

Le questionnaire d'une enquête en deux phases avec tirage d'un échantillon représentatif.

François Clanché, INSEE, France

1. Résumé

L'enquête « Propriétaires bailleurs » a été réalisée en 1996/97 par l'INSEE, auprès de 3 000 ménages qui avaient déclaré, au cours de l'enquête nationale Logement, posséder un patrimoine immobilier locatif ou en avoir détenu un depuis 1993. Son but est de faire décrire au ménage l'ensemble des logements locatifs qu'il possède, les revenus qu'il en tire, la façon dont il a constitué son patrimoine et ses projets.

La difficulté technique de cette enquête est la suivante : Les "gros" bailleurs, ceux qui possèdent plus de deux logements, représentent à peine 15 % des ménages propriétaires, mais ils possèdent près de la moitié du parc. On ne peut évidemment pas leur demander de décrire dans le détail la totalité de leur patrimoine, sinon l'enquête durerait plusieurs heures. On a donc mis au point le système suivant : au début du questionnaire, le ménage énumère la totalité des logements qu'il possède et en donne les caractéristiques principales. Avec ces quelques indications, 2 logements "représentatifs" de l'ensemble sont choisis par le logiciel, que le ménage décrit dans le détail. Grâce à cette procédure, l'enquête, qui dure moins d'une demi-heure quand le ménage possède un seul logement, dépasse rarement les 50 minutes, même dans le cas de patrimoines importants.

Dans le programme en Blaise-III, on a procédé en 3 phases :

- la première partie de l'enquête lance des blocks identiques, courts, qui permettent de noter une description sommaire de chaque logement (adresse, date de construction, situation juridique, taille). Pour les appartements situés dans des immeubles, on travaille à un double niveau : on décrit une seule fois l'adresse, la date de construction et la situation juridique du bâtiment, puis on pose, dans les petits immeubles, 3 questions pour chaque appartement (dont la taille et des précisions permettant de retrouver un appartement), et pour les grands immeubles on demande une répartition par taille des appartements. Parmi les logements ainsi repérés, seuls certains sont susceptibles de faire l'objet d'une description détaillée dans la deuxième partie (les logements « éligibles »).

- à la fin de cette première partie, le programme crée en AUXFIELDS une table ayant autant de lignes que de logements éligibles (il y a donc des duplications dans le cas des immeubles) et comportant en colonne les variables caractéristiques de la représentativité souhaitée. Les "individus/logements" sont ensuite triés selon ces caractéristiques grâce à un algorithme optimisé pour ne pas trop ralentir la collecte sur un ordinateur peu performant. Enfin vient le tirage lui-même par une méthode systématique utilisant la fonction RANDOM.

- Au début de la deuxième partie du questionnaire, le programme récupère les informations de la première partie, essentielles pour filtrer les questions et adapter les textes par la suite. Cette information récupérée permet en particulier d'indiquer au ménage quels logements ont été désignés, avec un message du type "nous allons maintenant décrire l'appartement de 3 pièces 1er étage gauche avenue Charles de Gaulle à Lyon". La difficulté de cette phase vient de ce qu'il n'y a aucun lien systématique entre les rangs des blocks décrivant un même logement entre les parties 1 et 2.

Nous allons examiner plus en détail ces trois phases de l'enquête en nous attachant aux syntaxes Blaise « originales » et à leurs justifications. Les extraits de programme proposés ne sont que des illustrations, extraites d'un programme global beaucoup plus long et complexe, que l'auteur tient à la disposition des personnes intéressées.

2. Première phase : description rapide de tout le patrimoine

Cette phase comporte en fait deux parties : les logements « isolés » et les immeubles.

1. Les logements isolés sont décrits de façon simple dans des blocks de type BIsolé dans lesquels on demande notamment la commune où est situé le logement, son adresse, sa taille, et son occupation actuelle. Ces réponses permettent de renseigner, toujours à l'intérieur du block Isolé, la variable Eligi (qui distingue les logements susceptibles de faire l'objet d'une description détaillée) et la variable Tri qui servira plus tard pour s'assurer de la représentativité de l'échantillon des logements décrits en détail. A la fin de chaque logement « Isolé », la question Autre permet de savoir si on doit une nouvelle fois entamer une description ou si l'on doit passer à la partie Immeubles. Des compteurs de logements décrits (Mlo) et de logement isolés éligibles (Mis) sont incrémentés.

