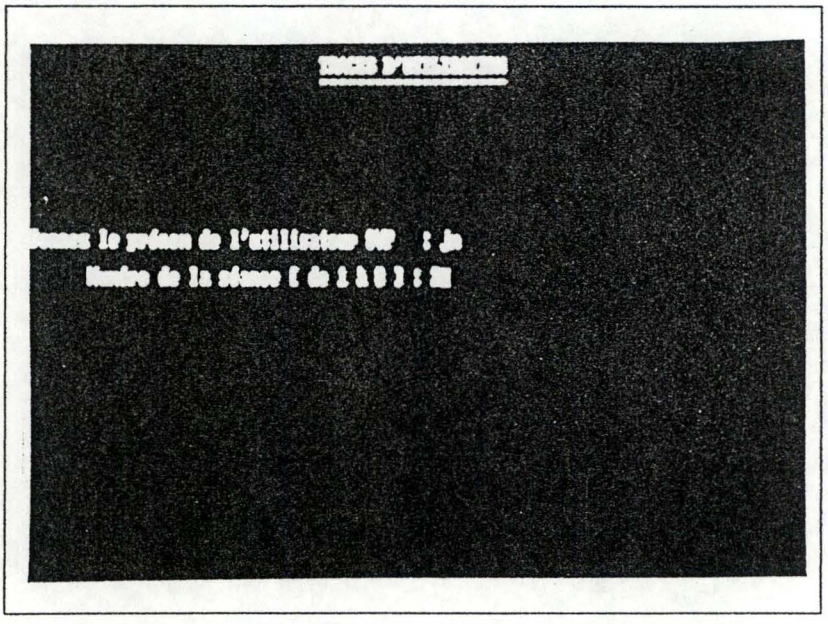


Logiciel Paramètres: Ecran 2

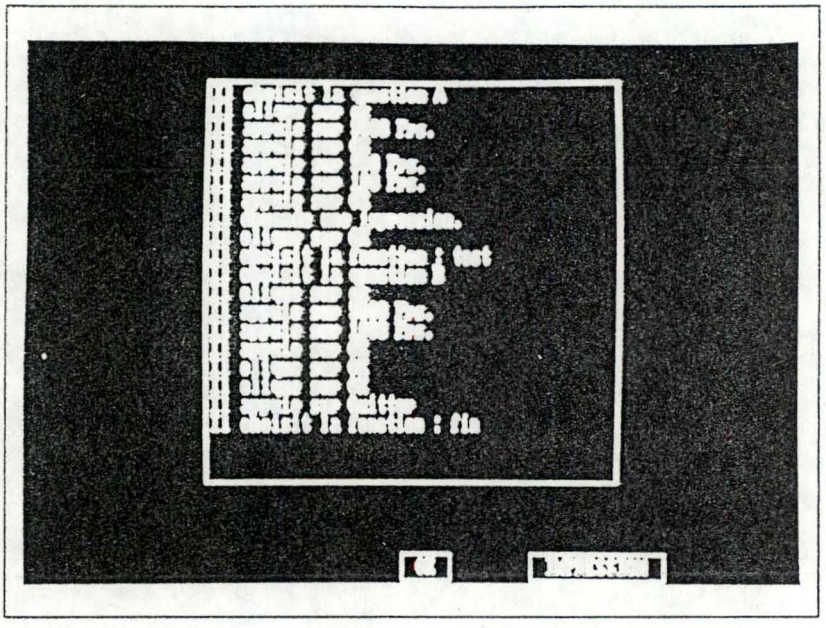
3. Le logiciel *Traces*

Une trace d'utilisation est une interaction entre la personne handicapée mentale et le logiciel *Budget*. A chaque utilisation du logiciel *Budget*, les traces d'utilisations de la personne sont enregistrées. Le logiciel *Traces* permet à l'éducateur de voir toutes les traces d'un utilisateur et de les imprimer.

L'éducateur doit d'abord entrer le prénom de la personne handicapée mentale; ensuite il rentre le numéro de la séance et voit apparaître les différentes traces. Pour les imprimer, il clique sur "impression" (coin inférieur droit).

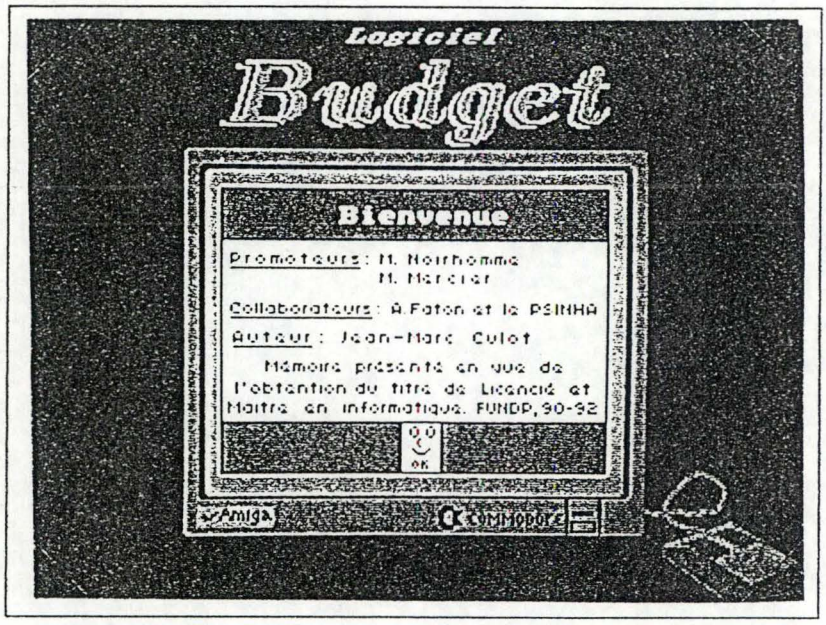


Logiciel Traces: Ecran 1



Logiciel Traces: Ecran 2

4. Logiciel *Budget*

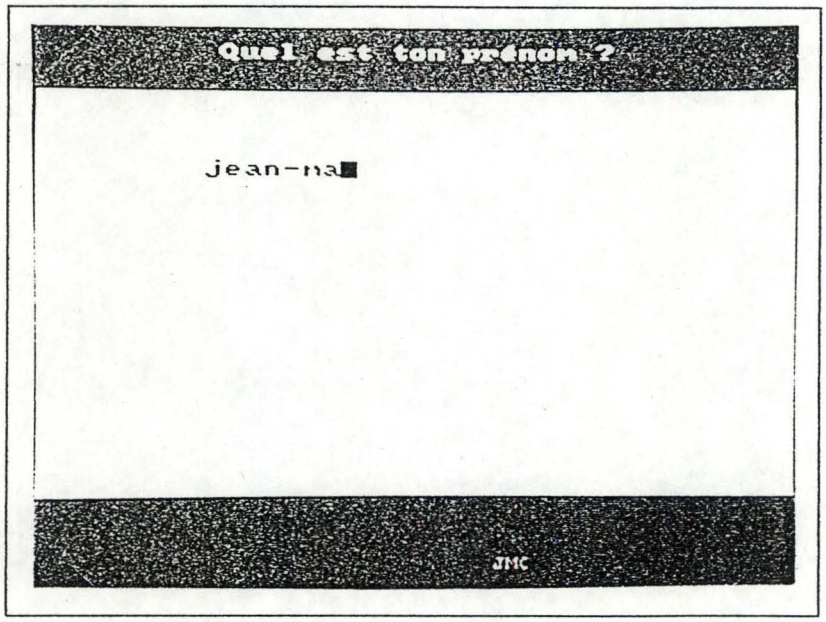


Ecran de présentation

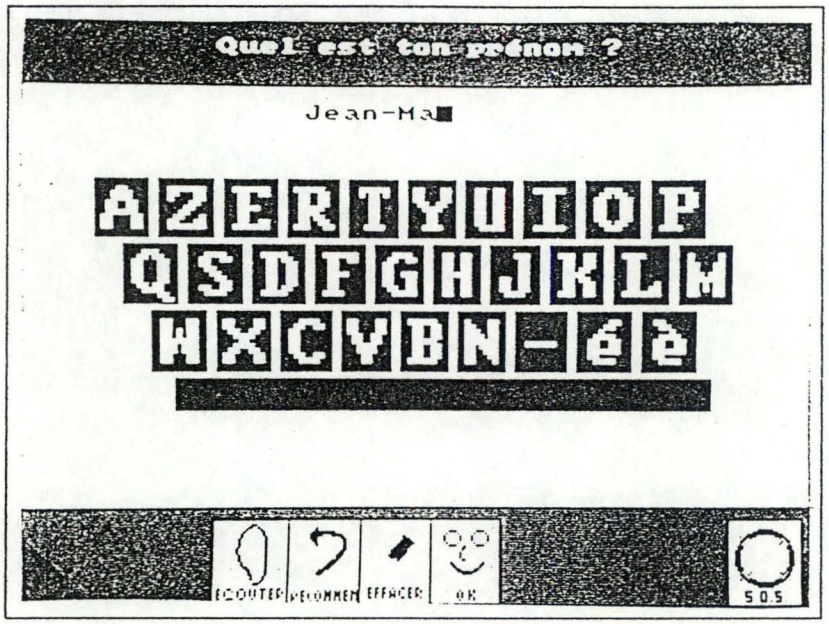
4.1. La saisie du prénom (1-2)

Suivant la valeur d'un certain paramètre, l'utilisateur pourra entrer son prénom soit, à l'aide du clavier, soit à l'aide de la souris. Le fait de connaître le prénom de l'utilisateur permettra à l'ordinateur d'aller récupérer sur la disquette les paramètres correspondant à ce prénom.

4.1.1. La saisie du prénom au moyen du clavier (1)



4.1.2. La saisie du prénom à l'aide de la souris (2)

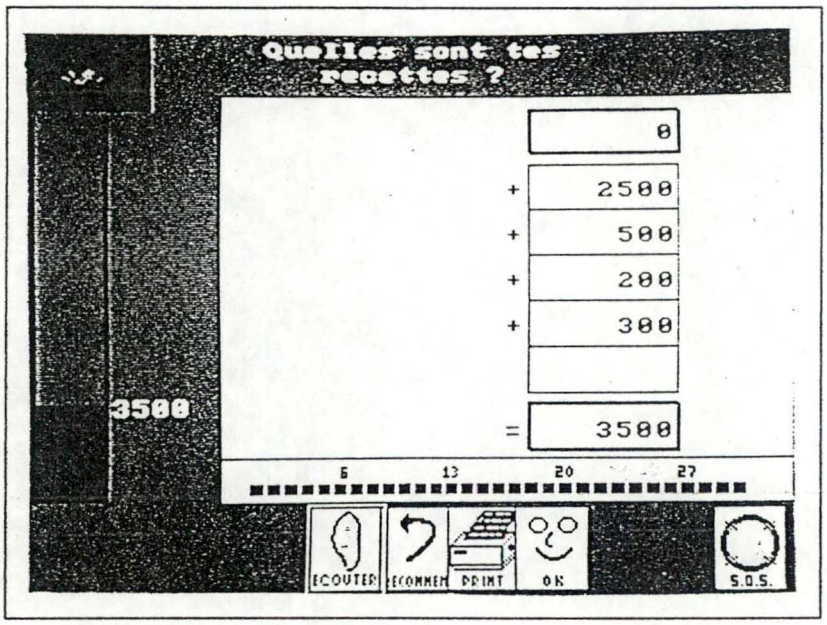


4.2. La saisie des recettes principales (3-4-5)

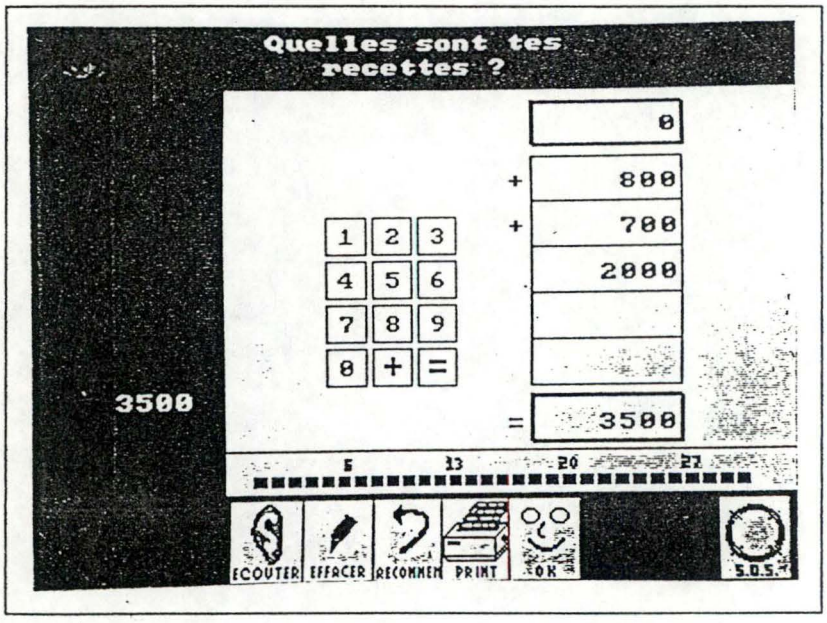
La saisie des recettes principales est une saisie du montant des recettes que la personne utilisatrice espère recevoir. On peut sélectionner jusqu'à 5 recettes sur la période de gestion.

D'après un paramètre, la saisie peut se faire à l'aide du clavier, de la calculette ou des billets.

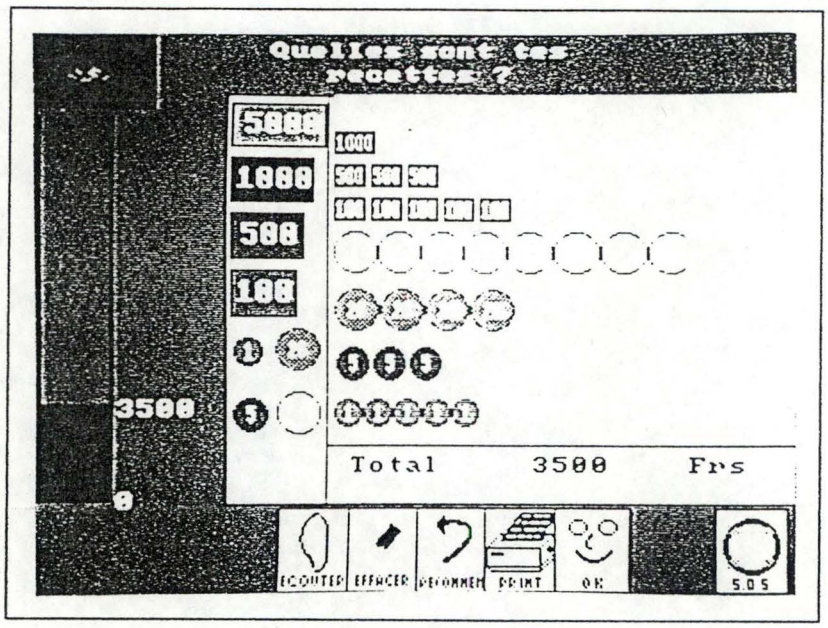
4.2.1. Saisie des recettes principales à l'aide du clavier (3)



4.2.2. Saisie des recettes à l'aide de la calculette (4)



4.2.3. Saisie des recettes principales à l'aide des billets (5)

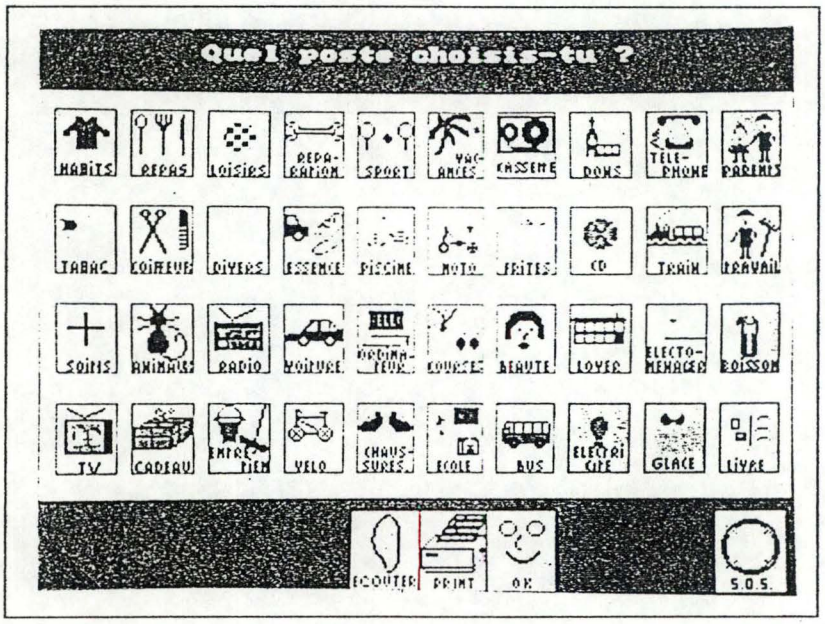


4.3. La prévision des dépenses (6)

4.3.1. Création d'une dépense

a) *Saisie du poste (7)*

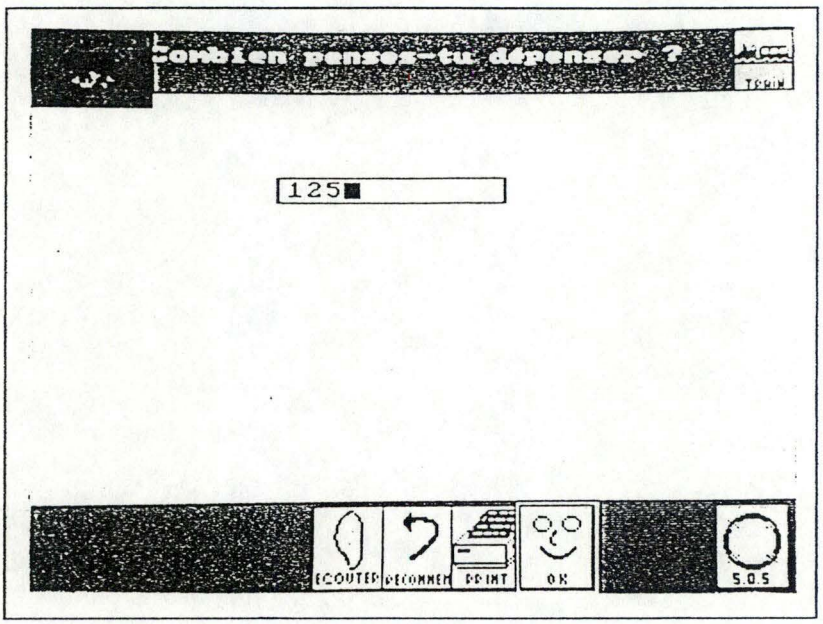
Lorsque la personne désire ajouter une prévision de dépenses, elle "clique" sur l'icône "créer". Cela entraîne une saisie, de la part de la personne, du poste pour lequel cette dépense est prévue.



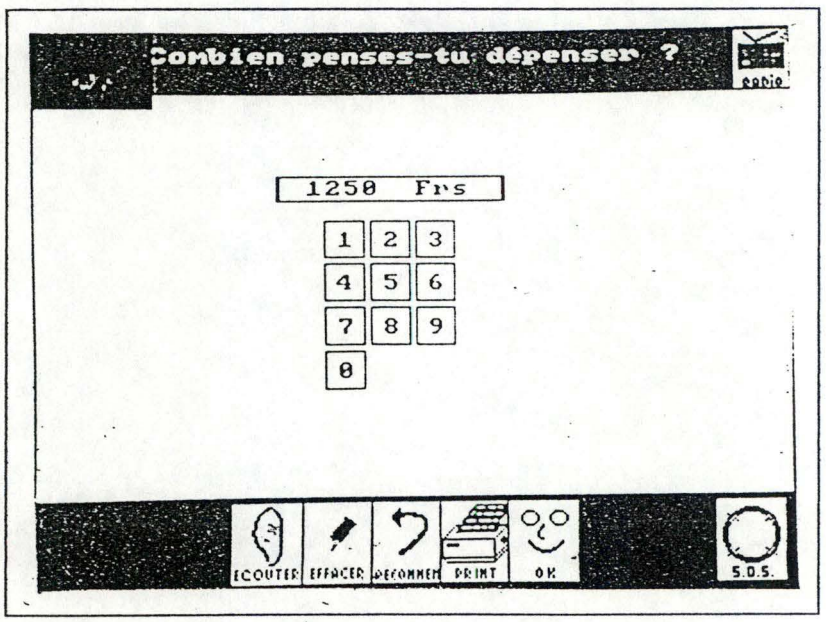
b) *Saisie du montant prévu de la dépense (8a-8b-8c)*

Similairement à la saisie des recettes principales plusieurs types de saisie du montant d'une dépense sont envisagées: la saisie à l'aide du clavier, au moyen d'une calculette et au moyen des billets.

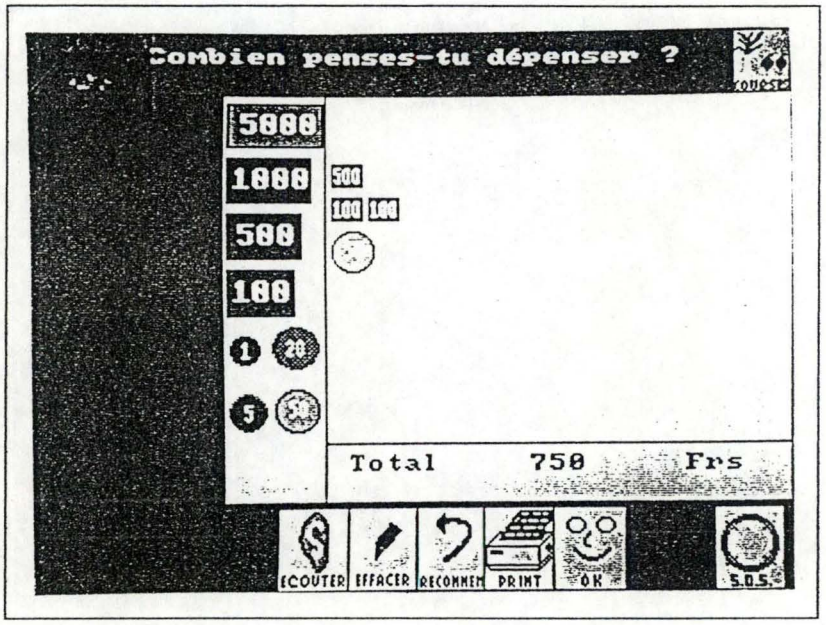
1. Saisie à l'aide du clavier (8a)



2. Saisie au moyen de la calculette (8b)



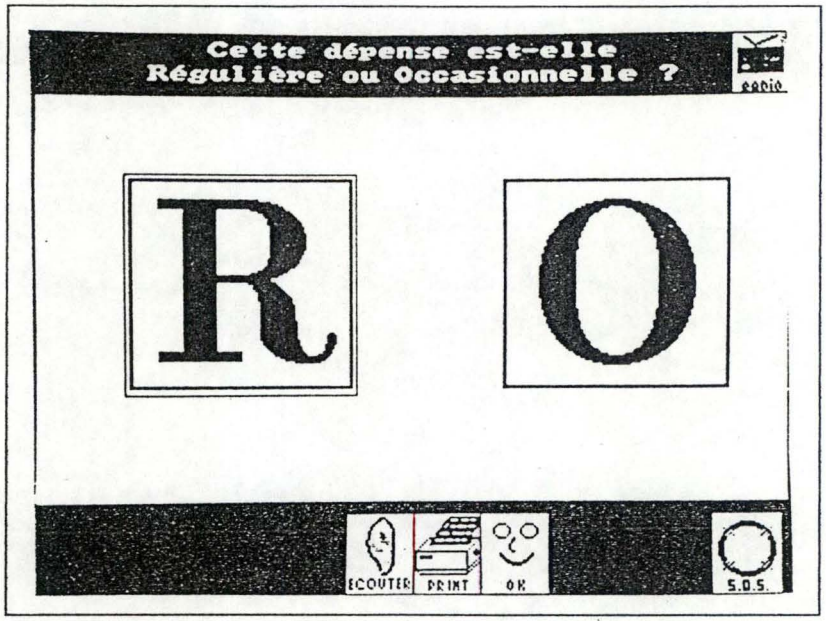
3. Saisie au moyen des billets (8c)



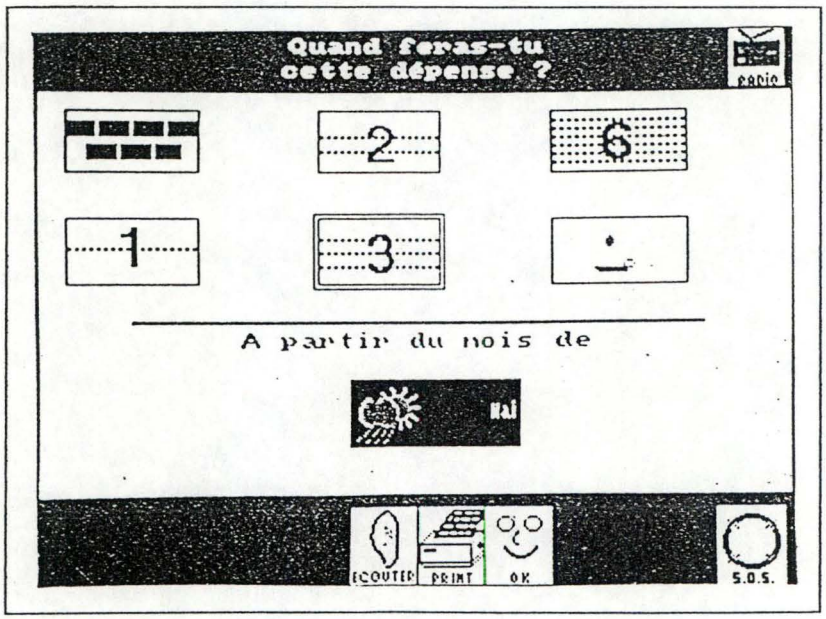
c) Saisie de la périodicité de la dépense (9-10)

Après avoir saisi le poste puis le montant de la dépense, il s'agit, maintenant, de savoir si la dépense se fera régulièrement. Dans ce cas, il nous faudra connaître la fréquence à laquelle s'effectuera la dépense. Cette fréquence sera automatiquement sauvée par l'ordinateur. De cette façon, la personne qui utilisera le logiciel Budget aura sous les yeux la liste des dépenses qu'elle devrait faire. Le fait de créer de nouvelles prévisions peut sembler lourd pour la personne mais est fait une fois pour toutes.

1. Saisie Régulière - Occasionnelle (9)



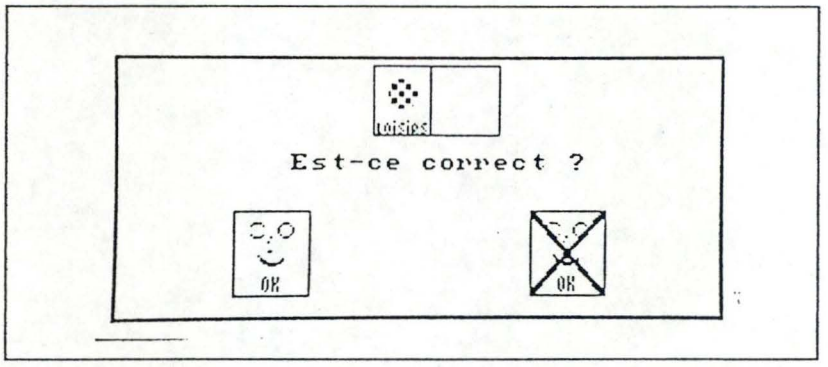
2. Saisie de la périodicité (10) (dépense régulière)



4.3.2. Le fait d'effectuer une dépense (initialement prévue)

Pour effectuer une dépense qu'elle avait prévu, la personne doit "cliquer" sur le pictogramme représentant la dépense puis sur l'icône "dépenser". Si la dépense est régulière et s'exécute toutes les semaines, alors on l'effectue pour le montant d'une semaine, sinon le montant affiché à côté du pictogramme est pris comme la somme prévue pour la dépense. Cela permet de constater que le montant prévu de la dépense est égal au montant réel de la dépense (8a-8b-8c).

Cette saisie apparaît sous forme de confirmation dans laquelle se trouvent le pictogramme de la dépense et son montant. La question "Est-ce correct ?" apparaît et est prononcée et la personne a le choix entre l'icône "OK", si la réponse est oui et l'icône "non OK", dans le cas contraire:



Si la personne choisis "non OK", elle est invitée à entrer sa somme réelle par une saisie de somme.

Dans les deux cas, la personne retourne à l'écran des prévisions de dépenses.

6.3.3. L'option "jeter"

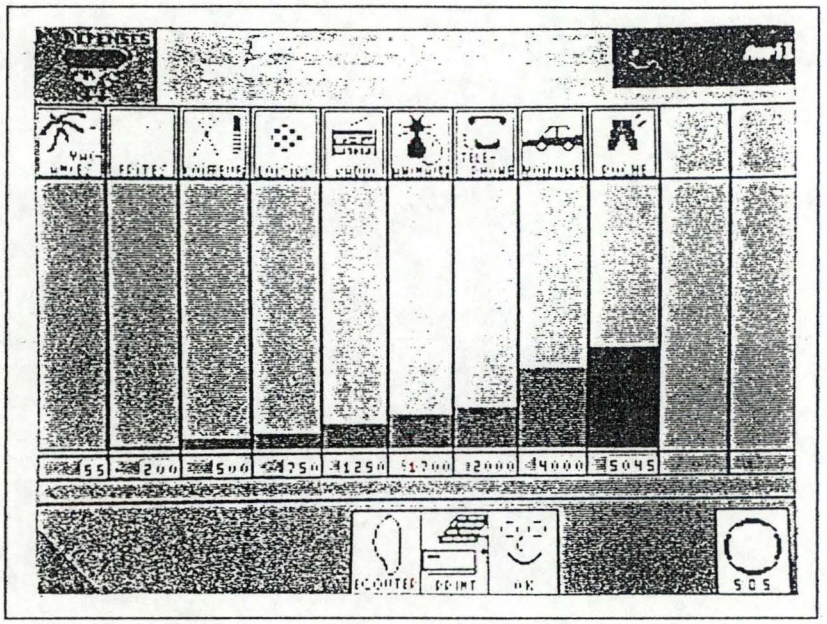
Si la personne a prévu régulièrement une dépense et que celle-ci ne peut se faire pour une raison ou une autre, alors, elle a la possibilité de la "jeter", c'est-à-dire de l'annuler provisoirement. Lors d'une autre période, cette dépense sera encore prévue.

Pour ce faire, elle a la possibilité de l'annuler soit une fois durant la période, soit durant la période entière. Pour la "jeter", il suffit de la sélectionner et, ensuite, de "cliquer" sur l'icône "jeter". Pour la sélectionner, il faut "cliquer" sur la partie pictogramme de la dépense, si on veut l'annuler une seule fois, et sur la partie chiffres, pour l'annuler durant la période entière. Cela implique que le montant d'argent réservé à cette dépense sera libre et ainsi, l'argent de poche augmentera.

Une fois la dépense annulée, la personne revient à l'écran des prévisions de dépenses avec l'argent de poche ajusté et, éventuellement, une dépense en moins.

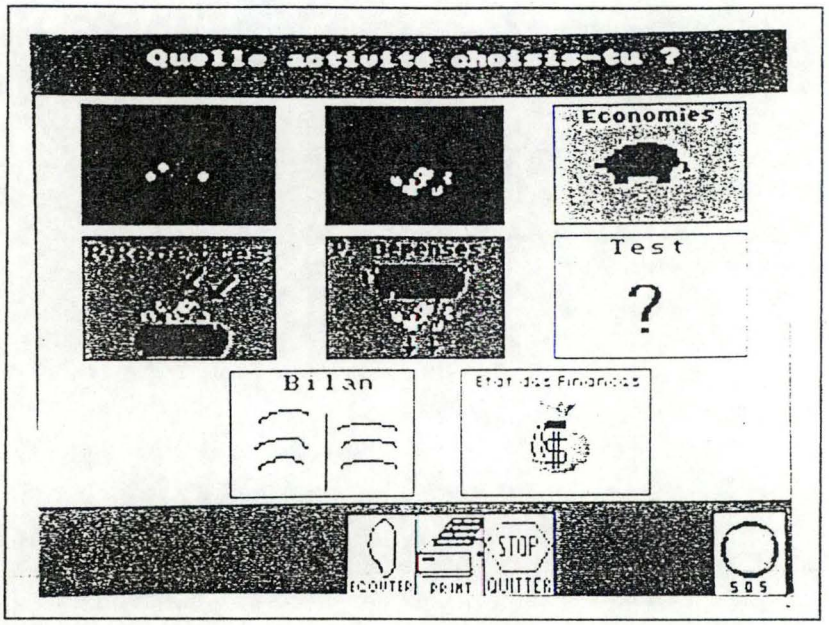
4.3.4. **Visualisation des prévisions de dépenses**

Lorsque la personne désire visualiser ses prévisions de dépenses, elle "clique" dans la jauge. Elle arrive à l'écran:



4.4. Le menu principal

Une fois la saisie du prénom, des recettes principales et des prévisions de dépense effectuées, l'utilisateur arrive au menu principal duquel il peut choisir entre quatre des huit activités proposées: Recettes, Dépenses, Prévisions de dépenses, et Test.



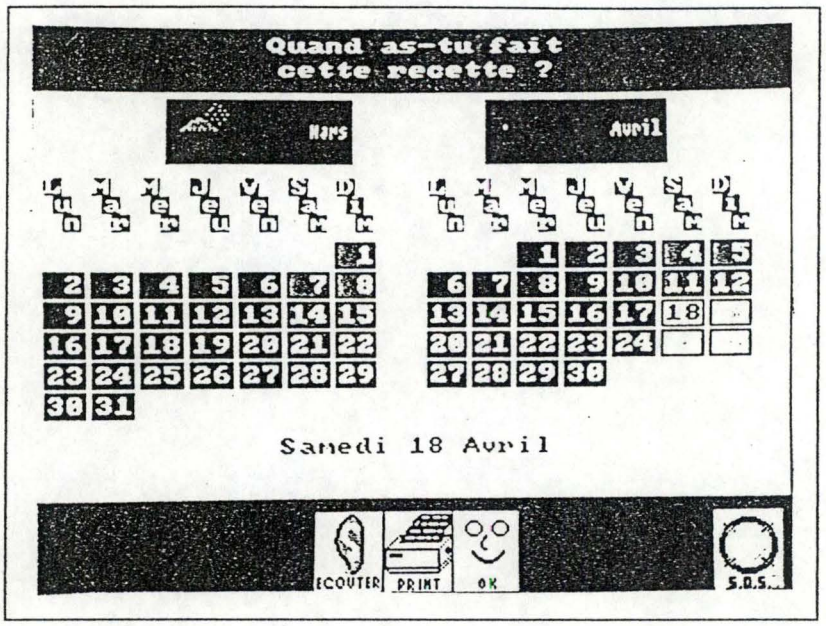
4.5. Effectuer une recette

Le fait d'effectuer une recette, qui n'a pas été, au préalable, prévue, se déroule en trois étapes:

★ La saisie du poste de la recette (7). Le même écran que lors du choix du poste d'une dépense prévue est présenté à la personne; excepté le cadre des pictogrammes qui est en vert.

★ La saisie du montant de la recette est une des 3 saisies de somme(8a-8b-8c): au moyen du clavier, de la calculette ou des billets. L'écran est le même sauf le titre qui devient: "A combien s'élève cette recette ?".

★ La saisie du moment de la recette (14). La question "Quand as-tu fait cette recette ?" apparaît à l'écran. Les représentations du mois passé et actuel sont présentes et la personne "clique" sur le jour de la recette. Elle confirme en "cliquant" sur l'icône "OK" et revient au menu principal.



4.6. Effectuer une dépense (7 - 8 - 14)

L'activité "effectuer une dépense" a pour but de faire une dépense qui n'a pas encore été prévue. Elle se déroule en trois étapes:

★ La saisie du poste de la dépense (7). Le même écran que lors du choix du poste d'une dépense prévue est présenté à la personne; excepté le cadre des pictogrammes qui est en rouge.

★ La saisie du montant de la dépense est une des 3 saisies de somme (8a-8b-8c): au moyen du clavier, de la calculette ou des billets. L'écran est le même sauf le titre qui devient: "A combien s'élève cette dépense ?".

★ La saisie du moment de la dépense (14). La question "Quand as-tu fait cette dépense ?" apparaît à l'écran. Les représentations du mois passé et actuel sont présentes et la personne "clique" sur le jour de la dépense. Elle confirme en "cliquant" sur l'icône "OK" et revient au menu principal.

4.7. La prévision des dépenses (6 et suivants)

La prévision des dépenses a déjà été expliquée précédemment (4.3.). Il y a, ici, sept écrans de prévisions de dépenses. Le premier est le même que dans le point 6.3. Les autres regroupent les prévisions de dépenses régulières par fréquence. Par exemple, dans le deuxième écran, sont rassemblées les prévisions de dépenses qui se font toutes les semaines, dans le troisième, tous les mois et ainsi de suite jusque tous les ans.

Ces écrans sont les mêmes, exceptés que ceux regroupés par fréquence, ont l'icône représentant la fréquence dans le coin supérieur droit.

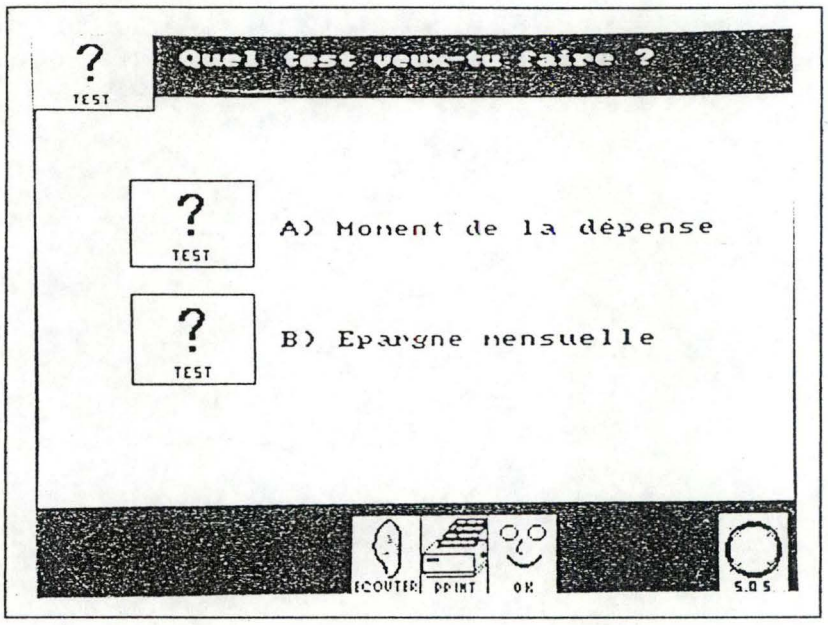
Pour passer d'un écran à l'autre, l'utilisateur "clique" sur l'icône-OK. Lorsque cette icône est activée dans le dernier écran (tous les ans) on revient au menu principal.

6.8. Test (24)

Deux questions nous paraissent fondamentales pour les personnes handicapées mentales. Le test tente d'y répondre.

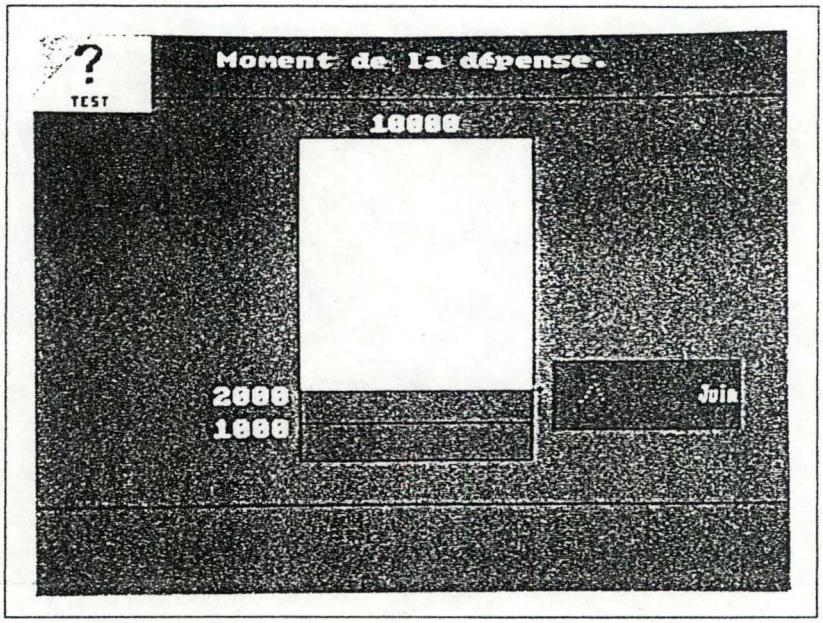
A: *"Quand pourra-t-on faire une dépense de X francs, si on épargne une certaine somme chaque mois ?"*

B: *"Combien faut-il épargner par mois, si on veut faire une dépense de X francs dans Y mois ?"*

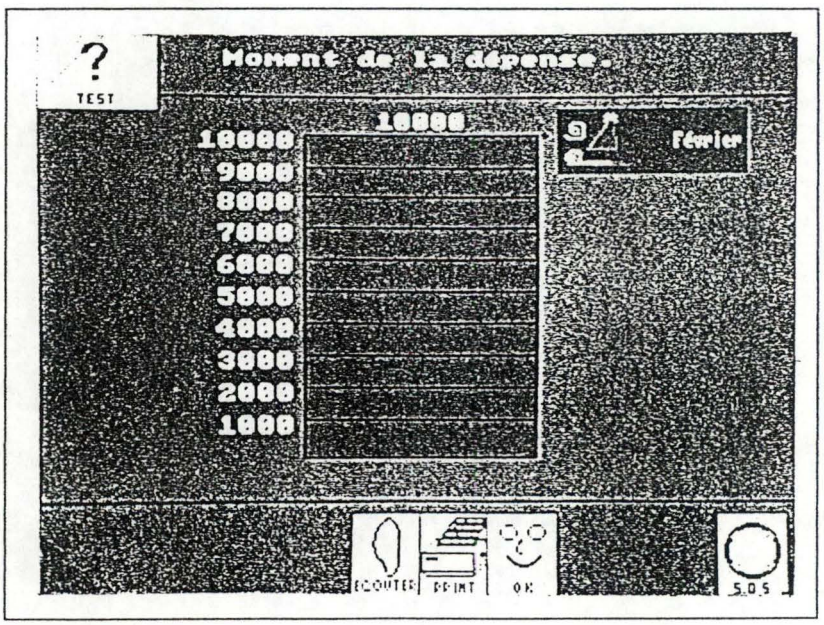


Si on choisit la première question, on doit d'abord faire une saisie du montant de la dépense à simuler (8a-8b-8c) puis, ensuite, on refait une saisie de somme (8a-8b-8c) pour connaître le montant à épargner par mois.