FIELDS

```
{ ** blocks de description des logements ** }  
Isole : ARRAY[0..10] OF BIsolé  
Mlo : 0..999 { nbre total de logt repérés }  
Mis : 0..10 {nbre d'isolés éligibles }
```

RULES

```
Isole[0]  
FOR n:=1 TO 10 DO  
  IF Isolé[n-1].Autre = oui  
  THEN Isole[n]  
    Mlo := Mlo + 1  
    IF Isole[n].Eligi = Oui  
    THEN Mis := Mis+1  
  ENDIF
```

```
ENDIF
ENDDO
```

2. Pour ce qui est des immeubles, le principe général est le même : on lance successivement un ou plusieurs blocks de type Bimme où certaines questions propres au bâtiment sont similaires au cas isolé. Si l'immeuble comporte moins de 10 appartements, l'occupation de l'appartement et sa taille sont détaillées appartement par appartement (sous-block Immeub.Appart), et on calcule une variable Tri pour chaque appartement. S'il compte plus de 9 appartements, on pose des questions sur la répartition entre les différentes tailles et on génère, toujours à l'intérieur du block, autant de variables Tri qu'il y a d'appartements dans l'immeuble en utilisant les données sur la répartition entre tailles. On calcule également le nombre de logements éligibles de l'immeuble (NbElig).

```
FIELDS
```

```
Immeub : ARRAY[0..3] OF Bimme
Elilmm  : 0..400 { nbre d'éligibles en immeubles }
Nim     : 0..3 {nombre d'immeubles repérés }
Neligm  : 0..350 { nbre total de logements éligibles : Mis + Elilmm }
```

```
RULES
```

```
Nelig := Mis
```

```
Immeub[0]
FOR z:= 1 TO 3 DO
IF Immeub[z-1].Autre = oui THEN
    Immeub[z]
    Nim := Nim + 1
    Mlo := Mlo + Immeub[z].Inolo
    IF Immeub[z].Eligi = Oui THEN
        Elilmm := Elilmm + Immeub[z].NbElig
        Neligm := Neligm + Immeub[z].NbElig
    ENDIF
ENDIF
ENDIF
ENDDO
```

3. Deuxième phase : choix des logements à décrire en détail

1. Pour rendre possible un tri et un tirage parmi tous les logements, on renseigne un vecteur d'auxfields comportant, pour chaque logement éligible (qu'il soit isolé ou situé dans un immeuble), la variable ATri. Cet ARRAY d'auxfields est indicé par i, i allant de 1 à Neligm, et la variable Aclasst est initialisée à i. On remplit cet ARRAY en deux temps, d'abord avec les logements isolés, ensuite avec les appartements.

```

AUXFIELDS
AClasst  : ARRAY[0..350] of 0..350
Anbcorr  : ARRAY[0..350] of 0..350
ATri     : ARRAY[1..350] of 1..7000
Anum     : ARRAY[1..350] of 0..2
AChoix   : ARRAY[1..350] of OuiNonType

```

RULES

```

FOR n := 1 TO 10 DO
  IF Isole[n].Eligi = Oui THEN
    FOR i:= 1 TO Mis DO
      IF Isole[n].k = i THEN
        ATri[i] := Isolé[n].Tri
        AClasst[i] := i
        ANbCorr[i] := Isolé[n].Nbc
      ENDIF
    ENDDO
  ENDIF
ENDDO

IF Nim > 0 THEN
  FOR z:= 1 to Nim DO
    IF Immeub[z].Eligi = Oui THEN
      For i:= Immeub[z].Debut TO Nelig DO
        IF i > (Immeub[z].Debut)
          AND i < (Immeub[z].Debut + Immeub[z].NbElig + 1)
          THEN
            AClasst[i] := i
            w := (i - Immeub[z].Debut)
            ATri[i] := Immeub[z].Tri[w]
          ENDIF
        ENDDO
      ENDDO
    ENDIF
  ENDDO
ENDIF

```

2. Tri des logements : un algorithme fait en sorte que la variable AClasst de chacun des Nelig logements corresponde au classement selon ATri. Pour cela on utilise la méthode de l'insertion successive des individus/logements, dont les principes sont les suivants:

Au début de chaque étape les (i-1) premiers individus sont triés, et on introduit le ième individu avec le classement i.

On compare la valeur de la variable ATri de l'individu i avec celle de l'individu classé (i-1)ème, c'est-à-dire dernier à l'issue de l'étape précédente.

Si elle est supérieure, le classement est alors entièrement bon et on peut passer à la phase suivante (introduction du (i+1)ème individu).

Sinon, on inverse les classements des deux derniers individus (i est classé (i-1)ème), et on compare son ATri avec celui de l'individu classé (i-2)ème, etc...

On ne passe à la phase suivante que quand l'individu i a trouvé sa « bonne » place.

```

RULES
FOR i:=2 TO nelig DO { évolution du niveau d'entrée : étapes }
  h := i -1
  FOR k:=1 TO h DO { évolution de la profondeur de comparaison }
    FOR q := 1 TO h DO { recherche du j avec qui comparer }
      IF AClasst[q] = (AClasst[i] - 1) THEN
        IF (ATri[q] > Atri[i])
          THEN devant := Aclasst[q]
              Aclasst[q] := Aclasst[i]
              AClasst[i] := devant
          ENDIF
        ENDIF
      ENDDO
    ENDDO
  ENDDO

```

Remarque : la syntaxe de l'algorithme est ici simplifiée. Pour gagner en performance on introduit des tests qui limitent le nombre de boucles parcourues aux seules comparaisons pertinentes.

3. Le choix des logements à décrire se fait ensuite par un tirage systématique, dont le pas est calculé en fonction du nombre de logements éligibles et le niveau de départ tiré au hasard par la fonction RANDOM. Notons la nécessité de « fixer » le valeur de Alea en utilisant l'ordre KEEP pour éviter qu'elle se modifie à tout moment.

Toujours dans la table d'auxfields, on parcourt l'ensemble des indices i de 1 à Nelig, et les logements « choisis » se distinguent par des variables Achoix renseignée à oui, et leur ordre de tirage est indiqué dans la variable Anum.

Si il y a moins de 3 logements éligibles, le nombre de logements détaillés décrits (NbDet) est égal à Nelig et le renseignement de Achoix et de Anum est plus aisé.

```

FIELDS
Alea , Pas , Debut : REAL , EMPTY
NbDet : 0..2 {nombre de descriptions détaillées}
Deta : ARRAY[1..2] of 0..350 {numéros des logements choisis}

RULES

{ ** Tirage de alea ** }
Alea.Keep
IF Alea = Empty THEN
  Alea := Random
ENDIF

{ ** calcul du nombre de blocks détaillés nbdet ** }
IF Nelig < 3 THEN NbDet := Nelig
  ELSE NbDet := 2
ENDIF

{ ** détermination des logements à décrire ** }
IF nelig < 3 THEN
  FOR i:= 1 TO nelig DO
    AChoix[i]:= oui Anum[i] := Aclasst[i]
    IF Deta[1] <> RESPONSE THEN Deta[1] := i
    ELSE Deta[2] := i
  ENDDO
ELSE
  Aclasst[0]:=0
  Pas := (Nelig/2)
  Debut := Pas*Alea
  FOR i:= 1 TO nelig DO
    IF ((Aclasst[i]-1 <= Debut) AND ( Debut < Aclasst[i]))
      THEN AChoix[i]:= oui Anum[i]:=1 deta[1] := i
    ELSEIF ((Aclasst[i]-1 <=(Debut+pas)) AND ((Debut+pas) < Aclasst[i]))
      THEN AChoix[i]:= oui Anum[i]:=2 Deta[2] := i
    ELSE AChoix[i]:= non Anum[i] := 0
  ENDDO
ENDIF