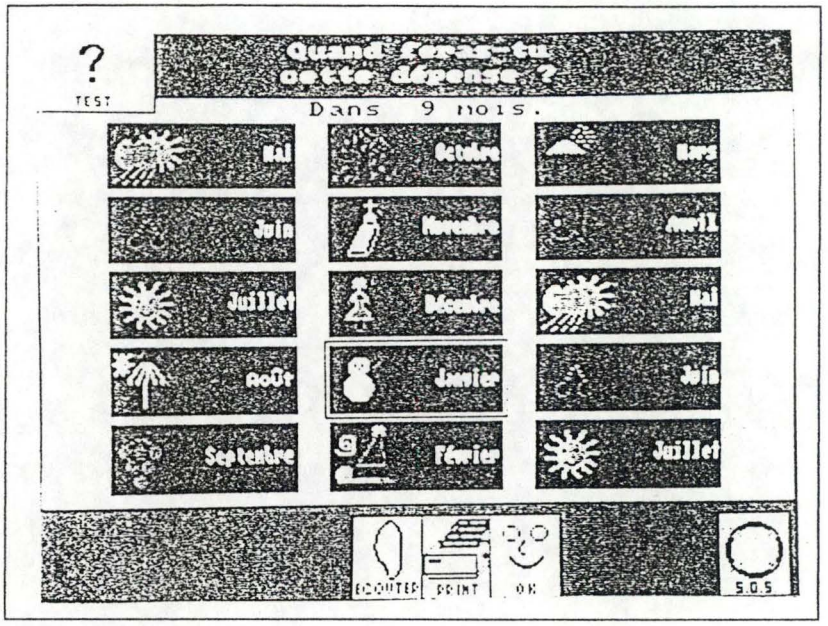
On obtient, par exemple, la réponse (26)



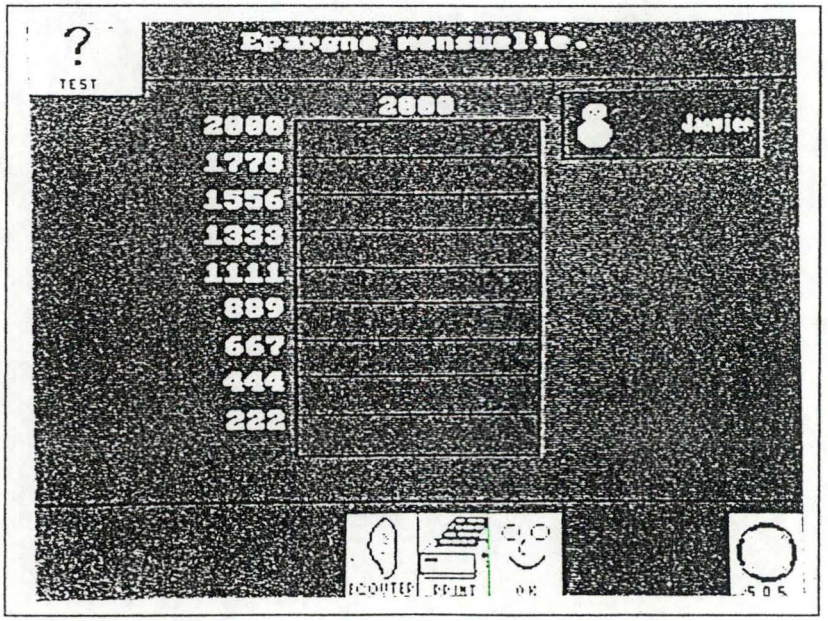
le rectangle représentant la dépense se remplit progressivement par tranches de montant à épargner par mois. Une fois le rectangle complètement rempli, on obtient, par exemple l'écran 26'.



Si on choisit la deuxième question, on doit d'abord faire une saisie du montant de la dépense à simuler (8a-8b-8c) puis, ensuite, on refait une saisie du mois (25).

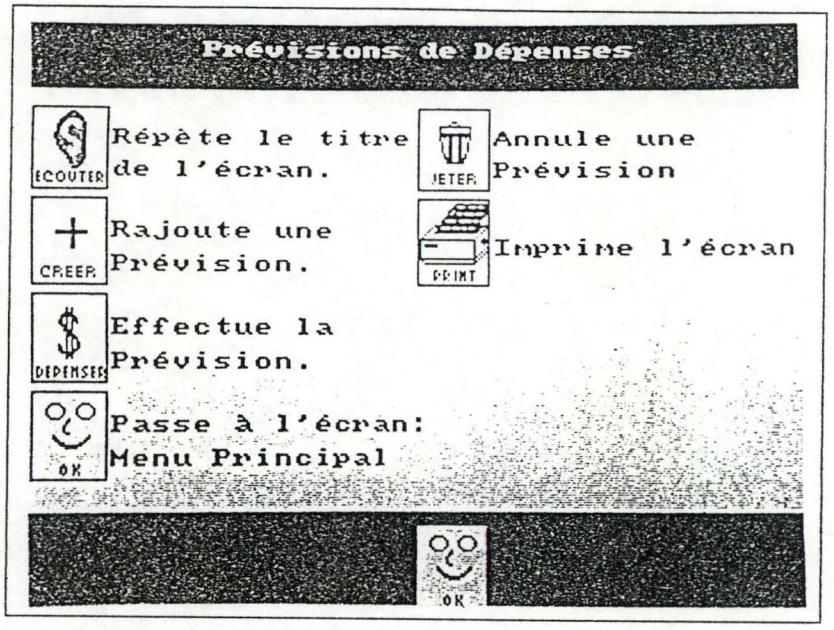


La réponse de l'ordinateur sera par exemple, l'écran 27.



6.9. L'aide (12)

A chaque écran, se trouve une icône-SOS qui, une fois activée, explique chaque commande de l'écran en cours.



★ ★ ★

★ ★

Exemple de traces

Exemple de trace d'utilisation

Jm : trace n° 0

Liste des paramètres choisis

Impression du résumé
Questions posées par écrit + parole
Saisie du Nom par souris
Recettes princ. saisies par billets
Dépenses saisies par billets
Notion du temps présente
Cycle d'un mois

Il appuie sur 5000 Frs.
Il appuie sur OK
Il arrive aux prévisions de dépenses
Il clique sur HELP
Il clique sur ok ... et quitte l'aide
Il clique sur Créer
Il arrive dans la librairie
Il choisit: Electro-ménager
Il clique sur ok
Il appuie sur 5000 Frs.
Il appuie sur 5000 Frs.
Il appuie sur OK
Il arrive au choix de la périodicité
Il choisit Occasionnelle
Il appuie sur ok
Il clique sur Dépenser
Il n'a pas sélectionné de poste
Il clique sur Frites
Il clique sur Jeter
Il demande de visualiser ses PD
Il appuie sur ok
Il clique sur OK
Il choisit la fonction : depenses
Il désire effectuer une dépense
Il choisit: Vacances
Il clique sur ok
Il appuie sur 1000 Frs.
Il appuie sur 1000 Frs.
Il appuie sur 1000 Frs.
Il appuie sur OK
Il choisit: Le Dimanche 19 Avril
Il clique sur OK
Il appuie sur Quitter
Il choisit la fonction : fin

Exemple de trace d'utilisation

Jm : trace n° 1

Liste des paramètres choisis

Impression du résumé
Questions posées par écrit + parole
Saisie du Nom par souris
Recettes princ. saisies par billets
Dépenses saisies par billets
Notion du temps présente
Cycle d'un mois

Il appuie sur 5000 Frs.
Il appuie sur 1000 Frs.
Il appuie sur 1000 Frs.
Il appuie sur 500 Frs.
Il appuie sur OK
Il arrive aux prévisions de dépenses
Il clique sur Electro-ménager
Il clique sur Dépenser
Il confirme la demande
Il clique sur Créer
Il arrive dans la librairie
Il choisit: Train
Il clique sur ok
Il appuie sur 500 Frs.
Il appuie sur 100 Frs.
Il appuie sur 100 Frs.
Il appuie sur OK
Il arrive au choix de la périodicité
Il choisit Régulière
Il appuie sur ok
Sa dépense se fait :
Tous les mois
Il clique sur ok
Il clique sur OK
Il choisit la fonction : test
Il choisit la question A
Il clique sur ok
Il appuie sur 5000 Frs.
Il appuie sur OK
Il appuie sur 500 Frs.
Il appuie sur 100 Frs.
Il appuie sur OK
Il demande une impression.
Il clique sur OK

Exemple de trace d'utilisation

Il choisit la fonction : test
Il choisit la question B
Il clique sur ok
Il appuie sur 5000 Frs.
Il appuie sur 1000 Frs.
Il appuie sur OK
Il choisit Janvier
Il clique sur OK
Il clique sur OK
Il appuie sur Quitter
Il choisit la fonction : fin

Exemple de trace d'utilisation

Jm : trace n° 2

Liste des paramètres choisis

Pas d'impression
Questions posées par écrit + parole
Saisie du Nom par souris
Recettes princ. saisies par clavier
Dépenses saisies par calculette
Notion du temps absente
Cycle d'une semaine
Début de cycle = lundi

Il arrive aux prévisions de dépenses
Il clique sur Créer
Il arrive dans la librairie
Il choisit: Glace
Il clique sur ok
Il appuie sur 1
Il appuie sur 0
Il appuie sur 0
Il appuie sur OK
Il arrive au choix de la périodicité
Il choisit Occasionnelle
Il appuie sur ok
Il demande de visualiser ses PD
Il appuie sur ok
Il clique sur HELP
Il clique sur ok ... et quitte l'aide
Il clique sur OK
Il choisit la fonction : depenses
Il désire effectuer une dépense
Il choisit: Bus
Il clique sur ok
Il choisit: Le Mercredi 15 Avril
Il clique sur OK
Il appuie sur Quitter
Il choisit la fonction : fin

Exemple de trace d'utilisation

JM : trace n° 3

Liste des paramètres choisis

Pas d'impression
Questions posées par écrit + parole
Saisie du Nom par clavier
Recettes princ. saisies par billets
Dépenses saisies par calculette
Notion du temps présente
Cycle d'un mois

Il appuie sur 100 Frs.
Il appuie sur OK
Il arrive aux prévisions de dépenses
Il clique sur HELP
Il clique sur ok ... et quitte l'aide
Il clique sur OK
Il choisit la fonction : recettes
Il a reçu de l'argent
Il choisit: Piscine
Il clique sur ok
Il appuie sur 100 Frs.
Il appuie sur OK
Il choisit: Le Lundi 20 Avril
Il clique sur OK
Il appuie sur Quitter
Il choisit la fonction : fin

★ ★ ★
★ ★

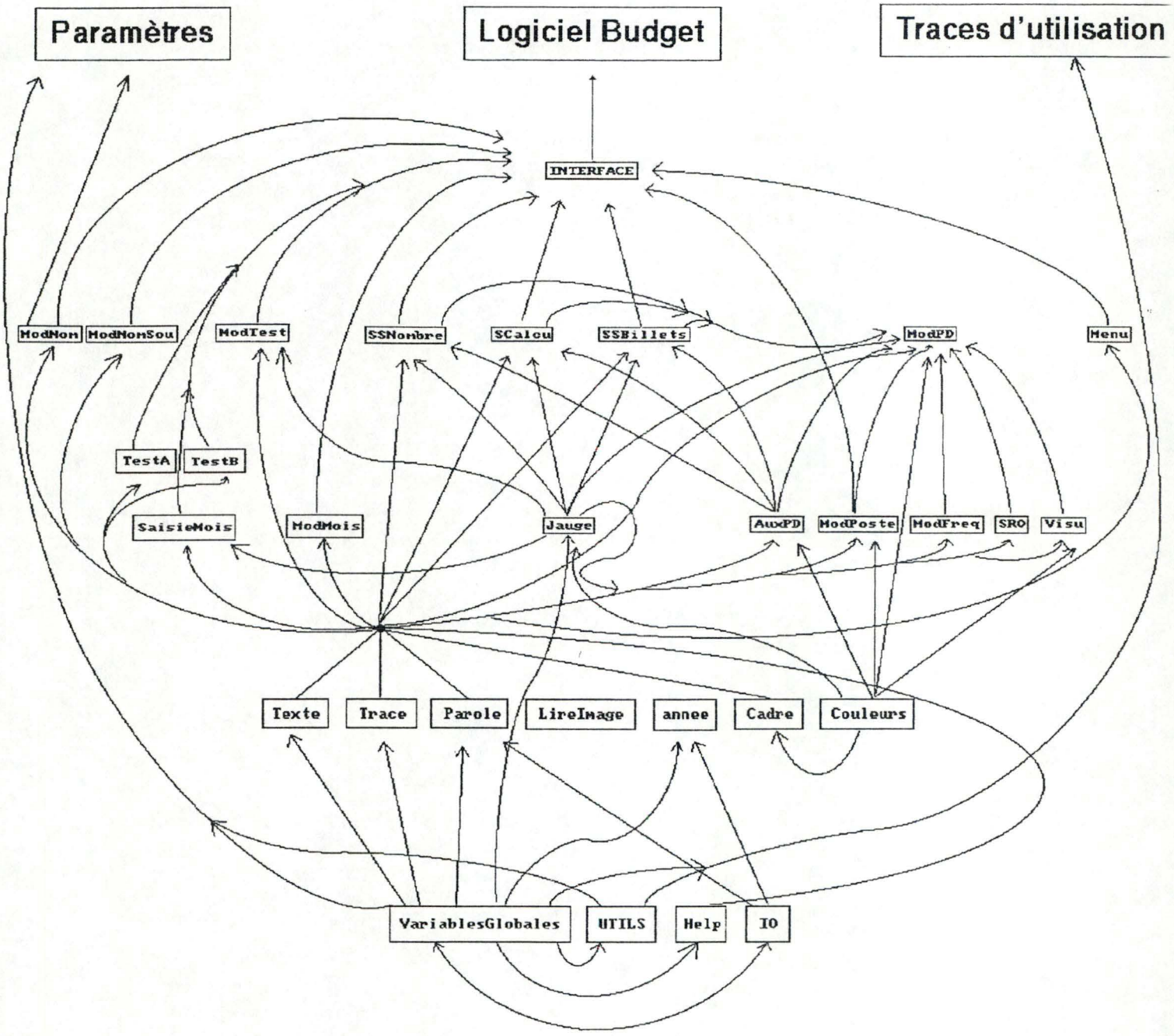
Implementation modules

LOCALISATION

MODULE Coordinateur 1
IMPLEMENTATION MODULE Interface 3
IMPLEMENTATION MODULE ModNom 6
IMPLEMENTATION MODULE ModNomSou 9
IMPLEMENTATION MODULE SSNombre 14
IMPLEMENTATION MODULE SCalcu 20
IMPLEMENTATION MODULE SSBillets 25
IMPLEMENTATION MODULE ModPD 31
IMPLEMENTATION MODULE ModMenu 38
IMPLEMENTATION MODULE ModMois 41
IMPLEMENTATION MODULE Jauge 46
IMPLEMENTATION MODULE AuxPD 50
IMPLEMENTATION MODULE ModPoste 56
IMPLEMENTATION MODULE ModFreq 59
IMPLEMENTATION MODULE ModSRO 62
IMPLEMENTATION MODULE Visu 65
IMPLEMENTATION MODULE ModTest 69
IMPLEMENTATION MODULE TestA 72
IMPLEMENTATION MODULE TestB 76
IMPLEMENTATION MODULE SaisieMois 80
IMPLEMENTATION MODULE Texte 83
IMPLEMENTATION MODULE Trace 85
IMPLEMENTATION MODULE Parole 87

IMPLEMENTATION MODULE LireImage 89
IMPLEMENTATION MODULE annee 93
IMPLEMENTATION MODULE Cadre 96
IMPLEMENTATION MODULE couleurs 97
IMPLEMENTATION MODULE VariablesGlobales 97
IMPLEMENTATION MODULE UTILS 99
IMPLEMENTATION MODULE Help 101
IMPLEMENTATION MODULE IO 105

Interaction entre modules



MODULE Coordinateur;

```
  FROM SYSTEM IMPORT ADR,SHORT;
  FROM Conversions IMPORT ConvNumberToString;
  FROM VariablesGlobales IMPORT Str40,TableauParametres,Nom,cmap,Str20,
                                Scr, TableauTraces,Liquide,Son;
  FROM Strings IMPORT ConcatString, CopyString,OverwriteWithSubString,
                      StringLength,CompareString,Relation;
  FROM RealConversions IMPORT ConvRealToString,RealToStringFormat;
  FROM Intuition IMPORT CloseScreen,ScreenPtr,Image;
  FROM SimpleScreens IMPORT CreateScreen;
  FROM Views IMPORT LoadRGB4;
  FROM Interface IMPORT SSaisieNom,SaisieSomme,SaisieRP,Menu,pd;
  FROM Trace IMPORT trace,TraceParametres,sauver_traces;
  FROM Libraries IMPORT OpenLibrary,CloseLibrary;
  FROM SoundLib IMPORT SBase,SoundName,ResetChan,LoadAIFF,ASampPtr;
  FROM Memory IMPORT FreeMem;
  FROM IO IMPORT LoadMax,LoadLiquide,sauver_Liquide,InitParametres;
  FROM InOut IMPORT WriteString,WriteLn,WriteInt;

VAR val : REAL;
    fonction  : Str20;
    i,F,M:INTEGER;
    es,ch:CHAR;
    im:Image;

PROCEDURE InitSons;
(*---------------*)
VAR NOM:Str20;
    istr:ARRAY[0..2] OF CHAR.

BEGIN
     SBase:=OpenLibrary(ADR(SoundName),LONGCARD(0));
     FOR i:=1 TO 24 DO
         NOM:="SON:Son";
         ConvNumberToString (istr,LONGINT(i),FALSE,10,2,"0");
         ConcatString(NOM,istr);
         Son[i]:=LoadAIFF(ADR(NOM"SON:Son01"));
     END;
END InitSons;

PROCEDURE FinSons;
(*--------------*)
BEGIN
   FOR i:=1 TO 24 DO
       FreeMem(Son[i]^.Samp,Son[i]^.Size);
   END;
   ResetChan;
   CloseLibrary(SBase^);
END FinSons;
```

```
PROCEDURE Initialisation;
(*--------------------*)

BEGIN

    Scr := CreateScreen (320,256,5,NIL);
    IF (Scr # NIL) THEN
        LoadRGB4 (Scr^.ViewPort,ADR (cmap),32);
        InitParametres;
        InitSons;
        SSaisieNom;
        LoadMax;
        LoadLiquide;
    END;
END Initialisation;

PROCEDURE Terminaison;
(*------------------*)

BEGIN
     sauver_traces;
     sauver_Liquide;
END Terminaison;

(************************ PROGRAMME PRINCIPAL ************************)

BEGIN
      Initialisation;
      SaisieRP(FALSE,val,es);
      Liquide:=Liquide+val;
      pd;
      Menu;
      FinSons;
      Terminaison;
      CloseScreen(Scr^);

END Coordinateur.
```

IMPLEMENTATION MODULE Interface;

```
FROM Strings IMPORT CopyString, ConcatString,CompareString,equal;
FROM VariablesGlobales IMPORT TableauParametres,TableauTraces,Nom,Str20,
                              Liquide,Str40;
FROM Intuition IMPORT Image;
FROM ModNom IMPORT SaisieNom;
FROM ModNomSou IMPORT SaisieNomSou;
FROM SCalcu IMPORT SaisieSommeCalcu;
FROM SSNombre IMPORT SaisieSommeNbre;
FROM SSBillets IMPORT SaisieSommeBillets;
IMPORT ModMenu;
FROM Trace IMPORT trace;
FROM ModPD IMPORT PD;
FROM IO IMPORT Str15,TabPD,NbPD,LirePD,EcrirePD;
FROM ModPoste IMPORT Poste;
FROM ModMois IMPORT QUAND;
FROM ModTest IMPORT TEST;
FROM TestA IMPORT TestA;
FROM TestB IMPORT TestB;
FROM SaisieMois IMPORT SMois;
VAR
    ES,EtatSortie:CHAR;
    p : Str20;
    j : INTEGER;
    im:Image;
    nom:Str15;
    OK:BOOLEAN;

PROCEDURE SSaisieNom;
(*******************)
BEGIN
    IF TableauParametres[2] = 0 THEN SaisieNom;
                                ELSE SaisieNomSou;
     END;
END SSaisieNom;

PROCEDURE SaisieSomme(ch:INTEGER; Q:BOOLEAN;VAR Val:REAL; VAR ES:CHAR);
(*****************************************************************)
BEGIN
     CASE TableauParametres[3] OF
         2: SaisieSommeCalcu(ch,Q,Val,ES,im); |
         0: SaisieSommeNbre (ch,Q,Val,ES,im); |
         1: SaisieSommeBillets(ch,Q,Val,ES,im); |
     END;
END SaisieSomme;

PROCEDURE SaisieRP(Q:BOOLEAN;VAR Val:REAL; VAR ES:CHAR);
(*****************************************************)
BEGIN
     SaisieSomme(1,Q,Val,ES);
END SaisieRP;
```

```
PROCEDURE pd2;
(************)
VAR X:CHAR;
    phrase : Str40;
    i,N:INTEGER;
    P:ARRAY[0..6] OF INTEGER;
BEGIN
     P[0]:=6; P[1]:=0; P[2]:=3; P[3]:=1; P[4]:=4; P[5]:=2; P[6]:=5;
     IF TableauParametres[9]=1 THEN N:=1;
                               ELSE N:=6;
     END;
     LirePD(NbPD,TabPD);
     FOR i:=0 TO N DO
     phrase:="Il arrive aux prévisions de dépenses ";
     trace(phrase);
        X:="X";
        WHILE X<>"A" DO
           PD(P[i],X);
        END;
     END;
     EcrirePD(NbPD,TabPD);
END pd2;

PROCEDURE dep;
(*----------*)
VAR phrase : Str40;
    es:CHAR;
    Val:REAL;
    N,JJ,MM,A:INTEGER;
BEGIN
     phrase:="Il désire effectuer une dépense ";
     trace(phrase);
     N:= 1;
     Poste(im,N);
     SaisieSomme(3,FALSE,Val,es);
     Liquide:=Liquide-Val;
     A:=1;
     QUAND(JJ,MM,A);
END dep;

PROCEDURE rec;
(*----------*)
VAR phrase : Str40;
    es:CHAR;
    Val:REAL;
    N,JJ,MM,A:INTEGER;
BEGIN
     phrase:="Il a  reçu de l'argent ";
     trace(phrase);
     N:=2;
     Poste(im,N);
     SaisieSomme(4,FALSE,Val,es);
     Liquide:=Liquide+Val;
     A:=2;
     QUAND(JJ,MM,A);
END rec;
```

```
PROCEDURE tst;
(*----------*)
VAR CH,ES:CHAR;
    Val,VPM:REAL;
    n:INTEGER;

BEGIN
     TEST(CH);
     IF CH="A" THEN SaisieSomme(5,FALSE,Val,ES);
                    SaisieSomme(6,FALSE,VPM,ES);
                    TestA(Val,VPM);
               ELSE SaisieSomme(5,FALSE,Val,ES);
                    SMois(n);
                    TestB(Val,n);
     END;
END tst;

PROCEDURE Menu;
(*************)

VAR phrase : Str40;
    fct:Str20;

BEGIN  fct:="a";
       WHILE CompareString(fct,"fin")#equal DO
            ModMenu.Menu (fct);
            CopyString (phrase,"Il choisit la fonction : ");
            ConcatString (phrase,fct);
            ConcatString (phrase," ");
            trace(phrase);
            IF CompareString(fct,"prevision-depenses")=equal THEN pd2
END;
            IF CompareString(fct,"depenses")=equal THEN dep END;
            IF CompareString(fct,"recettes")=equal THEN rec END;
            IF CompareString(fct,"test")=equal THEN tst END;
       END;
END Menu;

PROCEDURE pd;
(***********)
VAR X:CHAR;
    phrase : Str40;
BEGIN
     LirePD(NbPD,TabPD);
     phrase:="Il arrive aux prévisions de dépenses ";
     trace(phrase);
     X:="X";
     WHILE X<>"A" DO
           PD(6,X);
     END;
     EcrirePD(NbPD,TabPD);
END pd;

END Interface.
```

IMPLEMENTATION MODULE ModNom;

```
FROM SYSTEM IMPORT BYTE, ADR, SHORT;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr, WindowFlags, WindowFlagsSet,
IDCMPFlags, IDCMPFlagsSet, CloseWindow,ActivateWindow,DrawImage;
FROM SimpleWindows IMPORT CreateWindow;
FROM SimpleConsole IMPORT CreateConsole, DeleteConsole, PutStr, GetStr;
FROM SimpleConsoleCmds IMPORT wMove, wClrScr, wSetColor, wSetCursor,
                              wClrEndLine;
FROM InOut IMPORT OpenInputFile,CloseInput,Done;
FROM Strings IMPORT  CopyString, CompareString, Relation,ConcatString,
          StringLength,LocateChar,ExtractSubString,ConvStringToUpperCase;
FROM Conversions IMPORT ConvNumberToString;
FROM VariablesGlobales IMPORT Nom,cmap,TableauParametres,Str50,Scr,Str40,

                              ST10,Str20,Commande;
FROM Views IMPORT ViewPort, LoadRGB4;
FROM Drawing IMPORT SetAPen,RectFill,Move;
FROM annee IMPORT DATE,NOMMOIS;
FROM Parole  IMPORT SAY;
FROM Texte IMPORT ECRIT;
FROM cadre IMPORT FRAME2;
FROM Couleurs IMPORT rouge;
FROM Rasters IMPORT RastPortPtr,RastPort;
FROM Text IMPORT Text;
FROM Trace IMPORT trace;
FROM UTILS IMPORT EnleveBlancs;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Activate,Borderless};
  NomBlanc = "                         ";

VAR
 Nombre,Affichage: Str20;
  T,FileName : Str40;
  Efface:BOOLEAN;
  PP:Str50;
  Win,Won: WindowPtr;
  i : CARDINAL;
  rp:RastPortPtr;

PROCEDURE Erreur;
(*-------------*)

VAR
   Wun : WindowPtr;

BEGIN
      Wun := CreateWindow (15,80,290,90,WIDCMP,WFlags,NIL,Scr,NIL);
      rp:=Wun^.RPort;
      rp^.FgPen:=BYTE(1);
      rp^.BgPen:=BYTE(0);
      SetAPen (Wun^.RPort^,1);
```

```
      DrawImage(rp^,Commande[5],5,5);
      DrawImage(rp^,Commande[1],5,46);
      SAY(8);
      Move(rp^,44,22);
      Text(rp^,ADR("Je ne trouve pas ton prénom."),28);
      Move(rp^,96,58);
      Text( rp^,ADR("essaie encore !"),15);
      Delay(500);
      CloseWindow(Wun^);
END Erreur;


PROCEDURE TraceDate;
(*----------------*)

VAR  Mois:ST10;
     Bi:BOOLEAN;
     JJ,MM,AA:INTEGER;
     ANStr, JJstr:ST10;

BEGIN
     DATE(JJ,MM,AA,Bi);
     CopyString (T,"Date de la session : ");
      ConvNumberToString (JJstr,LONGINT(JJ),FALSE,10,2," ");
     ConcatString (T,JJstr);
     ConcatString (T," ");
     NOMMOIS(Mois,MM);
     ConcatString (T,Mois);
     ConcatString (T," ");
     ConvNumberToString (ANStr,LONGINT(AA),FALSE,10,2," ");
     ConcatString (T,ANStr);
     trace(T);
END TraceDate;

(*************************************************************************
(*                                  SaisieNom                            *
(*************************************************************************

PROCEDURE SaisieNom;
(*----------------*)

BEGIN
     Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,NIL,Scr,NIL);
     Won := CreateWindow (70,40,180,60,WIDCMP,WFlags2,NIL,Scr,NIL);

     IF (Won # NIL) THEN
       FRAME2(Win);
       ECRIT(Win,1);
       Efface := TRUE;
       WHILE Efface = TRUE DO
       ActivateWindow(Won^);
       IF CreateConsole (Won^) THEN
          wMove (Won^,1,4);
          IF (TableauParametres[1] = 2) THEN SAY(1); END;
          CopyString (Nom,"");
          GetStr (Won^,ADR (Nom), SIZE (Nom));
          IF CompareString (Nom,"") # equal THEN EnleveBlancs(Nom); END;
          ConvStringToUpperCase (Nom);
```

```
        CopyString (T,"Il donne le nom : ");
        ConcatString(T,Nom);
        trace(T);
        wSetCursor (Won^,FALSE);
        FileName := "Param:";
        ConcatString (FileName,Nom);
        ConcatString (FileName,".MAX");
        OpenInputFile (FileName);
        IF NOT Done THEN Efface := TRUE;
                         Erreur;
                    ELSE Efface:=FALSE
        END;
      CloseInput;
      DeleteConsole (Won^);
      END;
    END; (* while efface *)
      CloseWindow(Won^);
      END;
      CloseWindow(Win^);
END SaisieNom;

END ModNom.
```

IMPLEMENTATION MODULE ModNomSou;

```
FROM SYSTEM IMPORT BYTE, ADR, SHORT;
FROM Intuition IMPORT Window,WindowPtr, GadgetPtr,
    Gadget, WindowFlags, WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,
    IntuiMessagePtr, CloseWindow, Image, DrawImage, ActivateWindow,
    EndGadget,RelVerify,  GadgetFlagsSet, GadgetActivationSet,
GadgetMutualExcludeSet;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
    AddGadgetImageButton , GadgetTypeReq, GlobalGadgetOpt,
    GadgetOpt,AddGadgetTextButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM SimpleConsole IMPORT CreateConsole, DeleteConsole, PutStr;
FROM SimpleConsoleCmds IMPORT wMove,wSetColor, wSetCursor, wClrEndLine;
FROM InOut IMPORT OpenInputFile, CloseInput,Done;
FROM Strings IMPORT  CopyString, CompareString,ConcatString,LocateChar ,
                     ExtractSubString,Relation;
FROM VariablesGlobales IMPORT Nom,Str20,Str40,ST10,cmap,Scr,
                              TableauParametres,Str50,Commande;
FROM Cadre IMPORT FRAME2;
FROM Parole  IMPORT SAY;
FROM Texte IMPORT ECRIT;
FROM ModNom IMPORT Erreur,TraceDate;
FROM Trace IMPORT trace;
FROM LireImage IMPORT LoadLettre,TabLettre;
FROM UTILS IMPORT EnleveBlancs;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate,Borderless};
  WFlags2 = WindowFlagsSet {Activate,Borderless};
  NomBlanc = "                          ";
VAR
  Trace : Str40;
  Done3,Efface : BOOLEAN;
  Win,Won : WindowPtr;
  TabImages: TabLettre;
  i : CARDINAL;
  FileName,Nom2 : Str40;
  Gl3 : GadgetPtr;
  Wp3 : WindowProc;
  Sig : SignalSet;
  Msg3 : IntuiMessagePtr;
  Res,CWB : BOOLEAN;
  Nbrlettre ,pos: INTEGER;

PROCEDURE GadgetHandlerLettres (VAR w: Window; VAR Msg3 : MsgData;
                                VAR Gad: Gadget);

VAR CLAVIER,clavier : Str40;
    TOTO : ARRAY [0..1] OF CHAR;

BEGIN
```

```
    CopyString(CLAVIER,"AZERTYUIOPQSDFGHJKLMWXCVBN");
    CopyString(clavier,"azertyuiopqsdfghjklmwxcvbn");

  CopyString(TOTO,"");
  IF (Gad.GadgetID < 26) THEN
        Nbrlettre := Nbrlettre + 1;
        IF Nbrlettre <= 20 THEN
           pos:=LocateChar(Nom,"-",Nbrlettre-2,Nbrlettre);
           IF ((Nbrlettre=1) OR (pos # -1))
              THEN ExtractSubString(TOTO,CLAVIER,Gad.GadgetID,1);
                   ConcatString (Nom,TOTO);
              ELSE ExtractSubString(TOTO,clavier,Gad.GadgetID,1);
                   ConcatString (Nom,TOTO);
           END;
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));
        END;
    ELSE
CASE Gad.GadgetID OF

     26 : Nbrlettre := Nbrlettre + 1;
        IF Nbrlettre <= 20 THEN
            ConcatString (Nom,"-");
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));
        END; |

     27 : Nbrlettre := Nbrlettre + 1;
        IF Nbrlettre <= 20 THEN
           ConcatString (Nom," ");
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));
        END; |
     28 : Nbrlettre := Nbrlettre + 1;
        IF Nbrlettre <= 20 THEN
         ConcatString (Nom,"é");
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));
        END; |
     29 : Nbrlettre := Nbrlettre + 1;
        IF Nbrlettre <= 20 THEN
           ConcatString (Nom,"è");
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));
        END; |
     30 : Done3 := TRUE;
          Nbrlettre :=0;
          Efface:=FALSE; |
```

```
      31 : Nbrlettre :=0;
           CopyString (Nom,"");
           wMove (Won^,1,1);
           wClrEndLine (Won^); |

      32 : Nbrlettre :=Nbrlettre-1;
           ExtractSubString(Nom2,Nom,0,Nbrlettre);
           CopyString (Nom,Nom2);
           wMove (Won^,1,1);
           wClrEndLine (Won^);
           wMove (Won^,1,1);
           PutStr (Won^,ADR(Nom));  |
      33 : Trace:="Il clique sur HELP ";
           trace(Trace);
           SOS(1); |
      34 : SAY(1);

  END;
  END;
END GadgetHandlerLettres;

(*************************************************************************)
(*                      Saisie Nom par Souris                            *)
(*************************************************************************)

PROCEDURE SaisieNomSou;

BEGIN

WITH Wp3 DO
      procGadgetUp := GadgetHandlerLettres;
END;

TraceDate;
     LoadLettre(TabImages);
     BeginGadgetList ();
     AddGadgetImageButton (29,73,TabImages[0]);
     AddGadgetImageButton (54,73,TabImages[1]);
     AddGadgetImageButton (79,73,TabImages[2]);
     AddGadgetImageButton (104,73,TabImages[3]);
     AddGadgetImageButton (129,73,TabImages[4]);
     AddGadgetImageButton (154,73,TabImages[5]);
     AddGadgetImageButton (179,73,TabImages[6]);
     AddGadgetImageButton (204,73,TabImages[7]);
     AddGadgetImageButton (229,73,TabImages[8]);
     AddGadgetImageButton (254,73,TabImages[9]);
     AddGadgetImageButton (40,101,TabImages[10]);
     AddGadgetImageButton (65,101,TabImages[11]);
     AddGadgetImageButton (90,101,TabImages[12]);
     AddGadgetImageButton (115,101,TabImages[13]);
     AddGadgetImageButton (140,101,TabImages[14]);
     AddGadgetImageButton (165,101,TabImages[15]);
     AddGadgetImageButton (190,101,TabImages[16]);
     AddGadgetImageButton (215,101,TabImages[17]);
     AddGadgetImageButton (240,101,TabImages[18]);
     AddGadgetImageButton (265,101,TabImages[19]);
     AddGadgetImageButton (51,129,TabImages[20]);
     AddGadgetImageButton (76,129,TabImages[21]);
```

```
    AddGadgetImageButton (101,129,TabImages[22]);
    AddGadgetImageButton (126,129,TabImages[23]);
    AddGadgetImageButton (151,129,TabImages[24]);
    AddGadgetImageButton (176,129,TabImages[25]);
    AddGadgetImageButton (201,129,TabImages[26]);
    AddGadgetImageButton (60,157,TabImages[27]);
    AddGadgetImageButton (226,129,TabImages[28]);
    AddGadgetImageButton (251,129,TabImages[29]);
    AddGadgetImageButton (165,215,Commande[2]);
    AddGadgetImageButton (105,215,Commande[1]);
    AddGadgetImageButton (135,215,Commande[0]);
    AddGadgetImageButton (285,215,Commande[12]);
   IF (TableauParametres[1] = 2) THEN
       AddGadgetImageButton (75,215,Commande[4]); END;
    Gl3 := EndGadgetList ();
    IF ( Gl3 # NIL)  THEN
    Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl3,Scr,NIL);
    Won := CreateWindow (110,40,200,8,WIDCMP,WFlags2,NIL,Scr,NIL);
    FRAME2 (Win);
    ECRIT(Win,1);
    DrawImage (Win^.RPort^,Commande[2],165,215);
    DrawImage (Win^.RPort^,Commande[1],105,215);
    DrawImage (Win^.RPort^,Commande[0],135,215);
    DrawImage (Win^.RPort^,Commande[12],285,215);
    IF (TableauParametres[1] = 2) THEN
       DrawImage (Win^.RPort^,Commande[4],75,215);
    END;
  IF (Won # NIL) THEN
      Efface := TRUE;
      WHILE Efface = TRUE DO
        ActivateWindow (Won^);
        IF CreateConsole (Won^) THEN
            wMove (Won^,1,1);
            IF (TableauParametres[1] = 2) THEN SAY(1); END;
            Nbrlettre :=0;
            CopyString (Nom,"");
            Done3 := FALSE;
            WHILE (NOT Done3) DO
              Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
              LOOP
                Msg3 := GetMsg(Win^.UserPort^);
                IF (Msg3 = NIL) THEN EXIT; END;
                ProcIMsg (Wp3, Msg3);
              END;
            END;
            IF CompareString (Nom,"") # equal THEN EnleveBlancs (Nom);
END;
            CopyString (Trace,"Il donne le nom : ");
            ConcatString (Trace,Nom);
            trace(Trace);
            wSetCursor (Won^,FALSE);
            FileName := "Param:";
            ConcatString (FileName,Nom);
            ConcatString (FileName,".MAX");
            OpenInputFile (FileName);
            IF NOT Done THEN Efface := TRUE;
                             Erreur;
                        ELSE CloseInput;
```

```
            END;
          END;
       DeleteConsole (Won^);
      END; (* while efface *)
         CloseWindow(Won^);
       END;
       CloseWindow(Win^);
END;
END SaisieNomSou;

END ModNomSou.
```

IMPLEMENTATION MODULE SSNombre;

```
FROM SYSTEM IMPORT SHORT,BYTE, ADR;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,WindowFlags, WindowFlagsSet,
IDCMPFlags, IDCMPFlagsSet, CloseWindow, DrawImage,
ActivateWindow,WindowToBack,Image,
                        ClearPointer,WindowToFront;
FROM SimpleWindows IMPORT CreateWindow;
FROM SimpleConsole IMPORT CreateConsole, DeleteConsole, PutStr,
GetStr,GetCh;
FROM SimpleConsoleCmds IMPORT
wMove,wClrScr,wSetColor,wSetCursor,wClrEndLine;
FROM Strings IMPORT CopyString,Relation,ConcatString,StringLength,
                    DeleteSubString;
FROM VariablesGlobales IMPORT
Str20,Str40,Icone,Commande,TableauParametres,
                                Liquide,Scr;
FROM RealConversions IMPORT ConvRealToString,Decimal,ConvStringToReal;
FROM Rasters IMPORT RastPortPtr;
FROM Drawing IMPORT SetAPen,Move,Draw,RectFill;
FROM Parole IMPORT SAY;
FROM cadre IMPORT FRAME,FRAME2;
FROM Jauge IMPORT InitJauge,MAJ_Jauge,InitSomme,MAJ_Somme ,JaugeX,
                  JaugeY,cdr;
FROM couleurs IMPORT noir,blanc,rouge,orange,jaune,vert,gris;
FROM Texte IMPORT ECRIT;
FROM Trace IMPORT trace;
FROM UTILS IMPORT PRINT,Reel;
FROM AuxPD IMPORT LigneTemps;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Activate,Borderless};

VAR
  Nombre : Str20;
  OKNombre,Res,FiniSSN,ok :BOOLEAN;
  Win,Won: WindowPtr;
  W:ARRAY[1..5] OF WindowPtr;
  CH:CHAR;
  Somme:ARRAY[1..5] OF REAL;
  Val,Stotale : REAL;
  Rp : RastPortPtr;
  im,couleur,sens,ph,pp,i,Nfleche,Ncase,Length:INTEGER;
  Trace:Str40;
  pos:ARRAY[1..5] OF INTEGER;

PROCEDURE DrawCommande;
BEGIN
     SetAPen(Rp^,gris);
     RectFill(Rp^,110,211,319,254);
     IF TableauParametres[1]=2 THEN
DrawImage(Rp^,Commande[4],pos[1],212);
     END;
```

```
     DrawImage(Rp^,Commande[1],pos[2],212);
     DrawImage(Rp^,Commande[2],pos[4],212);
     DrawImage(Rp^,Commande[11],pos[3],212);
     DrawImage(Rp^,Commande[12],pos[5],212);
END DrawCommande;

PROCEDURE REC;
BEGIN
     DrawCommande;
     cdr(Win,pos[Nfleche],212,5,FALSE);
     cdr(Win,1+pos[Nfleche],213,6,FALSE);
END REC;


PROCEDURE INIT(ch:INTEGER);
BEGIN
       IF (ch=1) OR (ch=3) OR (ch=4) THEN FRAME(Win);
                                     ELSE FRAME2(Win);
       END;
       ECRIT(Win,ph);
       pos[1]:=115; pos[2]:=145; pos[3]:=175; pos[4]:=205; pos[5]:=285;
       IF (ch=1) OR (ch=3) OR (ch=4) THEN
          InitJauge(JaugeX,JaugeY,Win);
          MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
       END;
       Rp := Win^.RPort;
       DrawImage (Win^.RPort^,Icone[im],0,0);
       IF TableauParametres[1]=2 THEN SAY(pp); END;
       Nfleche:=1; Ncase:=1; REC;
       IF (TableauParametres[6]=1) AND (ch=1) THEN LigneTemps(Win); END;
       IF ch=1 THEN InitSomme(Win,5);
                    FOR i:=1 TO 5 DO
                       Somme[i]:=0.;
                    END;
      END;
END INIT;

PROCEDURE FL;
BEGIN
     WindowToFront(W[Ncase]^);
     wSetCursor(W[Ncase]^,TRUE);
     IF Somme[Ncase]>0.
        THEN ConvRealToString(Somme[Ncase],Nombre,0,Decimal);
             Length:=StringLength(Nombre);
        ELSE Nombre:=""; Length:=0;
     END;
     wClrScr(W[Ncase]^);
     wMove(W[Ncase]^,1,2);
     PutStr(W[Ncase]^,ADR(Nombre));
     wMove(W[Ncase]^,1,2);
END FL;

PROCEDURE LF;
BEGIN
     WindowToBack(W[Ncase]^);
     MAJ_Somme(Win,Ncase,Somme[Ncase]);
END LF;
```

```
PROCEDURE Gestion_Fleches;
BEGIN
   CH:=GetCh(W[Ncase]^);
   CASE ORD(CH) OF
    65: wSetCursor(W[Ncase]^,FALSE); IF Somme[Ncase]>0. THEN LF; END;
        Ncase:=Ncase-1;IF Ncase=0 THEN Ncase:=1;ELSE FL; END; |
    66: wSetCursor(W[Ncase]^,FALSE);IF Somme[Ncase]>0. THEN LF; END;
        Ncase:=Ncase+1;IF Ncase=6 THEN Ncase:=5;ELSE FL; END; |
    67: Nfleche:=Nfleche+1; IF Nfleche=6 THEN Nfleche:=1; END; REC; |
    68: Nfleche:=Nfleche-1; IF Nfleche=0 THEN Nfleche:=5; END; REC; |
   END;
END Gestion_Fleches;

PROCEDURE Gestion_Plus;
BEGIN
     Reel(Nombre,ok);
     IF ok THEN
        IF Somme[Ncase]=0. THEN Somme[Ncase]:=ConvStringToReal(Nombre);
                                Stotale:=Stotale+Somme[Ncase];
        END;
        wClrScr(W[Ncase]^); wSetCursor(W[Ncase]^,FALSE);
        WindowToBack(W[Ncase]^);
        MAJ_Somme(Win,10+Ncase,Somme[Ncase]);
        MAJ_Somme(Win,7,Stotale+Liquide);
        MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide+Stotale,vert,0.,0);
        Ncase:=Ncase+1;IF Ncase=6 THEN Ncase:=5; ELSE FL; END;
     ELSE wClrScr(W[Ncase]^);wMove(W[Ncase]^,1,1);
          wSetCursor(W[Ncase]^,TRUE);
     END;
END Gestion_Plus;

PROCEDURE Gestion_Return;
BEGIN
     CASE Nfleche OF
        1: SAY(pp); |
        2: Stotale:=0.; Nombre:=""; Length:=0;
           wSetCursor(W[Ncase]^,FALSE);
           Ncase:=1;
           InitJauge(JaugeX,JaugeY,Win);
           MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
           InitSomme(Win,5);
           FOR i:=1 TO 5 DO
               Somme[i]:=0.;
           END; |
        4: FiniSSN:=TRUE; |
        3: PRINT; |
        5: SOS(2);|
     END;
END Gestion_Return;

PROCEDURE Gestion_Chiffre;
BEGIN
     IF Length <6 THEN Length:=Length+1;
                       ConcatString(Nombre,CH);
                       wClrScr(W[Ncase]^);
                       wMove(W[Ncase]^,1,2);
                       PutStr(W[Ncase]^,ADR(Nombre));
     END;
```

```
END Gestion_Chiffre;

PROCEDURE Gestion_Del;
BEGIN
     IF Length>0 THEN Length:=Length-1;
                      DeleteSubString(Nombre,Length,1);
                      wClrScr(W[Ncase]^);
                      wMove(W[Ncase]^,1,2);
                      PutStr(W[Ncase]^,ADR(Nombre));
     END;
END Gestion_Del;

PROCEDURE Gestion_Eff;
BEGIN
     WindowToFront(W[Ncase]^);
     wClrScr(W[Ncase]^);
     wMove(W[Ncase]^,1,2);
     Length:=0;
     Stotale:=Stotale-Somme[Ncase];
     Somme[Ncase]:=0.;
     Nombre:="";
     MAJ_Somme(Win,7,Stotale+Liquide);
     MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide+Stotale,vert,0.,0);
END Gestion_Eff;

PROCEDURE Gestion_Fleches2;
BEGIN
   CH:=GetCh(Won^);
   CASE ORD(CH) OF
    67: Nfleche:=Nfleche+1; IF Nfleche=6 THEN Nfleche:=1; END; REC; |
    68: Nfleche:=Nfleche-1; IF Nfleche=0 THEN Nfleche:=5; END; REC; |
   END;
END Gestion_Fleches2;

PROCEDURE Gestion_Plus2(ch:INTEGER);
BEGIN
     Reel(Nombre,ok);
     IF ok THEN
        wSetCursor(Won^,FALSE);
        Stotale:=ConvStringToReal(Nombre);
        IF (ch=3) OR (ch=4) THEN
        MAJ_Jauge(JaugeX,JaugeY,Win,sens,Liquide,vert,Stotale,couleur);
        END;
        Delay(200);
     END;
END Gestion_Plus2;

PROCEDURE Gestion_Return2(ch:INTEGER);
BEGIN
     CASE Nfleche OF
        1: SAY(pp); |
        2: Stotale:=0.; Nombre:=""; Length:=0;
           IF (ch=3) OR (ch=4) THEN
              InitJauge(JaugeX,JaugeY,Win);
              MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
           END; |
        4: Gestion_Plus2(ch); FiniSSN:=TRUE; |
        3: PRINT; |
```

```
        5: SOS(8); |
    END;
END Gestion_Return2;

PROCEDURE Gestion_Chiffre2;
BEGIN
     IF Length <6 THEN Length:=Length+1;
                       ConcatString(Nombre,CH);
                       wClrScr(Won^);
                       wMove(Won^,1,1);
                       PutStr(Won^,ADR(Nombre));
     END;
END Gestion_Chiffre2;

PROCEDURE Gestion_Del2;
BEGIN
     IF Length>0 THEN Length:=Length-1;
                      DeleteSubString(Nombre,Length,1);
                      wClrScr(Won^);
                      wMove(Won^,1,1);
                      PutStr(Won^,ADR(Nombre));
     END;
END Gestion_Del2;

PROCEDURE GETNbr(ch:INTEGER);
BEGIN
     Won:=CreateWindow(100,70,100,12,WIDCMP,WFlags,NIL,Scr,NIL);
     ActivateWindow(Won^);
     IF CreateConsole (Won^) THEN
        Length:=0; Stotale:=0.; Nombre:="";
       FiniSSN := FALSE;
        wClrScr(Won^);
        wMove(Won^,1,1);
        WHILE NOT FiniSSN DO
             wSetCursor (Won^,TRUE);
              CH:=GetCh(Won^);
              CASE ORD (CH) OF
                155: Gestion_Fleches2; |
                43: Gestion_Plus2(ch); |
                13: Gestion_Return2(ch); |
                8: Gestion_Del2; |
              END;
              IF (ORD(CH)>45) AND (ORD(CH)<58) THEN Gestion_Chiffre2;
END;
       END;
    END;
    DeleteConsole (Won^);
    CloseWindow(Won^);
END GETNbr;

PROCEDURE SaisieSommeNbre(ch:INTEGER;Quit:BOOLEAN;VAR Valeur:REAL;
                                    VAR ES:CHAR;VAR poste:Image);
(*-------------------------------------------------------------*)

BEGIN
    CASE ch OF
    1: im:=0; pp:=2; ph:=2;
       couleur:=vert; sens:=1; |
```

```
      2: im:=2; pp:=5; ph:=5;
         couleur:=orange; sens:=2; |
      3: im:=1; pp:=23; ph:=13;
         couleur:=rouge; sens:=2; |
      4: im:=0; pp:=24; ph:=14;
         couleur:=vert; sens:=1; |
      5: im:=3; pp:=5; ph:=5;|
      6: im:=3; pp:=20; ph:=16; |
    END;
      ES := "A";
      Win := CreateWindow (0,0,320,256,WIDCMP,WFlags2,NIL,Scr,NIL);
      ClearPointer(Win^);
      INIT(ch);
         IF ch=2 THEN DrawImage(Win^.RPort^,poste,293,1); END;
         IF ch#1 THEN GETNbr(ch); Valeur:=Val;
         ELSE
             W[1]:=CreateWindow(211,69,58,13,WIDCMP,WFlags2,NIL,Scr,NIL);
             W[2]:=CreateWindow(211,89,58,13,WIDCMP,WFlags2,NIL,Scr,NIL);
             W[3]:=CreateWindow(211,109,58,13,WIDCMP,WFlags2,NIL,Scr,NIL);
             W[4]:=CreateWindow(211,129,58,13,WIDCMP,WFlags2,NIL,Scr,NIL);
             W[5]:=CreateWindow(211,149,58,13,WIDCMP,WFlags2,NIL,Scr,NIL);
             IF CreateConsole(W[1]^) THEN wSetCursor(W[1]^,FALSE);
             IF CreateConsole(W[2]^) THEN wSetCursor(W[2]^,FALSE);
             IF CreateConsole(W[3]^) THEN wSetCursor(W[3]^,FALSE);
             IF CreateConsole(W[4]^) THEN wSetCursor(W[4]^,FALSE);
             IF CreateConsole(W[5]^) THEN wSetCursor(W[5]^,FALSE);
           Stotale:=0.; Nombre:=""; Length:=0;
           FiniSSN:=FALSE;
           WHILE (NOT FiniSSN)  DO
                ActivateWindow(W[Ncase]^);
                wSetCursor(W[Ncase]^,TRUE);
                CH:=GetCh(W[Ncase]^);
                CASE ORD (CH) OF
                  155: Gestion_Fleches; |
                  43: Gestion_Plus; |
                  13: Gestion_Return; |
                  8: Gestion_Del; |
                  127: Gestion_Eff; |
                  61: FiniSSN:=TRUE; |
                END;
                IF (ORD(CH)>45) AND (ORD(CH)<58) THEN Gestion_Chiffre; END
           END;
           DeleteConsole(W[5]^); END;
           DeleteConsole(W[4]^); END;
           DeleteConsole(W[3]^); END;
           DeleteConsole(W[2]^); END;
           DeleteConsole(W[1]^); END;
           CloseWindow(W[5]^);
           CloseWindow(W[4]^);
           CloseWindow(W[3]^);
           CloseWindow(W[2]^);
           CloseWindow(W[1]^);
         END;
         CloseWindow(Win^);
         Valeur:=Stotale;
END SaisieSommeNbre;
END SSNombre.
```

IMPLEMENTATION MODULE Scalcu;

```
FROM SYSTEM IMPORT SHORT,BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,  GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet, IntuiMessagePtr, CloseWindow,
    Image,DrawImage, ActivateWindow,  EndGadget,RelVerify,
GadgetFlagsSet;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
                  AddGadgetImageButton ,GlobalGadgetOpt, GadgetOpt;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT SetAPen,Move;
FROM Tasks IMPORT SignalSet, Wait;
FROM SimpleConsole IMPORT CreateConsole, DeleteConsole, PutStr, GetStr;
FROM SimpleConsoleCmds IMPORT wMove,wClrScr,wSetColor,
                              wSetCursor,wClrEndLine;
FROM Strings IMPORT  CopyString, CompareString, Relation,ConcatString,
            StringLength,LocateChar,ExtractSubString;
FROM VariablesGlobales IMPORT Phrases, Str20, cmap,TableauParametres,
                              Icone,Str40,Liquide,TableauTraces,
                              Str50,Max,Scr,Commande;
FROM Rasters IMPORT RastPortPtr;
FROM RealConversions IMPORT ConvStringToReal;
FROM Parole IMPORT SAY;
FROM cadre IMPORT FRAME,FRAME2;
FROM Jauge IMPORT InitJauge,MAJ_Jauge,InitSomme,MAJ_Somme,JaugeX,JaugeY;
FROM couleurs IMPORT noir,blanc,rouge,orange,jaune,vert,gris;
FROM Texte IMPORT ECRIT;
FROM Trace IMPORT trace;
FROM UTILS IMPORT PRINT,Reel;
FROM AuxPD IMPORT LigneTemps;
FROM LireImage IMPORT LoadCalcu,TabCalcu;
FROM Text IMPORT Text;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Activate,Borderless};
  x=2;
  y=40;
VAR
  Trace : Str40;
  DoneSSC,Recom,Res,bi,Done_next,OKS : BOOLEAN;
  Win,Won: WindowPtr;
  i,pos,ph : INTEGER;
  Gl  : GadgetPtr;
  Wp  : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Fait,I,IJK,im,couleur,pp,sens,jj,mm,aa,CompChiffre,FOIS,CH,
  Nbrlettre:INTEGER;
  Somme,Stotale: REAL;
  ES:CHAR;
  StrSomme,strsomme : Str20;
```

```
  Rp : RastPortPtr;
  Chiffre:TabCalcu;

PROCEDURE PutStrSomme;
BEGIN
            IF FOIS=1 THEN wMove (Won^,1,1);
                           wClrEndLine (Won^);
                           wMove (Won^,2,1);
                           PutStr (Won^,ADR(StrSomme));
                           wMove (Won^,8,1);
                           PutStr (Won^,ADR("Frs"));
                           wSetCursor (Won^,FALSE);
                      ELSE Somme := ConvStringToReal(StrSomme);
                           MAJ_Somme(Win,IJK,Somme);
            END;
END PutStrSomme;

PROCEDURE GadgetHandlerCA (VAR w: Window; VAR Msg:MsgData; VAR
Gad:Gadget);

VAR Liste:ARRAY [0..10] OF CHAR;
    N:ARRAY[0..1] OF CHAR;

BEGIN
   CopyString(Liste,"01234567899");
   IF (Gad.GadgetID < 10) THEN
       IF (CompChiffre <=8) THEN
            CompChiffre := CompChiffre + 1;
            ExtractSubString(N,Liste,Gad.GadgetID,1);
            ConcatString (StrSomme,N);
            PutStrSomme;
            Trace := "Il appuie sur ";
            ConcatString(Trace,N);
            ConcatString(Trace," ");
            trace(Trace);
        END
    ELSE
    CASE Gad.GadgetID OF

 11 : Trace := "Il appuie sur EGAL";
      trace(Trace);
      Somme := ConvStringToReal(StrSomme);
      Stotale:=Stotale+Somme;
      IF (CH=1) THEN
         MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide+Stotale,vert,0.,0);
         MAJ_Somme(Win,IJK,Somme); Delay(50);
         MAJ_Somme(Win,7,Stotale+Liquide); Delay(50);
      END;
      IF (CH=4) THEN
         MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide+Stotale,vert,0.,0);
      END;
      Done_next := TRUE;
 12 : Trace := "Il appuie sur PLUS";
      trace(Trace);
      Somme := ConvStringToReal(StrSomme);
      Stotale:=Stotale+Somme;
      IF (CH=1) THEN
         MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide+Stotale,vert,0.,0);
```

```
        MAJ_Somme(Win,10+IJK,Somme); Delay(50);
        MAJ_Somme(Win,7,Stotale+Liquide); Delay(50);
      END;

      CompChiffre:=0;
      Done_next := TRUE; |

  13 : Trace := "Il appuie sur Del ";
       Reel (StrSomme,OKS);
       trace(Trace);
       Nbrlettre:=StringLength(StrSomme);
       Nbrlettre :=Nbrlettre-1;
       CompChiffre:=CompChiffre-1;
       ExtractSubString(strsomme,StrSomme,0,Nbrlettre);
       CopyString (StrSomme,strsomme);
       PutStrSomme; |
    14 : Trace := "Il appuie sur OK  ";
         trace(Trace);
         IF FOIS<5 THEN Stotale := ConvStringToReal(StrSomme); END;
         IF (CH=4) OR (CH=3) THEN

MAJ_Jauge(JaugeX,JaugeY,Win,sens,Liquide,vert,Stotale,couleur);
            Delay(200);
         END;
         DoneSSC := TRUE |

    15 :   Trace := "Il appuie sur RECOMMENCER ";
           trace(Trace);
           Recom:=TRUE; Done_next:=TRUE;
           Stotale:=0.;
           IF FOIS=5 THEN InitJauge(JaugeX,JaugeY,Win);

MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
                          InitSomme(Win,5);
           END;
           IJK:=0;|

    16 :  Trace := "Il clique sur S.O.S. ";
          trace(Trace);
          IF FOIS=5 THEN SOS(3); ELSE SOS(9); END; |

      17 : SAY(pp);|
      18 : PRINT;
  END;
END;

END GadgetHandlerCA;

(*************************************************************************)
(*                        SaisieSommeCalculette                          *)
(*************************************************************************)

PROCEDURE SaisieSommeCalcu (ch:INTEGER; Quit : BOOLEAN; VAR Valeur :
REAL;
                            VAR EtatSortie : CHAR;VAR poste:Image);


BEGIN
```

```
    CASE ch OF
     1: im:=0; pp:=2; ph:=2;                          (* 5 Recettes *)
        couleur:=vert; sens:=1; |
     2: im:=2; pp:=5; ph:=5;                          (* 1 PD       *)
        couleur:=orange; sens:=2; |
     3: im:=1; pp:=23; ph:=13;                        (* 1 Dépense  *)
        couleur:=rouge; sens:=2; |
     4: im:=0; pp:=24; ph:=14;                        (* 1 Recette  *)
        couleur:=vert; sens:=1; |
     5: im:=3; pp:=5; ph:=5; |                        (* test: 1    *)
     6: im:=3; pp:=20; ph:=16;|                       (* test: 2    *)
    END;
     ES := "A";
     CH:=ch;
     IF ch=1 THEN FOIS:=5; ELSE FOIS:=1; END;
     WITH Wp DO
       procGadgetUp := GadgetHandlerCA;
     END;
LoadCalcu(Chiffre);
          pos:=0;
          BeginGadgetList();
          AddGadgetImageButton (pos+120,147,Chiffre[0]);
          AddGadgetImageButton (pos+120,90,Chiffre[1]);
          AddGadgetImageButton (pos+140,90,Chiffre[2]);
          AddGadgetImageButton (pos+160,90,Chiffre[3]);
          AddGadgetImageButton (pos+120,109,Chiffre[4]);
          AddGadgetImageButton (pos+140,109,Chiffre[5]);
          AddGadgetImageButton (pos+160,109,Chiffre[6]);
          AddGadgetImageButton (pos+120,128,Chiffre[7]);
          AddGadgetImageButton (pos+140,128,Chiffre[8]);
          AddGadgetImageButton (pos+160,128,Chiffre[9]);
          AddGadgetImageButton (1,1,Commande[9]);
          IF (ch = 1) THEN AddGadgetImageButton (pos+160,147,Chiffre[10])
                           AddGadgetImageButton (pos+140,147,Chiffre[11])
                      ELSE AddGadgetImageButton (1,1,Commande[9]);
                           AddGadgetImageButton (1,1,Commande[9]); END;
          AddGadgetImageButton (110,212,Commande[0]);
          AddGadgetImageButton (200,212,Commande[2]);
          AddGadgetImageButton (140,212,Commande[1]);
          AddGadgetImageButton (285,212,Commande[12]);
          IF TableauParametres[1] = 2
             THEN AddGadgetImageButton (80,212,Commande[4]);
             ELSE AddGadgetImageButton (1,1,Commande[9]);
          END;
          IF TableauParametres[0] = 1
             THEN AddGadgetImageButton (170,212,Commande[11]);
             ELSE AddGadgetImageButton (1,1,Commande[9]);
          END;
     Gl := EndGadgetList();
        Win := CreateWindow (0,0,320,256,WIDCMP,WFlags2,Gl,Scr,NIL);
        IF FOIS=5
        THEN Won := CreateWindow (pos+100,70,100,12,WIDCMP,WFlags2,NIL,
                                  Scr,NIL);
        ELSE Won := CreateWindow (pos+100,70,100,12,WIDCMP,WFlags,NIL,
                                  Scr,NIL);
        END;
        IF (Win # NIL) AND (Won # NIL)  THEN
         IF (ch=1) OR (ch=4) OR (ch=3)
```

```
                              THEN FRAME(Win);
                              ELSE FRAME2(Win);
        END;
        ECRIT(Win,ph);
        Rp := Win^.RPort;
        IF ch=2 THEN DrawImage(Rp^,poste,293,1); END;
        DrawImage (Rp^,Commande[0],110,212);
        DrawImage (Rp^,Commande[2],200,212);
        DrawImage (Rp^,Commande[1],140,212);
        DrawImage (Rp^,Commande[12],285,212);
        IF TableauParametres[1]=2 THEN
           DrawImage(Rp^,Commande[4],80,212);END;
        IF TableauParametres[0]=1 THEN
           DrawImage(Rp^,Commande[11],170,212); END;
        DrawImage (Win^.RPort^,Icone[im],0,0);
        IF (TableauParametres[6]=1) AND (ch=1) THEN LigneTemps(Win);END;
        IF CreateConsole (Won^) THEN
    IF ch=1 THEN InitJauge(JaugeX,JaugeY,Win);
                 MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
                 InitSomme(Win,5);
    END;
    IF (ch=4) OR (ch=3) THEN
       InitJauge(JaugeX,JaugeY,Win);
       MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
    END;
    Stotale:=0.;
    IJK:=1;
    WHILE (IJK<=FOIS) DO
       Recom:=FALSE; CompChiffre:=0; StrSomme:="";
       IF FOIS=1 THEN wClrScr (Won^);wMove (Won^,8,1);
                      PutStr (Won^,ADR("Frs"));
       END;
       wSetCursor (Won^,FALSE);
       IF TableauParametres[1] = 2 THEN SAY(pp); END;
       DoneSSC := FALSE;
       Done_next:=FALSE;
       WHILE (NOT Done_next) DO
            Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
            LOOP
              Msg := GetMsg(Win^.UserPort^);
              IF (Msg = NIL) THEN EXIT; END;
              ProcIMsg (Wp, Msg);
            END;
            IF DoneSSC THEN  IJK:=FOIS; Done_next:=TRUE; END;
       END;
       INC(IJK);
    END;
    DeleteConsole (Won^);
 END;
    EtatSortie := ES;
    Valeur := Stotale;
    CloseWindow(Won^);
    CloseWindow(Win^);
    FreeGadgetList (Gl^);
END;
END SaisieSommeCalcu;
END Scalcu.
```

IMPLEMENTATION MODULE SSBillets;

```
FROM SYSTEM IMPORT SHORT,BYTE, ADR;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,  GadgetPtr,IntuiMessagePtr,
  Gadget, WindowFlags, WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,
   CloseWindow, Image, DrawImage, ActivateWindow, EndGadget;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton, GlobalGadgetOpt;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM SimpleConsole IMPORT CreateConsole, DeleteConsole, PutStr, GetStr;
FROM SimpleConsoleCmds IMPORT wMove, wClrScr, wSetColor,
wSetCursor,wClrEndLine;
FROM Strings IMPORT  CopyString, CompareString, Relation,ConcatString,
   StringLength,LocateChar,ExtractSubString;
FROM VariablesGlobales IMPORT Nom, Str20,Str40,Icone,Commande,Max,
                              TableauParametres,Liquide,Scr,Str3;
FROM Help IMPORT SOS;
FROM Rasters IMPORT SetRast,RastPortPtr;
FROM RealConversions IMPORT ConvRealToString, Decimal,ConvStringToReal;
FROM Drawing IMPORT RectFill,SetAPen,Draw,Move;
FROM Parole IMPORT SAY;
FROM cadre IMPORT FRAME;
FROM Jauge IMPORT InitJauge,MAJ_Jauge,JaugeX,JaugeY;
FROM couleurs IMPORT noir,blanc,rouge,orange,jaune,vert,gris;
FROM Texte IMPORT ECRIT;
FROM Trace IMPORT trace;
FROM UTILS IMPORT PRINT;
FROM LireImage IMPORT LoadMonnaie,TabMonnaie;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Activate,Borderless};

TYPE
  Dernier = RECORD
      Type : Str3;
      X    : CARDINAL;
      Y    : CARDINAL;
      Last : REAL;
       END;
  NB = RECORD
      Nb : CARDINAL;
      val: REAL;
      pos,ligne,longueur,petit:CARDINAL;
      type:Str3;
  END;

VAR
  N:ARRAY [0..7] OF NB;
  Nombre,StrSomme,Strval : Str20;
  Trace : Str40;
  DoneSSB,OK,OKNombre,AuMoinsUn,ActivEff,QR,Quit,bi: BOOLEAN;
  Der : Dernier;
```

```
  Win,Won : WindowPtr;
  Monnaie : TabMonnaie;
  list6,list7, i,tab,car : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Somme : REAL;
  CH,Nbillets,couleur,sens,jj,mm,aa,Fait,I,pp,ph,im : INTEGER;
  Rp :RastPortPtr;
  ES:CHAR;
PROCEDURE Init;
BEGIN
     N[0].ligne:=10;   N[0].val:=5000.; N[0].pos:=33; N[0].longueur:=19;
     N[0].type:="BIL"; N[0].petit:=8;

     N[1].ligne:=10;   N[1].val:=1000.; N[1].pos:=48; N[1].longueur:=19;
     N[1].type:="BIL"; N[1].petit:=9;

     N[2].ligne:=12;   N[2].val:=500.;  N[2].pos:=63; N[2].longueur:=16;
     N[2].type:="BIL"; N[2].petit:=10;

     N[3].ligne:=12;   N[3].val:=100.;  N[3].pos:=78; N[3].longueur:=16;
     N[3].type:="BIL"; N[3].petit:=11;

     N[4].ligne:=10;   N[4].val:=50.;   N[4].pos:=93; N[4].longueur:=19;
     N[4].type:="PA";  N[4].petit:=4;

     N[5].ligne:=9;    N[5].val:=20.;   N[5].pos:=118;N[5].longueur:=20;
     N[5].type:="PA";  N[5].petit:=5;

     N[6].ligne:=12;   N[6].val:=5.;    N[6].pos:=142;N[6].longueur:=16;
     N[6].type:="PA";  N[6].petit:=6;

     N[7].ligne:=15;   N[7].val:=1.;    N[7].pos:=165;N[7].longueur:=13;
     N[7].type:="PA";  N[7].petit:=7;

END Init;

PROCEDURE TOTAL;
(*------------*)
BEGIN
     wClrScr (Won^);
     PutStr (Won^,ADR(" Total"));
     wMove (Won^,12,1);
     PutStr (Won^,ADR(StrSomme));
     wMove (Won^,20,1);
     PutStr (Won^,ADR("Frs"));
     wSetCursor (Won^,FALSE);
END TOTAL;

PROCEDURE GadgetHandlerBB(VAR w:Window;VAR Msg:MsgData;VAR Gad: Gadget);

VAR id:INTEGER;
BEGIN
     IF (Gad.GadgetID < 8) THEN
          id:=Gad.GadgetID;
          Init;
```

```
            Nbillets := Nbillets + 1;
            ActivEff := TRUE;
            N[id].Nb := N[id].Nb + 1;
            Somme := Somme + N[id].val;
            IF N[id].Nb <= N[id].ligne THEN
               Der.Y := N[id].pos;
               Der.X := 123 + ((N[id].Nb - 1) * N[id].longueur);
               Der.Type := N[id].type;
               Der.Last := N[id].val;
               AuMoinsUn := TRUE;
               Delay (25);
               DrawImage (Win^.RPort^,Monnaie[N[id].petit],Der.X,Der.Y)
            ELSE
               AuMoinsUn := FALSE;
               Der.Last := N[id].val
            END;
            ConvRealToString (Somme, StrSomme, 0, Decimal);
            TOTAL;
            IF (CH=1) OR (CH=3) OR (CH=4) THEN

MAJ_Jauge(JaugeX,JaugeY,Win,sens,Liquide,vert,Somme,couleur);
            END;
            ConvRealToString (N[id].val, Strval, 0, Decimal);
            Trace := "Il appuie sur ";
            ConcatString(Trace,Strval);
            ConcatString(Trace," Frs.");
            trace(Trace);
         ELSE
   CASE Gad.GadgetID OF

      8 : IF ActivEff THEN

       Trace := "Il l'efface ";
       trace( Trace);
       Delay(25);
       FOR i :=0 TO 7 DO
          IF Der.Last = N[i].val THEN N[i].Nb:=N[i].Nb-1; END;
       END;
       ActivEff := FALSE;
       IF ((CompareString (Der.Type,"BIL") = equal) AND (AuMoinsUn))
         THEN
                  SetAPen (Rp^,0);
                  RectFill (Rp^,Der.X,Der.Y,Der.X+19,Der.Y+13);
                  SetAPen (Rp^,1);
              IF (Der.X # 123) THEN
           FOR i:=0 TO 3 DO
              IF Der.Last = N[i].val
                THEN Der.Y := N[i].pos;
                     Der.X := 123 + ((N[i].Nb - 1) * N[i].longueur);
                     DrawImage
(Win^.RPort^,Monnaie[N[i].petit],Der.X,Der.Y)
                  END;
           END;

              END
         ELSIF ((CompareString (Der.Type,"PA") = equal) AND (AuMoinsUn))
              THEN SetAPen (Rp^,0);
                   IF ((Der.Last=5.) OR (Der.Last=1.)) THEN
```

```
                    RectFill (Rp^,Der.X+1,Der.Y,Der.X +16,Der.Y+16);
                 ELSE
                    RectFill (Rp^,Der.X,Der.Y,Der.X +20,Der.Y+20);
                 END;
                 SetAPen (Rp^,1);

            END;
       Somme := Somme - Der.Last;
       ConvRealToString (Somme, StrSomme, 0, Decimal);
       TOTAL;
       IF (CH=1) OR (CH=3) OR (CH=4) THEN
          MAJ_Jauge(JaugeX,JaugeY,Win,sens,Liquide,vert,Somme,couleur);
       END;
          END;
          |

     9:
    Trace := "Il appuie sur OK  ";
    trace(Trace);
    Delay(25);
    DoneSSB := TRUE; |

    10 :
      Trace := "Il appuie sur RECOMMENCER ";
      trace( Trace);

           Delay(25);
           SetAPen (Rp^,0);
           RectFill (Rp^,123,31,318,184);
           SetAPen (Rp^,1);
           Somme := 0.;
           ConvRealToString (Somme, StrSomme, 2, Decimal);
           TOTAL;
           IF (CH=1) OR (CH=3) OR (CH=4) THEN

MAJ_Jauge(JaugeX,JaugeY,Win,sens,Liquide,vert,Somme,couleur);
           END;
           FOR i:=0 TO 7 DO
              N[i].Nb:=0;
           END;
           Der.Last := 0.;
           Der.X := 150;
           AuMoinsUn := FALSE;
           ActivEff := FALSE; |

      11 : SAY(pp); |
      12:  Trace := "Il clique sur HELP ";
           trace(Trace);
           IF pp=2 THEN SOS(4); ELSE SOS(10); END;|
      13:  PRINT;
END;
  END;
END GadgetHandlerBB;
```

```
(*************************************************************************)
(*                        SaisieSommeBillets                             *)
(*************************************************************************)

PROCEDURE SaisieSommeBillets (ch:INTEGER; Quit:BOOLEAN; VAR Valeur:REAL;
                              VAR EtatSortie : CHAR;VAR poste:Image);


BEGIN
    CASE ch OF
    1: im:=0; pp:=2; ph:=2;
       couleur:=vert; sens:=1; |
    2: im:=2; pp:=5; ph:=5;
       couleur:=orange; sens:=2; |
    3: im:=1; pp:=23; ph:=13;
       couleur:=rouge; sens:=2; |
    4: im:=0; pp:=24; ph:=14;                    (* 1 Recette  *)
       couleur:=vert; sens:=1; |
    5: im:=3; pp:=5; ph:=5; couleur:=orange;|
    6: im:=3; pp:=20; ph:=16; couleur:=orange;|
    END;
    CH:=ch;
    ES := "A"; QR := Quit; Somme := 0.;
    FOR i:=0 TO 7 DO
       N[i].Nb:=0;
    END;
    Der.Last := 0.; Der.X := 150; AuMoinsUn := FALSE; ActivEff := FALSE;
    Nbillets := 0;
     WITH Wp DO
      procGadgetUp := GadgetHandlerBB;
    END;
       LoadMonnaie(Monnaie);
       BeginGadgetList();
         AddGadgetImageButton (80,35,Monnaie[0]);
         AddGadgetImageButton (80,60,Monnaie[1]);
         AddGadgetImageButton (80,85,Monnaie[2]);
         AddGadgetImageButton (80,110,Monnaie[3]);
         AddGadgetImageButton (99,162,Monnaie[4]);
         AddGadgetImageButton (98,135,Monnaie[5]);
         AddGadgetImageButton (80,164,Monnaie[6]);
         AddGadgetImageButton (81,139,Monnaie[7]);
         AddGadgetImageButton (130,212,Commande[0]);
         AddGadgetImageButton (220,212,Commande[2]);
         AddGadgetImageButton (160,212,Commande[1]);
         IF TableauParametres[1]=2
            THEN AddGadgetImageButton (100,212,Commande[4]);
            ELSE AddGadgetImageButton (1,1,Commande[9]); END;
         AddGadgetImageButton (285,212,Commande[12]);
         IF TableauParametres[0]=1
            THEN AddGadgetImageButton (190,212,Commande[11]);
            ELSE AddGadgetImageButton (1,1,Commande[9]); END;

        Gl := EndGadgetList();
         IF (Gl # NIL) THEN
       Win := CreateWindow (0,0,320,256,WIDCMP,WFlags2,Gl,Scr,NIL);
       Won := CreateWindow (122,190,196,15,WIDCMP,WFlags2,NIL,Scr,NIL);
       IF (Win # NIL) AND (Won # NIL) THEN
         FRAME(Win);
```

```
        ECRIT(Win,ph);
        Rp := Win^.RPort;
        IF ch=2 THEN DrawImage(Rp^,poste,293,1); END;
        DrawImage (Rp^,Commande[0],130,212);
        DrawImage (Rp^,Commande[2],220,212);
        DrawImage (Rp^,Commande[1],160,212);
        DrawImage (Rp^,Commande[4],100,212);
        DrawImage (Rp^,Commande[12],285,212);
        IF TableauParametres[0]=1
           THEN DrawImage (Rp^,Commande[11],190,212);
        END;
        IF (CH=1) OR (CH=3) OR (CH=4) THEN
              InitJauge(JaugeX,JaugeY,Win);
              MAJ_Jauge(JaugeX,JaugeY,Win,0,Liquide,vert,0.,0);
        END;
      IF CreateConsole (Won^) THEN
           TOTAL;
           Rp := Win^.RPort;
           DrawImage (Win^.RPort^,Icone[im],0,0);
           SetAPen(Rp^,1);
           Move (Rp^,120,30);
           Draw (Rp^,120,210);
           Move (Rp^,121,185);
           Draw (Rp^,319,185);
           IF TableauParametres[1] = 2 THEN SAY(pp);  END;
      END;
      DoneSSB := FALSE;
      WHILE (NOT DoneSSB) DO
        Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
        LOOP
            Msg := GetMsg(Win^.UserPort^);
            IF (Msg = NIL) THEN EXIT; END;
            ProcIMsg (Wp, Msg);
        END;
      END;
      DeleteConsole (Won^);
    END;
    EtatSortie := ES;
    Valeur := Somme;
    CloseWindow(Won^);
    CloseWindow(Win^);
  END;
  FreeGadgetList (Gl^);
END SaisieSommeBillets;
END SSBillets.
```

IMPLEMENTATION MODULE ModPD;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM Tasks IMPORT SignalSet,Wait;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow
  NewWindow, Image,  DrawImage, ActivateWindow;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT RectFill,SetAPen,Draw,Move;
FROM VariablesGlobales IMPORT Str40,
Scr,TableauParametres,Commande,Icone,
                                  Liquide,Str20,Win,Cash;
FROM cadre IMPORT FRAME;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM Jauge IMPORT  InitJauge,MAJ_Jauge,JaugeX,JaugeY,cdr;
FROM LireImage IMPORT
POSTE,TP,LoadPostes,LireMois,TabMois,LoadFreq,TabFreq;
FROM ModPoste IMPORT Poste;
FROM ModSRO IMPORT SRO;
FROM ModFreq IMPORT Frequence;
FROM IO IMPORT TabPD,NbPD,TABPOSTE,LIRENOMS_POSTES,Str15,TPD;
FROM couleurs IMPORT vert,orange,rouge,gris;
FROM annee IMPORT DATE;
FROM SCalcu IMPORT SaisieSommeCalcu;
FROM SSNombre IMPORT SaisieSommeNbre;
FROM SSBillets IMPORT SaisieSommeBillets;
FROM AuxPD IMPORT Somme2,LigneTemps,CONF,ENLEVERPD,DeterminerPD,VERIF;
FROM Strings IMPORT ConcatString;
FROM Visu IMPORT Visualisation;
FROM InOut IMPORT WriteString;
FROM UTILS IMPORT PRINT;
FROM Help IMPORT SOS;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate,Borderless};

VAR
  N:CARDINAL;
  i,PosteSelectionne,mois_actuel,jj,aa,K,RF,ancien:INTEGER;
  Trace : Str40;
  ok,Done,bi,oke,rep,oksomme,okposte: BOOLEAN;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  PDFreq:TPD;
  NbFreq:ARRAY[0..6] OF INTEGER;
  Mois:TabMois;
```

```
  freq:TabFreq;
  NomPostes:TABPOSTE;
  image,po:Image;
  Val,S:REAL;
  es,ch,CH:CHAR;
  nom:Str15;
  Pos:ARRAY[1..20] OF RECORD
                          X,Y:INTEGER;
                      END;
PROCEDURE ListeGadgets;
(*-------------------*)
BEGIN
    BeginGadgetList();
        AddGadgetImageButton (235,215,Commande[2]);
        IF (NbFreq[RF]#0) AND (RF=6)
           THEN AddGadgetImageButton (175,215,Commande[7]);
                AddGadgetImageButton (145,215,Commande[8]);
           ELSE AddGadgetImageButton (1,1,Commande[9]);
                AddGadgetImageButton (1,1,Commande[9]);
        END;
        AddGadgetImageButton (115,215,Commande[6]);
        AddGadgetImageButton (JaugeX,JaugeY,Commande[10]);
        IF TableauParametres[1]=2
           THEN AddGadgetImageButton(85,215,Commande[4]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        IF TableauParametres[0]=1
           THEN AddGadgetImageButton(205,215,Commande[11]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        AddGadgetImageButton (285,215,Commande[12]);
        S:=0.;
        FOR i:=1 TO NbFreq[RF] DO
           N:=PDFreq[RF][i].NoPoste;
           IF (TableauParametres[9]=2) AND (RF=0)
                  THEN S:=S+PDFreq[RF][i].SommeDprt;
                  ELSE S:=S+PDFreq[RF][i].Somme;
           END;
           AddGadgetImageButton(Pos[i].X,Pos[i].Y,POSTE[N]);
        END;
        IF RF=6 THEN Cash:=Liquide+S; END;
        FOR i:=1 TO NbFreq[RF] DO
           AddGadgetImageButton(Pos[i].X+27,Pos[i].Y,POSTE[12]);
        END;
     Gl := EndGadgetList();
END ListeGadgets;