```

```
ENDDO
ENDIF
```

4. Troisième phase : lancement des descriptions détaillées

1. Une message paramétré indique à l'enquêté (et à l'enquêteur) le nombre de logements qu'il va maintenant avoir à décrire en détail.

```
FIELDS
```

```
Intro "Nous allons décrire en détail ^tt vous venez d'évoquer. ": STRING[1]
```

```
RULES
```

```
IF NbDet = Mlo THEN
```

```
  IF Mlo = 1 THEN tt := ' le logement que'
```

```
  ELSE tt := ' les '+str(Nbdet)+' logements que'
```

```
  ENDF
```

```
ELSE
```

```
  IF NbDet = 1 THEN tt := ' un des '+str(Mlo)+' logements que'
```

```
  ELSE tt := ' '+str(NbDet)+' des '+str(Mlo)+' logements que'
```

```
  ENDF
```

```
ENDIF
```

```
Intro
```

2. Remarque : Il est absolument nécessaire de «verrouiller» le choix des logements au début de cette deuxième partie. En effet, si après avoir commencé la partie détaillée on modifie dans la première partie ne serait-ce qu'une caractéristique des logements décrits, les paramètres du tirage des logements vont être modifiés, et éventuellement le choix des logements tirés. Il y a donc un risque que deux variables d'un même block détaillé se réfèrent en fait à deux logements différents ! L'utilisation de l'ordre KEEP et de filtres du type « IF Intro = Empty THEN » fait en sorte que le choix des logements à décrire n'est plus remis en cause une fois la deuxième partie entamée.

3. Avant de poser les questions du block BDetaill (questions beaucoup plus nombreuses que dans le block BIsol: environ 20 minutes d'entretien contre moins de 5), il faut récupérer et noter dans ce block les informations recueillies dans la première partie concernant ce logement.

La difficulté de cette opération vient de ce qu'il n'y a aucun lien systématique entre les rangs des blocks décrivant un même logement entre les parties : le premier logement décrit en détail peut être le second, ou le troisième, décrit dans la première partie. Ce peut être un logement isolé, un appartement situé dans un petit immeuble, ou encore situé dans un grand immeuble.

Pour chaque indice j (j étant égal à 1, puis éventuellement à 2), on va donc d'abord parcourir les indices i (i allant de 1 à Nelig) pour trouver celui qui, à l'issue du tirage, a été choisi pour être décrit en jème (chercher i tel que Achoix[i] = oui et ANum[i] = j).

Si l'indice i concerné est inférieur ou égal au nombre d'isolés éligibles ($i < Mis$), on va aller chercher dans la première partie, parmi les logements isolés, celui qui a été décrit en nème position, n tenant compte des logements non éligibles éventuellement décrits avant ce logement. Ce

sont les caractéristiques de ce logement isolé qui seront affectées au block détaillé j. En particulier, son adresse et sa taille seront indiquées au tout début du block détaillé, lorsqu'il faudra indiquer au ménage sur quel logement portera la description (variable Detail[j].tex).

Si $i > \text{Mis}$, c'est que le logement choisi est issu d'un immeuble. On va alors parcourir les immeubles (z allant de 1 à Nim) pour repérer celui qui contient le logement en question. Une fois l'immeuble repéré, il faut trouver à l'intérieur de l'immeuble l'appartement concerné (indice k). Quand l'immeuble est petit, on est ramené à la situation des logements isolés et la récupération des caractéristiques ne pose pas de difficultés. Quand l'immeuble est grand, on détermine la taille de l'appartement à décrire en fonction de la répartition par tailles donnée dans la première partie et on génère un message du type « Nous allons maintenant décrire le 2ème appartement de 4 pièces de l'immeuble situé.... ».

Une fois ces opérations réalisées, le lancement du block détaillé lui-même est à la portée d'un programmeur BLAISE débutant !