PROCEDURE REL(n:INTEGER; VAR i:INTEGER);
(*-----------------------------------*)

VAR ok:BOOLEAN;

BEGIN
     i:=0;
     ok:=FALSE;
     WHILE ((i<NbPD) AND (NOT ok)) DO
          i:=i+1;
          IF ((PDFreq[RF][n].NoPoste= TabPD[i].NoPoste) AND
```

```
                (PDFreq[RF][n].Somme=   TabPD[i].Somme)) THEN ok:=TRUE;
END;
    END;
END REL;

PROCEDURE SaisieSomme(Q:BOOLEAN;VAR Val:REAL;VAR ES:CHAR;
                      VAR poste:Image);
(*****************************************************)

BEGIN
     CASE TableauParametres[4] OF
         2: SaisieSommeCalcu(2,Q,Val,ES,poste);
         0: SaisieSommeNbre (2,Q,Val,ES,poste);
         1: SaisieSommeBillets(2,Q,Val,ES,poste);
     END;
END SaisieSomme;

PROCEDURE GadgetHandlerME(VAR w:Window;VAR Msg:MsgData; VAR Gad: Gadget)

BEGIN
      IF (Gad.GadgetID<8) THEN
        CASE Gad.GadgetID OF
         0: Trace:="Il clique sur OK ";
            trace(Trace);CH:="A";  Done:=TRUE;
         1: Trace:="Il clique sur Jeter ";
            trace(Trace);
            IF (NOT okposte) AND (NOT oksomme)
              THEN Trace:="Il n'a pas sélectionné de poste ";
                   trace(Trace);
              ELSE IF (okposte) AND ( TabPD[PosteSelectionne].RO="O")
                      THEN okposte:=FALSE;
                           Liquide:=Liquide+
                                    TabPD[PosteSelectionne].Somme;
                           ENLEVERPD(PosteSelectionne);
                      ELSE oksomme:=FALSE;
                           Liquide:=Liquide+
                                    TabPD[PosteSelectionne].Somme;
                           TabPD[PosteSelectionne].Faite:="O";
                           TabPD[PosteSelectionne].Quand:=mois_actuel;
                   END;
            END; Done:=TRUE;
         2: Trace := "Il clique sur Dépenser ";
            trace(Trace);
            IF (NOT okposte) AND (NOT oksomme)
               THEN Trace:="Il n'a pas sélectionné de poste ";
                    trace(Trace);
               ELSE CONF(POSTE[TabPD[PosteSelectionne].NoPoste],
                                 TabPD[PosteSelectionne].Somme,rep);
                    IF NOT rep THEN
                       po:=POSTE[TabPD[PosteSelectionne].NoPoste];
                       SaisieSomme(TRUE,Val,es,po);
                       IF es="A"
                          THEN IF (TabPD[PosteSelectionne].Frequence=0)
                                  AND
                                  (TabPD[PosteSelectionne].RO="R") AND
                                   oksomme
                                  THEN TabPD[PosteSelectionne].Somme:=
                                       TabPD[PosteSelectionne].Somme-Val
```

```
                              ELSE
                                   Liquide:=Liquide-Val+
                                   TabPD[PosteSelectionne].Somme;
                              END;
                         END;
                    END;
               IF (TabPD[PosteSelectionne].Frequence=0) AND
                  (TabPD[PosteSelectionne].RO="R") AND (oksomme)
                  THEN
                  ELSE TabPD[PosteSelectionne].Faite:="O";
                       TabPD[PosteSelectionne].Quand:=mois_actuel;
                       IF TabPD[PosteSelectionne].RO="O"
                          THEN ENLEVERPD(PosteSelectionne);
                       END;
               END;
               okposte:=FALSE; oksomme:=FALSE;
      END; Done:=TRUE;|
3:    Trace := "Il clique sur Créer  ";
      trace(Trace);
      NbPD:=NbPD+1;
      Trace:="Il arrive dans la librairie ";
      trace(Trace);
      TabPD[NbPD].NoPoste:=3;
      Poste (image,TabPD[NbPD].NoPoste);
      SaisieSomme(TRUE,Val,es,image);
      IF es="A" THEN
         TabPD[NbPD].Somme:=Val;
         TabPD[NbPD].SommeDprt:=Val;
         TabPD[NbPD].Faite:="N";
         TabPD[NbPD].Quand:=mois_actuel;
         Trace:="Il arrive au choix de la périodicité ";
         trace(Trace);
         SRO (image,TabPD[NbPD].RO);
         IF TableauParametres[9]=2 THEN
         oke:=TRUE;
         IF (TabPD[NbPD].RO="R")
         THEN Trace:="Sa dépense se fait :  ";
              trace(Trace);
              Frequence(TabPD[NbPD].Frequence,
                        TabPD[NbPD].MoisDepart,image);
              CASE TabPD[NbPD].Frequence OF
                1: VERIF(TabPD[NbPD].MoisDepart,2,oke);|
                2: VERIF(TabPD[NbPD].MoisDepart,6,oke);|
                4: VERIF(TabPD[NbPD].MoisDepart,3,oke);|
                5: IF TabPD[NbPD].MoisDepart#mois_actuel THEN
                                              oke:=FALSE;END;
              END;
         ELSE TabPD[NbPD].Frequence:=0;
              TabPD[NbPD].MoisDepart:=0;
         END;
         IF oke THEN Liquide:=Liquide-TabPD[NbPD].Somme; END;
         ELSE Liquide:=Liquide-TabPD[NbPD].Somme;
              TabPD[NbPD].Frequence:=0; END;
      ELSE NbPD:=NbPD-1;
      END;
      Done:=TRUE; |
4:    Trace:="Il demande de visualiser ses PD ";
      trace(Trace);
```

```
            Visualisation(RF,PDFreq,NbFreq[RF]);  |
         5: IF TableauParametres[9]=2 THEN SAY(3);SAY(15);
                                      ELSE SAY(3);SAY(9);
            END; |
         6: PRINT; |
         7: Trace:="Il clique sur HELP ";
            trace(Trace);
            IF NbFreq[RF]=0 THEN IF RF=6 THEN SOS(5);ELSE SOS(15); END;
                            ELSE IF RF=6 THEN SOS(12); ELSE SOS(16); END
            END;|
      END;
   ELSE IF (NbFreq[RF]>0)
         THEN FOR i:=1 TO NbFreq[RF] DO
                N:=i+7;
                IF (N=Gad.GadgetID) THEN
                  IF ancien<99
                     THEN cdr(Win,1+Pos[ancien].X,1+Pos[ancien].Y,
                              1,TRUE);
                  END;
                  ancien:=i;
                  REL(i,PosteSelectionne);
                  cdr(Win,1+Pos[i].X,1+Pos[i].Y,1,FALSE);
                  okposte:=TRUE; oksomme:=FALSE;
                  Trace:="Il clique sur ";
                  ConcatString(Trace,NomPostes[
                                      PDFreq[RF][i].NoPoste]);
                  ConcatString(Trace," ");
                  trace(Trace);
                END;
              END;
              FOR i:=1 TO NbFreq[RF] DO
                 N:=i+7+NbFreq[RF];
                 IF (N=Gad.GadgetID) THEN
                   IF ancien<99
                      THEN cdr(Win,1+Pos[ancien].X,1+Pos[ancien].Y,
                               1,TRUE);
                   END;
                   ancien:=i;
                   REL(i,PosteSelectionne);
                   cdr(Win,1+Pos[i].X,1+Pos[i].Y,1,FALSE);
                   oksomme:=TRUE; okposte:=FALSE;
                   Trace:="Il clique sur la somme ";
                   ConcatString(Trace,
                                NomPostes[PDFreq[RF][i].NoPoste]);
                   ConcatString(Trace," ");
                   trace(Trace);
                END;
            END;
         END;
      END;
END GadgetHandlerME;
```

```
(*****************************************************************)
(*                              PD                               *)
(*****************************************************************)

PROCEDURE PD(FR:INTEGER;VAR ES:CHAR);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   Pos[1].X:=81;  Pos[6].X:=141;  Pos[11].X:=201;
   Pos[1].Y:=75;  Pos[6].Y:=115;  Pos[11].Y:=155;
   Pos[2].X:=81;  Pos[7].X:=141;  Pos[12].X:=261;
   Pos[2].Y:=115; Pos[7].Y:=155;  Pos[12].Y:=35;
   Pos[3].X:=81;  Pos[8].X:=201;  Pos[13].X:=261;
   Pos[3].Y:=155; Pos[8].Y:=35;   Pos[13].Y:=75;
   Pos[4].X:=141; Pos[9].X:=201;  Pos[14].X:=261;
   Pos[4].Y:=35;  Pos[9].Y:=75;   Pos[14].Y:=115;
   Pos[5].X:=141; Pos[10].X:=201; Pos[15].X:=261;
   Pos[5].Y:=75;  Pos[10].Y:=115; Pos[15].Y:=155;
   CH:="X"; ancien:=99;
   RF:=FR;
   LoadPostes(POSTE);
   LoadFreq(freq);
   LIRENOMS_POSTES(NomPostes);
   LireMois(Mois);
   FOR i:=0 TO 6 DO
      DeterminerPD(i,PDFreq,NbFreq[i]);
   END;
   ListeGadgets;
     IF (Gl # NIL) THEN
         Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
         IF (Win # NIL) THEN
             FRAME(Win);
             IF TableauParametres[9]=2
                THEN IF TableauParametres[1] = 2 THEN SAY(3);SAY(15);
                     END;
                     ECRIT(Win,3);
                ELSE IF TableauParametres[1] = 2 THEN SAY(3);SAY(9); END;

                     ECRIT(Win,8);
             END;
             DrawImage (Win^.RPort^,Icone[2],0,0);
             DATE(jj,mois_actuel,aa,bi);
             IF TableauParametres[6]=1 THEN LigneTemps(Win); END;
             InitJauge(JaugeX,JaugeY,Win);
             IF FR=6
                THEN
MAJ_Jauge(JaugeX,JaugeY,Win,3,Cash,orange,Liquide,vert);
                     IF TableauParametres[9]=2 THEN
                        DrawImage(Win^.RPort^,Mois[mois_actuel-1],243,1);
                     END;
                ELSE IF (FR=0) OR (FR=3)
                        THEN MAJ_Jauge(JaugeX,JaugeY,Win,3,Cash,orange,
                                       Liquide,vert);
                        ELSE MAJ_Jauge(JaugeX,JaugeY,Win,1,Liquide,vert,
                                       0.,0);
                     END;
```

```
                    DrawImage(Win^.RPort^,freq[FR],263,1);
                END;
                DrawImage (Win^.RPort^,Commande[2],235,215);
                DrawImage (Win^.RPort^,Commande[12],285,215);
                IF NbFreq[FR]#0
                   THEN DrawImage (Win^.RPort^,Commande[7],175,215);
                        DrawImage (Win^.RPort^,Commande[8],145,215);
                END;
                DrawImage (Win^.RPort^,Commande[6],115,215);
                IF (TableauParametres[1]=2)
                   THEN DrawImage (Win^.RPort^,Commande[4],85,215);
                END;
                IF (TableauParametres[0]=1)
                   THEN DrawImage (Win^.RPort^,Commande[11],205,215);
                END;
                DrawImage (Win^.RPort^,POSTE[40],80,35);
                Somme2(Win,105,35,Liquide);
                FOR i:=1 TO NbFreq[RF] DO
                   Somme2(Win,25+Pos[i].X,Pos[i].Y,PDFreq[RF][i].Somme);
                   IF ((TableauParametres[9]=2) AND (RF=0)) THEN
                     FOR K:=1 TO 4 DO
                       Somme2(Win,30*(K-1)+25+Pos[i].X,Pos[i].Y,
                              PDFreq[RF][i].SommeDprt/4.);
                     END;
                  END;
                END;
                SetAPen(Win^.RPort^,0);
                Done:=FALSE;
                ok:=FALSE;
                WHILE (NOT Done) DO
                   Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
                   LOOP
                      Msg := GetMsg(Win^.UserPort^);
                      IF (Msg = NIL) THEN EXIT; END;
                      ProcIMsg (Wp, Msg);
                   END;
                END;
                ES:=CH;
                CloseWindow (Win^);
             END;
             FreeGadgetList (Gl^);
       END;
  END PD;
END ModPD.
```

IMPLEMENTATION MODULE ModMenu;

```
FROM SYSTEM IMPORT BYTE, ADR, SHORT;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow,
  NewWindow, Image,  DrawImage, ActivateWindow,IntuiText,IntuiTextPtr;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM VariablesGlobales IMPORT Str20,Str40, Scr,TableauParametres,
                              Commande,cmap;
FROM FindImages IMPORT ImageDescTablePtr, FindImageTable;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM Help IMPORT SOS;
FROM UTILS IMPORT PRINT;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
 Trace : Str40;
  ok: BOOLEAN;
  Win : WindowPtr;
  icone: ARRAY [0..7] OF Image;
  ImgPtr: ImageDescTablePtr;
  ImgCount,Indice, i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  fonc : Str20;

PROCEDURE GadgetHandlerME(VAR w:Window;VAR Msg:MsgData;VAR Gad:Gadget);

BEGIN
    ok:=TRUE;
  CASE Gad.GadgetID OF
      0: fonc := "recettes";   |
      1: fonc := "depenses";   |
      2: fonc := "economies";  |
      3: fonc := "prevision-recettes"; |
      4: fonc := "prevision-depenses"; |
      5: fonc := "test";  |
      6: fonc := "bilan";|
      7: fonc := "etat des finances"; |
      8:      Trace := "Il appuie sur Quitter ";
             trace (Trace);
              fonc := "fin";   |
      9: SOS(6); ok:=FALSE;|
```

```
      10: PRINT; ok:=FALSE; |
      11: SAY(14); ok:=FALSE;
    END;
END GadgetHandlerME;


(*************************************************************************)
(*                                  Menu                                 *)
(*************************************************************************)

PROCEDURE Menu (VAR FonctionChoisie : Str20);

BEGIN

      WITH Wp DO
         procGadgetUp := GadgetHandlerME;
      END;

   ImgPtr := FindImageTable (22222222H,ImgCount);
   IF (ImgPtr # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH icone [Indice] DO
       LeftEdge   := 0;
       TopEdge    := 0;
       Width      := ImgPtr^[Indice].Width;
       Height     := ImgPtr^[Indice].Height;
       Depth      := ImgPtr^[Indice].Depth;
       ImageData  := ImgPtr^[Indice].Data;
       PlanePick  := BYTE(31);
       PlaneOnOff := BYTE(31);
       NextImage  := NIL;
      END;
      Indice :=Indice+1;
     END;
   END;
   BeginGadgetList();
        AddGadgetImageButton (80,35,icone[0]);
        AddGadgetImageButton (180,35,icone[1]);
        AddGadgetImageButton (1,1,Commande[9]);
        AddGadgetImageButton (20,94,icone[3]);
        AddGadgetImageButton (120,94,icone[4]);
        AddGadgetImageButton (220,94,icone[5]);
        AddGadgetImageButton (80,153,icone[6]);
        AddGadgetImageButton (180,153,icone[7]);
        AddGadgetImageButton (190,215,Commande[3]);
        AddGadgetImageButton (285,215,Commande[12]);
     IF TableauParametres[0]=1
         THEN AddGadgetImageButton(160,215,Commande[11]);
         ELSE AddGadgetImageButton(1,1,Commande[9]);
     END;
     IF TableauParametres[1]=2
         THEN AddGadgetImageButton(130,215,Commande[4]);
         ELSE AddGadgetImageButton(1,1,Commande[9]);
     END;
        Gl := EndGadgetList();
        IF (Gl # NIL) THEN
         Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
```

```
        IF (Win # NIL) THEN
            FRAME2(Win);
            IF TableauParametres[1] = 2 THEN SAY(14); END;
            ECRIT(Win,9);
            DrawImage (Win^.RPort^,Commande[3],190,215);
            DrawImage (Win^.RPort^,Commande[12],285,215);
            IF (TableauParametres[0] = 1) THEN
               DrawImage (Win^.RPort^,Commande[11],160,215);
            END;
            IF (TableauParametres[1] = 2) THEN
               DrawImage (Win^.RPort^,Commande[4],130,215);
            END;
            ok:=FALSE;
            WHILE (NOT ok) DO
               Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                   Msg := GetMsg(Win^.UserPort^);
                   IF (Msg = NIL) THEN EXIT; END;
                   ProcIMsg (Wp, Msg);
               END;
            END;
            FonctionChoisie := fonc;
            CloseWindow (Win^);
          END;
          FreeGadgetList (Gl^);
      END;
  END Menu;
END ModMenu.
```

IMPLEMENTATION MODULE ModMois;

```
FROM SYSTEM IMPORT BYTE, ADR, SHORT;
FROM MathLib0 IMPORT entier,real;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr, GadgetPtr,Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow
NewWindow, Image,DrawImage, ActivateWindow,GadgetFlagsSet;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList
  AddGadgetTextButton,  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM Strings IMPORT  CopyString, CompareString, Relation,ConcatString,
     StringLength,LocateChar,ExtractSubString;
FROM Conversions IMPORT ConvNumberToString;
FROM Rasters IMPORT RastPortPtr,RastPort;
FROM VariablesGlobales IMPORT
ST10,Str20,Str40,TableauParametres,Commande,Scr;
FROM RealConversions IMPORT ConvRealToString, RealToStringFormat,
                    ConvStringToReal;
FROM Drawing IMPORT SetAPen,Move,Draw,RectFill;
FROM cadre IMPORT FRAME2;
FROM annee IMPORT DATE,NOMMOIS,RechJour,NbJours,TABNJ;
FROM Texte IMPORT ECRIT;
FROM Text IMPORT Text;
FROM Parole IMPORT SAY;
FROM UTILS IMPORT PRINT;
FROM Trace IMPORT trace;
FROM Help IMPORT SOS;
FROM LireImage IMPORT LireMois,TabMois;
FROM Couleurs IMPORT bleu,bleuclair,gris;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate,Borderless};

VAR
  Trace,Phrase : Str40;
  DoneMois,Bi,DEP,OKMois : BOOLEAN;
  Win : WindowPtr;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  St:ARRAY[0..2] OF CHAR;
  i,j,m,a,MM,Num,mActuel,mPrec,Ligne,N,JJ:INTEGER;
  Mois:TabMois;
  TabJours:TABNJ;
  MOIS:ST10;
  Jour:ARRAY[1..7] OF ST10;
```

```
PROCEDURE GadgetHandlerMO (VAR w: Window;VAR Msg : MsgData;VAR Gad:
Gadget);
BEGIN
     IF Gad.GadgetID<4 THEN
        CASE Gad.GadgetID OF
             0: IF OKMois THEN DoneMois:=TRUE; END;
                Trace:="Il clique sur OK";
                trace(Trace);|
             1: IF DEP THEN SAY(17);
                       ELSE SAY(18);
                END; |
             2: PRINT; |
             3: IF DEP THEN SOS(18);
                       ELSE SOS(19);
                END; |
        END;
        ELSE Num:=Gad.GadgetID-3;
             IF Num<TabJours[mPrec]+1 THEN MM:=mPrec;
                                      ELSE MM:=mActuel;
                                           Num:=Num-TabJours[mPrec];
             END;
             NOMMOIS(MOIS,MM+1);
             RechJour(Num,MM+1,a,N);
             CopyString(Phrase,Jour[N]);
             ConcatString(Phrase," ");
             ConvNumberToString(St,LONGINT(Num),FALSE,10,2," ");
             ConcatString(Phrase,St);
             ConcatString(Phrase," ");
             ConcatString(Phrase,MOIS);
             SetAPen(Win^.RPort^,0);
             RectFill(Win^.RPort^,1,180,318,200);
             IF (MM=mActuel) AND (Num>j) THEN OKMois:=FALSE; ELSE
             Win^.RPort^.FgPen:=BYTE(bleu);
             Win^.RPort^.BgPen:=BYTE(0);
             SetAPen(Win^.RPort^,bleu);
             Move(Win^.RPort^,160-((8*StringLength(Phrase)) DIV 2),190);
             Text(Win^.RPort^,ADR(Phrase),StringLength(Phrase));
             OKMois:=TRUE;
             JJ:=Num;
             END;
             Trace:="Il choisit: Le ";
             ConcatString(Trace,Phrase);
             trace(Trace);
    END;
END GadgetHandlerMO;

(*************************************************************************)
(*                                Quand                                  *)
(*************************************************************************)


PROCEDURE QUAND ( VAR JJ,MM,AA:INTEGER);

VAR S1,S2,S3:ARRAY[1..7] OF CHAR;
BEGIN
    WITH Wp DO
      procGadgetUp := GadgetHandlerMO;
    END;
```

```
S1:="LMMJVSD";
S2:="uaeeeai";
S3:="nrrunmm";
IF AA=0 THEN DEP:=TRUE;
        ELSE DEP:=FALSE;
END;
OKMois:=FALSE;
DATE(j,m,a,Bi);
IF m=1 THEN mActuel:=0;  mPrec:=11;
       ELSE mActuel:=m-1;mPrec:=m-2;
END;
NbJours(TabJours);
LireMois(Mois);
Jour[1]:="Lundi";
Jour[2]:="Mardi";
Jour[3]:="Mercredi";
Jour[4]:="Jeudi";
Jour[5]:="Vendredi";
Jour[6]:="Samedi";
Jour[7]:="Dimanche";
BeginGadgetList();
    AddGadgetImageButton (175,215,Commande[2]);
    IF TableauParametres[1]=2
       THEN AddGadgetImageButton(115,215,Commande[4]);
       ELSE AddGadgetImageButton(1,1,Commande[9]);
    END;
    IF TableauParametres[0]=1
       THEN AddGadgetImageButton(145,215,Commande[11]);
       ELSE AddGadgetImageButton(1,1,Commande[9]);
    END;
    AddGadgetImageButton (285,215,Commande[12]);
    RechJour(1,mActuel,a,N);
    Ligne:=0;
    FOR i:=1 TO TabJours[mPrec] DO
       AddGadgetTextButton(5+20*(N-1),100+Ligne*13,ADR(" "));
       IF N=7 THEN N:=1; Ligne:=Ligne+1;
              ELSE N:=N+1;
       END;
    END;
    RechJour(1,m,a,N);
    Ligne:=0;
    FOR i:=1 TO TabJours[m-1] DO
       AddGadgetTextButton(165+20*(N-1),100+Ligne*13,ADR(" "));
       IF N=7 THEN N:=1; Ligne:=Ligne+1;
              ELSE N:=N+1;
       END;
    END;
G1:=EndGadgetList();
IF (G1 # NIL) THEN
   Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,G1,Scr,NIL);
   IF (Win # NIL) THEN
      FRAME2(Win);
      IF DEP THEN ECRIT(Win,11);
             ELSE ECRIT(Win,12);
      END;
      Win^.RPort^.FgPen:=BYTE(0);
      Win^.RPort^.BgPen:=BYTE(bleu);
      SetAPen(Win^.RPort^,0);
```

```
  FOR i:=1 TO 7 DO
      Move(Win^.RPort^,5+20*(i-1),77);
      Text(Win^.RPort^,ADR(S1[i]),1);
      Move(Win^.RPort^,165+20*(i-1),77);
      Text(Win^.RPort^,ADR(S1[i]),1);
      Move(Win^.RPort^,9+20*(i-1),85);
      Text(Win^.RPort^,ADR(S2[i]),1);
      Move(Win^.RPort^,169+20*(i-1),85);
      Text(Win^.RPort^,ADR(S2[i]),1);
      Move(Win^.RPort^,13+20*(i-1),93);
      Text(Win^.RPort^,ADR(S3[i]),1);
      Move(Win^.RPort^,173+20*(i-1),93);
      Text(Win^.RPort^,ADR(S3[i]),1);
  END;
  DrawImage (Win^.RPort^,Mois[mPrec],55,35);
  DrawImage (Win^.RPort^,Mois[mActuel],190,35);
  RechJour(1,mActuel,a,N);
  Ligne:=0;
  Win^.RPort^.FgPen:=BYTE(0);
  FOR i:=1 TO TabJours[mPrec] DO
   ConvNumberToString(St,LONGINT(i),FALSE,10,2," ");
   IF (N=6) OR (N=7) THEN  Win^.RPort^.BgPen:=BYTE(bleuclair);
                           SetAPen(Win^.RPort^,bleuclair);
                     ELSE  Win^.RPort^.BgPen:=BYTE(bleu);
                           SetAPen(Win^.RPort^,bleu);
   END;
   RectFill(Win^.RPort^,5+20*(N-1),100+Ligne*13,20+20*(N-1),
                                    109+Ligne*13);
   Move(Win^.RPort^,5+20*(N-1),107+Ligne*13);
   SetAPen(Win^.RPort^,0);
   Text(Win^.RPort^,ADR(St),StringLength(St));
   IF N=7 THEN N:=1; Ligne:=Ligne+1;
          ELSE N:=N+1;
   END;
END;
  RechJour(1,m,a,N);
  Ligne:=0;
  Win^.RPort^.FgPen:=BYTE(0);
  FOR i:=1 TO TabJours[m-1] DO
   ConvNumberToString(St,LONGINT(i),FALSE,10,2," ");
   IF (N=6) OR (N=7) THEN  IF i<j
                              THEN Win^.RPort^.BgPen:=
                                   BYTE(bleuclair);
                                   SetAPen(Win^.RPort^,
                                           bleuclair);
                              ELSE Win^.RPort^.BgPen:=
                                   BYTE(gris+3);
                                   SetAPen(Win^.RPort^,gris+3);
                           END;
                     ELSE  IF i<j
                              THEN Win^.RPort^.BgPen:=BYTE(bleu);
                                   SetAPen(Win^.RPort^,bleu);
                              ELSE Win^.RPort^.BgPen:=BYTE(gris);
                                   SetAPen(Win^.RPort^,gris);
                           END;
   END;
   IF i=j THEN Win^.RPort^.BgPen:=BYTE(0);
               SetAPen(Win^.RPort^,0);
```

```
            END;
            RectFill(Win^.RPort^,165+20*(N-1),100+Ligne*13,180+20*(N-1),
                                                  109+Ligne*13);
            Move(Win^.RPort^,165+20*(N-1),107+Ligne*13);
            SetAPen(Win^.RPort^,0);
            IF i=j THEN SetAPen(Win^.RPort^,1);END;
            Text(Win^.RPort^,ADR(St),StringLength(St));
            IF N=7 THEN N:=1; Ligne:=Ligne+1;
                   ELSE N:=N+1;
            END;
         END;

          DrawImage (Win^.RPort^,Commande[2],175,215);
          IF TableauParametres[1]=2
           THEN DrawImage (Win^.RPort^,Commande[4],115,215); END;
          IF TableauParametres[0]=1
           THEN DrawImage(Win^.RPort^,Commande[11],145,215); END;
          DrawImage(Win^.RPort^,Commande[12],285,215);

          DoneMois := FALSE;
          WHILE (NOT DoneMois) DO
             Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
             LOOP
                Msg := GetMsg(Win^.UserPort^);
                IF (Msg = NIL) THEN EXIT; END;
                ProcIMsg (Wp, Msg);
             END;
          END;
          CloseWindow(Win^);
       END;
       FreeGadgetList (Gl^);
END ;
END QUAND;

END ModMois.
```

IMPLEMENTATION MODULE Jauge;

```
FROM Intuition IMPORT WindowPtr;
FROM SYSTEM IMPORT ADR,BYTE ;
FROM Drawing IMPORT RectFill,Move,SetAPen,Draw;
FROM Rasters IMPORT RastPortPtr,RastPort;
FROM Text IMPORT Text;
FROM Conversions IMPORT ConvNumberToString;
FROM Strings IMPORT StringLength;
FROM Couleurs IMPORT gris,vert,orange,rouge;
FROM MathLib0 IMPORT entier;
FROM RealConversions IMPORT ConvRealToString,RealToStringFormat;
FROM VariablesGlobales IMPORT Max,Liquide;

CONST L=30;
      l=168;
      longueur=168.;

TYPE Str20=ARRAY [0..19] OF CHAR;

VAR
   rp:RastPortPtr;
   toto,tata,titi:INTEGER;
   DEBORDE:BOOLEAN;
   bidon:REAL;

PROCEDURE InitJauge(x,y:INTEGER;win:WindowPtr);
(*-----------------------------------------*)
BEGIN
   rp:=win^.RPort;
   SetAPen(rp^,1);
   Move(rp^,x,y+2);
   Draw(rp^,x,y+1);
   Draw(rp^,x+L,y+1);
   Draw(rp^,x+L,y+2);
   IF (x=2) AND (y=40) THEN SetAPen(rp^,gris);
                            RectFill(rp^,x+L+1,y,77,y+1+2);
                            RectFill(rp^,x+1,y,x+L-1,y+1-1);
   END;
END InitJauge;

PROCEDURE MAJ_Jauge(x,y:INTEGER;win:WindowPtr;sens:CARDINAL;s1:REAL;
                    c1:CARDINAL;s2:REAL;c2:CARDINAL);
(*-------------------------------------------------------------*)

VAR ss:ARRAY[0..10] OF CHAR;

PROCEDURE nb(Y:INTEGER;s:REAL);
(*--------------------------*)

BEGIN
     rp^.BgPen:=BYTE(gris);
     rp^.FgPen:=BYTE(0);
     SetAPen(rp^,0);
     Move(rp^,x+L+2,Y+1);
     ConvRealToString(s,ss,0,Decimal);
```

```
     Text(rp^,ADR(ss),StringLength(ss));
END nb;

PROCEDURE FLECHE;
(*-------------*)

BEGIN
     SetAPen(rp^,0);
     RectFill(rp^,x+14,y+5,x+15,y+14);
     RectFill(rp^,x+10,y+9,x+19,y+10);
END FLECHE;

PROCEDURE REC1;
BEGIN
     bidon:=s1 / (Max/longueur); toto:=entier(bidon);
     IF (y+5 > y+l-1-toto) THEN toto:=l-2; DEBORDE:=TRUE; END;
     RectFill(rp^,x+1,y+l-1-toto,L+1,y+l-1);
     nb(y+l+2-toto,s1);
END REC1;

PROCEDURE REC2;
BEGIN
     bidon:=s1 / (Max/longueur); toto:=entier(bidon);
     bidon:=s2 / (Max/longueur); tata:=entier(bidon);titi:=toto+tata;
     IF (y+5 > y+l-1-titi) THEN titi:=l-2; DEBORDE:=TRUE; END;
     RectFill(rp^,x+1,y+l-1-titi,L+1,y+l-1);
     nb(y+l+2-titi,s1+s2);
     nb(y+l+2-toto,s1);
END REC2;

BEGIN
     DEBORDE:=FALSE;
     rp:=win^.RPort;
     IF (x=2) AND (y=40) THEN SetAPen(rp^,gris);
                         ELSE SetAPen(rp^,0);
     END;
     RectFill(rp^,x+1,y,x+L-1,y+l-1);
     IF (x=2) AND (y=40) THEN RectFill(rp^,L+3,y-1,77,y+l+4);
     END;
     SetAPen(rp^,c1);
     CASE sens OF
       2:   REC1;
            SetAPen(rp^,c2);
            bidon:=s2 / (Max/longueur); tata:=entier(bidon);
            IF (tata > toto) THEN tata:=toto; END;
            RectFill(rp^,x+1,y+l-1-toto,L+1,y+l-1-toto+tata);
            nb(y+l+2-toto+tata,s1-s2);
       1:   REC2;
       0:   REC1;
       3:   REC1;
            SetAPen(rp^,c2);
            bidon:=s2 / (Max/longueur); tata:=entier(bidon);
            IF (y+5 > y+l-1-tata) THEN tata:=l-2; DEBORDE:=TRUE;
            END;
            RectFill(rp^,x+1,y+l-1-tata,L+1,y+l-1);
            nb(y+l+2-tata,s2);
     END;
     IF DEBORDE THEN FLECHE; END;
```

```
 END MAJ_Jauge;

PROCEDURE MAJ_Somme(Win:WindowPtr;i:INTEGER;VAL:REAL);

VAR ss:Str20;
    New:BOOLEAN;

BEGIN
     New:=FALSE;
     rp^.FgPen:=BYTE(vert);
     rp^.BgPen:=BYTE(0);
     SetAPen(rp^,vert);
     ConvRealToString(VAL,ss,0,Decimal);
     IF i=0 THEN Move(rp^,269-(StringLength(ss)*8),51);
                 Text(rp^,ADR(ss),StringLength(ss));
            ELSE IF i=7 THEN Move(rp^,269-(StringLength(ss)*8),182);
                             Text(rp^,ADR(ss),StringLength(ss));
                             Move(rp^,200,182);
                             Text(rp^,ADR("="),1);
                        ELSE IF i>10 THEN i:=i-10;New:=TRUE; END;
                             SetAPen(rp^,0);
                             RectFill(rp^,211,64+(i-1)*20,
                                          268,82+(i-1)*20);
                             SetAPen(rp^,vert);
                             Move(rp^,269-(StringLength(ss)*8),
                                      77+(i-1)*20);
                             Text(rp^,ADR(ss),StringLength(ss));
                             IF (New) AND (i#5)
                                THEN Move(rp^,200,77+(i-1)*20);
                                     Text(rp^,ADR("+"),1);
                             END;
                 END;
     END;
END MAJ_Somme;

PROCEDURE InitSomme(Win:WindowPtr; F:INTEGER);

VAR i,X,Y:INTEGER;

PROCEDURE DrawRec(x,y:INTEGER);
BEGIN
     Move(rp^,x,y);
     Draw(rp^,x+60,y);
     Draw(rp^,x+60,y+20);
     Draw(rp^,x,y+20);
     Draw(rp^,x,y);
END DrawRec;

BEGIN

     rp:=Win^.RPort;
     SetAPen(rp^,0);
     RectFill(rp^,199,41,270,189);
     SetAPen(rp^,1);
     FOR i:=1 TO F DO
        X:=210;
        Y:=63+(i-1)*20;
       DrawRec(X,Y);
```

```
      END;
      DrawRec(210,167);
      DrawRec(211,168);
      DrawRec(210,37);
      DrawRec(211,38);
      MAJ_Somme(Win,0,Liquide);
      MAJ_Somme(Win,7,Liquide);
END InitSomme;

PROCEDURE cdr(Win:WindowPtr;x,y,N:INTEGER; ON:BOOLEAN);
VAR Longueur,Largeur,Couleur:INTEGER;
BEGIN
      CASE N OF
        1:Longueur:=53; Largeur :=28; Couleur :=orange; |
        2:Longueur:=92; Largeur :=92; Couleur :=1;      |
        3:Longueur:=54; Largeur :=29; Couleur :=1;      |
        4:Longueur:=25; Largeur :=30; Couleur :=rouge;  |
        5:Longueur:=31; Largeur :=37; Couleur :=1;      |
        6:Longueur:=29; Largeur :=35; Couleur :=0;      |
        7:Longueur:=49; Largeur :=38; Couleur :=1;      |
        8:Longueur:=76; Largeur :=29; Couleur :=1;      |
        9:Longueur:=25; Largeur :=30; Couleur :=rouge;  |
       10:Longueur:=25; Largeur :=30; Couleur :=vert;   |
       11:Longueur:=25; Largeur :=30; Couleur :=orange; |
       12:Longueur:=21; Largeur :=26; Couleur :=rouge;  |
       13:Longueur:=21; Largeur :=26; Couleur :=vert;   |
       14:Longueur:=21; Largeur :=26; Couleur :=orange; |
      END;
      IF ON THEN Couleur:=0; END;
      SetAPen(Win^.RPort^,Couleur);
      Move(Win^.RPort^,x-2,y-2);
      Draw(Win^.RPort^,x+Longueur+2,y-2);
      Draw(Win^.RPort^,x+Longueur+2,y+Largeur+2);
      Draw(Win^.RPort^,x-2,y+Largeur+2);
      Draw(Win^.RPort^,x-2,y-2);
END cdr;

BEGIN
    JaugeX:=2;
    JaugeY:=40;
 END Jauge.
```

IMPLEMENTATION MODULE AuxPD;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM Tasks IMPORT SignalSet,Wait;
FROM MathLib0 IMPORT real;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget,
WindowFlagsSet,  IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow,Image,
DrawImage, Activate,GadgetUp;
FROM Drawing IMPORT RectFill,SetAPen,Draw,Move;
FROM RealConversions IMPORT ConvRealToString,RealToStringFormat;
FROM Text IMPORT Text;
FROM Strings IMPORT StringLength;
FROM Conversions IMPORT ConvStringToNumber;
FROM couleurs IMPORT bleu,bleuclair,gris,rouge;
FROM LireImage IMPORT Mini,TabMini,LoadMini;
FROM annee IMPORT DATE,RechJour,TABNJ,NbJours;
FROM VariablesGlobales IMPORT Scr,Commande,Str40,TableauParametres,Str3;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Trace IMPORT trace;
FROM parole IMPORT SAY;
FROM IO IMPORT TabPD,NbPD,TPD;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR jj,mois_actuel,aa:INTEGER;
    bi,Done:BOOLEAN;
    Trace:Str40;

PROCEDURE CONV(s:REAL; VAR l:INTEGER; VAR j:TABJ);
(*-------------------------------------------*)

VAR str:ARRAY [0..6] OF CHAR;
    i:INTEGER;
BEGIN
     ConvRealToString(s,str,0,Decimal);
     l:=StringLength(str);
     FOR i:=0 TO l-1 DO
        CASE str[i] OF
           "0": j[i]:=0; |
           "1": j[i]:=1; |
           "2": j[i]:=2; |
           "3": j[i]:=3; |
           "4": j[i]:=4; |
           "5": j[i]:=5; |
           "6": j[i]:=6; |
           "7": j[i]:=7; |
           "8": j[i]:=8; |
           "9": j[i]:=9; |
        END;
     END;
```

```
END CONV;

PROCEDURE Somme2(VAR Win:WindowPtr;x,y:INTEGER;s:REAL);
(*-------------------------------------------------*)
VAR Longueur,i,k:INTEGER;
    J:TABJ;
BEGIN
     SetAPen(Win^.RPort^,rouge);
     Move(Win^.RPort^,x,y);
     Draw(Win^.RPort^,x+30,y);
     Draw(Win^.RPort^,x+30,y+30);
     Draw(Win^.RPort^,x,y+30);
     SetAPen(Win^.RPort^,gris+3);
     RectFill(Win^.RPort^,x+1,y+1,x+29,y+29);
     CONV(s,Longueur,J);
     FOR i:=0 TO Longueur-1 DO
         k:=x+29-((Longueur-i)*5);
         DrawImage(Win^.RPort^,Mini[J[i]],k,y+13);
     END;
END Somme2;


PROCEDURE DrawLundi(VAR Win:WindowPtr;jour:INTEGER);
(*-------------------------------------------------*)
VAR x,i,L:INTEGER;
    J:TABJ;
    ir:REAL;
BEGIN
     ir:=real(jour);
     CONV(ir,L,J);
     FOR i:=0 TO L-1 DO
         x:=92-(L-1)*4+i*4+(jour-1)*7;
         DrawImage(Win^.RPort^,Mini[J[i]],x,194);
     END;
END DrawLundi;

PROCEDURE LigneTpsSem(VAR Win:WindowPtr);
(*-------------------------------------*)
VAR jour:ARRAY [0..6] OF Str3;
    i,n,JActuel,JD:INTEGER;
BEGIN
     RechJour(jj,mois_actuel,aa,n);
     jour[0]:="Lun";
     jour[1]:="Mar";
     jour[2]:="Mer";
     jour[3]:="Jeu";
     jour[4]:="Ven";
     jour[5]:="Sam";
     jour[6]:="Dim";
     JD:=TableauParametres[10];
     n:=n-1;
     FOR i:=0 TO 6 DO
        IF JD>6 THEN JD:=0; END;
        IF n=JD THEN SetAPen(Win^.RPort^,1);
                     RectFill(Win^.RPort^,88+i*32,197,118+i*32,207);
                     Win^.RPort^.FgPen:=BYTE(0);
                     Win^.RPort^.BgPen:=BYTE(1);
                     SetAPen(Win^.RPort^,0);
```

```
                ELSE SetAPen(Win^.RPort^,bleu);
                     RectFill(Win^.RPort^,88+i*32,197,118+i*32,207);
                     Win^.RPort^.BgPen:=BYTE(bleu);
                     SetAPen(Win^.RPort^,1);
          END;
          Move(Win^.RPort^,91+i*32,204);
          Text(Win^.RPort^,ADR(jour[JD]),3);
          INC(JD);
       END;
       SetAPen(Win^.RPort^,1);
       RectFill(Win^.RPort^,78,194,319,194);
END LigneTpsSem;

PROCEDURE LigneTemps(VAR Win:WindowPtr);
(*-------------------------------------*)

VAR i,j,n:INTEGER;
    NJOURS:TABNJ;
BEGIN
   IF TableauParametres[9]=1 THEN LigneTpsSem(Win);
   ELSE
     RechJour(1,mois_actuel,aa,n);
     NbJours(NJOURS);
     j:=n;
     FOR i :=1 TO NJOURS[mois_actuel-1] DO
        IF j=1 THEN DrawLundi(Win,i); END;
        IF (j>5) THEN SetAPen(Win^.RPort^,bleuclair);
                 IF j>6 THEN j:=0; END;
                 ELSE SetAPen(Win^.RPort^,bleu);
        END;
        RectFill(Win^.RPort^,90+(i-1)*7,202,95+(i-1)*7,206);
        INC(j);
     END;
     SetAPen(Win^.RPort^,1);
     RectFill(Win^.RPort^,78,191,319,191);
     RectFill(Win^.RPort^,90+(jj-1)*7,202,95+(jj-1)*7,206);
   END;
END LigneTemps;

PROCEDURE GadgetHandlerConf(VAR w:Window;VAR Msg:MsgData;VAR Gad:Gadget);
BEGIN
        CASE Gad.GadgetID OF
         0: Trace:="Il confirme la demande ";
            trace(Trace);bi:=TRUE;|
         1: Trace:="Il infirme la demande ";
            trace(Trace);bi:=FALSE;|
        END;
        Done:=TRUE;
END GadgetHandlerConf;

PROCEDURE CONF(Im:Image;s:REAL;VAR Reponse:BOOLEAN);
(*-----------------------------------------------*)
VAR
   Win : WindowPtr;
   MsgConf:IntuiMessagePtr;
   WpConf : WindowProc;
   SigConf : SignalSet;
   GlConf:GadgetPtr;
```

```
BEGIN
      WITH WpConf DO
         procGadgetUp := GadgetHandlerConf;
      END;
      BeginGadgetList();
        AddGadgetImageButton (100,55,Commande[2]);
        AddGadgetImageButton (160,55,Commande[5]);
      GlConf:=EndGadgetList();
      Win := CreateWindow (15,80,290,100,WIDCMP,WFlags,GlConf,Scr,NIL);
      Win^.RPort^.FgPen:=BYTE(1);
      Win^.RPort^.BgPen:=BYTE(0);
      DrawImage(Win^.RPort^,Im,115,5);
      Somme2(Win,140,5,s);
      SAY(10);
      SetAPen (Win^.RPort^,1);
      Move(Win^.RPort^,92,45);
      Text(Win^.RPort^,ADR("Est-ce correct ?"),16);
      Done:=FALSE;
      WHILE (NOT Done) DO
           SigConf := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)})
           LOOP
               MsgConf := GetMsg(Win^.UserPort^);
               IF (MsgConf = NIL) THEN EXIT; END;
               ProcIMsg (WpConf, MsgConf);
           END;
      END;
      Reponse:=bi;
      CloseWindow(Win^);
      FreeGadgetList (GlConf^);

END CONF;

PROCEDURE ENLEVERPD(n:INTEGER);
(*--------------------------*)
VAR i:INTEGER;
BEGIN
     FOR i:=n TO NbPD-1 DO
          TabPD[i].NoPoste:=TabPD[i+1].NoPoste;
          TabPD[i].Somme  :=TabPD[i+1].Somme;
          TabPD[i].RO     :=TabPD[i+1].RO;
          TabPD[i].MoisDepart:=TabPD[i+1].MoisDepart;
          TabPD[i].Frequence :=TabPD[i+1].Frequence;
          TabPD[i].Faite     :=TabPD[i+1].Faite;
          TabPD[i].Quand     :=TabPD[i+1].Quand;
          TabPD[i].SommeDprt :=TabPD[i+1].SommeDprt;
    END;
    NbPD:=NbPD-1;
END ENLEVERPD;


PROCEDURE VERIF(mois,freq:INTEGER;VAR  ok:BOOLEAN);
(*-----------------------------------------------*)
VAR FIN:BOOLEAN;
    I,J:INTEGER;

BEGIN
     J:=12 DIV freq;
     I:=0;
```

```
      FIN:=FALSE;
      ok:=FALSE;
      WHILE (NOT ok) AND (NOT FIN) DO
          I:=I+1;
          IF mois=mois_actuel THEN ok:=TRUE;
                              ELSE mois:=mois+freq;
                                   IF mois>12 THEN mois:=mois-12 END;
          END;
          IF J=I THEN FIN:=TRUE; END;
      END;
END VERIF;

PROCEDURE DeterminerPD(N:INTEGER;VAR PDFreq:TPD;VAR Nb:INTEGER);
(*-------------------------------------------------------------*)

VAR i:INTEGER;
    ok:BOOLEAN;

PROCEDURE MAJTPD(pos:INTEGER);
(*-------------------------*)
BEGIN
      Nb:=Nb+1;
      PDFreq[N][Nb].NoPoste:=TabPD[pos].NoPoste;
      PDFreq[N][Nb].Somme  :=TabPD[pos].Somme;
      PDFreq[N][Nb].RO     :=TabPD[pos].RO;
      PDFreq[N][Nb].MoisDepart:=TabPD[pos].MoisDepart;
      PDFreq[N][Nb].Frequence :=TabPD[pos].Frequence;
      PDFreq[N][Nb].Faite  :=TabPD[pos].Faite;
      PDFreq[N][Nb].Quand  :=TabPD[pos].Quand;
      PDFreq[N][Nb].SommeDprt :=TabPD[pos].SommeDprt;
END MAJTPD;

PROCEDURE DebutPD;
(*--------------*)

VAR i:INTEGER;

BEGIN
      FOR i:=1 TO NbPD DO
          IF (TabPD[i].Faite="O") AND (TabPD[i].Quand # mois_actuel) THEN
              TabPD[i].Faite:="N";
              TabPD[i].Quand:=mois_actuel;
          END;
      END;
END DebutPD;

BEGIN
      DebutPD;
      Nb:=0;
      IF (N<6) THEN
                  FOR i:=1 TO NbPD DO
                    IF TabPD[i].Faite="N" THEN
                      IF TabPD[i].RO= "R" THEN
                        IF TabPD[i].Frequence=N THEN MAJTPD(i); END;
                      END;
                    END;
                  END;
              ELSE
```

```
                    FOR i:= 1 TO NbPD DO
                     IF TabPD[i].Faite="N" THEN
                     IF TabPD[i].RO="O" THEN MAJTPD(i);
                                        ELSE
                       CASE TabPD[i].Frequence OF
                          0: MAJTPD(i); |
                          1: VERIF(TabPD[i].MoisDepart,2,ok);
                             IF ok THEN MAJTPD(i); END; |

                          2: VERIF(TabPD[i].MoisDepart,6,ok);
                             IF ok THEN MAJTPD(i); END; |

                          3: MAJTPD(i); |
                          4: VERIF(TabPD[i].MoisDepart,3,ok);
                             IF ok THEN MAJTPD(i); END; |

                          5: IF TabPD[i].MoisDepart=mois_actuel
                             THEN MAJTPD(i);END;|
                       END;
                     END;
                     END;
                    END;
    END;
END DeterminerPD;

BEGIN
     LoadMini(Mini);
     DATE(jj,mois_actuel,aa,bi);

END AuxPD.
```

IMPLEMENTATION MODULE ModPoste;

```
FROM SYSTEM IMPORT BYTE;
FROM AmigaDosProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow,
  NewWindow, Image,  DrawImage, ActivateWindow;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton;
FROM Strings IMPORT ConcatString,CopyString;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM VariablesGlobales IMPORT Str40, Scr,TableauParametres,Commande;
FROM FindImages IMPORT ImageDescTablePtr, FindImageTable;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY,SayPoste;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM IO IMPORT TABPOSTE,LIRENOMS_POSTES,Str15;
FROM InOut IMPORT WriteString,WriteLn;
FROM UTILS IMPORT PRINT;
FROM Jauge IMPORT cdr;
FROM LireImage IMPORT LoadPostes,POSTE;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  ok,Done: BOOLEAN;
  nom:TABPOSTE;
  Win:WindowPtr;
  i,j,oui: CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  x,y,P:INTEGER;
PROCEDURE Det(n:INTEGER; VAR x,y:INTEGER);

BEGIN
     y:=n DIV 10;
     y:=40+(y*43);
     x:=n MOD 10;
     x:=8+(x*31);
END Det;

PROCEDURE GadgetHandlerME (VAR w: Window; VAR Msg:MsgData;
                           VAR Gad:Gadget);

BEGIN
     IF (Gad.GadgetID <40) THEN IF ok THEN Det(oui,x,y);
                                           cdr(Win,x,y,8+P,TRUE);
```

```
                                    END;
                                    oui:=Gad.GadgetID;
                                    Det(oui,x,y);
                                    cdr(Win,x,y,8+P,FALSE);
                                    ok:=TRUE;
                                    Trace:="Il choisit: ";
                                    ConcatString(Trace,nom[Gad.GadgetID]);
                                    trace(Trace);
                                    SayPoste(nom[Gad.GadgetID]);
     ELSE CASE Gad.GadgetID OF
             40: Trace := "Il clique sur ok ";
                 trace (Trace);
                 IF ok THEN Done:=TRUE; END;|
             41: Trace := "Il clique sur HELP ";
                 trace (Trace);
                 SOS(7); |
             42: SAY(4); |
             43: PRINT;|
          END;
    END;
END GadgetHandlerME;

(*************************************************************)
(*                          Poste                            *)
(*************************************************************)

PROCEDURE Poste (VAR poste_choisi:Image; VAR No:INTEGER);

BEGIN
   LIRENOMS_POSTES(nom);
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   P:=No;
   LoadPostes(POSTE);
   BeginGadgetList();
        FOR j:=0 TO 3 DO
           FOR i:=0 TO 9 DO
              AddGadgetImageButton (8+(i*31),40+(j*43),POSTE[i+(10*j)]);
           END;
        END;
        AddGadgetImageButton (190,215,Commande[2]);
        AddGadgetImageButton (285,215,Commande[12]);
        IF TableauParametres[1]=2
            THEN AddGadgetImageButton(130,215,Commande[4]);
            ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        IF TableauParametres[0]=1
           THEN AddGadgetImageButton(160,215,Commande[11]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
      Gl := EndGadgetList();

      IF (Gl # NIL) THEN
         Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
         IF (Win # NIL) THEN
             FRAME2(Win);
             FOR j:=0 TO 3 DO
```

```
                FOR i:=0 TO 9 DO
                   cdr(Win,10+(i*31),42+(j*43),11+No,FALSE);
                END;
             END;
             IF TableauParametres[1] = 2 THEN SAY(4);END;
             ECRIT(Win,4);
             DrawImage (Win^.RPort^,Commande[2],190,215);
             DrawImage (Win^.RPort^,Commande[12],285,215);
             IF TableauParametres[1]=2
                THEN DrawImage(Win^.RPort^,Commande[4],130,215);
             END;
             IF TableauParametres[0]=1
                THEN DrawImage(Win^.RPort^,Commande[11],160,215);
             END;
             Done:=FALSE;
             ok:=FALSE;
             WHILE (NOT Done) DO
               Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                    Msg := GetMsg(Win^.UserPort^);
                    IF (Msg = NIL) THEN EXIT; END;
                    ProcIMsg (Wp, Msg);
               END;
             END;
             poste_choisi:=POSTE[oui];
             No:=oui;
             CloseWindow (Win^);
          END;
          FreeGadgetList (Gl^);
       END;
  END Poste;
END ModPoste.
```

IMPLEMENTATION MODULE ModFreq;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow
  NewWindow, Image,  DrawImage, ActivateWindow,IntuiText,IntuiTextPtr;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM Drawing IMPORT RectFill,SetAPen,Move;
FROM VariablesGlobales IMPORT Str40, Scr,TableauParametres,Commande;
FROM Text IMPORT Text;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY,SayMois,SayFreq;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM LireImage IMPORT LireMois,TabMois,LoadFreq,TabFreq;
FROM annee IMPORT DATE,NOMMOIS;
FROM UTILS IMPORT PRINT;
FROM Jauge IMPORT cdr;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  pos:ARRAY[0..5] OF RECORD
                          X,Y:INTEGER;
                     END;
  Trace : Str40;
  T:LONGINT;
  jj,FR,mois,aa:INTEGER;
  Win:WindowPtr;
  ok,Done,Aff,bi: BOOLEAN;
  i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Tab:TabMois;
  freq:TabFreq;
  TF:ARRAY[0..5] OF Str40;

PROCEDURE GadgetHandlerME (VAR w: Window; VAR Msg : MsgData; VAR Gad:
Gadget);

BEGIN
     IF Gad.GadgetID < 6
                THEN trace(TF[Gad.GadgetID]);
                     IF FR<9 THEN cdr(Win,pos[FR].X,pos[FR].Y,3,TRUE);
                     END;
                     FR:=Gad.GadgetID;
```

```
                    SayFreq(Gad.GadgetID);
                    cdr(Win,pos[FR].X,pos[FR].Y,3,FALSE);
               ELSE CASE Gad.GadgetID OF
                    6: IF mois=12 THEN mois:=1;
                                  ELSE INC(mois);
                       END;
                       DrawImage(Win^.RPort^,Tab[mois-1],130,160);
                       SayMois(mois);
                       Trace:="Il change de mois ";
                       trace(Trace);
                    7: Trace := "Il clique sur ok ";
                       trace (Trace);
                       ok:=TRUE;
                    8: SAY(7);
                    9: PRINT;
                   10: Trace := "Il clique sur HELP ";
                       trace (Trace);
                       SOS(13);
                   END;
     END;
END GadgetHandlerME;

(*************************************************************************)
(*                              FREQ                                     *)
(*************************************************************************)

PROCEDURE Frequence (VAR Freq,start_mois:INTEGER; poste:Image);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   TF[0]:="Toutes les semaines ";
   TF[1]:="Tous les deux mois ";
   TF[2]:="Tous les six mois ";
   TF[3]:="Tous les mois";
   TF[4]:="Tous les trois mois ";
   TF[5]:="Tous les ans ";
   FR:=9;
   LoadFreq(freq);
   LireMois(Tab);
   DATE(jj,mois,aa,bi);
   pos[0].X:=13;pos[0].Y:=40;pos[1].X:=116;pos[1].Y:=40;pos[2].X:=219;
   pos[2].Y:=40;pos[3].X:=13;pos[3].Y:=90;pos[4].X:=116;pos[4].Y:=90;
   pos[5].X:=219;pos[5].Y:=90;
   BeginGadgetList();
        FOR i:=0 TO 5 DO
           AddGadgetImageButton (pos[i].X,pos[i].Y,freq[i]);
        END;
        AddGadgetImageButton (130,160,Tab[mois-1]);
        AddGadgetImageButton (190,215,Commande[2]);
        IF TableauParametres[1]=2
           THEN AddGadgetImageButton(130,215,Commande[4]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        IF TableauParametres[0]=1
           THEN AddGadgetImageButton(160,215,Commande[11]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
```

```
        END;
        AddGadgetImageButton (285,215,Commande[12]);
    Gl := EndGadgetList();
        IF (Gl # NIL) THEN
       Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
        IF (Win # NIL) THEN
            FRAME2(Win);
            IF TableauParametres[1] = 2 THEN SAY(7);  END;
            DrawImage (Win^.RPort^,Commande[2],190,215);
            DrawImage (Win^.RPort^,Commande[12],285,215);
            IF (TableauParametres[1] = 2) THEN
               DrawImage (Win^.RPort^,Commande[4],130,215);
            END;
            IF (TableauParametres[0] = 1) THEN
               DrawImage (Win^.RPort^,Commande[11],160,215);
            END;
            SetAPen(Win^.RPort^,1);
            RectFill(Win^.RPort^,40,135,280,136);
            Move(Win^.RPort^,84,146);
            Win^.RPort^.FgPen:=BYTE(1);
            Win^.RPort^.BgPen:=BYTE(0);
            Text(Win^.RPort^,ADR("A partir du mois de"),19);
            ECRIT(Win,7);
            DrawImage (Win^.RPort^,poste,290,0);
            Done:=FALSE;
            ok:=FALSE;
            WHILE (NOT Done) DO
               Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                  Msg := GetMsg(Win^.UserPort^);
                  IF (Msg = NIL) THEN EXIT; END;
                  ProcIMsg (Wp, Msg);
               END;
               IF ok AND (FR<>9) THEN Done:=TRUE; ELSE ok:=FALSE; END;
            END;
            Freq:=FR;
            start_mois:=mois;
            CloseWindow (Win^);
          END;
          FreeGadgetList (Gl^);
       END;
  END Frequence;
END ModFreq.
```

IMPLEMENTATION MODULE ModSRO;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM Tasks IMPORT SignalSet,Wait;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow,
  NewWindow, Image,  DrawImage, ActivateWindow,IntuiText,IntuiTextPtr;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT RectFill,SetAPen;
FROM VariablesGlobales IMPORT Str40, Scr,TableauParametres,Commande,cmap;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM UTILS IMPORT PRINT;
FROM Jauge IMPORT cdr;
FROM Help IMPORT SOS;
FROM LireImage IMPORT LoadRO,TabRO;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  T:LONGINT;
  Cl:INTEGER;
  Win:WindowPtr;
  ok,Done,Aff,Fin: BOOLEAN;
  image: TabRO;
  i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  CH:CHAR;

PROCEDURE GadgetHandlerME(VAR w: Window; VAR Msg:MsgData; VAR Gad:
Gadget);

BEGIN
     CASE Gad.GadgetID OF
      0: Trace:="Il choisit Régulière";
         trace(Trace); CH:="R"; SAY(11);
         cdr(Win,40,70,2,FALSE); cdr(Win,200,70,2,TRUE);|
      1: Trace:="Il choisit Occasionnelle";
         trace(Trace); CH:="O"; SAY(12);
         cdr(Win,40,70,2,TRUE); cdr(Win,200,70,2,FALSE);|
      2: Trace := "Il appuie sur ok ";
       trace (Trace);
         ok:=TRUE;  |
      3: SAY(6); SAY(11); SAY(13); SAY(12); |
```

```
      4: PRINT; |
      5: Trace := "Il clique sur HELP ";
         trace (Trace);
         IF TableauParametres[9]=1 THEN SOS(11); ELSE SOS(14); END; |
  END;
END GadgetHandlerME;

(*****************************************************************)
(*                              SRO                              *)
(*****************************************************************)

PROCEDURE SRO (poste:Image; VAR ch:CHAR);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   CH:="X";
   LoadRO(image);
   BeginGadgetList();
        AddGadgetImageButton (40,70,image[0]);
        AddGadgetImageButton (200,70,image[1]);
        AddGadgetImageButton (190,215,Commande[2]);
        IF TableauParametres[1]=2
           THEN AddGadgetImageButton(130,215,Commande[4]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        IF TableauParametres[0]=1
           THEN AddGadgetImageButton(160,215,Commande[11]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        AddGadgetImageButton (285,215,Commande[12]);
   Gl := EndGadgetList();
   IF (Gl # NIL) THEN
       Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
         IF (Win # NIL) THEN
             FRAME2(Win);
             IF TableauParametres[1] = 2 THEN
                          SAY(6); Delay(10); SAY(11); Delay(10);
                          SAY(13);Delay(10); SAY(12); Delay(10);
             END;
             ECRIT(Win,6);
             DrawImage (Win^.RPort^,poste,293,1);
             DrawImage (Win^.RPort^,Commande[2],190,215);
             DrawImage (Win^.RPort^,Commande[12],285,215);
             IF (TableauParametres[1] = 2) THEN
                DrawImage (Win^.RPort^,Commande[4],130,215);
             END;
             IF (TableauParametres[0] = 1) THEN
                DrawImage (Win^.RPort^,Commande[11],160,215);
             END;
             SetAPen(Win^.RPort^,0);
             Done:=FALSE;
             ok:=FALSE;
             WHILE (NOT Done) DO
                Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
                LOOP
                   Msg := GetMsg(Win^.UserPort^);
```

```
                IF (Msg = NIL) THEN EXIT; END;
                ProcIMsg (Wp, Msg);
              END;
              IF ok AND (CH<>"X") THEN Done:=TRUE; ELSE ok:=FALSE; END;
            END;
            ch:=CH;
            CloseWindow (Win^);
          END;
          FreeGadgetList (Gl^);
      END;
  END SRO;
END ModSRO.
```

IMPLEMENTATION MODULE Visu;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM MathLib0 IMPORT entier;
FROM AmigaDosProcess IMPORT Delay;
FROM Tasks IMPORT SignalSet,Wait;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow
  NewWindow, Image,  DrawImage, ActivateWindow,IntuiText,IntuiTextPtr;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT RectFill,SetAPen,Draw,Move;
FROM VariablesGlobales IMPORT Str40, Scr,TableauParametres,Commande,cmap
                              Liquide,Max,Icone;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM IO IMPORT TPD;
FROM LireImage IMPORT POSTE,Mini,LoadFreq,TabFreq,LireMois,TabMois;
FROM AuxPD IMPORT CONV,TABJ;
FROM couleurs IMPORT orange,gris,vert;
FROM UTILS IMPORT PRINT;
FROM annee IMPORT DATE;
FROM Help IMPORT SOS;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR Trace:Str40;
    Win:WindowPtr;
    Msg:IntuiMessagePtr;
    Wp : WindowProc;
    Sig : SignalSet;
    Gl:GadgetPtr;
    Done,B,DeuxEcrans,OK:BOOLEAN;
    Tpos:ARRAY [1..15] OF BOOLEAN;
    freq:TabFreq;
    Mois:TabMois;
    JJ,MM,AA,Top:INTEGER;
    TAB:ARRAY[1..15] OF INTEGER;

PROCEDURE DrawJauge(x,Nb,N,i,couleur:INTEGER;VAR Tab:TPD);
(*-------------------------------------------------------*)
VAR RE,t,k,Longueur:INTEGER;
    R,RR:REAL;
    J:TABJ;

BEGIN
     IF couleur=vert THEN CONV(Liquide,Longueur,J);
                     ELSE CONV(Tab[N][i].Somme,Longueur,J);
     END;
     FOR t:=0 TO Longueur-1 DO
```

```
        k:=x*29-((Longueur-t)*5);
        DrawImage(Win^.RPort^,Mini[J[t]],k,193);
     END;
     IF couleur=vert
        THEN DrawImage(Win^.RPort^,POSTE[40],2+29*(x-1),40);
        ELSE DrawImage(Win^.RPort^,POSTE[Tab[N][i].NoPoste],
                                              2+29*(x-1),40);
     END;
     R:=130./Max;
     SetAPen(Win^.RPort^,couleur);
     IF couleur=vert THEN RR:=R*Liquide;
                     ELSE RR:=R*Tab[N][i].Somme;
     END;
     RE:=entier(RR);
     IF RE>120 THEN RE:=120; END;
     RectFill(Win^.RPort^,1+29*(x-1),188-RE,(29*x)-1,188);
END DrawJauge;

PROCEDURE Affiche(Nb,N:INTEGER;VAR Tab:TPD);
(*---------------------------------------*)
VAR i:INTEGER;

BEGIN
     IF Nb>10 THEN Top:=10; DeuxEcrans:=TRUE;
              ELSE Top:=Nb; DeuxEcrans:=FALSE;
     END;
     FOR i:=1 TO Top DO
        DrawJauge(i,Nb,N,TAB[i],orange,Tab);
     END;
     IF N=6 THEN  DrawJauge(Top+1,Nb,N,Top+1,vert,Tab); END;
END Affiche;

PROCEDURE AfficheSuivant(Nb,N:INTEGER;VAR Tab:TPD);
(*-----------------------------------------------*)
VAR i:INTEGER;

BEGIN
     FOR i:=Top TO Nb DO
        DrawJauge(i-10,Nb,N,TAB[i],orange,Tab);
     END;
     IF N=6 THEN  DrawJauge(Nb-9,Nb,N,Nb+1,vert,Tab); END;
END AfficheSuivant;

PROCEDURE GadgetHandlerME (VAR w: Window;VAR Msg:MsgData; VAR Gad:
Gadget);
BEGIN
     CASE Gad.GadgetID OF
      0: Trace := "Il appuie sur ok ";
        trace (Trace);
        IF DeuxEcrans THEN Trace:="et arrive au second écran ";
                           trace(Trace);
                           IF NOT OK THEN OK:=TRUE; END;
                     ELSE OK:=FALSE;Done:=TRUE;
        END;|
      1: Trace := "Il appuie sur HELP ";
         trace (Trace);
         SOS(17);|
      2: SAY(16);|
```

```
      3:PRINT; |
    END;
END GadgetHandlerME;

(*************************************************************************)
(*                                  Visu                                 *)
(*************************************************************************)

PROCEDURE Visualisation(N:INTEGER;VAR Tab:TPD;Nb:INTEGER);

PROCEDURE TRI;
(*----------*)
VAR i,j,pos:INTEGER;
    max:REAL;
BEGIN
     FOR j:=1 TO Nb DO
        max:=100000.;
        FOR i:=1 TO Nb DO
           IF ((Tab[N][i].Somme<=max) AND (NOT Tpos[i]))
                                    THEN max:=Tab[N][i].Somme;
                                         pos:=i;
           END;
        END;
        Tpos[pos]:=TRUE;
        TAB[j]:=pos;
     END;
END TRI;

PROCEDURE DessinerTableau;
(*----------------------*)
VAR i:INTEGER;
BEGIN
     SetAPen(Win^.RPort^,gris);
     RectFill(Win^.RPort^,1,31,318,209);
     DrawImage (Win^.RPort^,Icone[2],0,0);
     SetAPen(Win^.RPort^,1);
     Move(Win^.RPort^,1,71);
     Draw(Win^.RPort^,319,71);
     Move(Win^.RPort^,1,38);
     Draw(Win^.RPort^,319,38);
     Move(Win^.RPort^,1,200);
     Draw(Win^.RPort^,319,200);
     Move(Win^.RPort^,1,190);
     Draw(Win^.RPort^,319,190);
     FOR i:=1 TO 12 DO
        Tpos[i]:=FALSE;
        Move(Win^.RPort^,29*(i-1),38);
        Draw(Win^.RPort^,29*(i-1),200);
     END;
END DessinerTableau;

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   BeginGadgetList();
        AddGadgetImageButton (190,215,Commande[2]);
        AddGadgetImageButton (285,215,Commande[12]);
```

```
      IF TableauParametres[1]=2
         THEN AddGadgetImageButton(130,215,Commande[4]);
         ELSE AddGadgetImageButton(1,1,Commande[9]);
      END;
      IF TableauParametres[0]=1
         THEN AddGadgetImageButton(160,215,Commande[11]);
         ELSE AddGadgetImageButton(1,1,Commande[9]);
      END;
  Gl := EndGadgetList();
  LoadFreq(freq);
  LireMois(Mois);
  IF (Gl # NIL) THEN
       Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
       IF (Win # NIL) THEN
           FRAME2(Win);
           IF TableauParametres[1] = 2 THEN  SAY(16); END;
           ECRIT(Win,10);
           DrawImage (Win^.RPort^,Commande[2],190,215);
           DrawImage (Win^.RPort^,Commande[12],285,215);
           IF (TableauParametres[1] = 2) THEN
              DrawImage (Win^.RPort^,Commande[4],130,215);
           END;
           IF (TableauParametres[0] = 1) THEN
              DrawImage (Win^.RPort^,Commande[11],160,215);
           END;
           DessinerTableau;
           TRI;
           IF (N=6) AND (TableauParametres[9]#1)
                THEN DATE(JJ,MM,AA,B);
                     DrawImage(Win^.RPort^,Mois[MM-1],243,0);
                ELSE IF N<6 THEN DrawImage(Win^.RPort^,freq[N],
                                           263,0);
                     END;
           END;
           Affiche(Nb,N,Tab);
           Done:=FALSE; OK:=FALSE;
           WHILE (NOT Done) DO
              Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
              LOOP
                 Msg := GetMsg(Win^.UserPort^);
                 IF (Msg = NIL) THEN EXIT; END;
                 ProcIMsg (Wp, Msg);
              END;
              IF OK AND DeuxEcrans THEN DeuxEcrans:=FALSE;
                                        DessinerTableau;
                                        AfficheSuivant(Nb,N,Tab);
              END;
           END;
           CloseWindow (Win^);
        END;
        FreeGadgetList (Gl^);
     END;
  END Visualisation;
END Visu.
```

```
IMPLEMENTATION MODULE ModTest;

FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM Tasks IMPORT SignalSet,Wait;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
                WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,
                IntuiMessagePtr, CloseWindow,  DrawImage;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT RectFill,SetAPen,Move;
FROM VariablesGlobales IMPORT Str40,Scr,TableauParametres,Commande,cmap,
                        Icone;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Texte IMPORT ECRIT;
FROM UTILS IMPORT PRINT;
FROM Jauge IMPORT cdr;
FROM Help IMPORT SOS;
FROM Text IMPORT Text;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  T:LONGINT;
  Cl:INTEGER;
  Win:WindowPtr;
  ok,Done,Aff,Fin: BOOLEAN;
  i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  CH:CHAR;

PROCEDURE GadgetHandlerME(VAR w: Window;VAR Msg:MsgData; VAR Gad:
Gadget);

BEGIN

     CASE Gad.GadgetID OF
      0: Trace:="Il choisit la question A";
         trace(Trace); CH:="A"; SAY(11);
         cdr(Win,40,70,7,FALSE); cdr(Win,40,120,7,TRUE);|
      1: Trace:="Il choisit la question B";
         trace(Trace); CH:="B"; SAY(12);
         cdr(Win,40,70,7,TRUE); cdr(Win,40,120,7,FALSE);|
      2: Trace := "Il clique sur ok ";
         trace (Trace);
```

```
        ok:=TRUE;  |
      3: SAY(19); |
      4: PRINT; |
      5: Trace := "Il clique sur HELP ";
         trace (Trace);
         SOS(20); |
  END;
END GadgetHandlerME;

(*************************************************************************)
(*                                TEST                                   *)
(*************************************************************************)

PROCEDURE TEST(VAR ch:CHAR);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   CH:="X";
   BeginGadgetList();
        AddGadgetImageButton (40,70,Icone[3]);
        AddGadgetImageButton (40,120,Icone[3]);
        AddGadgetImageButton (190,215,Commande[2]);
        IF TableauParametres[1]=2
           THEN AddGadgetImageButton(130,215,Commande[4]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        IF TableauParametres[0]=1
           THEN AddGadgetImageButton(160,215,Commande[11]);
           ELSE AddGadgetImageButton(1,1,Commande[9]);
        END;
        AddGadgetImageButton (285,215,Commande[12]);
        Gl := EndGadgetList();
        IF (Gl # NIL) THEN
       Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
         IF (Win # NIL) THEN
             FRAME2(Win);
             IF TableauParametres[1] = 2 THEN SAY(19); END;
             ECRIT(Win,15);
             DrawImage (Win^.RPort^,Icone[3],0,0);
             DrawImage (Win^.RPort^,Commande[2],190,215);
             DrawImage (Win^.RPort^,Commande[12],285,215);
             IF (TableauParametres[1] = 2) THEN
                DrawImage (Win^.RPort^,Commande[4],130,215);
             END;
             IF (TableauParametres[0] = 1) THEN
                DrawImage (Win^.RPort^,Commande[11],160,215);
             END;
             Win^.RPort^.BgPen:=BYTE(0);
             Win^.RPort^.FgPen:=BYTE(1);
             SetAPen(Win^.RPort^,1);
             Move(Win^.RPort^,100,94);
             Text(Win^.RPort^,ADR("A) Moment de la dépense"),23);
             Move(Win^.RPort^,100,142);
             Text(Win^.RPort^,ADR("B) Epargne mensuelle"),20);
             Done:=FALSE;
             ok:=FALSE;
```

```
            WHILE (NOT Done) DO
               Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                  Msg := GetMsg(Win^.UserPort^);
                  IF (Msg = NIL) THEN EXIT; END;
                  ProcIMsg (Wp, Msg);
               END;
               IF ok AND (CH<>"X") THEN Done:=TRUE; ELSE ok:=FALSE; END
            END;
            ch:=CH;
            CloseWindow (Win^);
         END;
         FreeGadgetList (Gl^);
      END;
   END TEST;
END ModTest.
```

```
IMPLEMENTATION MODULE TestA;

FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM MathLib0 IMPORT entier,real;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget,WindowFlags,
               WindowFlagsSet,IDCMPFlags,IDCMPFlagsSet,IntuiMessagePtr,
               CloseWindow,DrawImage;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
                  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM Drawing IMPORT RectFill,SetAPen,Move,Draw;
FROM VariablesGlobales IMPORT Str40,Icone,Scr,TableauParametres,Commande;
FROM Text IMPORT Text;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY,SayMois;
FROM Trace IMPORT trace;
FROM Strings IMPORT StringLength;
FROM Texte IMPORT ECRIT;
FROM LireImage IMPORT LireMois,TabMois;
FROM annee IMPORT DATE,NOMMOIS;
FROM UTILS IMPORT PRINT;
FROM RealConversions IMPORT Decimal,ConvRealToString;
FROM Help IMPORT SOS;
FROM couleurs IMPORT gris,orange;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  T:LONGINT;
  YYY,jj,FR,mois,aa:INTEGER;
  Win:WindowPtr;
  ok,Done,Aff,bi: BOOLEAN;
  i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Tab:TabMois;

PROCEDURE GadgetHandler(VAR w: Window; VAR Msg : MsgData; VAR Gad:
Gadget);

BEGIN
    CASE Gad.GadgetID OF
      0: Done:=TRUE;
         Trace:="Il clique sur OK";
         trace(Trace); |
      1: SAY(21); |
      2: PRINT; |
      3: SOS(21); |
```

```
    END;
END GadgetHandler;

PROCEDURE AfficherCommandes;

BEGIN
     DrawImage (Win^.RPort^,Commande[2],190,215);
     DrawImage (Win^.RPort^,Commande[12],285,215);
     IF (TableauParametres[1] = 2) THEN
        DrawImage (Win^.RPort^,Commande[4],130,215);
     END;
     IF (TableauParametres[0] = 1) THEN
        DrawImage (Win^.RPort^,Commande[11],160,215);

     END;
END AfficherCommandes;

PROCEDURE ECRAN(Val:REAL);
VAR str:ARRAY [0..8] OF CHAR;
BEGIN
     SetAPen(Win^.RPort^,gris);
     RectFill(Win^.RPort^,1,1,319,255);
     FRAME2(Win);
     ECRIT(Win,17);
     IF TableauParametres[1] = 2 THEN SAY(21);  END;
     DrawImage (Win^.RPort^,Icone[3],0,0);
     SetAPen(Win^.RPort^,0);
     RectFill(Win^.RPort^,110,50,210,190);
     Move(Win^.RPort^,109,49);
     SetAPen(Win^.RPort^,1);
     Draw(Win^.RPort^,211,49);
     Draw(Win^.RPort^,211,191);
     Draw(Win^.RPort^,109,191);
     Draw(Win^.RPort^,109,49);
     ConvRealToString(Val,str,0,Decimal);
     SetAPen(Win^.RPort^,0);
     Win^.RPort^.FgPen:=BYTE(0);
     Win^.RPort^.BgPen:=BYTE(gris);
     Move(Win^.RPort^,160-4*StringLength(str),45);
     Text(Win^.RPort^,ADR(str),StringLength(str));
END ECRAN;

PROCEDURE REMPLIR(Val,VPM:REAL);

VAR i,N,echelle,m,Y,YY:INTEGER;
    echReel,S:REAL;
    str:ARRAY[0..8] OF CHAR;
    reste:CARDINAL;
BEGIN
     N:=entier(Val/VPM);
     echReel:=Val/140.;
     S:=0.;
     m:=mois;
     FOR i:=1 TO N DO
         SetAPen(Win^.RPort^,orange);
         S:=S+VPM;
         echelle:=entier(real(i)*VPM/echReel);
         Y:=190-echelle;
```

```
        YY:= 190-echelle+(echelle DIV i);
        RectFill(Win^.RPort^,110,Y,210,YY);
        SetAPen(Win^.RPort^,1);
        Move(Win^.RPort^,110,Y-1);
        Draw(Win^.RPort^,210,Y-1);
        Move(Win^.RPort^,110,YYY);
        Draw(Win^.RPort^,210,YYY);
        YYY:=Y-1;
        SetAPen(Win^.RPort^,gris);
        RectFill(Win^.RPort^,213,45,317,205);
        SetAPen(Win^.RPort^,1);
        Move(Win^.RPort^,213,Y-1);
        Draw(Win^.RPort^,215,Y-1);
        Draw(Win^.RPort^,214,Y-2);
        Draw(Win^.RPort^,214,Y);
        IF m=12 THEN m:=1; ELSE m:=m+1; END;
        DrawImage(Win^.RPort^,Tab[m-1],220,Y-13);
        ConvRealToString(S,str,0,Decimal);
        SetAPen(Win^.RPort^,0);
        Win^.RPort^.FgPen:=BYTE(0);
        Win^.RPort^.BgPen:=BYTE(gris);
        Move(Win^.RPort^,105-8*StringLength(str),Y+4);
        Text(Win^.RPort^,ADR(str),StringLength(str));
        Delay(150);
    END;
    reste:=entier(Val) MOD (entier(VPM));
    IF reste>0 THEN SetAPen(Win^.RPort^,orange);
                    S:=Val;
                    YY:=Y; Y:=50;
                    RectFill(Win^.RPort^,110,50,210,YY);
                    SetAPen(Win^.RPort^,1);
                    Move(Win^.RPort^,110,YYY);
                    Draw(Win^.RPort^,210,YYY);
                    SetAPen(Win^.RPort^,gris);
                    RectFill(Win^.RPort^,213,45,317,205);
                    SetAPen(Win^.RPort^,1);
                    Move(Win^.RPort^,213,Y-1);
                    Draw(Win^.RPort^,215,Y-1);
                    Draw(Win^.RPort^,214,Y-2);
                    Draw(Win^.RPort^,214,Y);
                    IF m=12 THEN m:=1; ELSE m:=m+1; END;
                    DrawImage(Win^.RPort^,Tab[m-1],220,Y-13);
                    ConvRealToString(S,str,0,Decimal);
                    SetAPen(Win^.RPort^,0);
                    Win^.RPort^.FgPen:=BYTE(0);
                    Win^.RPort^.BgPen:=BYTE(gris);
                    Move(Win^.RPort^,105-8*StringLength(str),Y+4);
                    Text(Win^.RPort^,ADR(str),StringLength(str));
    END;
END REMPLIR;
```

```
(*************************************************************************)
(*                                TEST-A                                 *)
(*************************************************************************)