FIELDS

Detail : ARRAY[1..2] Of BDetail

RULES

FOR j:= 1 TO Nbdet DO

{ ** récupération des données de la première partie ** }

FOR i:= 1 TO nelig DO

IF AChoix[i] = oui AND ANum[i] = j THEN

IF $i \leq \text{Mis}$ THEN { données issues de logements isolés }

n := (i+ ANbCorr[i])

Detail[j].num := j

Detail[j].numlo := n

Detail[j].tex := 'Nous allons maintenant décrire '+Isole[n].tex+

Isole[n].ladres+ 'à '+Isole[n].Commune+Isole[n].LCom+'.'

Detail[j].app := Isole[n].Appma

Detail[j].Inopi := Isole[n].Nopi

ELSEIF $i > \text{Mis}$ THEN { données issues d'immeubles }

FOR z := 1 TO nim DO

IF ($i > \text{Immeub}[z].\text{debut}$) AND ($i < (\text{Immeub}[z].\text{debut} + \text{Immeub}[z].\text{Nbelig} + 1)$)

THEN

Detail[j].num := j

Detail[j].numlo := (Mis+z)

k := (i - immeub[z].debut)

IF $\text{Immeub}[z].\text{Inolo} < 10$ THEN { petits immeubles }

FOR y := 1 TO 10 DO

IF $\text{Immeub}[z].\text{Appart}[y].\text{ClassEli} = k$

THEN v := y

ENDIF

ENDDO

IF $\text{Immeub}[z].\text{Appart}[v].\text{INop} = 1$ THEN tx := 'le studio'

ELSE tx := 'l'appartement de+STR(Immeub[z].Appart[v].INop)+' pièces'

ENDIF

Detail[j].tex := 'Nous allons maintenant décrire '+tx+' situé '+

Immeub[z].appart[v].lbatesc + ' de l'immeuble situé '+ Immeub[z].MAdres+ 'à '+

Immeub[z].Commune+

Immeub[z].LCom+'.'

Detail[j].LNopi := Immeub[z].Appart[v].INop

ELSE { grands immeubles }

y := INT(k/Immeub[z].max)

IF $y \leq \text{Immeub}[z].\text{Inop1}$ THEN v:= y tx:= 'studio' Detail[j].LNopi := 1

ELSEIF $y \leq \text{Immeub}[z].\text{Inop22}$ THEN

v:=(y-Immeub[z].Inop1) tx := 'appartement de 2 pièces' Detail[j].LNopi := 2

```

ELSEIF y <= Immeub[z].Inop33 THEN
  v:=(y-Immeub[z].Inop22) tx := 'appartement de 3 pièces' Detail[j].Lnopi := 3
ELSEIF y <= Immeub[z].Inop44 THEN
  v:=(y-Immeub[z].Inop33) tx := 'appartement de 4 pièces' Detail[j].Lnopi := 4
ELSEIF y <= Immeub[z].Inop55 THEN
  v:=(y-Immeub[z].Inop44) tx := 'appartement de 5 pièces' Detail[j].Lnopi := 5
ELSE v:= (y-Immeub[z].Inop55) tx := 'appartement de plus de 5 pièces'
ENDIF
IF (v) = 1 THEN ième := 'er '
  ELSE ième := ' ième '
ENDIF
Detail[j].tex := 'Nous allons maintenant décrire le '+str(v)
+ième+tx+' de l'immeuble situé '+Immeub[z].MAdres+' à '
+Immeub[z].Commune+Immeub[z].LCom+'.'
ENDIF { fin du filtre sur la taille de l'immeuble }

Detail[j].Linet := Immeub[z].Inbet
ENDIF { fin du filtre sur z}
Detail[j].imm := Oui
Detail[j].App := Ap
ENDDO {fin de la boucle sur z}

ENDIF { fin du filtre la différence isolés / immeubles }

ENDIF { fin du filtre sur i = j }

ENDDO {fin de la boucle sur i }

{** Lancement du block de description détaillée **}
Detail[j]

ENDDO {fin de la boucle sur j}

```


Le questionnaire d'une enquête en deux phases avec tirage d'un échantillon représentatif.