PROCEDURE TestA(Val,ValeurParMois:REAL);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandler;
   END;
   YYY:=191;
   LireMois(Tab);
   DATE(jj,mois,aa,bi);
   BeginGadgetList();
       AddGadgetImageButton (190,215,Commande[2]);
       IF TableauParametres[1]=2
          THEN AddGadgetImageButton(130,215,Commande[4]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       IF TableauParametres[0]=1
          THEN AddGadgetImageButton(160,215,Commande[11]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       AddGadgetImageButton (285,215,Commande[12]);
   Gl := EndGadgetList();
   IF (Gl # NIL) THEN
      Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
      IF (Win # NIL) THEN
         ECRAN(Val);
         REMPLIR(Val,ValeurParMois);
         AfficherCommandes;
         Done:=FALSE;
         WHILE (NOT Done) DO
             Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
             LOOP
                Msg := GetMsg(Win^.UserPort^);
                IF (Msg = NIL) THEN EXIT; END;
                ProcIMsg (Wp, Msg);
             END;
         END;
         CloseWindow (Win^);
      END;
      FreeGadgetList (Gl^);
   END;
END TestA;
END TestA.
```

IMPLEMENTATION MODULE TestB;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM MathLib0 IMPORT entier,real;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget,WindowFlags,
               WindowFlagsSet,IDCMPFlags,IDCMPFlagsSet,IntuiMessagePtr,
               CloseWindow,DrawImage;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
                   AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM Drawing IMPORT RectFill,SetAPen,Move,Draw;
FROM VariablesGlobales IMPORT Str40,Icone,Scr,TableauParametres,Commande;
FROM Text IMPORT Text;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY,SayMois;
FROM Trace IMPORT trace;
FROM Strings IMPORT StringLength;
FROM Texte IMPORT ECRIT;
FROM LireImage IMPORT LireMois,TabMois;
FROM annee IMPORT DATE,NOMMOIS;
FROM UTILS IMPORT PRINT;
FROM RealConversions IMPORT Decimal,ConvRealToString;
FROM Help IMPORT SOS;
FROM couleurs IMPORT gris,orange;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  T:LONGINT;
  YYY,jj,FR,mois,aa:INTEGER;
  Win:WindowPtr;
  ok,Done,Aff,bi: BOOLEAN;
  i : CARDINAL;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Tab:TabMois;

PROCEDURE GadgetHandler(VAR w: Window; VAR Msg : MsgData; VAR Gad:
Gadget);

BEGIN
     CASE Gad.GadgetID OF
       0: Done:=TRUE;
          Trace:="Il clique sur OK";
          trace(Trace); |
       1: SAY(22); |
       2: PRINT; |
       3: SOS(22); |
```

```
    END;
END GadgetHandler;

PROCEDURE AfficherCommandes;

BEGIN
     DrawImage (Win^.RPort^,Commande[2],190,215);
     DrawImage (Win^.RPort^,Commande[12],285,215);
     IF (TableauParametres[1] = 2) THEN
        DrawImage (Win^.RPort^,Commande[4],130,215);
     END;
     IF (TableauParametres[0] = 1) THEN
        DrawImage (Win^.RPort^,Commande[11],160,215);

     END;
END AfficherCommandes;

PROCEDURE ECRAN(Val:REAL);
VAR str:ARRAY [0..8] OF CHAR;
BEGIN
     SetAPen(Win^.RPort^,gris);
     RectFill(Win^.RPort^,1,1,319,255);
     FRAME2(Win);
     ECRIT(Win,18);
     IF TableauParametres[1] = 2 THEN SAY(22);  END;
     DrawImage (Win^.RPort^,Icone[3],0,0);
     SetAPen(Win^.RPort^,0);
     RectFill(Win^.RPort^,110,50,210,190);
     Move(Win^.RPort^,109,49);
     SetAPen(Win^.RPort^,1);
     Draw(Win^.RPort^,211,49);
     Draw(Win^.RPort^,211,191);
     Draw(Win^.RPort^,109,191);
     Draw(Win^.RPort^,109,49);
     ConvRealToString(Val,str,0,Decimal);
     SetAPen(Win^.RPort^,0);
     Win^.RPort^.FgPen:=BYTE(0);
     Win^.RPort^.BgPen:=BYTE(gris);
     Move(Win^.RPort^,160-4*StringLength(str),45);
     Text(Win^.RPort^,ADR(str),StringLength(str));
END ECRAN;

PROCEDURE REMPLIR(Val:REAL;N:INTEGER);

VAR i,echelle,m,Y,YY:INTEGER;
    echReel,S:REAL;
    str:ARRAY[0..8] OF CHAR;

BEGIN
     echReel:=Val/140.;
     S:=0.;
     m:=mois;
     FOR i:=1 TO N DO
         SetAPen(Win^.RPort^,orange);
         S:=S+(Val/real(N));
         echelle:=entier(S/echReel);
         Y:=190-echelle;
         YY:= 190-echelle+(echelle DIV i);
```

```
        RectFill(Win^.RPort^,110,Y,210,YY);
        SetAPen(Win^.RPort^,1);
        Move(Win^.RPort^,110,Y-1);
        Draw(Win^.RPort^,210,Y-1);
        SetAPen(Win^.RPort^,1);
        Move(Win^.RPort^,110,YYY);
        Draw(Win^.RPort^,210,YYY);
        YYY:=Y-1;
        SetAPen(Win^.RPort^,gris);
        RectFill(Win^.RPort^,213,45,317,205);
        SetAPen(Win^.RPort^,1);
        Move(Win^.RPort^,213,Y-1);
        Draw(Win^.RPort^,215,Y-1);
        Draw(Win^.RPort^,214,Y-2);
        Draw(Win^.RPort^,214,Y);
        IF m=12 THEN m:=1; ELSE m:=m+1; END;
        DrawImage(Win^.RPort^,Tab[m-1],220,Y-13);
        ConvRealToString(S,str,0,Decimal);
        SetAPen(Win^.RPort^,0);
        Win^.RPort^.FgPen:=BYTE(0);
        Win^.RPort^.BgPen:=BYTE(gris);
        Move(Win^.RPort^,105-8*StringLength(str),Y+4);
        Text(Win^.RPort^,ADR(str),StringLength(str));
        Delay(150);
    END;
END REMPLIR;

(*************************************************************************)
(*                              TEST-B                                   *)
(*************************************************************************)

PROCEDURE TestB(Val:REAL;N:INTEGER);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandler;
   END;
   YYY:=191;
   LireMois(Tab);
   DATE(jj,mois,aa,bi);
   BeginGadgetList();
       AddGadgetImageButton (190,215,Commande[2]);
       IF TableauParametres[1]=2
          THEN AddGadgetImageButton(130,215,Commande[4]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       IF TableauParametres[0]=1
          THEN AddGadgetImageButton(160,215,Commande[11]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       AddGadgetImageButton (285,215,Commande[12]);
   Gl := EndGadgetList();
   IF (Gl # NIL) THEN
      Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
      IF (Win # NIL) THEN
         ECRAN(Val);
         REMPLIR(Val,N);
         AfficherCommandes;
```

```
            Done:=FALSE;
            WHILE (NOT Done) DO
                Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
                LOOP
                   Msg := GetMsg(Win^.UserPort^);
                   IF (Msg = NIL) THEN EXIT; END;
                   ProcIMsg (Wp, Msg);
                END;
            END;
            CloseWindow (Win^);
        END;
        FreeGadgetList (Gl^);
    END;
END TestB;
END TestB.
```

IMPLEMENTATION MODULE SaisieMois;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDosProcess IMPORT Delay;
FROM MathLib0 IMPORT entier,real;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget,WindowFlags,
               WindowFlagsSet,IDCMPFlags,IDCMPFlagsSet,IntuiMessagePtr,
               CloseWindow,DrawImage;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList,
                  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM Drawing IMPORT RectFill,SetAPen,Move,Draw;
FROM VariablesGlobales IMPORT Str40,Icone,Scr,TableauParametres,Commande;
FROM Text IMPORT Text;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY,SayMois;
FROM Trace IMPORT trace;
FROM Jauge IMPORT cdr;
FROM Texte IMPORT ECRIT;
FROM LireImage IMPORT LireMois,TabMois;
FROM annee IMPORT DATE,NOMMOIS;
FROM UTILS IMPORT PRINT;
FROM Help IMPORT SOS;
FROM Conversions IMPORT ConvNumberToString;
FROM Strings IMPORT ConcatString;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  i,jj,FR,mois,aa,X,Y:INTEGER;
  n:CARDINAL;
  Win:WindowPtr;
  ok,Done,bi: BOOLEAN;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Tab:TabMois;

PROCEDURE GadgetHandler(VAR w: Window;VAR Msg:MsgData;VAR Gad:Gadget);

VAR str:ARRAY[0..2] OF CHAR;
    Ph:Str40;
BEGIN
     IF Gad.GadgetID <4 THEN CASE Gad.GadgetID OF
                                0: IF ok THEN Done:=TRUE; END;
                                   Trace:="Il clique sur OK";
                                   trace(Trace); |
                                1: SAY(7); |
                                2: PRINT; |
                                3: SOS(23); |
```

```
                        END;
                    ELSE cdr(Win,X,Y,8,TRUE);
                         n:=Gad.GadgetID-3;
                         ok:=TRUE;
                         X:=30+90*((n-1) DIV 5);
                         Y:=43+33*((n-1) MOD 5);
                         cdr(Win,X,Y,8,FALSE);
                         Move(Win^.RPort^,112,39);
                         ConvNumberToString(str,LONGINT(n),FALSE,
                                            10,2," ");
                         Ph:="Dans ";
                         ConcatString(Ph,str);
                         ConcatString(Ph," mois.  ");
                         SetAPen(Win^.RPort^,1);
                         Win^.RPort^.FgPen:=BYTE(1);
                         Win^.RPort^.BgPen:=BYTE(0);
                         SetAPen(Win^.RPort^,1);
                         Text(Win^.RPort^,ADR(Ph),14);
    END;
END GadgetHandler;

PROCEDURE AfficherCommandes;
(*------------------------*)

BEGIN
     DrawImage (Win^.RPort^,Commande[2],190,215);
     DrawImage (Win^.RPort^,Commande[12],285,215);
     IF (TableauParametres[1] = 2) THEN
        DrawImage (Win^.RPort^,Commande[4],130,215);
     END;
     IF (TableauParametres[0] = 1) THEN
        DrawImage (Win^.RPort^,Commande[11],160,215);

     END;
END AfficherCommandes;

PROCEDURE ECRAN;
(*------------*)

BEGIN
     FRAME2(Win);
     ECRIT(Win,7);
     IF TableauParametres[1] = 2 THEN SAY(7);  END;
     DrawImage (Win^.RPort^,Icone[3],0,0);
END ECRAN;
```

```
(*************************************************************************)
(*                              SaisieMois                               *)
(*************************************************************************)

PROCEDURE SMois(VAR N:INTEGER);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandler;
   END;
   ok:=FALSE;
   LireMois(Tab);
   DATE(jj,mois,aa,bi);
   BeginGadgetList();
       AddGadgetImageButton (190,215,Commande[2]);
       IF TableauParametres[1]=2
          THEN AddGadgetImageButton(130,215,Commande[4]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       IF TableauParametres[0]=1
          THEN AddGadgetImageButton(160,215,Commande[11]);
          ELSE AddGadgetImageButton(1,1,Commande[9]);
       END;
       AddGadgetImageButton (285,215,Commande[12]);
       FOR i:=1 TO 15 DO
           IF mois=12 THEN mois:=1 ELSE mois:=mois+1; END;
           AddGadgetImageButton (30+90*((i-1) DIV 5),43+33*((i-1) MOD 5),
                                 Tab[mois-1]);
       END;
   Gl := EndGadgetList();
   X:=30; Y:=43;
   IF (Gl # NIL) THEN
      Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
      IF (Win # NIL) THEN
         ECRAN;
         AfficherCommandes;
         Done:=FALSE;
         WHILE (NOT Done) DO
             Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
             LOOP
                Msg := GetMsg(Win^.UserPort^);
                IF (Msg = NIL) THEN EXIT; END;
                ProcIMsg (Wp, Msg);
             END;
         END;
         N:=n;
         CloseWindow (Win^);
      END;
      FreeGadgetList (Gl^);
   END;
END SMois;
END SaisieMois.
```

IMPLEMENTATION MODULE Texte;

```
FROM SYSTEM IMPORT ADR,BYTE ;
FROM Intuition IMPORT WindowPtr;
FROM Strings IMPORT StringLength,LocateChar,ExtractSubString;
FROM SimpleConsole IMPORT CreateConsole,PutStr;
FROM SimpleConsoleCmds IMPORT wMove,wSetCursor;
FROM Couleurs IMPORT gris;
FROM Text IMPORT Text;
FROM Drawing IMPORT RectFill,Move,SetAPen,Draw;
FROM Rasters IMPORT RastPortPtr,RastPort;
FROM InOut IMPORT WriteString,WriteLn,WriteInt;
FROM VariablesGlobales IMPORT Str60,Str40,Phrases,cmap;
FROM MathLib0 IMPORT entier,real;

VAR
    lc:CARDINAL;
    rp:RastPortPtr;
    n,I,ent:INTEGER;
    p1,p2:Str40;
    tab:REAL;

PROCEDURE DIVISE (P:Str60; VAR P1,P2 : Str40; VAR NbP : INTEGER);
(*-------------------------------------------------------------*)

VAR res,l : INTEGER;


BEGIN
     NbP := 1;
     res:=LocateChar(P,"@",1,40);
     IF (res # -1) THEN
       NbP := 2;
       l:=StringLength(P);
       ExtractSubString(P1,P,0,res);
       ExtractSubString(P2,P,res+1,l);
     END;
END DIVISE;

PROCEDURE ECRIT(FEN:WindowPtr;X:INTEGER);
(*-----------------------------------*)

VAR p1,p2:Str40;
    n,i:INTEGER;

BEGIN
DIVISE(Phrases[X],p1,p2,n);

 rp:=FEN^.RPort;
 rp^.FgPen:=BYTE(0);
 rp^.BgPen:=BYTE(gris);
 SetAPen(rp^,gris);
 RectFill(rp^,1,1,318,30);
 SetAPen(rp^,0);

IF n=1 THEN
```

```
    lc:=StringLength(Phrases[X]);
    tab:=20.-(real(lc) / 2.);
    ent:=entier(tab*8.);
    Move(rp^,ent,15);
    Text(rp^,ADR(Phrases[X]),StringLength(Phrases[X]));
 ELSE
    lc:=StringLength(p1);
    tab:=20.-(real(lc) / 2.);
    ent:=entier(tab*8.);
    Move(rp^,ent,11);
    Text(rp^,ADR(p1),StringLength(p1));
    lc:=StringLength(p2);
    tab:=20.-(real(lc) / 2.);
    ent:=entier(tab*8.);
    Move(rp^,ent,23);
    Text(rp^,ADR(p2),StringLength(p2));
END
END ECRIT;

END Texte.
```

IMPLEMENTATION MODULE Trace;

```
FROM VariablesGlobales IMPORT
Nom,Str3,Str20,Str40,TableauTraces,TableauParametres;
FROM Strings IMPORT CopyString,ConcatString;
FROM InOut IMPORT OpenInputFile, CloseInput, OpenOutputFile,
                  CloseOutput, Done, WriteString,WriteLn;
FROM RealInOut IMPORT ReadReal, WriteReal;
FROM RealConversions IMPORT ConvRealToString,RealToStringFormat;

PROCEDURE trace( T : Str40 );
(*-------------------------*)
BEGIN
   IF TableauTraces.Indice <= 499 THEN
      CopyString ( TableauTraces.Tab [TableauTraces.Indice] ,T);
      TableauTraces.Indice :=  TableauTraces.Indice + 1;
   END;
END trace;

PROCEDURE TraceParametres;
(*-----------------------*)
BEGIN  TableauTraces.Indice := 1; trace(" ");
       trace("Liste des paramètres choisis");
       IF TableauParametres[0] = 0
          THEN trace("Pas d'impression")
          ELSE trace("Impression du résumé")
       END;
       IF TableauParametres[1] = 0
          THEN trace("Questions posées par écrit")
          ELSE trace("Questions posées par écrit + parole")
       END;
       IF TableauParametres[2] = 0
          THEN trace("Saisie du Nom par clavier")
          ELSE trace("Saisie du Nom par souris")
       END;
       IF TableauParametres[3] = 0
          THEN trace("Recettes saisies par clavier");
       END;
       IF TableauParametres[3] = 1
          THEN trace("Recettes saisies par billets");
        END;
       IF TableauParametres[3] = 2
          THEN trace("Recettes saisies par calculette");
       END;
       IF TableauParametres[4] = 0
          THEN trace("Dépenses saisies par clavier");
       END;
       IF TableauParametres[4] = 1
          THEN trace("Dépenses saisies par billets");
       END;
       IF TableauParametres[4] = 2
          THEN trace("Dépenses saisies par calculette");
      END;
      IF TableauParametres[6] = 0
         THEN trace("Notion du temps présente")
         ELSE trace("Notion du temps absente")
```

```
      END;
      IF TableauParametres[9] = 2
         THEN trace("Cycle d'un mois")
      END;
      IF TableauParametres[9] = 1
         THEN trace("Cycle d'une semaine")
      END;
      IF TableauParametres[9] = 1
          THEN
            CASE TableauParametres[10] OF
              0: trace("Début de cycle = lundi")|
              1: trace("Début de cycle = mardi")|
              2: trace("Début de cycle = mercredi")|
              3: trace("Début de cycle = jeudi")|
              4: trace("Début de cycle = vendredi")|
              5: trace("Début de cycle = samedi")|
              6: trace("Début de cycle = dimanche")
            END;
               trace(" ")
         ELSE  trace(" ");
       END;
END TraceParametres;

PROCEDURE sauver_traces;
(*--------------------*)
VAR FileName,FileName2,Ph:Str40;
    i:INTEGER;
    DernVers:REAL;
    St1:Str3;
BEGIN   FileName := "Param:";
        ConcatString (FileName,Nom);
        CopyString (FileName2,FileName);
        ConcatString (FileName2,".DV");
        OpenInputFile (FileName2);
        IF Done THEN ReadReal (DernVers);
                     DernVers := DernVers + 1.;
               ELSE DernVers := 0.
        END; CloseInput;
        OpenOutputFile (FileName2);
        WriteReal (DernVers,4);
        CloseOutput;
        ConvRealToString (DernVers,St1,0,Decimal);
        ConcatString (FileName,St1); ConcatString (FileName,".TRACE");
        CopyString (Ph,Nom); ConcatString (Ph," : trace n° ");
        ConcatString (Ph,St1); CopyString (TableauTraces.Tab[0],Ph);
        OpenOutputFile (FileName);
        i := 0;
        WHILE (i <= TableauTraces.Indice) DO
           WriteString (TableauTraces.Tab[i]); WriteLn;
           i := i + 1;
        END;
        CloseOutput;
END sauver_traces;
END Trace.
```

IMPLEMENTATION MODULE Parole;

```
FROM SYSTEM IMPORT ADR;
FROM SoundLib IMPORT PlaySamp,ASampPtr,LoadAIFF;
FROM VariablesGlobales IMPORT Son;
FROM IO IMPORT Str15;
FROM Strings IMPORT ConcatString;
FROM Memory IMPORT FreeMem;
FROM annee IMPORT NOMMOIS;

PROCEDURE SAY(X:INTEGER);
(*----------------------*)
BEGIN
     PlaySamp(Son[X]^.Samp,3,480,32,Son[X]^.Size,1,1);
END SAY;

PROCEDURE SayPoste(nom:Str15);
(*--------------------------*)

VAR SON:ASampPtr;
    fichier:Str40;

BEGIN
     fichier:="Son:son-";
     ConcatString(fichier,nom);
     SON:=LoadAIFF(ADR(fichier));
     PlaySamp(SON^.Samp,3,480,32,SON^.Size,1,1);
     FreeMem(SON^.Samp,SON^.Size);
END SayPoste;

PROCEDURE SayMois(x:INTEGER);
(*-------------------------*)

VAR SON:ASampPtr;
    fichier:Str40;
    mois:ST10;

BEGIN
     fichier:="Son:sonMois";
     NOMMOIS(mois,x);
     ConcatString(fichier,mois);
     SON:=LoadAIFF(ADR(fichier));
     PlaySamp(SON^.Samp,3,480,32,SON^.Size,1,1);
     FreeMem(SON^.Samp,SON^.Size);
END SayMois;

PROCEDURE SayFreq(x:INTEGER);
(*------------------------*)

VAR SON,SON2,SON3:ASampPtr;

BEGIN
     CASE x OF
         0: SON:=LoadAIFF(ADR("son:sonfreq1"));
            SON2:=NIL;
            SON3:=NIL;
```

```
        1: SON:=LoadAIFF(ADR("son:sonfreq2"));
           SON2:=LoadAIFF(ADR("son:sonfreq3"));
           SON3:=LoadAIFF(ADR("son:sonfreq6")); |
        2: SON:=LoadAIFF(ADR("son:sonfreq2"));
           SON2:=LoadAIFF(ADR("son:sonfreq5"));
           SON3:=LoadAIFF(ADR("son:sonfreq6")); |
        3: SON:=LoadAIFF(ADR("son:sonfreq2"));
           SON2:=LoadAIFF(ADR("son:sonfreq6"));
           SON3:=NIL; |
        4: SON:=LoadAIFF(ADR("son:sonfreq2"));
           SON2:=LoadAIFF(ADR("son:sonfreq4"));
           SON3:=LoadAIFF(ADR("son:sonfreq6")); |
        5: SON:=LoadAIFF(ADR("son:sonfreq2"));
           SON2:=LoadAIFF(ADR("son:sonfreq7"));
           SON3:=NIL; |
    END;
    PlaySamp(SON^.Samp,3,470,32,SON^.Size,1,1);
    IF (SON2 # NIL) THEN PlaySamp(SON2^.Samp,3,470,32,SON2^.Size,1,1);
    END;
    IF (SON3 # NIL) THEN PlaySamp(SON3^.Samp,3,470,32,SON3^.Size,1,1);
    END;
    FreeMem(SON^.Samp,SON^.Size);
    FreeMem(SON2^.Samp,SON2^.Size);
    FreeMem(SON3^.Samp,SON3^.Size);
END SayFreq;

END Parole.
```

IMPLEMENTATION MODULE LireImage;

```
FROM SYSTEM IMPORT BYTE,ADR;
FROM Intuition IMPORT Image;
FROM FindImages IMPORT ImageDescTablePtr, FindImageTable;

VAR ImgPtr,ImgPtr2,ImgPtr3,ImgPtr4,ImgPtr5,ImgPtr6,
    ImgPtr7,ImgPtr8:ImageDescTablePtr;
    ImgCount,Indice:CARDINAL;

PROCEDURE LoadPostes(VAR POSTE:TP);
(*-------------------------------*)
BEGIN
   ImgPtr := FindImageTable (87878787H,ImgCount);
   IF (ImgPtr # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH POSTE [Indice] DO
       LeftEdge   := 0;
       TopEdge    := 0;
       Width      := ImgPtr^[Indice].Width;
       Height     := ImgPtr^[Indice].Height;
       Depth      := ImgPtr^[Indice].Depth;
       ImageData  := ImgPtr^[Indice].Data;
       PlanePick  := BYTE(31);
       PlaneOnOff := BYTE(31);
       NextImage  := NIL;
      END;
      Indice :=Indice+1;
     END;
   END;
END LoadPostes;

PROCEDURE LoadMini(VAR Mini:TabMini);
(*---------------------------------*)
BEGIN
   ImgPtr2 := FindImageTable (11122233H,ImgCount);
   IF (ImgPtr2 # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH Mini [Indice] DO
       LeftEdge   := 0;
       TopEdge    := 0;
       Width      := ImgPtr2^[Indice].Width;
       Height     := ImgPtr2^[Indice].Height;
       Depth      := ImgPtr2^[Indice].Depth;
       ImageData  := ImgPtr2^[Indice].Data;
       PlanePick  := BYTE(31);
       PlaneOnOff := BYTE(31);
       NextImage  := NIL;
      END;
      Indice :=Indice+1;
     END;
   END;
END LoadMini;
```

```
PROCEDURE LoadCalcu(VAR Chiffre:TabCalcu);
BEGIN
      ImgPtr4 := FindImageTable(45454545H, ImgCount);
      IF (ImgPtr4 # NIL) AND (ImgCount # 0) THEN
        Indice := 0;
        WHILE (Indice <= ImgCount-1) DO
          WITH Chiffre[Indice] DO
       LeftEdge    := 0;
       TopEdge     := 0;
       Width       := ImgPtr4^[Indice].Width;
       Height      := ImgPtr4^[Indice].Height;
       Depth       := ImgPtr4^[Indice].Depth;
       ImageData   := ImgPtr4^[Indice].Data;
       PlanePick   := BYTE(31);
       PlaneOnOff  := BYTE(31);
       NextImage := NIL;
          END;
          Indice := Indice + 1;
        END;
      END;
END LoadCalcu;

PROCEDURE LoadLettre(VAR TabImages:TabLettre);
BEGIN
   ImgPtr5 := FindImageTable (33335555H,ImgCount);
   IF (ImgPtr5 # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH TabImages [Indice] DO
       LeftEdge    := 0;
       TopEdge     := 0;
       Width       := ImgPtr5^[Indice].Width;
       Height      := ImgPtr5^[Indice].Height;
       Depth       := ImgPtr5^[Indice].Depth;
       ImageData   := ImgPtr5^[Indice].Data;
       PlanePick   := BYTE(31);
       PlaneOnOff  := BYTE(31);
       NextImage   := NIL;
      END;
      Indice :=Indice+1;
     END;
   END;
END LoadLettre;

PROCEDURE LoadRO(VAR image:TabRO);
BEGIN
   ImgPtr8 := FindImageTable (89898989H,ImgCount);
   IF (ImgPtr8 # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH image [Indice] DO
       LeftEdge    := 0;
       TopEdge     := 0;
       Width       := ImgPtr8^[Indice].Width;
       Height      := ImgPtr8^[Indice].Height;
       Depth       := ImgPtr8^[Indice].Depth;
       ImageData   := ImgPtr8^[Indice].Data;
       PlanePick   := BYTE(31);
```

```
        PlaneOnOff := BYTE(31);
        NextImage  := NIL;
       END;
       Indice :=Indice+1;
      END;
    END;
END LoadRO;

PROCEDURE LoadFreq(VAR freq:TabFreq);
(*--------------------------------*)
BEGIN
   ImgPtr7 := FindImageTable (52525252H,ImgCount);
   IF (ImgPtr7 # NIL) AND (ImgCount # 0) THEN
     Indice :=0;
     WHILE (Indice <= ImgCount-1) DO
      WITH freq [Indice] DO
       LeftEdge   := 0;
       TopEdge    := 0;
       Width      := ImgPtr7^[Indice].Width;
       Height     := ImgPtr7^[Indice].Height;
       Depth      := ImgPtr7^[Indice].Depth;
       ImageData  := ImgPtr7^[Indice].Data;
       PlanePick  := BYTE(31);
       PlaneOnOff := BYTE(31);
       NextImage  := NIL;
      END;
      Indice :=Indice+1;
     END;
   END;
END LoadFreq;

PROCEDURE LoadMonnaie(VAR Monnaie:TabMonnaie);
BEGIN
    ImgPtr6 := FindImageTable(12111111H, ImgCount);
      IF (ImgPtr6 # NIL) AND (ImgCount # 0) THEN
        Indice := 0;
        WHILE (Indice <= ImgCount - 1) DO
          WITH Monnaie[Indice] DO
            LeftEdge    := 0;
             TopEdge     := 0;
       Width       := ImgPtr6^[Indice].Width;
       Height      := ImgPtr6^[Indice].Height;
       Depth       := ImgPtr6^[Indice].Depth;
       ImageData   := ImgPtr6^[Indice].Data;
       PlanePick   := BYTE(31);
       PlaneOnOff  := BYTE(31);
       NextImage := NIL;
          END;
          Indice := Indice + 1;
         END; END;
END LoadMonnaie;
```

```
PROCEDURE LireMois(VAR Mois: TabMois);
(*-----------------------------------*)
VAR ImgPtr2 : ImageDescTablePtr;
    ImgCount,Indice : CARDINAL;

BEGIN
  ImgPtr3 := FindImageTable(54444444H, ImgCount);
      IF (ImgPtr3 # NIL) AND (ImgCount # 0) THEN
        Indice := 0;
        WHILE (Indice <= ImgCount-1) DO
          WITH Mois[Indice] DO
             LeftEdge    := 0;
             TopEdge     := 0;
             Width       := ImgPtr3^[Indice].Width;
             Height      := ImgPtr3^[Indice].Height;
             Depth       := ImgPtr3^[Indice].Depth;
             ImageData   := ImgPtr3^[Indice].Data;
             PlanePick   := BYTE(31);
             PlaneOnOff  := BYTE(31);
             NextImage   := NIL;
          END;
          Indice := Indice + 1;
        END;
      END;
END LireMois;

END LireImage.
```

IMPLEMENTATION MODULE annee;

```
FROM AmigaDOS IMPORT DateStampRecord,DateStamp;
FROM SYSTEM IMPORT BYTE, ADR, SHORT;
FROM MathLib0 IMPORT entier,real;
FROM VariablesGlobales IMPORT ST10;

VAR
   Bi : BOOLEAN;
   X,AA,JJ,MM:INTEGER;
   NJOURS:TABNJ;

PROCEDURE MOIS( VAR MM:INTEGER);
VAR I: INTEGER;
    TrouveMOIS : BOOLEAN;
BEGIN
   TrouveMOIS := FALSE;
   I:=0;
   WHILE (NOT TrouveMOIS) AND (I<=11) DO
        IF (X>NJOURS[I]) THEN X:=X-NJOURS[I];
                              I:=I+1;
                         ELSE MM:=I+1;
                              TrouveMOIS:=TRUE;
                         END;
   END;
END MOIS;

PROCEDURE RechJour(j,m,a:INTEGER;VAR n:INTEGER);
(*-----------------------------------------*)
VAR

BEGIN
     a:=a+1900;
     n:=(a DIV 100) - (a DIV 400);
     IF m<3 THEN m:=m+12;
                 a:=a-1;
     END;
     n:=j-1+(13*(m+1) DIV 5) + (5*a DIV 4) - n;
     n:=n-(7*(n DIV 7));
     IF n=0 THEN n:=7; END;
END RechJour;

PROCEDURE DATE ( VAR JJ,MM,AA:INTEGER; VAR Bi:BOOLEAN);

VAR
   Y :INTEGER;
   En : INTEGER;
   an,Quatre,Re,Re2:REAL;
   TrouveAn : BOOLEAN;
   DSR : DateStampRecord;

BEGIN

   DateStamp (DSR);
   X := SHORT (DSR.dsDays);
   Y := X;
```

```
   X := X + 1;
   AA :=78;
   TrouveAn := FALSE;
   WHILE NOT TrouveAn DO
      an := real (AA);
      Quatre := 4.0;
      Re := an/Quatre;
      En := entier (Re);
      Re2 := real (En);
      IF Re=Re2
      THEN
         IF X>366
         THEN
            X := X - 366;
            AA := AA + 1;
         ELSE
            TrouveAn := TRUE;
            Bi:=TRUE;
            MOIS(MM);
            JJ:=X;
         END;
      ELSE
         IF X>365
         THEN
            X := X-365;
            AA := AA+1;
         ELSE
            TrouveAn := TRUE;
            Bi:=FALSE;
            MOIS(MM);
            JJ:=X;
         END;
      END;
   END;
END DATE;

PROCEDURE NbJours(VAR NJOURS:TABNJ);
(*--------------------------------*)
VAR J,M,A:INTEGER;
    B:BOOLEAN;
BEGIN
   DATE(J,M,A,B);
   NJOURS[0]:=31;
   IF B THEN NJOURS[1]:=29;
        ELSE NJOURS[1]:=28;
   END;
   NJOURS[2]:=31;
   NJOURS[3]:=30;
   NJOURS[4]:=31;
   NJOURS[5]:=30;
   NJOURS[6]:=31;
   NJOURS[7]:=31;
   NJOURS[8]:=30;
   NJOURS[9]:=31;
   NJOURS[10]:=30;
   NJOURS[11]:=31;
END NbJours;
```

```
PROCEDURE NOMMOIS (VAR MOIS:ST10; I:INTEGER);

BEGIN
   CASE I OF

        1: MOIS:="Janvier";
        2: MOIS:="Février";
        3: MOIS:="Mars";
        4: MOIS:="Avril";
        5: MOIS:="Mai";
        6: MOIS:="Juin";
        7: MOIS:="Juillet";
        8: MOIS:="Aôut";
        9: MOIS:="Septembre";
        10: MOIS:="Octobre";
        11: MOIS:="Novembre";
        12: MOIS:="Décembre";
  END;
END NOMMOIS;

BEGIN
     NbJours(NJOURS);
END annee.
```

IMPLEMENTATION MODULE Cadre;

```
FROM Drawing   IMPORT Flood,SetAPen,Draw,Move,RectFill;
FROM Intuition IMPORT WindowPtr;
FROM Rasters   IMPORT RastPortPtr;
FROM Couleurs IMPORT gris,blanc,noir;

PROCEDURE FRAME (win:WindowPtr);

VAR
   RP :RastPortPtr;

BEGIN
   RP:=win^.RPort;
   SetAPen(RP^,blanc);
   Move(RP^,0,0);
   Draw(RP^,0,255);
   Draw(RP^,319,255);
   Draw(RP^,319,0);
   Draw(RP^,0,0);
   Move(RP^,78,31);
   Draw(RP^,78,210);
   Draw(RP^,319,210);
   Draw(RP^,319,31);
   Draw(RP^,78,31);
   SetAPen(RP^,gris);
   RectFill(RP^,1,31,77,210);
   RectFill(RP^,1,1,318,29);
   RectFill(RP^,1,211,318,254);

END FRAME;

PROCEDURE FRAME2 (win:WindowPtr);

VAR
   RP :RastPortPtr;
BEGIN
   RP:=win^.RPort;
   SetAPen(RP^,blanc);
   Move(RP^,0,0);
   Draw(RP^,0,255);
   Draw(RP^,319,255);
   Draw(RP^,319,0);
   Draw(RP^,0,0);
   Move(RP^,0,31);
   Draw(RP^,320,31);
   Move(RP^,0,210);
    Draw(RP^,320,210);
   SetAPen(RP^,gris);
   RectFill(RP^,1,1,318,29);
   RectFill(RP^,1,211,318,254);
END FRAME2;
END Cadre.
```

IMPLEMENTATION MODULE couleurs;

```
BEGIN
   noir:=0;
   blanc:=1;
   rouge:=2;
   jaune:=5;
   vert:=7;
   bleu:=12;
   gris:=26; (* 28 *)
   orange:=17;
   bleuclair:=10;
END couleurs.
```

IMPLEMENTATION MODUL VariablesGlobales;

```
FROM SYSTEM IMPORT BYTE;
FROM FindImages IMPORT ImageDescTablePtr, FindImageTable;
FROM Intuition IMPORT Image;
FROM IO IMPORT LirePhrases;

VAR ImgPtr: ImageDescTablePtr;
    ImgCount,Indice: CARDINAL;

BEGIN
      LirePhrases ;
  cmap[00] := 0FFFH;  (* 0000H *)
  cmap[01] := 0000H;  (* 0FFFH *)
  cmap[02] := 0E00H;
  cmap[03] := 0A00H;
  cmap[04] := 0D80H;
  cmap[05] := 0FE0H;
  cmap[06] := 08F0H;
  cmap[07] := 0080H;
  cmap[08] := 00B6H;
  cmap[09] := 00DDH;
  cmap[10] := 00AFH;
  cmap[11] := 007CH;
  cmap[12] := 000FH;
  cmap[13] := 070FH;
  cmap[14] := 0C0EH;
  cmap[15] := 0C08H;
  cmap[16] := 0620H;
  cmap[17] := 0E52H;
  cmap[18] := 0A52H;
  cmap[19] := 0FCAH;
  cmap[20] := 0DB9H;
  cmap[21] := 0444H;
  cmap[22] := 0555H;
  cmap[23] := 0666H;
  cmap[24] := 0777H;
  cmap[25] := 0888H;
  cmap[26] := 0999H;
  cmap[27] := 0AAAH;
  cmap[28] := 0CCCH;
  cmap[29] := 0DDDH;
```

```
  cmap[30] := 0EEEH;
  cmap[31] := 0000H; (* 0FFFH *)

    ImgPtr := FindImageTable(33333333H, ImgCount);
    IF (ImgPtr # NIL) AND (ImgCount # 0) THEN
      Indice := 0;
      WHILE (Indice <= ImgCount -1) DO
        WITH Commande[Indice] DO
          LeftEdge    := 0;
         TopEdge      := 0;
         Width        := ImgPtr^[Indice].Width;
         Height       := ImgPtr^[Indice].Height;
         Depth        := ImgPtr^[Indice].Depth;
         ImageData    := ImgPtr^[Indice].Data;
         PlanePick    := BYTE(31);
         PlaneOnOff   := BYTE(31);
         NextImage := NIL;
        END;
        Indice := Indice + 1;
      END;
    END;

    ImgPtr := FindImageTable(14111111H, ImgCount);
    IF (ImgPtr # NIL) AND (ImgCount # 0) THEN
      Indice := 0;
      WHILE (Indice <= ImgCount -1) DO
        WITH Icone[Indice] DO
          LeftEdge    := 0;
         TopEdge      := 0;
         Width        := ImgPtr^[Indice].Width;
         Height       := ImgPtr^[Indice].Height;
         Depth        := ImgPtr^[Indice].Depth;
         ImageData    := ImgPtr^[Indice].Data;
         PlanePick    := BYTE(31);
         PlaneOnOff   := BYTE(31);
         NextImage := NIL;
        END;
        Indice := Indice + 1;
      END;
    END;

END VariablesGlobales.
```

IMPLEMENTATION MODULE UTILS;

```
FROM SYSTEM IMPORT ADR;
FROM Intuition IMPORT  CloseScreen,WindowPtr;
FROM SimpleScreens IMPORT CreateScreen;
FROM VariablesGlobales IMPORT cmap,Scr,Str20,Str40;
FROM Views IMPORT ViewPort, LoadRGB4;
FROM Strings IMPORT CompareString,equal, ConvStringToUpperCase,
                    StringLength,CopyString;
FROM SimpleConsole IMPORT PutStr, GetStr;
FROM SimpleConsoleCmds IMPORT wMove;
FROM Trace IMPORT trace;

PROCEDURE EnleveBlancs (VAR S : Str20);
  VAR Indice1,Indice2 : CARDINAL;
      S2 : Str20;
  BEGIN
    Indice1 := 0;
    Indice2 := 0;
    WHILE (Indice1 <= (StringLength (S) - 1)) DO
      IF (S[Indice1] # " ") AND (S[Indice1] # "-") AND (S[Indice1] # "_"
      THEN
        IF (S[Indice1] = "é") OR (S[Indice1] = "è")
        THEN S2[Indice2] := "E";
             Indice2 := Indice2 + 1;
        ELSE
           S2[Indice2] := S[Indice1];
           Indice2 := Indice2 + 1;
        END;
      END (* IF *);
      Indice1 := Indice1 + 1;
    END (* WHILE *);
    S2[Indice2] := S[Indice1];
    CopyString (S,S2);
END EnleveBlancs;

PROCEDURE Reel (VAR St : ARRAY OF CHAR ;  VAR OK : BOOLEAN);
  VAR Indice : CARDINAL;
      Premier : BOOLEAN;
  BEGIN
    Indice := 0;
    OK := TRUE;
    Premier := TRUE;
    IF (CompareString(St,"") = equal) THEN OK := FALSE END;
    IF (CompareString(St,"-") = equal) THEN OK := FALSE END;
    WHILE ((Indice < StringLength (St)) AND OK) DO
      IF ((St[Indice] < "0") OR (St[Indice] > "9"))
         THEN IF (((St[Indice] = ",") OR (St[Indice] = ".")) AND Premier
                 THEN St[Indice] := ".";
                      Indice := Indice + 1;
                      Premier := FALSE;
                 ELSE IF ((St[Indice] = "-") AND
                          (Indice=StringLength (St)-1))
              THEN St[Indice] := " ";Indice := Indice + 1;
              ELSE  OK := FALSE END;
          END;
```

```
        ELSE Indice := Indice + 1 END;
    END;

  END Reel;

PROCEDURE Init;
BEGIN
      Scr := CreateScreen (640,240,4,NIL);
      LoadRGB4 (Scr^.ViewPort,ADR (cmap),16);

END Init;

PROCEDURE Term;
BEGIN
     CloseScreen(Scr^);
END Term;

PROCEDURE GetNom(Win:WindowPtr; VAR NomPHM:Str20);
(*-------------------------------------------*)
BEGIN wMove (Win^,1,12);
      PutStr (Win^,ADR ("Donnez le prénom de l'utilisateur SVP   : "));
      NomPHM := "";
      WHILE (CompareString(NomPHM,"") = equal) DO
        GetStr (Win^,ADR (NomPHM),SIZE(NomPHM));
      END;
      ConvStringToUpperCase (NomPHM);
      EnleveBlancs (NomPHM);

END GetNom;

PROCEDURE PRINT;
VAR Trace:Str40;

BEGIN
     Trace:="Il demande une impression. ";
     trace(Trace);
END PRINT;

END UTILS.
```

IMPLEMENTATION MODULE Help;

```
FROM SYSTEM IMPORT BYTE, ADR;
FROM Tasks IMPORT SignalSet,Wait;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr, Gadget, WindowFlags,
WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,  IntuiMessagePtr, CloseWindow
  NewWindow, Image,  DrawImage, ActivateWindow,IntuiText,IntuiTextPtr;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList, EndGadgetList, FreeGadgetList
  AddGadgetImageButton;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Drawing IMPORT RectFill,SetAPen,Move;
FROM VariablesGlobales IMPORT Str40, Scr,TableauParametres,Commande;
FROM cadre IMPORT FRAME2;
FROM Parole IMPORT SAY;
FROM Trace IMPORT trace;
FROM Couleurs IMPORT gris;
FROM MathLib0 IMPORT entier,real;
FROM Text IMPORT Text;
FROM Strings IMPORT StringLength;
FROM Rasters IMPORT RastPort;
CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};

VAR
  Trace : Str40;
  Win:WindowPtr;
  ok,DoneHelp,Creer,Eff,Recom,Dep,Jeter: BOOLEAN;
  Gl : GadgetPtr;
  Wp : WindowProc;
  Sig : SignalSet;
  Msg : IntuiMessagePtr;
  Ecran:ARRAY[1..12] OF Str40;
  EcranSuivant,EcranCourant,NbrComm,n:INTEGER;

PROCEDURE GadgetHandlerME(VAR w: Window;VAR Msg : MsgData; VAR Gad:
Gadget);

BEGIN

     CASE Gad.GadgetID OF
      0: Trace := "Il clique sur ok ... et quitte l'aide ";
         trace (Trace);
         DoneHelp:=TRUE;|
      1: CASE n OF
             1:SAY(1); |
             2:SAY(2); |
             3:SAY(2); |
             4:SAY(2); |
             5:SAY(3);IF TableauParametres[9]=2 THEN SAY(15);
                                                ELSE SAY(9);END;|
             6:SAY(14);|
             7:SAY(4); |
             8:SAY(5); |
```

```
            9:SAY(5); |
           10:SAY(5); |
           11:SAY(6);SAY(11);SAY(13);SAY(12); |
           12:SAY(3);IF TableauParametres[9]=2 THEN SAY(15);
                                               ELSE SAY(9);END;|
           13:SAY(7); |
           14:SAY(6);SAY(11);SAY(13);SAY(12); |
           15:SAY(3);IF TableauParametres[9]=2 THEN SAY(15);
                                               ELSE SAY(9);END;|
           16:SAY(3);IF TableauParametres[9]=2 THEN SAY(15);
                                               ELSE SAY(9);END;|
           17:SAY(16); |
         END; |
  END;
END GadgetHandlerME;

PROCEDURE affiche(x,y:INTEGER; QQCH:ARRAY OF CHAR);
BEGIN
    Win^.RPort^.FgPen:=BYTE(1);
    Win^.RPort^.BgPen:=BYTE(0);
    SetAPen(Win^.RPort^,1);
    Move(Win^.RPort^,x,y);
    Text(Win^.RPort^,ADR(QQCH),StringLength(QQCH));
END affiche;

PROCEDURE TITRE;
VAR lc,ent:INTEGER;
    tab:REAL;
BEGIN
    Win^.RPort^.FgPen:=BYTE(0);
    Win^.RPort^.BgPen:=BYTE(gris);
    SetAPen(Win^.RPort^,gris);
    RectFill(Win^.RPort^,1,1,318,30);
    SetAPen(Win^.RPort^,0);
    lc:=StringLength(Ecran[EcranCourant]);
    tab:=20.-(real(lc) / 2.);
    ent:=entier(tab*8.);
    Move(Win^.RPort^,ent,15);
    Text(Win^.RPort^,ADR(Ecran[EcranCourant]),
                                    StringLength(Ecran[EcranCourant]));
END TITRE;

(*************************************************************************)
(*                                  SOS                                  *)
(*************************************************************************)

PROCEDURE SOS (N:INTEGER);

BEGIN
   WITH Wp DO
      procGadgetUp := GadgetHandlerME;
   END;
   n:=N;
   Ecran[1]:="Saisie du Prénom";
   Ecran[2]:="Saisie des Recettes";
   Ecran[3]:="Prévisions de Dépenses";
   Ecran[4]:="Menu Principal";
   Ecran[5]:="Choix d'un poste";
```

```
Ecran[6]:="Saisie du montant d'une prévision";
Ecran[7]:="Périodicité d'une prévision";
Ecran[8]:="Fréquence d'une prévision";
Ecran[9]:="Visualisation des prévisions";
Ecran[10]:="Fin du Programme";
Ecran[11]:="Saisie du moment d'une dépense";
Ecran[12]:="Saisie du moment d'une recette";
Eff:=FALSE;Recom:=FALSE;Creer:=FALSE;Jeter:=FALSE;Dep:=FALSE;
CASE N OF
  1: EcranCourant:=1; EcranSuivant:=2; Eff:=TRUE; Recom:=TRUE; |
  2: EcranCourant:=2; EcranSuivant:=3;            Recom:=TRUE; |
  3: EcranCourant:=2; EcranSuivant:=3; Eff:=TRUE; Recom:=TRUE; |
  4: EcranCourant:=2; EcranSuivant:=3; Eff:=TRUE; Recom:=TRUE; |
  5: EcranCourant:=3; EcranSuivant:=4; Creer:=TRUE; |
  6: EcranCourant:=4; EcranSuivant:=10; |
  7: EcranCourant:=5; EcranSuivant:=6;  |
  8: EcranCourant:=6; EcranSuivant:=7;  |
  9: EcranCourant:=6; EcranSuivant:=7; Eff:=TRUE; Recom:=TRUE; |
 10: EcranCourant:=6; EcranSuivant:=7; Eff:=TRUE; Recom:=TRUE; |
 11: EcranCourant:=7; EcranSuivant:=3;  |
 12: EcranCourant:=3; EcranSuivant:=4;Creer:=TRUE;Jeter:=TRUE;
                                                  Dep:=TRUE; |
 13: EcranCourant:=8; EcranSuivant:=3; |
 14: EcranCourant:=7; EcranSuivant:=8; |
 15: EcranCourant:=3; EcranSuivant:=3; Creer:=TRUE; |
 16: EcranCourant:=3; EcranSuivant:=3; Creer:=TRUE;Jeter:=TRUE;
                                                    Dep:=TRUE;|
 17: EcranCourant:=9; EcranSuivant:=3;  |
 18: EcranCourant:=11;EcranSuivant:=4; |
 19: EcranCourant:=12;EcranSuivant:=4; |
END;
BeginGadgetList();
  AddGadgetImageButton (160,215,Commande[2]);
  IF TableauParametres[1]=2
     THEN AddGadgetImageButton(5,40,Commande[4]);
     ELSE AddGadgetImageButton(1,1,Commande[9]);
 END;
        Gl := EndGadgetList();
     IF (Gl # NIL) THEN
    Win := CreateWindow (0,0,320,256,WIDCMP,WFlags,Gl,Scr,NIL);
      IF (Win # NIL) THEN
          NbrComm:=0;
          FRAME2(Win);
          TITRE;
          DrawImage (Win^.RPort^,Commande[2],160,215);
          IF TableauParametres[1]=2 THEN
                     INC(NbrComm);
                     affiche(37,16+40*NbrComm,"Répète le titre");
                     affiche(37,30+40*NbrComm,"de l'écran.");
                     END;
          IF Eff THEN INC(NbrComm);
                     DrawImage (Win^.RPort^,Commande[0],
                                5,40*NbrComm);
                     affiche(37,16+40*NbrComm,"Efface le der-");
                     affiche(37,30+40*NbrComm,"nier caractère.");
          END;
          IF Recom THEN INC(NbrComm);
                        DrawImage (Win^.RPort^,Commande[1],
```

```
                                    5,40*NbrComm);
                        affiche(37,16+40*NbrComm,"Recommence la");
                        affiche(37,30+40*NbrComm,"saisie.");
            END;
            IF Creer THEN INC(NbrComm);
                        DrawImage (Win^.RPort^,Commande[6],
                                    5,40*NbrComm);
                        affiche(37,16+40*NbrComm,"Rajoute une");
                        affiche(37,30+40*NbrComm,"Prévision.");
            END;
            IF Dep THEN INC(NbrComm);
                        DrawImage (Win^.RPort^,Commande[8],
                                    5,40*NbrComm);
                        affiche(37,16+40*NbrComm,"Effectue la");
                        affiche(37,30+40*NbrComm,"Prévision.");
            END;
            IF Jeter THEN DrawImage(Win^.RPort^,Commande[7],160,40);
                        affiche(192,56,"Annule une");
                        affiche(192,70,"Prévision");
            END;
            IF (TableauParametres[0]=1) AND (N#1) THEN
            IF (NbrComm>2)
               THEN IF Jeter
                        THEN DrawImage(Win^.RPort^,Commande[11],160,80);
                             affiche(192,103,"Imprime l'écran");
                        ELSE DrawImage(Win^.RPort^,Commande[11],160,40);
                             affiche(192,63,"Imprime l'écran");
                    END;
               ELSE DrawImage (Win^.RPort^,Commande[11],
                                5,40*(1+NbrComm));
                    affiche(37,23+40*(NbrComm+1),"Imprime l'écran");
            END;
            END;
            IF N#6 THEN DrawImage (Win^.RPort^,Commande[2],5,160);
                        affiche(37,176,"Passe à l'écran:");
                        affiche(37,190,Ecran[EcranSuivant]);
                   ELSE DrawImage (Win^.RPort^,Commande[3],5,160);
                        affiche(37,183,Ecran[EcranSuivant]);
            END;
            SetAPen(Win^.RPort^,0);
            DoneHelp:=FALSE;
            ok:=FALSE;
            WHILE (NOT DoneHelp) DO
               Sig:=Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                  Msg := GetMsg(Win^.UserPort^);
                  IF (Msg = NIL) THEN EXIT; END;
                  ProcIMsg (Wp, Msg);
               END;
            END;
            CloseWindow (Win^);
         END;
         FreeGadgetList (Gl^);
      END;
   END SOS;
END Help.
```

IMPLEMENTATION MODULE IO;

```
FROM InOut IMPORT
OpenInputFile,CloseInput,OpenOutputFile,CloseOutput,Done,
                  ReadString,WriteString,ReadInt,WriteInt,WriteLn;
FROM FileSystem IMPORT Lookup,File,ReadChar,Close;
FROM Strings IMPORT ConcatString,CopyString;
FROM VariablesGlobales IMPORT Max,Str40,Liquide,Scr,TableauParametres,
                              Phrases, Nom,Str100,Cash;
FROM Intuition IMPORT ScreenToFront;
FROM RealInOut IMPORT ReadReal, WriteReal;

VAR f:File;
    FileName:Str40;
    i:INTEGER;

PROCEDURE LIRENOMS_POSTES(VAR nom:TABPOSTE);
(*---------------------------------------*)
BEGIN
     OpenInputFile("POSTES");
     FOR i:=0 TO 39 DO
         ReadString(nom[i]);
     END;
     CloseInput;
END LIRENOMS_POSTES;

PROCEDURE ReadLn (VAR P:Str100);
(*----------------------------*)

VAR ch:CHAR;
    FIN:BOOLEAN;
    i:INTEGER;

BEGIN
     FIN:=FALSE;
     i:=0;
     CopyString(P,"");
     WHILE ((ch # 12C) AND (NOT f.eof)) DO
           ReadChar(f,ch);
           IF (ch # 12C) THEN ConcatString(P,ch); END;
           INC(i);
     END;
END ReadLn;

PROCEDURE LirePhrases;
(*------------------*)

VAR i,j,l : INTEGER;
    TOTO:Str100;

   BEGIN
     i:=1;
     Lookup(f,"Phrases",FALSE);
     WHILE (NOT f.eof) DO
           CopyString(Phrases[i],"");
           ReadLn(TOTO);
```

```
            CopyString(Phrases[i],TOTO);
            i:=i+1;
      END;
      Close(f);
END LirePhrases;

PROCEDURE LoadMax;
(*--------------*)
BEGIN
        FileName := "Param:";
        ConcatString (FileName,Nom);
        ConcatString (FileName,".MAX");
        OpenInputFile (FileName);
        ScreenToFront (Scr^);
        IF NOT Done THEN Max := 300.
                    ELSE ReadReal (Max);
        END;
        CloseInput;
END LoadMax;

PROCEDURE LoadLiquide;
(*------------------*)

BEGIN
        FileName := "Param:";
        ConcatString (FileName,Nom);
        ConcatString (FileName,".SOMME");
        OpenInputFile (FileName);
        ScreenToFront (Scr^);
        IF NOT Done THEN Liquide := 0. ;
                    ELSE
                         ReadReal (Liquide);
                         CloseInput;
        END;
        Cash:=Liquide;
END LoadLiquide;

PROCEDURE sauver_Liquide;
BEGIN
        FileName := "Param:";
        ConcatString (FileName,Nom);
        ConcatString (FileName,".SOMME");
        OpenOutputFile (FileName);
        ScreenToFront (Scr^);
        WriteReal (Liquide,8);
        CloseOutput;
END sauver_Liquide;

PROCEDURE InitParametres;
(*---------------------*)

VAR j : INTEGER;
    FileName:Str40;
    OK:BOOLEAN;
    param:LONGINT;

PROCEDURE parametres_par_defaut;
BEGIN
```

```
      TableauParametres[0] := 0;
      TableauParametres[1] := 2;
      TableauParametres[2] := 0;
      TableauParametres[3] := 1;
      TableauParametres[4] := 2;
      TableauParametres[5] := 0;
      TableauParametres[6] := 1;
      TableauParametres[7] := 0;
      TableauParametres[8] := 0;
      TableauParametres[9] := 1;
      TableauParametres[10] := 0;
      TableauParametres[11] := 1;
      TableauParametres[12] := 2;
END parametres_par_defaut;


BEGIN
        FileName := "Param:UTILISATEUR.PARAM";
        OpenInputFile (FileName);
        ScreenToFront (Scr^);
        IF NOT Done THEN parametres_par_defaut;
                    ELSE FOR j:=0 TO 11 DO
                           ReadInt(TableauParametres[j]);
                         END;
                         CloseInput;
        END;
END InitParametres;

PROCEDURE LirePD(VAR NbPD:INTEGER; VAR TabPD:TABPD);
(*-----------------------------------------------*)
VAR i:INTEGER;

BEGIN FileName := "Param:";
      ConcatString (FileName,Nom);
      ConcatString (FileName,".PD");
      OpenInputFile (FileName);
      ScreenToFront (Scr^);
      IF NOT Done THEN NbPD:=0;
                  ELSE ReadInt (NbPD);
                       IF (NbPD > 0) THEN
                         FOR i:=1 TO NbPD DO
                           ReadInt(TabPD[i].NoPoste);
                           ReadReal(TabPD[i].Somme);
                           ReadString(TabPD[i].RO);
                           ReadInt(TabPD[i].MoisDepart);
                           ReadInt(TabPD[i].Frequence);
                           ReadString(TabPD[i].Faite);
                           ReadInt(TabPD[i].Quand);
                           ReadReal(TabPD[i].SommeDprt);
                         END;
                       END;
                       CloseInput;
      END;
END LirePD;
```

```
PROCEDURE EcrirePD(NbPD:INTEGER; VAR TabPD:TABPD);
(*-----------------------------------------------*)
VAR i:INTEGER;

BEGIN FileName := "Param:";
      ConcatString (FileName,Nom);
      ConcatString (FileName,".PD");
      OpenOutputFile (FileName);
      ScreenToFront (Scr^);
      WriteInt(NbPD,2);
      WriteLn;
      IF (NbPD > 0) THEN
         FOR i:=1 TO NbPD DO
             WriteInt(TabPD[i].NoPoste,2);
             WriteLn;
             WriteReal(TabPD[i].Somme,6);
             WriteLn;
             WriteString(TabPD[i].RO);
             WriteLn;
             WriteInt(TabPD[i].MoisDepart,2);
             WriteLn;
             WriteInt(TabPD[i].Frequence,2);
             WriteLn;
             WriteString(TabPD[i].Faite);
             WriteLn;
             WriteInt(TabPD[i].Quand,2);
             WriteLn;
             WriteReal(TabPD[i].SommeDprt,6);
             WriteLn;
         END;
      END;
      CloseOutput;
END EcrirePD;
END IO.
```

★ ★ ★
★ ★

Definition modules

```
DEFINITION MODULE Interface;

FROM VariablesGlobales IMPORT Str20;
FROM Intuition IMPORT Image;

PROCEDURE SSaisieNom;

(*************************************************************)
(*                                                           *)
(* IN : /                                                    *)
(*                                                           *)
(* OUT: /                                                    *)
(*                                                           *)
(* BUT: effectue la saisie du prénom au clavier, si le paramètre 2  *)
(*      est égal à 0 et à l'aide de la souris sinon.         *)
(*                                                           *)
(*************************************************************)

PROCEDURE SaisieSomme(ch:INTEGER; Q:BOOLEAN;VAR Val:REAL; VAR ES:CHAR);

(*************************************************************)
(*                                                           *)
(* IN : ch: type de la saisie 1=recette; 2=dépense; 3=prévision     *)
(*      Q: TRUE-> on peut annuler la saisie; FALSE-> on ne peut pas *)
(*                                                           *)
(* OUT: Val: montant saisi.                                  *)
(*      ES: ="Q" si on a annulé la saisie et "A" sinon.      *)
(*                                                           *)
(* BUT: effectue une saisie de somme au clavier, à la calculette ou *)
(*      à l'aide des billets selon le paramètre 4            *)
(*                                                           *)
(*************************************************************)

PROCEDURE SaisieRP(Q:BOOLEAN;VAR Val:REAL; VAR ES:CHAR);

(*************************************************************)
(*                                                           *)
(* IN :  Q: TRUE-> on peut annuler la saisie; FALSE-> on ne peut pas*)
(*                                                           *)
(* OUT: Val: montant saisi.                                  *)
(*      ES: ="Q" si on a annulé la saisie et "A" sinon.      *)
(*                                                           *)
(* BUT: effectue une saisie des recettes principales au clavier, à  *)
(*      la calculette ou à l'aide des billets selon le paramètre 3  *)
(*                                                           *)
(*************************************************************)

PROCEDURE Menu;

(*************************************************************)
(*                                                           *)
(* IN : /                                                    *)
(*                                                           *)
(* OUT: /                                                    *)
(*                                                           *)
(* BUT: appel la procedure menu, garnit les traces et appel la      *)
(*      procédure correspondant à l'activité choisie.        *)
(*                                                           *)
(*************************************************************)
```

```
PROCEDURE pd;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: lit et écrit les prévisions de dépenses du mois courant et       *)
(*      appel la procedure de gestion des PD.                            *)
(*                                                                       *)
(*************************************************************************)

END Interface.
```

DEFINITION MODULE ModNom;

```
PROCEDURE SaisieNom;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: effectue la saisie du prénom au clavier (écran 1)                *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE Erreur;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: affiche un message d'erreur annonçant que l'utilisateur          *)
(*      s'est trompé de nom. Le message disparaît après 12 secondes      *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE TraceDate;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: met dans le tableaau des traces, la première trace de la         *)
(*      session, à savoir la date (j,m,a) et le n° de session.           *)
(*                                                                       *)
(*************************************************************************)

END ModNom.
```

```
DEFINITION MODULE ModNomSou;

PROCEDURE SaisieNomSou;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT:  effectue la saisie du prénom à la souris (écran 2)              *)
(*                                                                       *)
(*************************************************************************)


END ModNomSou.
```

```
DEFINITION MODULE SSNombre;

FROM Intuition IMPORT Image;

PROCEDURE SaisieSommeNbre (ch:INTEGER; Quit : BOOLEAN ; VAR Valeur : REAL ;
                           VAR EtatSortie : CHAR;VAR im:Image);

(*************************************************************************)
(*                                                                       *)
(* IN : ch: type de la saisie 1=recette; 2=dépense; 3=prévision          *)
(*                            4=recettes principales                     *)
(*      Quit: TRUE->on peut annuler la saisie; FALSE->on ne peut pas     *)
(*      im: icône éventuelle du poste dont on saisit le montant.         *)
(*          = NIL si on ne saisit pas le montant d'un poste              *)
(*                                                                       *)
(* OUT: Valeur: montant saisi.                                           *)
(*      EtatSortie: ="Q" si on a annulé la saisie et "A" sinon.          *)
(*                                                                       *)
(* BUT: effectue une saisie de somme à l'aide du clavier (écran 8a)      *)
(*                                                                       *)
(*************************************************************************)

END SSNombre.
```

```
DEFINITION MODULE Scalcu;

FROM Intuition IMPORT Image;

PROCEDURE SaisieSommeCalcu (ch:INTEGER;Quit:BOOLEAN;VAR Valeur:REAL;
                            VAR EtatSortie:CHAR;VAR im:Image);

(*************************************************************************)
(*                                                                       *)
(* IN : ch: type de la saisie 1=recette; 2=dépense; 3=prévision          *)
(*                            4=recettes principales                     *)
(*      Quit: TRUE->on peut annuler la saisie; FALSE->on ne peut pas     *)
(*      im: icône éventuelle du poste dont on saisit le montant.         *)
(*          = NIL si on ne saisit pas le montant d'un poste              *)
(*                                                                       *)
(* OUT: Valeur: montant saisi.                                           *)
(*      EtatSortie: ="Q" si on a annulé la saisie et "A" sinon.          *)
(*                                                                       *)
(* BUT: effectue une saisie de somme à l'aide  de la calculette          *)
(*      (écran 8b)                                                       *)
(*                                                                       *)
(*************************************************************************)

END Scalcu.

DEFINITION MODULE SSBillets;

FROM Intuition IMPORT Image;

PROCEDURE SaisieSommeBillets (ch:INTEGER;Quit:BOOLEAN;VAR Valeur:REAL;
                              VAR EtatSortie : CHAR;VAR im:Image);

(*************************************************************************)
(*                                                                       *)
(* IN : ch: type de la saisie 1=recette; 2=dépense; 3=prévision          *)
(*                            4=recettes principales                     *)
(*      Quit: TRUE->on peut annuler la saisie; FALSE->on ne peut pas     *)
(*      im: icône éventuelle du poste dont on saisit le montant.         *)
(*          = NIL si on ne saisit pas le montant d'un poste              *)
(*                                                                       *)
(* OUT: Valeur: montant saisi.                                           *)
(*      EtatSortie: ="Q" si on a annulé la saisie et "A" sinon.          *)
(*                                                                       *)
(* BUT: effectue une saisie de somme à l'aide des billets (écran 8c)     *)
(*                                                                       *)
(*************************************************************************)


END SSBillets.
```

```
DEFINITION MODULE ModPD;

PROCEDURE PD(FR:INTEGER;VAR ES:CHAR);

(*************************************************************************)
(*                                                                       *)
(* IN : FR:Fréquence des prévissions: 0 = toutes les semaines            *)
(*                                    1 = tous les 2 mois                *)
(*                                    2 = tous les 6 mois                *)
(*                                    3 = tous mois                      *)
(*                                    4 = tous les 3 mois                *)
(*                                    5 = tous les ans                   *)
(*                                    6 = du mois (semaine) actuel       *)
(*                                                                       *)
(* OUT: ES="Q" quand on a fini les prévisions                            *)
(*                                                                       *)
(* BUT: Gère les prévisions de dépenses (éran 6 et 6')                   *)
(*                                                                       *)
(*************************************************************************)


END ModPD.

DEFINITION MODULE ModMenu;

FROM VariablesGlobales IMPORT Str20;

PROCEDURE Menu (VAR FonctionChoisie : Str20);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: FonctionChoisie est égal à une des 8 activités: recettes,        *)
(*      depenses, PR, PD, économies, test, bilan, etat_finances          *)
(*                                                                       *)
(* BUT: permet de choisir l'activité voulue (écran 13)                   *)
(*                                                                       *)
(*************************************************************************)


END ModMenu.

DEFINITION MODULE ModMois;

PROCEDURE QUAND(VAR JJ,MM,AA:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : AA=0 s'il s'agit d'une dépense et 1 d'une recette                *)
(*                                                                       *)
(* OUT: JJ:jour où s'est faite la dépense ou recette (1..31)             *)
(*      MM:mois où s'est faite la dépense ou recette (0..11)             *)
(*                                                                       *)
(* BUT: saisit le moment où s'est faite une R ou D (écran 14)            *)
(*                                                                       *)
(*************************************************************************)
END ModMois.
```

```
DEFINITION MODULE Jauge;

FROM Intuition IMPORT WindowPtr;

PROCEDURE InitJauge(x,y:INTEGER;win:WindowPtr);

(*****************************************************************)
(*                                                               *)
(* IN : x,y:coordonnées du coin supérieur gauche de la jauge     *)
(*      win: Fenêtre dans laquelle s'affiche la jauge            *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: affiche une jauge vide dans la fenetre win en (x,y)      *)
(*                                                               *)
(*****************************************************************)

PROCEDURE  MAJ_Jauge(x,y:INTEGER;win:WindowPtr;sens:CARDINAL;somme1:REAL;
                     c1:CARDINAL;somme2:REAL;c2:CARDINAL);

(*****************************************************************)
(*                                                               *)
(* IN : x,y:coordonnées du coin supérieur gauche de la jauge     *)
(*      win: Fenêtre dans laquelle s'affiche la jauge            *)
(*      sens: 0->recette 1-> dépense 2 -> recettes principales   *)
(*      c1,c2: couleurs des montants  (0..31)                    *)
(*      somme1,somme2: montants à mettre dans la jauge           *)
(*                                                               *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: remplit la jauge avec somme1 de couleur c1 et somme2 de  *)
(*      couleur c2 d'aprés le sens                               *)
(*                                                               *)
(*****************************************************************)

PROCEDURE InitSomme(win:WindowPtr;FOIS:INTEGER);

(*****************************************************************)
(*                                                               *)
(* IN : win: fenêtre destinée à recevoir les rectangles          *)
(*      FOIS:Nombre de rectangles à afficher                     *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: dessine FOIS rectangles destinés à recevoir les recettes *)
(*      principales (écran 3 et 4)                               *)
(*                                                               *)
(*****************************************************************)
```

```
PROCEDURE MAJ_Somme(win:WindowPtr;I:INTEGER;val:REAL);

(*****************************************************************)
(*                                                               *)
(* IN : win: fenetre sur laquelle on travaille                   *)
(*      I: numéro du rectangle dans lequel la somme va s'afficher*)
(*      val: somme qui va s'afficher dans le Ième rectangle      *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: affiche le nombre val dans la fenêtre win (écran 3 et 4) *)
(*                                                               *)
(*****************************************************************)

PROCEDURE cdr(Win:WindowPtr;x,y,N:INTEGER; ON:BOOLEAN);

(*****************************************************************)
(*                                                               *)
(* IN : Win:fenêtre dans laquelle on travaille                   *)
(*      x,y: coordonnées du coin supérieur gauche du cadre       *)
(*      N: type de cadre; 1=PD; 2=RO; 3=Freq; 4=Postes;          *)
(*      ON: =TRUE -> efface le cadre; FALSE -> affiche le cadre  *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: place un cadre autour d'une icône quand celle-ci a été   *)
(*      sélectionnée.                                            *)
(*                                                               *)
(*****************************************************************)


VAR JaugeX,JaugeY:INTEGER;

END Jauge.
```

DEFINITION MODULE AuxPD;

```
FROM Intuition IMPORT WindowPtr,Image;
FROM IO IMPORT TPD;

TYPE TABJ=ARRAY[0..6] OF INTEGER;

PROCEDURE Somme2(VAR W:WindowPtr;x,y:INTEGER;s:REAL);

(*****************************************************************)
(*                                                               *)
(* IN : Win:fenêtre de travail                                   *)
(*      x,y: coordonnées du coin supérieur gauche du poste.      *)
(*      s: nombre à afficher à la droite du poste.               *)
(*                                                               *)
(* OUT: /                                                        *)
(*                                                               *)
(* BUT: afficher le montant d'une prévision de dépenses à sa droite *)
(*      (cf écran 6)                                             *)
(*                                                               *)
(*****************************************************************)
```

```
PROCEDURE CONV(s:REAL; VAR l:INTEGER; VAR j:TABJ);

(*************************************************************)
(*                                                           *)
(* IN : s:nombre à convertir                                 *)
(*                                                           *)
(* OUT: l:nombre de chiffres de s                            *)
(*      j:contient la représentation de s sous forme d'images*)
(*                                                           *)
(* BUT: transforme un nombre en images (une image par chiffre).*)
(*                                                           *)
(*************************************************************)

PROCEDURE LigneTemps(VAR W:WindowPtr);

(*************************************************************)
(*                                                           *)
(* IN : W:fenetre destinnée à recevoir la ligne du temps     *)
(*                                                           *)
(* OUT: /                                                    *)
(*                                                           *)
(* BUT: trace une ligne du temps (parametre 6) de la semaine ou du *)
(*      mois (parametre 9) (cf écran 6).                     *)
(*                                                           *)
(*************************************************************)

PROCEDURE ENLEVERPD(N:INTEGER);

(*************************************************************)
(*                                                           *)
(* IN : N: numéro de la PD à enlever.                        *)
(*                                                           *)
(* OUT: /                                                    *)
(*                                                           *)
(* BUT: enlève une PD du tableau des PD.                     *)
(*                                                           *)
(*************************************************************)

PROCEDURE CONF(Im:Image;s:REAL;VAR Reponse:BOOLEAN);

(*************************************************************)
(*                                                           *)
(* IN : Im: icône de la PD.                                  *)
(*      s: montant de la PD.                                 *)
(*                                                           *)
(* OUT: Reponse=TRUE si on a répondu oui; FALSE sinon.       *)
(*                                                           *)
(* BUT: affiche une confirmation (CONF 1) qui demande si le montant *)
(*      dépensé d'une PD est le même que le montant prévu.   *)
(*                                                           *)
(*************************************************************)
```

```
PROCEDURE DeterminerPD(N:INTEGER;VAR PDFreq:TPD;VAR Nb:INTEGER);

(*****************************************************************)
(*                                                               *)
(* IN : N: n° de la fréquence (0..6)                             *)
(*                                                               *)
(* OUT: PDFreq:Tableau des PD dont la frequence est N            *)
(*      Nb: nombre de PD dont la fréquence est N                 *)
(*                                                               *)
(* BUT: extrait de TABPD la liste des PD dont la frequence est N *)
(*      et la range dans PDFreq. Si N=6 (toutes les PD échéant ce*)
(*      mois) PDFreq contient: les PD occasionnelles, toutes les *)
(*      semaines, tous les mois, et les autres si elles tombent ce*)
(*      mois.                                                    *)
(*                                                               *)
(*****************************************************************)

PROCEDURE VERIF(I,J:INTEGER; VAR OK:BOOLEAN);

(*****************************************************************)
(*                                                               *)
(* IN : I: mois à partir duquel la PD à été faite.               *)
(*      J: Fréquence de la PD. (2,3,4,6)                         *)
(*                                                               *)
(* OUT: OK=TRUE si tous les J mois, à partir du mois I tombe ce mois*)
(*         FALSE sinon.                                          *)
(*                                                               *)
(* BUT: vérifie si une PD avec une fréquence quelconque se fait ce*)
(*      mois.                                                    *)
(*                                                               *)
(*****************************************************************)

END AuxPD.
```

DEFINITION MODULE ModPoste;

```
FROM Intuition IMPORT Image;

PROCEDURE Poste (VAR poste:Image;VAR No:INTEGER);

(*****************************************************************)
(*                                                               *)
(* IN : /                                                        *)
(*                                                               *)
(* OUT: poste: icône sélectionnée                                *)
(*      No: numéro du poste sélectionné.                         *)
(*                                                               *)
(* BUT: affiche les différents postes (40) pour qu'on en sélectionne*)
(*      (écran 7).                                               *)
(*                                                               *)
(*****************************************************************)


END ModPoste.
```

```
DEFINITION MODULE ModFreq;

FROM Intuition IMPORT Image;

PROCEDURE Frequence(VAR freq,start: INTEGER;poste:Image);

(*************************************************************)
(*                                                           *)
(* IN : poste: icône de la PD.                               *)
(*                                                           *)
(* OUT: freq: fréquence de la PD (0..5)                      *)
(*      start: mois à partir duquel se fera la PD (0..11)    *)
(*                                                           *)
(* BUT: saisie de la fréquence et du mois de départ d'une PD.*)
(*      (écran 10)                                           *)
(*                                                           *)
(*************************************************************)

END ModFreq.
```

```
DEFINITION MODULE ModSRO;

FROM Intuition IMPORT Image;

PROCEDURE SRO (poste:Image;VAR ch:CHAR);

(*************************************************************)
(*                                                           *)
(* IN : poste:icône de la PD.                                *)
(*                                                           *)
(* OUT: ch: "O" si la PD est occasionnelle et "R" si régulière*)
(*                                                           *)
(* BUT: saisie de la périodicité d'une PD. (écran 9)         *)
(*                                                           *)
(*************************************************************)


END ModSRO.
```

```
DEFINITION MODULE Visu;

FROM IO IMPORT TPD;

PROCEDURE Visualisation(N:INTEGER;VAR Tab:TPD;Nb:INTEGER);

(*************************************************************)
(*                                                           *)
(* IN : N:fréquence des PD à visualiser.                     *)
(*      Tab: Tableau contenant les PD.                       *)
(*      Nb: nombre de PD à visualiser.                       *)
(*                                                           *)
(* OUT: /                                                    *)
(*                                                           *)
(* BUT: visualise les PD (selon leur fréquence) (écran 11)   *)
(*                                                           *)
(*************************************************************)


END Visu.
```

```
DEFINITION MODULE Texte;

FROM Intuition IMPORT WindowPtr;

PROCEDURE ECRIT(FEN:WindowPtr;X:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : FEN: Fenêtre dans laquelle on travaille                          *)
(*      X: n° de la phrase à afficher. (1..16)                           *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: affiche le titre de l'écran. (voir tous les écrans)              *)
(*                                                                       *)
(*************************************************************************)

END Texte.

DEFINITION MODULE Trace;

FROM VariablesGlobales IMPORT Str40;

PROCEDURE trace( T : Str40 );

(*************************************************************************)
(*                                                                       *)
(* IN : T: action effectuée par  l'utilisateur.                          *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: range dans le tableau de traces la dernière trace d'utilis.      *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE sauver_traces;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: enregistre sur disque le tableau des traces d'utilisation        *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE TraceParametres;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: met dans le tableau des traces les paramètres choisis            *)
(*                                                                       *)
(*************************************************************************)
END Trace.
```

```
DEFINITION MODULE Parole;

FROM IO IMPORT Str15;

PROCEDURE SAY(X:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : X:n° de la phrase à prononcer (1..19)                            *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: énonce oralement un des titres                                   *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE SayPoste(nom:Str15);

(*************************************************************************)
(*                                                                       *)
(* IN : nom: nom du poste à prononcer.                                   *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: énonce oralement le poste nom                                    *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE SayMois(x:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : x: numéro du mois à prononcer (1..12)                            *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: énonce oralement le(x-1)ème mois.                                *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE SayFreq(x:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : x: numéro de la fréquence à prononcer(0..5)                      *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: énonce oralement la xème fréquence                               *)
(*                                                                       *)
(*************************************************************************)


END Parole.
```

```
DEFINITION MODULE LireImage;

FROM Intuition IMPORT Image;

TYPE TP=ARRAY [0..40] OF Image;
     TabMini=ARRAY[0..9] OF Image;
     TabMois=ARRAY[0..11] OF Image;
     TabCalcu= ARRAY [0..12] OF Image;
     TabLettre=ARRAY [0..29] OF Image;
     TabMonnaie=ARRAY [0..12] OF Image;
     TabFreq=ARRAY [0..5] OF Image;
     TabRO=ARRAY [0..1] OF Image;
VAR  POSTE: TP;
     Mini:TabMini;
PROCEDURE LoadPostes(VAR POSTE:TP);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: POSTE: tableau d'images contenant les 40 postes                  *)
(*                                                                       *)
(* BUT: lit les icônes des postes                                        *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadMini(VAR Mini:TabMini);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: Mini: tableau d'images contenant les 10 chiffres                 *)
(*                                                                       *)
(* BUT: lit les icônes des 10 chiffres                                   *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LireMois(VAR Mois: TabMois);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: Mois: tableau d'images contenant les 12 mois                     *)
(*                                                                       *)
(* BUT: lit les icônes des mois                                          *)
(*                                                                       *)
(*************************************************************************)
```

```
PROCEDURE LoadCalcu(VAR Chiffre:TabCalcu);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: Chiffre: tableau d'images contenant les 12 chiffres              *)
(*                                                                       *)
(* BUT: lit les icônes des chiffres de la calculette                     *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadLettre(VAR TabImages:TabLettre);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: TabImages: tableau d'images contenant les 30 lettres             *)
(*                                                                       *)
(* BUT: lit les icônes des lettres                                       *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadMonnaie(VAR Monnaie:TabMonnaie);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: Monnaie: tableau d'images contenant les 12 billets               *)
(*                                                                       *)
(* BUT: lit les icônes des billets                                       *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadFreq(VAR freq:TabFreq);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: freq: tableau d'images contenant les 6 fréquences                *)
(*                                                                       *)
(* BUT: lit les icônes des fréquences                                    *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadRO(VAR image:TabRO);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: image:  tableau d'images contenant les 2 icônes "R" "O"          *)
(*                                                                       *)
(* BUT: lit les icônes régulière et occasionnelle                        *)
(*                                                                       *)
(*************************************************************************)
END LireImage.
```

```
DEFINITION MODULE annee;

FROM VariablesGlobales IMPORT ST10;

TYPE TABNJ=ARRAY[0..11] OF INTEGER;

PROCEDURE DATE (VAR JJ,MM,AA:INTEGER; VAR Bi: BOOLEAN);

(*****************************************************************)
(*                                                               *)
(* IN : /                                                        *)
(*                                                               *)
(* OUT: JJ: jour actuel (1..31)                                  *)
(*      MM: mois actuel (0..11)                                  *)
(*      AA: année actuelle (4 chiffres)                          *)
(*      Bi: TRUE si année bisextile; FALSE sinon                 *)
(*                                                               *)
(* BUT: renvoie la date actuelle                                 *)
(*                                                               *)
(*****************************************************************)

PROCEDURE NbJours(VAR NJOURS:TABNJ);

(*****************************************************************)
(*                                                               *)
(* IN : /                                                        *)
(*                                                               *)
(* OUT: NJOURS: contient le nombre de jours des 12 mois          *)
(*                                                               *)
(* BUT: calcule le nombre de jours des différents mois.          *)
(*                                                               *)
(*****************************************************************)

PROCEDURE NOMMOIS (VAR MOIS:ST10; I:INTEGER);

(*****************************************************************)
(*                                                               *)
(* IN : I: numéro du mois (0..11)                                *)
(*                                                               *)
(* OUT: MOIS: nom du Ième mois                                   *)
(*                                                               *)
(* BUT: à partir d'un chiffre renvoie le nom du mois             *)
(*                                                               *)
(*****************************************************************)

PROCEDURE RechJour(j,m,a:INTEGER;VAR n:INTEGER);

(*****************************************************************)
(*                                                               *)
(* IN : j:jour à rechercher (1..31)                              *)
(*      m:mois à rechercher (0..11)                              *)
(*      a:année à rechercher (4 chiffres)                        *)
(*                                                               *)
(* OUT: n: jour de la semaine (1..7)                             *)
(*                                                               *)
(* BUT: à partir d'une date, renvoie le jour de la semaine       *)
(*                                                               *)
(*****************************************************************)
END annee.
```

DEFINITION MODULE Cadre;

```
FROM Intuition IMPORT WindowPtr;

PROCEDURE FRAME (win:WindowPtr);

(*************************************************************************)
(*                                                                       *)
(* IN : Win: fenetre de travail                                          *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: dessine un des deux types d'écrans (voir chaque écran)           *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE FRAME2(win:WindowPtr);

(*************************************************************************)
(*                                                                       *)
(* IN : win: fenetre de travail                                          *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: dessine un des deux types d'écrans (voir chaque écrans)          *)
(*                                                                       *)
(*************************************************************************)


END Cadre.
```

DEFINITION MODULE couleurs;

```
VAR
   noir,blanc,rouge,jaune,vert,bleu,gris,orange,bleuclair:INTEGER;

END couleurs.
```

DEFINITION MODULE VariablesGlobales;

```
FROM Intuition IMPORT Image,ScreenPtr,WindowPtr;
FROM SoundLib IMPORT ASampPtr;

TYPE Str3 = ARRAY [0..2] OF CHAR;
TYPE Str9 = ARRAY [0..8] OF CHAR;
TYPE ST10 = ARRAY [0..9] OF CHAR;
TYPE Str20 = ARRAY [0..19] OF CHAR;
TYPE Str40 = ARRAY [0..39] OF CHAR;
TYPE Str50 = ARRAY [0..49] OF CHAR;
TYPE Str60 = ARRAY [0..59] OF CHAR;
TYPE Str100 = ARRAY [0..99] OF CHAR;

VAR TableauParametres : ARRAY [0..19] OF INTEGER;
    TableauTraces : RECORD
                      Tab : ARRAY [0..499] OF Str40;
                      Indice : INTEGER;
                    END;
    Liquide,Cash, Max : REAL;
    Nom:Str20;
    Scr:ScreenPtr;
```

```
    Win:WindowPtr;
    cmap : ARRAY [0..31] OF CARDINAL;
    nomuser,Numseance : Str20;
    Phrases: ARRAY [1..14] OF Str60;
    Icone : ARRAY [0..3] OF Image;
    Commande : ARRAY[0..12] OF Image;
    Son:ARRAY[1..17] OF ASampPtr;
END VariablesGlobales.
```

DEFINITION MODULE UTILS;

```
FROM VariablesGlobales IMPORT Str20;
FROM Intuition IMPORT WindowPtr;

PROCEDURE Init;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: Initialise une fenetre pour les logiciels paramètres et          *)
(*      traces.                                                          *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE Term;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: ferme une fenetre (log. parametres et traces)                    *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE GetNom(win:WindowPtr; VAR NN:Str20);

(*************************************************************************)
(*                                                                       *)
(* IN : win: fenetre de travail                                          *)
(*                                                                       *)
(* OUT: NN:nom saisi au clavier                                          *)
(*                                                                       *)
(* BUT: saisit un nom tapé au clavier                                    *)
(*                                                                       *)
(*************************************************************************)
```

```
PROCEDURE PRINT;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: Imprime l'écran en cours (hardcopie)                             *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE Reel (VAR St : ARRAY OF CHAR ;  VAR OK : BOOLEAN);

(*************************************************************************)
(*                                                                       *)
(* IN : St: chaine de  caractères devant représenter un réel             *)
(*                                                                       *)
(* OUT: OK: TRUE si St représente un réel et FALSE sinon                 *)
(*                                                                       *)
(* BUT: vérifie si une chaine de caractères représente un réel           *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE EnleveBlancs (VAR s:Str20);

(*************************************************************************)
(*                                                                       *)
(* IN : s: chaine  de caracteres avec, éventuellement des blancs         *)
(*                                                                       *)
(* OUT: s: chaine de caractères sans blancs                              *)
(*                                                                       *)
(* BUT: enlève des blancs, éventuellement présents dans une chaine       *)
(*      de caractères.                                                   *)
(*                                                                       *)
(*************************************************************************)

END UTILS.
```

DEFINITION MODULE Help;

```
PROCEDURE SOS(s:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : s: numéro de l'écran courant                                     *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: explique les commandes de l'écran courant (écran 12)             *)
(*                                                                       *)
(*************************************************************************)

END Help.
```

```
DEFINITION MODULE IO;