François Clanché, INSEE, France

1. Résumé

L'enquête « Propriétaires bailleurs » a été réalisée en 1996/97 par l'INSEE, auprès de 3 000 ménages qui avaient déclaré, au cours de l'enquête nationale Logement, posséder un patrimoine immobilier locatif ou en avoir détenu un depuis 1993. Son but est de faire décrire au ménage l'ensemble des logements locatifs qu'il possède, les revenus qu'il en tire, la façon dont il a constitué son patrimoine et ses projets.

La difficulté technique de cette enquête est la suivante : Les "gros" bailleurs, ceux qui possèdent plus de deux logements, représentent à peine 15 % des ménages propriétaires, mais ils possèdent près de la moitié du parc. On ne peut évidemment pas leur demander de décrire dans le détail la totalité de leur patrimoine, sinon l'enquête durerait plusieurs heures. On a donc mis au point le système suivant : au début du questionnaire, le ménage énumère la totalité des logements qu'il possède et en donne les caractéristiques principales. Avec ces quelques indications, 2 logements "représentatifs" de l'ensemble sont choisis par le logiciel, que le ménage décrit dans le détail. Grâce à cette procédure, l'enquête, qui dure moins d'une demi-heure quand le ménage possède un seul logement, dépasse rarement les 50 minutes, même dans le cas de patrimoines importants.

Dans le programme en Blaise-III, on a procédé en 3 phases :

- la première partie de l'enquête lance des blocks identiques, courts, qui permettent de noter une description sommaire de chaque logement (adresse, date de construction, situation juridique, taille). Pour les appartements situés dans des immeubles, on travaille à un double niveau : on décrit une seule fois l'adresse, la date de construction et la situation juridique du bâtiment, puis on pose, dans les petits immeubles, 3 questions pour chaque appartement (dont la taille et des précisions permettant de retrouver un appartement), et pour les grands immeubles on demande une répartition par taille des appartements. Parmi les logements ainsi repérés, seuls certains sont susceptibles de faire l'objet d'une description détaillée dans la deuxième partie (les logements « éligibles »).

- à la fin de cette première partie, le programme crée en AUXFIELDS une table ayant autant de lignes que de logements éligibles (il y a donc des duplications dans le cas des immeubles) et comportant en colonne les variables caractéristiques de la représentativité souhaitée. Les "individus/logements" sont ensuite triés selon ces caractéristiques grâce à

un algorithme optimisé pour ne pas trop ralentir la collecte sur un ordinateur peu performant. Enfin vient le tirage lui-même par une méthode systématique utilisant la fonction RANDOM.

- Au début de la deuxième partie du questionnaire, le programme récupère les informations de la première partie, essentielles pour filtrer les questions et adapter les textes par la suite. Cette information récupérée permet en particulier d'indiquer au ménage quels logements ont été désignés, avec un message du type "nous allons maintenant décrire l'appartement de 3 pièces 1er étage gauche avenue Charles de Gaulle à Lyon". La difficulté de cette phase vient de ce qu'il n'y a aucun lien systématique entre les rangs des blocks décrivant un même logement entre les parties 1 et 2.

Nous allons examiner plus en détail ces trois phases de l'enquête en nous attachant aux syntaxes Blaise « originales » et à leurs justifications. Les extraits de programme proposés ne sont que des illustrations, extraites d'un programme global beaucoup plus long et complexe, que l'auteur tient à la disposition des personnes intéressées.

2. Première phase : description rapide de tout le patrimoine

Cette phase comporte en fait deux parties : les logements « isolés » et les immeubles.