TYPE Str15=ARRAY[0..15] OF CHAR;
     TABPOSTE=ARRAY[0..39] OF Str15;
     TABPD=ARRAY[1..50] OF RECORD
                                NoPoste:INTEGER;
                                Somme   :REAL;
                                RO      :CHAR;
                                MoisDepart:INTEGER;
                                Frequence :INTEGER;
                                Faite   :CHAR;
                                Quand   :INTEGER;
                                SommeDprt :REAL;
                           END;
    TPD=ARRAY[0..6] OF TABPD;
VAR NbPD: INTEGER;
    TabPD:TABPD;

PROCEDURE LIRENOMS_POSTES(VAR nom:TABPOSTE);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: nom: noms des 40 postes                                          *)
(*                                                                       *)
(* BUT: lit sur le disque les noms des 40 postes                         *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LirePhrases;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: lit sur disque les titres dess écrans                            *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE LoadMax;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: lit sur disque param: l'argent maximum que l'utilisateur         *)
(*      peut avoir.                                                      *)
(*                                                                       *)
(*************************************************************************)
```

```
PROCEDURE LoadLiquide;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: lit sur le dique param: le solde de l'utilisateur de la          *)
(*      fois passée.                                                     *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE sauver_Liquide;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: enregistre sur param: l'argent disponible.                       *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE InitParametres;

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: lit sur le disque param: les différents paramètres.              *)
(*                                                                       *)
(*************************************************************************)

PROCEDURE EcrirePD(NbPD:INTEGER; VAR TabPD:TABPD);

(*************************************************************************)
(*                                                                       *)
(* IN : NbPD: nombre de prévisions de dépenses.                          *)
(*      TabPD: tableau des PD.                                           *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: enregistre sur param: les PD.                                    *)
(*                                                                       *)
(*************************************************************************)
```

```
PROCEDURE LirePD(VAR NbPD:INTEGER; VAR TabPD:TABPD);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: NbPD:nombre de PD.                                               *)
(*      TabPD:tableau des PD.                                            *)
(*                                                                       *)
(* BUT: lit sur param: les PD.                                           *)
(*                                                                       *)
(*************************************************************************)

END IO.
```

DEFINITION MODULE ModTest;

```
PROCEDURE TEST(VAR CH:CHAR);

(*************************************************************************)
(*                                                                       *)
(* IN : /                                                                *)
(*                                                                       *)
(* OUT: CH:="A" si l'utilisateur choisit le test A; "B" sinon            *)
(*                                                                       *)
(* BUT: Permet à l'utilisateur de choisir son test.                      *)
(*                                                                       *)
(*************************************************************************)

END ModTest.
```

DEFINITION MODULE TestA;

```
PROCEDURE TestA(Val,ValeurParMois:REAL);

(*************************************************************************)
(*                                                                       *)
(* IN : Val: montant de la dépense à simuler.                            *)
(*      ValeurParMois: montant que l'utilisateur épargne par mois.       *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: Visualise le nombre de fois qu'il faudra épargner un montant*)
(*      pour atteindre un autre.                                         *)
(*                                                                       *)
(*************************************************************************)

END TestA.
```

```
DEFINITION MODULE TestB;

PROCEDURE TestB(Val:REAL;N:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN : Val: montant de la dépense à simuler.                            *)
(*      N: Nombre de mois avant que l'utilisateur fasse sa dépense.      *)
(*                                                                       *)
(* OUT: /                                                                *)
(*                                                                       *)
(* BUT: Visualise le montant à épargner par mois avant de faire une      *)
(*      dépense.                                                         *)
(*                                                                       *)
(*************************************************************************)

END TestB.
```

```
DEFINITION MODULE SaisieMois;

PROCEDURE SMois(VAR N:INTEGER);

(*************************************************************************)
(*                                                                       *)
(* IN :                                                                  *)
(*                                                                       *)
(* OUT: N: Nombre de mois avant que l'utilisateur fasse sa dépense.      *)
(*                                                                       *)
(* BUT: Effectue une saisie pour savoir quand l'utilisateur veut         *)
(*      faire un test.                                                   *)
(*                                                                       *)
(*************************************************************************)

END SaisieMois.
```

★ ★ ★
★ ★

Modules:
Paramètres & Traces

Module Paramètres

```
MODULE Param;

FROM SYSTEM IMPORT BYTE, ADR;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window,WindowPtr,GadgetPtr, Gadget,WindowFlags,
     WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet, IntuiMessagePtr,
     CloseWindow, ActivateWindow,Requester, InitRequester, Request,
     RequesterFlagsSet, EndRequest, EndGadget,RelVerify,
     GadgetFlagsSet,GadgetActivationSet, GadgetMutualExcludeSet;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList,EndGadgetList,GadgetOpt,
     FreeGadgetList, AddGadgetTextButton, GadgetTypeReq,
     GlobalGadgetOpt;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM SimpleConsole IMPORT CreateConsole,DeleteConsole,PutStr,GetStr;
FROM SimpleConsoleCmds IMPORT wMove, wClrScr, wSetCursor,wClrEndLine;
FROM InOut IMPORT WriteInt,WriteLn,OpenOutputFile,CloseOutput;
FROM Strings IMPORT  CopyString, CompareString, Relation,ConcatString,
     StringLength, ConvStringToUpperCase;
FROM VariablesGlobales IMPORT cmap,Scr,Str3, Str20, Str40,
     TableauParametres;
FROM Rasters IMPORT SetRast;
FROM RealInOut IMPORT WriteReal;
FROM RealConversions IMPORT ConvRealToString,ConvStringToReal,
RealToStringFormat;
FROM MathLib0 IMPORT real;
FROM UTILS IMPORT Init,Term,Reel,EnleveBlancs;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Borderless};

VAR
  Win,Won,SubWin,Win1,Win2,Win3 : WindowPtr;
  le,I, i : CARDINAL;
  Gl,GlOK,Gl3 : GadgetPtr;
  Wp,WpOK,Wp3 : WindowProc;
  Sig : SignalSet;
  ReqOK : Requester;
  Msg,MsgOK : IntuiMessagePtr;
  Done1,Done2,Efface,Done,DoneOK,OK,Conf,UneSemaine,OKNombre,Win,Fen,
  PremPassage : BOOLEAN;
  Int : Gadget;
  st,Nom2,Max,L1,L2,L3,L4,L5,L6,L7,L8,NomInt,MaxInt : Str20;
  MAX,lene : REAL;
  FileName : Str40;
  For : RealToStringFormat;
  len : INTEGER;
```

```
PROCEDURE GadgetHandlerOK(VAR w:Window;VAR Msg:MsgData;
                          VAR Gad: Gadget);
BEGIN
  CASE Gad.GadgetID OF
     0 : OK := TRUE;DoneOK := TRUE |
     1 : OK := FALSE;DoneOK := TRUE |
     ELSE
  END;
END GadgetHandlerOK;

PROCEDURE OKRequester (FenWin : BOOLEAN);
BEGIN
     BeginGadgetList ();
    GadgetTypeReq := TRUE;
    GlobalGadgetOpt (GadgetFlagsSet {},GadgetActivationSet
                     {EndGadget,RelVerify});
    AddGadgetTextButton (10,40, ADR("OK"));
    AddGadgetTextButton (195,40, ADR("ANNULER"));
    AddGadgetTextButton (15,10,ADR("Introduire la disquette"));
    AddGadgetTextButton (87,21,ADR (" Param"));
    GadgetOpt (GadgetFlagsSet {}, GadgetActivationSet
               {RelVerify},GadgetMutualExcludeSet {});
    GlOK := EndGadgetList ();
    InitRequester (ReqOK);
    WITH ReqOK DO
         OlderRequest := NIL;
         LeftEdge := 185;
         TopEdge := 20;
         Width := 300;
         Height := 60;
         ReqGadget := GlOK;
         ReqBorder := NIL;
         ReqText := NIL;
         Flags := RequesterFlagsSet {};
         BackFill := BYTE(10);
         ReqLayer := NIL;
         ImageBMap := NIL;
    END;
    IF FenWin THEN
       IF Request (ReqOK, Win^) THEN
          DoneOK := FALSE;
          WHILE NOT DoneOK DO
                Sig:=Wait(SignalSet{
                     CARDINAL(Win^.UserPort^.mpSigBit)});
               LOOP
                MsgOK := GetMsg (Win^.UserPort^);
                IF (MsgOK = NIL) THEN EXIT; END;
                ProcIMsg (WpOK, MsgOK);
               END;
       END;
     END;
     ELSE IF Request (ReqOK, Win1^) THEN
            DoneOK := FALSE;
            WHILE NOT DoneOK DO
             Sig :=Wait(SignalSet{
                   CARDINAL(Win1^.UserPort^.mpSigBit)});
```

```
                LOOP
                    MsgOK := GetMsg (Win1^.UserPort^);
                    IF (MsgOK = NIL) THEN EXIT; END;
                    ProcIMsg (WpOK, MsgOK);
                END;
            END;
        END;
    END;
    FreeGadgetList (GlOK^);
END OKRequester;


PROCEDURE GadgetHandler3 (VAR w: Window; VAR Msg : MsgData;
                          VAR Gad: Gadget);
BEGIN
  CASE Gad.GadgetID OF
     0 : Done := TRUE |
     1 : Done := TRUE; Efface := FALSE; Done1 := FALSE |
     2 : FenWin := FALSE;
         OKRequester (FenWin);
         IF OK THEN
            Done:=TRUE;Done1:=TRUE;Done2:=TRUE;Efface:=FALSE;
         END;
  END;
END GadgetHandler3;


PROCEDURE GadgetHandler (VAR w: Window; VAR Msg : MsgData;
                         VAR Gad: Gadget);
BEGIN
  CASE Gad.GadgetID OF
     0 :TableauParametres[0] := 1;
        wMove (Win3^,1,2);
        wClrEndLine (Win3^);
        wMove (Win3^,1,2);
        PutStr (Win3^,ADR ("OUI"));
        L1 := "OUI" |
     1 :TableauParametres[0] := 0;
        wMove (Win3^,1,2);
        wClrEndLine (Win3^);
        wMove (Win3^,1,2);
        PutStr (Win3^,ADR ("NON"));
        L1 := "NON" |
     2 :TableauParametres[1] := 0;
        wMove (Win3^,1,5);
        wClrEndLine (Win3^);
        wMove (Win3^,1,5);
        PutStr (Win3^,ADR ("ECRIT"));
        L2 := "ECRIT" |
     3 :TableauParametres[1] := 2;
        wMove (Win3^,1,5);
        wClrEndLine (Win3^);
        wMove (Win3^,1,5);
        PutStr (Win3^,ADR ("ECRIT + PAROLE"));
        L2 := "ECRIT + PAROLE" |
```

```
 4 :TableauParametres[2] := 0;
    wMove (Win3^,1,9);
    wClrEndLine (Win3^);
    wMove (Win3^,1,9);
    PutStr (Win3^,ADR ("CLAVIER"));
    L3 := "CLAVIER" |
 5 :TableauParametres[2] := 1;
    wMove (Win3^,1,9);
    wClrEndLine (Win3^);
    wMove (Win3^,1,9);
    PutStr (Win3^,ADR ("SOURIS"));
    L3 := "SOURIS" |
 6 :TableauParametres[3] := 0;
    wMove (Win3^,1,12);
    wClrEndLine (Win3^);
    wMove (Win3^,1,12);
    PutStr (Win3^,ADR ("CLAVIER"));
    L4 := "CLAVIER" |
 7 :TableauParametres[3] := 1;
    wMove (Win3^,1,12);
    wClrEndLine (Win3^);
    wMove (Win3^,1,12);
    PutStr (Win3^,ADR ("BILLETS"));
    L4 := "BILLETS" |
 8 :TableauParametres[3] := 2;
    wMove (Win3^,1,12);
    wClrEndLine (Win3^);
    wMove (Win3^,1,12);
    PutStr (Win3^,ADR ("CALCULETTE"));
    L4 := "CALCULETTE" |
 9 :TableauParametres[4] := 0;
    wMove (Win3^,1,15);
    wClrEndLine (Win3^);
    wMove (Win3^,1,15);
    PutStr (Win3^,ADR ("CLAVIER"));
    L5 := "CLAVIER" |
10 :TableauParametres[4] := 1;
    wMove (Win3^,1,15);
    wClrEndLine (Win3^);
    wMove (Win3^,1,15);
    PutStr (Win3^,ADR ("BILLETS"));
    L5 := "BILLETS" |
11 :TableauParametres[4] := 2;
    wMove (Win3^,1,15);
    wClrEndLine (Win3^);
    wMove (Win3^,1,15);
    PutStr (Win3^,ADR ("CALCULETTE"));
    L5 := "CALCULETTE" |
12 : TableauParametres[6] := 0;
   wMove (Win3^,1,18);
   wClrEndLine (Win3^);
   wMove (Win3^,1,18);
   PutStr (Win3^,ADR ("OUI"));
    L6 := "OUI" |
13 : TableauParametres[6] := 1;
   wMove (Win3^,1,18);
   wClrEndLine (Win3^);
   wMove (Win3^,1,18);
   PutStr (Win3^,ADR ("NON"));
    L6 := "NON" |
```

```
14 : TableauParametres[9] := 1;
     wMove (Win3^,1,22);
     wClrEndLine (Win3^);
     wMove (Win3^,1,22);
     PutStr (Win3^,ADR ("UNE SEMAINE"));
     L7 := "UNE SEMAINE" |
15:  TableauParametres[9] := 2;
     wMove (Win3^,1,22);
     wClrEndLine (Win3^);
     wMove (Win3^,1,22);
     PutStr (Win3^,ADR ("UN MOIS"));
     L7 := "UN MOIS";
     L8 :="";
     wMove (Win3^,1,25);
     wClrEndLine (Win3^); |
16:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
        TableauParametres[10] := 0;
        wMove (Win3^,1,25);
        wClrEndLine (Win3^);
        wMove (Win3^,1,25);
        PutStr (Win3^,ADR ("LUNDI"));
        L8 := "LUNDI";
     END |
17:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
        TableauParametres[10] := 1;
        wMove (Win3^,1,25);
        wClrEndLine (Win3^);
        wMove (Win3^,1,25);
        PutStr (Win3^,ADR ("MARDI"));
        L8 := "MARDI";
     END |
18:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
        TableauParametres[10] := 2;
        wMove (Win3^,1,25);
        wClrEndLine (Win3^);
        wMove (Win3^,1,25);
        PutStr (Win3^,ADR ("MERCREDI"));
        L8 := "MERCREDI";
     END |
19:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
        TableauParametres[10] := 3;
        wMove (Win3^,1,25);
        wClrEndLine (Win3^);
        wMove (Win3^,1,25);
        PutStr (Win3^,ADR ("JEUDI"));
        L8:= "JEUDI";
     END |
20:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
        TableauParametres[10] := 4;
        wMove (Win3^,1,25);
        wClrEndLine (Win3^);
        wMove (Win3^,1,25);
        PutStr (Win3^,ADR ("VENDREDI"));
        L8 := "VENDREDI";
     END |
```

```
    21:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
            TableauParametres[10] := 5;
            wMove (Win3^,1,25);
            wClrEndLine (Win3^);
            wMove (Win3^,1,25);
            PutStr (Win3^,ADR ("SAMEDI"));
             L8 := "SAMEDI";
         END |
    22:  IF (CompareString(L7,"UNE SEMAINE") = equal) THEN
            TableauParametres[10] := 6;
            wMove (Win3^,1,25);
            wClrEndLine (Win3^);
            wMove (Win3^,1,25);
            PutStr (Win3^,ADR ("DIMANCHE"));
             L8:= "DIMANCHE";
         END |
    23:  FenWin := TRUE;
         OKRequester (FenWin);
         IF OK THEN
            Done := TRUE; Done1 := TRUE; Done2 := TRUE;
         END;
      END;
END GadgetHandler;

PROCEDURE Par_Defaut;
BEGIN
      L1  := "NON";
      L2  := "ECRIT + PAROLE";
      L3  := "CLAVIER";
      L4  := "BILLETS";
      L5  := "CALCULETTE";
      L6  := "OUI";
      L7  := "UNE SEMAINE";
      L8  := "LUNDI";
      TableauParametres [0] := 0;
      TableauParametres [1] := 2;
      TableauParametres [2] := 0;
      TableauParametres [3] := 1;
      TableauParametres [4] := 2;
      TableauParametres [5] := 0;
      TableauParametres [6] := 0;
      TableauParametres [7] := 0;
      TableauParametres [8] := 0;
      TableauParametres [9] := 1;
      TableauParametres [10] := 0;
      TableauParametres [11] := 1;
      TableauParametres [12] := 2;
END Par_Defaut;
```

```
BEGIN
    WITH Wp DO
      procGadgetUp := GadgetHandler;
    END;
    WITH WpOK DO
      procGadgetUp := GadgetHandlerOK;
    END;
    WITH Wp3 DO
      procGadgetUp := GadgetHandler3;
    END;
    Init;
      Nom2 := "";
      Max := "";
      Par_Defaut;
      Conf := FALSE;
      PremPassage := TRUE;
      Done := FALSE;
      Done1 := FALSE;
      Done2 := FALSE;
      WHILE (NOT Done) OR (NOT Done1) OR (NOT Done2) DO
       IF NOT Done THEN
        Efface := TRUE;
        WHILE Efface DO
         BeginGadgetList();
         AddGadgetTextButton (10,67,ADR("EFFACER TOUT"));
         AddGadgetTextButton (160,67,ADR("ECRAN SUIVANT"));
         AddGadgetTextButton (599,67,ADR("FIN"));
      Gl3 := EndGadgetList();
      Win:=CreateWindow(0,0,640,145,WIDCMP,WFlags,NIL,Scr,NIL);
         IF (Win # NIL) THEN
          IF CreateConsole (Win^) THEN
          wClrScr (Win^);
          PutStr (Win^,ADR("Bonjour, les deux écrans suivants vous"));
          PutStr (Win^,ADR(" demandent de sélectionner des valeurs"));
          wMove (Win^,1,2);
          PutStr(Win^,ADR("pour 12 paramètres. Ensuite, le programme
                           se"));
          PutStr (Win^,ADR (" déroulera en fonction" ));
          wMove (Win^,1,3);
          PutStr (Win^,ADR ("de vos choix. Pour"));
          PutStr (Win^,ADR (" sélectionner une valeur, cliquez sur la
                             case"));
          PutStr (Win^,ADR (" adéquate."));
          wMove (Win^,1,4);
          PutStr (Win^,ADR ("A droite se trouvent les valeurs"));
          PutStr (Win^,ADR (" par défaut, puis celles réellement
                             choisies."));
          wMove (Win^,1,5);
          PutStr (Win^,ADR ("Pour garnir les autres paramètres,
                            cliquez"));
          PutStr (Win^,ADR (" sur 'ECRAN SUIVANT'."));
          wMove (Win^,1,6);
          PutStr (Win^,ADR ("Quand tous les choix sont"));
          PutStr (Win^,ADR (" faits, cliquez sur 'FIN'."));
          wMove (Win^,1,7);
          PutStr (Win^,ADR ("Avant ces choix, nous vous demandons"));
          PutStr (Win^,ADR (" des renseignements généraux sur"));
          wMove (Win^,1,8);
```

```
PutStr (Win^,ADR ("la personne pour qui les paramètres "));
PutStr (Win^,ADR (" seront valables."));
wMove (Win^,1,9);
PutStr(Win^,ADR("Après cela, cliquez sur 'ECRAN SUIVANT'"));
PutStr (Win^,ADR (" et choisissez les paramètres."));
wMove (Win^,1,12);
PutStr(Win^,ADR("Donnez le prénom de l'utilisateur SVP: "));
IF (CompareString (Nom2,"") <> equal) THEN
     PutStr (Win^,ADR("("));
     PutStr (Win^,ADR(Nom2));
     PutStr (Win^,ADR(") "));
END;
NomInt := "";
IF ((CompareString(Nom2,"") = equal)
      AND (CompareString(NomInt,"") = equal))
  THEN
     WHILE (CompareString(NomInt,"") = equal) DO
         GetStr (Win^,ADR (NomInt),SIZE(NomInt));
     END
  ELSIF  (CompareString(Nom2,"") <> equal)
      THEN   GetStr (Win^,ADR (NomInt),SIZE(NomInt));
END;
IF (CompareString (NomInt,"") <> equal) THEN
     CopyString (Nom2,NomInt);
END;
ConvStringToUpperCase (Nom2);
 EnleveBlancs (Nom2);
 le:=StringLength(Nom2);
 len := le;
 lene := real (le);
 ConvRealToString (lene,st,20,For);
 wMove (Win^,1,13);
 wMove (Win^,1,14);
PutStr (Win^,ADR ("SVP, donnez le montant d'argent"));
wMove (Win^,1,15);
PutStr(Win^,ADR("maximum que l'utilisateur peut avoir : "));
IF (CompareString (Max,"") <> equal) THEN
     PutStr (Win^,ADR("("));
     PutStr (Win^,ADR(Max));
     PutStr (Win^,ADR(") "));
END;
OKNombre := FALSE; MAX := 0.;
WHILE NOT OKNombre DO
  IF StringLength (Max) = 0
     THEN wMove (Win^,45,15)
     ELSE wMove (Win^,45 + StringLength (Max) + 2,15)
  END;
  wClrEndLine (Win^);
  GetStr (Win^,ADR (MaxInt),SIZE (MaxInt));
  Reel (MaxInt,OKNombre);
  IF OKNombre THEN
     MAX := ConvStringToReal (MaxInt);
        IF (CompareString (MaxInt,"") <> equal) THEN
        IF (MAX = 0.) THEN
           OKNombre := FALSE;
        ELSE
           CopyString (Max,MaxInt);
        END;
        END;
```

```
      END;
    END;
    PremPassage := FALSE;
    wSetCursor (Win^,FALSE);
    Win1:=CreateWindow (0,145,640,85,WIDCMP,WFlags2,Gl3,Scr,NIL);
      IF Win1 # NIL THEN
       Done := FALSE;
       WHILE (NOT Done) DO
              Sig:=Wait(SignalSet{CARDINAL
                                  (Win1^.UserPort^.mpSigBit)});
              LOOP
                Msg := GetMsg(Win1^.UserPort^);
                IF (Msg = NIL) THEN EXIT; END;
                ProcIMsg (Wp3, Msg);
              END;
          END;
       CloseWindow (Win1^);
     END;
        DeleteConsole (Win^);
       END;
       CloseWindow (Win^);
  END;
END; (* du WHILE Efface *)
  END;  (* du IF NOT Done  *)


    BeginGadgetList();
    AddGadgetTextButton (270,7,ADR("OUI"));
    AddGadgetTextButton (422,7,ADR("NON"));

    AddGadgetTextButton (200,32,ADR("ECRIT"));
    AddGadgetTextButton (280,32,ADR("ECRIT + PAROLE"));

    AddGadgetTextButton (250,61,ADR("CLAVIER"));
    AddGadgetTextButton (410,61,ADR("SOURIS"));

    AddGadgetTextButton (200,88,ADR("CLAVIER"));
    AddGadgetTextButton (280,88,ADR("BILLETS"));
    AddGadgetTextButton (360,88,ADR("CALCULETTE"));

    AddGadgetTextButton (200,112,ADR("CLAVIER"));
    AddGadgetTextButton (280,112,ADR("BILLETS"));
    AddGadgetTextButton (360,112,ADR("CALCULETTE"));

    AddGadgetTextButton (270,139,ADR("OUI"));
    AddGadgetTextButton (422,139,ADR("NON"));

    AddGadgetTextButton (230,172,ADR("UNE SEMAINE"));
    AddGadgetTextButton (390,172,ADR("UN MOIS"));

    AddGadgetTextButton (200,190,ADR("LUNDI"));
    AddGadgetTextButton (280,190,ADR("MARDI"));
    AddGadgetTextButton (360,190,ADR("MERCREDI"));
    AddGadgetTextButton (440,190,ADR("JEUDI"));
    AddGadgetTextButton (200,205,ADR("VENDREDI"));
    AddGadgetTextButton (280,205,ADR("SAMEDI"));
    AddGadgetTextButton (360,205,ADR("DIMANCHE"));

    AddGadgetTextButton (599,225,ADR("FIN"));
    Gl := EndGadgetList();
```

```
IF (Gl # NIL) THEN
  IF NOT Done1 THEN
   Win := CreateWindow (0,0,640,240,WIDCMP,WFlags,Gl,Scr,NIL);
   Win2 := CreateWindow (0,0,180,220,WIDCMP,WFlags,NIL,Scr,NIL);
   Win3:=CreateWindow (520,0,120,220,WIDCMP,WFlags,NIL,Scr,NIL);
    IF (Win # NIL) AND  (Win2 # NIL)  AND  (Win3 # NIL)  THEN
   IF CreateConsole (Win2^) THEN
    PutStr (Win2^,ADR("Utilisez-vous"));
    wMove (Win2^,1,2);
    PutStr (Win2^,ADR ("une imprimante ?"));

    wMove (Win2^,1,4);
    PutStr (Win2^,ADR("Les questions sont "));
    wMove (Win2^,1,5);
    PutStr (Win2^,ADR ("posées par écrit ou"));
    wMove (Win2^,1,6);
    PutStr (Win2^,ADR ("par écrit et parole ?"));

     wMove (Win2^,1,8);
     PutStr (Win2^,ADR ("L'utilisateur entrera"));
     wMove (Win2^,1,9);
     PutStr (Win2^,ADR ("son prénom par :"));

     wMove (Win2^,1,11);
    PutStr (Win2^,ADR ("L'utilisateur entrera"));
    wMove (Win2^,1,12);
    PutStr (Win2^,ADR ("les recettes par :"));

     wMove (Win2^,1,14);
     PutStr (Win2^,ADR ("L'utilisateur entrera"));
    wMove (Win2^,1,15);
    PutStr (Win2^,ADR ("les prévisions par :"));

    wMove (Win2^,1,17);
    PutStr (Win2^,ADR("Voulez-vous la ligne"));
    wMove (Win2^,1,18);
    PutStr (Win2^,ADR ("du temps ?"));

     wMove (Win2^,1,20);
    PutStr (Win2^,ADR ("Le cycle est-il une"));
     wMove (Win2^,1,21);
    PutStr (Win2^,ADR ("semaine ou un mois?"));

     wMove (Win2^,1,23);
    PutStr (Win2^,ADR ("Si cycle = semaine,"));
     wMove (Win2^,1,24);
    PutStr (Win2^,ADR ("donnez le jour de"));
     wMove (Win2^,1,25);
    PutStr (Win2^,ADR ("début de cycle."));
    wSetCursor (Win2^,FALSE);
    DeleteConsole (Win2^);
    END;
    IF CreateConsole (Win3^) THEN
    wMove (Win3^,1,2);
    PutStr (Win3^,ADR (L1));
    wMove (Win3^,1,5);
    PutStr (Win3^,ADR (L2));
    wMove (Win3^,1,9);
    PutStr (Win3^,ADR (L3));
    wMove (Win3^,1,12);
```

```
            PutStr (Win3^,ADR (L4));
            wMove (Win3^,1,15);
            PutStr (Win3^,ADR (L5));
            wMove (Win3^,1,18);
            PutStr (Win3^,ADR (L6));
            wMove (Win3^,1,22);
            PutStr (Win3^,ADR (L7));
            wMove (Win3^,1,25);
            PutStr (Win3^,ADR (L8));
            wSetCursor (Win3^,FALSE);
            WHILE (NOT Done1) DO
              Sig := Wait(SignalSet{CARDINAL(Win^.UserPort^.mpSigBit)});
                 LOOP
                   Msg := GetMsg(Win^.UserPort^);
                   IF (Msg = NIL) THEN EXIT; END;
                   ProcIMsg (Wp, Msg);
                 END;
               END;
          DeleteConsole (Win3^);
          END;
             CloseWindow(Win^);CloseWindow(Win2^); CloseWindow(Win3^);
       END;
  FreeGadgetList (Gl^);
  END; (* du IF Gl^ ... *)
END ;  (* du WHILE *)
Term;
FileName := "Param:";
ConcatString (FileName,"UTILISATEUR");
ConcatString (FileName,".PARAM");
OpenOutputFile (FileName);

IF Done THEN
    I := 0;
    WHILE (I <= 12) DO
      WriteInt (TableauParametres[I],1);
      WriteLn;
      I := I + 1;
    END;
    CloseOutput;
END;
FileName := "Param:";
ConcatString (FileName,Nom2);
ConcatString (FileName,".MAX");
IF MAX # 0. THEN
  OpenOutputFile (FileName);
  IF Done THEN
  WriteReal(MAX,6);
  CloseOutput;
  END;
END;
END;
END Param.
```

Module Traces

```
MODULE TU;

FROM ASCII IMPORT CharIsControl;
FROM SYSTEM IMPORT SHORT,BYTE, ADR;
FROM Conversions IMPORT ConvNumberToString;
FROM AmigaDOS IMPORT IoErr,CreateDir,FileLock;
FROM AmigaDOSProcess IMPORT Delay;
FROM Intuition IMPORT Window, WindowPtr,GadgetPtr,Gadget, WindowFlags,
     WindowFlagsSet, IDCMPFlags, IDCMPFlagsSet,IntuiMessagePtr,
     CloseWindow,NewWindow, Image, CustomScreen, ActivateWindow,
     Requester, InitRequester, Request, RequesterFlagsSet, EndRequest,
     EndGadget, RelVerify,  GadgetFlagsSet, GadgetActivationSet,
     GadgetMutualExcludeSet, OnGadget, OffGadget,ScreenToFront;
FROM Ports IMPORT GetMsg;
FROM SimpleGadgets IMPORT BeginGadgetList,EndGadgetList,GadgetOpt,
     FreeGadgetList,AddGadgetTextButton,GadgetTypeReq,GlobalGadgetOpt;
FROM SimpleIDCMP IMPORT MsgData, WindowProc, ProcIMsg;
FROM SimpleWindows IMPORT CreateWindow;
FROM Tasks IMPORT SignalSet, Wait;
FROM SimpleConsole IMPORT CreateConsole,DeleteConsole,PutStr,GetStr,
     PutCh;
FROM SimpleConsoleCmds IMPORT wMove, wClrScr, wSetColor, wSetCursor,
     wClrEndLine;
FROM InOut IMPORT WriteString,WriteInt,Write,WriteLn,WriteWrd,
     OpenOutputFile,OpenInputFile,CloseOutput,CloseInput,Done,
     ReadInt,ReadString,termCH,Read;
FROM Intuition IMPORT StringInfoPtr;
FROM Strings IMPORT  CopyString,CompareString,Relation, ConcatString,
     StringLength, ConvStringToUpperCase,OverwriteWithSubString;
FROM VariablesGlobales IMPORT Str3,Str9,Str20,Str40,TableauParametres,
     Numseance,Scr,cmap,TableauTraces;
FROM Rasters IMPORT SetRast,RastPortPtr;
FROM Drawing IMPORT SetAPen,Move,Draw;
FROM RealInOut IMPORT WriteReal,ReadReal;
FROM RealConversions IMPORT ConvRealToString,RealToStringFormat,
     ConvStringToReal ;
FROM MathLib0 IMPORT entier,real;
FROM ModNom IMPORT TraceDate;
FROM UTILS IMPORT Init,Term,GetNom;

CONST
  WIDCMP = IDCMPFlagsSet {GadgetUp};
  WFlags = WindowFlagsSet {Activate};
  WFlags2 = WindowFlagsSet {Borderless};
  Blanc15 = "               ";
TYPE
  Dernier = RECORD
          Type : Str3;
          X    : CARDINAL;
          Y    : CARDINAL;
        END;
```

```
VAR
  Der : Dernier;
  Nw : NewWindow;
  Win,Win1,Win2,Win3,Win4 : WindowPtr;
  i : CARDINAL;
  Gl,GlOK,Gl2,Gl3,Gl4,Gl5 : GadgetPtr;
  Wp,Wp2,WpOK,Wp3,Wp4,WpImp : WindowProc;
  Sig : SignalSet;
  ReqOK,Req, Req2,Req3,Req5 : Requester;
  Msg,MsgOK,Msg2,Msg3,Msg4,Msg5 : IntuiMessagePtr;
  Done1,Done2,Done3,Done4,Efface,DoneOK,OK,Conf,UneSemaine,OKNombre,
  FenWin,PremPassage,Done5,Imp2 : BOOLEAN;
  Int : Gadget;
  Confirmation : Str40;
  Seance,Nom2,Max,NomInt,MaxInt : Str20;
  MAX : REAL;
  FileName : Str40;
  Fl : FileLock;
  le : CARDINAL;
  st : Str20;
  For : RealToStringFormat;
  len : INTEGER;
  lene : REAL;
  passe : Str20;
  reconnu : BOOLEAN;
  DV,essai : INTEGER;
  DoneNom : BOOLEAN;
  DernVers : REAL;
  LigneBlanc : ARRAY [0..77] OF CHAR;
  LigneSouligne : ARRAY [0..77] OF CHAR;
  imprime : BOOLEAN;
  Tableau : ARRAY [0..20000] OF CHAR;
  ca : CHAR;
  longfich : INTEGER;
  caract : CHAR;

PROCEDURE JOUR (VAR toto:Str40);
BEGIN
    TableauTraces.Indice:=1;
    TraceDate;
    CopyString(toto,TableauTraces.Tab[TableauTraces.Indice-1]);
END JOUR;

PROCEDURE GadgetHandler4(VAR W:Window;VAR Msg4:MsgData;VAR
                                Gad:Gadget);

BEGIN

   Done4 := TRUE;
   IF Gad.GadgetID = 1 THEN imprime := TRUE; END;

END GadgetHandler4;
```

```
PROCEDURE Saisieok (impr : BOOLEAN);

BEGIN
      BeginGadgetList ();
      AddGadgetTextButton (1,1,ADR(" OK "));
      IF impr = TRUE
      THEN  AddGadgetTextButton (110,1,ADR(" IMPRESSION "));
      END;
      Gl4 := EndGadgetList ();
      Win3:=CreateWindow (312,226,320,12,WIDCMP,WFlags2,Gl4,Scr,NIL);
      Done4 := FALSE;
      WHILE (NOT Done4) DO
          Sig := Wait(SignalSet{CARDINAL(Win3^.UserPort^.mpSigBit)});
          LOOP
            Msg4 := GetMsg(Win3^.UserPort^);
            IF (Msg4 = NIL) THEN EXIT; END;
            ProcIMsg (Wp4, Msg4);
          END;
      END;
      CloseWindow(Win3^);
      FreeGadgetList (Gl4^);

END Saisieok;

PROCEDURE GadgetHandlerImp (VAR w:Window; VAR Msg5 : MsgData;
                            VAR Gad: Gadget);

BEGIN

  CASE Gad.GadgetID OF

    0 : |
    1 : |
    2 : Done5 := TRUE; Imp2 := TRUE |
    3 : Done5 := TRUE;
  END;

END GadgetHandlerImp;


PROCEDURE ImpParam ;

VAR StTemps : Str40;
    J : INTEGER;
    I : INTEGER;

BEGIN

    Imp2 := FALSE;
    WITH WpImp DO
      procGadgetUp := GadgetHandlerImp;
    END;
    BeginGadgetList();
    GadgetTypeReq := TRUE;
    GlobalGadgetOpt (GadgetFlagsSet{},GadgetActivationSet{EndGadget,
                     RelVerify});
    AddGadgetTextButton (60,10,ADR(" L'imprimante est allumée "));
    GadgetOpt (GadgetFlagsSet {},GadgetActivationSet {RelVerify},
               GadgetMutualExcludeSet {});
```

```
AddGadgetTextButton (110,30,ADR(" et 'ON-LINE' ? "));
GadgetOpt (GadgetFlagsSet {},GadgetActivationSet {RelVerify},
           GadgetMutualExcludeSet {});
AddGadgetTextButton (110,50,ADR(" OUI "));
AddGadgetTextButton (195,50,ADR(" NON "));
G15 := EndGadgetList();
InitRequester (Req5);
WITH Req5 DO
        OlderRequest := NIL;
        LeftEdge := 150;
        TopEdge := 140;
        Width := 340;
        Height := 80;
        ReqGadget := G15;
        ReqBorder := NIL;
        ReqText := NIL;
        Flags := RequesterFlagsSet {};
        BackFill := BYTE(8);
        ReqLayer := NIL;
        ImageBMap := NIL;
END;

IF Request (Req5,Win2^)
THEN
Done5 := FALSE;
WHILE (NOT Done5) DO
      Sig := Wait(SignalSet{CARDINAL(Win2^.UserPort^.mpSigBit)});
      LOOP
        Msg5 := GetMsg(Win2^.UserPort^);
        IF (Msg5 = NIL) THEN EXIT; END;
        ProcIMsg (WpImp, Msg5);
      END;
END;
END;
IF Imp2 = TRUE
THEN

OpenOutputFile ("prt:");
IF Done
THEN
   WriteLn;
   WriteLn;
   WriteLn;
   WriteString("Nom de l'utilisateur : ");
   WriteString(nomuser);
   WriteLn;
   WriteLn;
   WriteString("Numéro de la séance : ");
   WriteString(Numseance);
   WriteLn;
   WriteLn;
   StTemps :="";
   JOUR (StTemps);
   WriteString("Date de l'impression : ");
   WriteString(StTemps);
   WriteLn;
   WriteLn;
   WriteLn;
   WriteLn;
```

```
      WriteString("                                   TRACES D'UTILISATION");
      WriteLn;
      WriteString("                                   --------------------");
      WriteLn;
      WriteLn;
      CloseOutput ();
      I := 0;
      J := 0;
      OpenOutputFile ("prt:");
      WHILE (I < longfich) DO
                IF (J > 110) THEN
                     CloseOutput ();OpenOutputFile("prt:");J:=0;
                END;
                Write (Tableau[I]);
                I := I + 1;
                J := J + 1;
       END;
       WriteLn;
       WriteLn;
       WriteLn;
       CloseOutput ();
   END;
   END;
   FreeGadgetList(Gl5^);
END ImpParam;


VAR I,J,Long,Valact,Valprec : INTEGER;
    St : Str20;
    Filechaine :Str40;
    ent : INTEGER;
    ree,ree2 : REAL;
    OKSeance : BOOLEAN;

BEGIN

    WITH Wp4 DO
      procGadgetUp := GadgetHandler4;
    END;
      Init;
      Win2 := CreateWindow (0,0,640,240,WIDCMP,WFlags,NIL,Scr,NIL);
      IF (Win2 # NIL)
      THEN IF CreateConsole(Win2^)
           THEN
            OKSeance := FALSE;
            WHILE (NOT OKSeance) DO
                wClrScr(Win2^);
                wMove(Win2^,30,2);
                PutStr(Win2^,ADR("TRACES D'UTILISATION"));
                wMove(Win2^,30,3);
                PutStr(Win2^,ADR("--------------------"));
                GetNom(Win2,nomuser);
       FileName := "Param:";
       ConcatString (FileName,nomuser);
       ConcatString (FileName,".DV");
       OpenInputFile (FileName);
       IF Done THEN ReadReal (DernVers);
          DernVers := DernVers + 1.;
       ELSE DernVers := 0.
       END;
```

```
CloseInput;
        wMove(Win2^,7,14);
        PutStr(Win2^,ADR("Numéro de la séance [ de 1 à "));
        ConvRealToString(DernVers,St,0,Decimal);
        DernVers := DernVers - 1.;
        PutStr(Win2^,ADR(St));
        PutStr(Win2^,ADR(" ] : "));
        wSetCursor(Win2^,TRUE);
        GetStr(Win2^,ADR(Numseance),SIZE(Numseance));
        wSetCursor(Win2^,FALSE);
        ree := ConvStringToReal (Numseance);
        ree := ree - 1.;
        ent := entier (ree);
        ree2 := real(ent);
        IF (ree2 = ree) AND (ent >= 0) AND (ree <= DernVers )
        THEN OKSeance := TRUE;
             ConvRealToString(ree,Seance,0,Decimal);
        END;
    END;
    wClrScr(Win2^);
    Filechaine := "Param:";
    ConcatString(Filechaine,nomuser);
 ConcatString(Filechaine,Seance);
 ConcatString(Filechaine,".TRACE");
    OpenInputFile(Filechaine);
    ScreenToFront(Scr^);
    IF Done
    THEN
       Read (caract);
       WHILE (NOT CharIsControl(caract)) DO
         Read (caract);
       END;
       I := 0;
       longfich := 0;
       WHILE (Done) DO
            Read(Tableau[I]);
            I := I + 1;
            longfich := longfich + 1;
       END;
       CloseInput;

       I := 0;
       J := 0;
 Win4:=CreateWindow(150,22,340,175,WIDCMP,WFlags,NIL,Scr,NIL);
       IF (Win4 # NIL)
       THEN
         IF CreateConsole(Win4^)
         THEN
          wClrScr(Win4^);
          wSetCursor(Win4^,FALSE);
          wMove(Win4^,1,1);
          WHILE (I < longfich ) DO
              IF CharIsControl (Tableau[I]) THEN INC(J); END;
              IF (J > 16) THEN Saisieok(FALSE); J :=0; END;
              PutCh(Win4^,Tableau[I]);
              INC(I);
          END;
          imprime := FALSE;
```

```
                Saisieok(TRUE);
                DeleteConsole(Win4^);
              END;
              CloseWindow(Win4^);
            END;

            IF imprime = TRUE THEN ImpParam; END;
          END;
          DeleteConsole(Win2^);
        END;
        CloseWindow(Win2^);
    END;
    Term;
END TU.
```