1. Les logements isolés sont décrits de façon simple dans des blocks de type BIsolé dans lesquels on demande notamment la commune où est situé le logement, son adresse, sa taille, et son occupation actuelle. Ces réponses permettent de renseigner, toujours à l'intérieur du block Isolé, la variable Eligi (qui distingue les logements susceptibles de faire l'objet d'une description détaillée) et la variable Tri qui servira plus tard pour s'assurer de la représentativité de l'échantillon des logements décrits en détail. A la fin de chaque logement « Isolé », la question Autre permet de savoir si on doit une nouvelle fois entamer une description ou si l'on doit passer à la partie Immeubles. Des compteurs de logements décrits (Mlo) et de logement isolés éligibles (Mis) sont incrémentés.

```
FIELDS
{ ** blocks de description des logements ** }
Isole : ARRAY[0..10] OF BIsolé
Mlo   : 0..999 { nbre total de logt repérés }
Mis   : 0..10  { nbre d'isolés éligibles }

RULES

Isole[0]
FOR n:=1 TO 10 DO
  IF Isolé[n-1].Autre = oui
    THEN Isole[n]
      Mlo := Mlo + 1
      IF Isole[n].Eligi = Oui
        THEN Mis := Mis+1
    ENDIF
  ENDIF
ENDDO
```

2. Pour ce qui est des immeubles, le principe général est le même : on lance successivement un ou plusieurs blocks de type Bimme où certaines

questions propres au bâtiment sont similaires au cas isolé. Si l'immeuble comporte moins de 10 appartements, l'occupation de l'appartement et sa taille sont détaillées appartement par appartement (sous-block Immeub.Appart), et on calcule une variable Tri pour chaque appartement. S'il compte plus de 9 appartements, on pose des questions sur la répartition entre les différentes tailles et on génère, toujours à l'intérieur du block, autant de variables Tri qu'il y a d'appartements dans l'immeuble en utilisant les données sur la répartition entre tailles. On calcule également le nombre de logements éligibles de l'immeuble (NbElig).

FIELDS

```
Immeub : ARRAY[0..3] OF Bimme
Elilmm  : 0..400 { nbre d'éligibles en immeubles }
Nim     : 0..3 {nombre d'immeubles repérés }
Neligi  : 0..350 { nbre total de logements éligibles : Mis + Elilmm }
```

RULES

```
Nelig := Mis

Immeub[0]
FOR z:= 1 TO 3 DO
IF Immeub[z-1].Autre = oui THEN
    Immeub[z]
    Nim := Nim + 1
    Mlo := Mlo + Immeub[z].Inolo
    IF Immeub[z].Eligi =Oui THEN
        Elilmm := Elilmm + Immeub[z].NbElig
        Neligi := Neligi + Immeub[z].NbElig
    ENDIF
ENDIF
ENDDO
```

3. Deuxième phase : choix des logements à décrire en détail

1. Pour rendre possible un tri et un tirage parmi tous les logements, on renseigne un vecteur d'auxfields comportant, pour chaque logement éligible (qu'il soit isolé ou situé dans un immeuble), la variable ATri. Cet ARRAY d'auxfields est indicé par i, i allant de 1 à Neligi, et la variable Aclasst est initialisée à i. On remplit cet ARRAY en deux temps, d'abord avec les logements isolés, ensuite avec les appartements.

```

AUXFIELDS
AClasst  : ARRAY[0..350] of 0..350
Anbcorr  : ARRAY[0..350] of 0..350
ATri     : ARRAY[1..350] of 1..7000
Anum     : ARRAY[1..350] of 0..2
AChoix   : ARRAY[1..350] of OuiNonType

```

RULES

```

FOR n := 1 TO 10 DO
  IF Isole[n].Eligi = Oui THEN
    FOR i:= 1 TO Mis DO
      IF Isole[n].k = i THEN
        ATri[i] := Isolé[n].Tri
        AClasst[i] := i
        ANbCorr[i] := Isolé[n].Nbc
      ENDIF
    ENDDO
  ENDIF
ENDDO

IF Nim > 0 THEN
  FOR z:= 1 to Nim DO
    IF Immeub[z].Eligi = Oui THEN
      For i:= Immeub[z].Debut TO Nelig DO
        IF i > (Immeub[z].Debut)
          AND i < (Immeub[z].Debut + Immeub[z].NbElig + 1)
          THEN
            AClasst[i] := i
            w := (i - Immeub[z].Debut)
            ATri[i] := Immeub[z].Tri[w]
          ENDIF
        ENDDO
      ENDDO
    ENDIF
  ENDDO
ENDIF

```

2. Tri des logements : un algorithme fait en sorte que la variable AClasst de chacun des Nelig logements corresponde au classement selon ATri. Pour cela on utilise la méthode de l'insertion successive des individus/logements, dont les principes sont les suivants:

Au début de chaque étape les (i-1) premiers individus sont triés, et on introduit le ième individu avec le classement i.

On compare la valeur de la variable ATri de l'individu i avec celle de l'individu classé (i-1)ème, c'est-à-dire dernier à l'issue de l'étape précédente.

Si elle est supérieure, le classement est alors entièrement bon et on peut passer à la phase suivante (introduction du (i+1)ème individu).

Sinon, on inverse les classements des deux derniers individus (i est classé (i-1)ème), et on compare son ATri avec celui de l'individu classé (i-2)ème, etc...

On ne passe à la phase suivante que quand l'individu i a trouvé sa « bonne » place.

```

RULES
FOR i:=2 TO nelig DO { évolution du niveau d'entrée : étapes }
  h := i -1
  FOR k:=1 TO h DO { évolution de la profondeur de comparaison }
    FOR q := 1 TO h DO { recherche du j avec qui comparer }
      IF AClasst[q] = (AClasst[i] - 1) THEN
        IF (ATri[q] > Atri[i])
          THEN devant := Aclasst[q]
              Aclasst[q] := Aclasst[i]
              AClasst[i] := devant
        ENDIF
      ENDIF
    ENDDO
  ENDDO
ENDDO

```

Remarque : la syntaxe de l'algorithme est ici simplifiée. Pour gagner en performance on introduit des tests qui limitent le nombre de boucles parcourues aux seules comparaisons pertinentes.

3. Le choix des logements à décrire se fait ensuite par un tirage systématique, dont le pas est calculé en fonction du nombre de logements éligibles et le niveau de départ tiré au hasard par la fonction RANDOM. Notons la nécessité de « fixer » le valeur de Alea en utilisant l'ordre KEEP pour éviter qu'elle se modifie à tout moment.

Toujours dans la table d'auxfields, on parcourt l'ensemble des indices i de 1 à Nelig, et les logements « choisis » se distinguent par des variables Achoix renseignée à oui, et leur ordre de tirage est indiqué dans la variable Anum.

Si il y a moins de 3 logements éligibles, le nombre de logements détaillés décrits (NbDet) est égal à Nelig et le renseignement de Achoix et de Anum est plus aisé.

```

FIELDS
Alea , Pas , Debut : REAL , EMPTY
NbDet : 0..2 {nombre de descriptions détaillées}
Deta : ARRAY[1..2] of 0..350 {numéros des logements choisis}

RULES

{ ** Tirage de alea ** }
Alea.Keep
IF Alea = Empty THEN
  Alea := Random
ENDIF

{ ** calcul du nombre de blocks détaillés nbdet ** }
IF Nelig < 3 THEN NbDet := Nelig
  ELSE NbDet := 2
ENDIF

{ ** détermination des logements à décrire ** }
IF nelig < 3 THEN
  FOR i:= 1 TO nelig DO
    AChoix[i]:= oui Anum[i] := Aclasst[i]
    IF Deta[1] <> RESPONSE THEN Deta[1] := i
    ELSE Deta[2] := i
  ENDIF
ENDDO
ELSE
  Aclasst[0]:=0
  Pas := (Nelig/2)
  Debut := Pas*Alea
  FOR i:= 1 TO nelig DO
    IF ((Aclasst[i]-1 <= Debut) AND ( Debut < Aclasst[i]))
      THEN AChoix[i]:= oui Anum[i]:=1 deta[1] := i
    ELSEIF ((Aclasst[i]-1 <=(Debut+pas)) AND ((Debut+pas) < Aclasst[i]))
      THEN AChoix[i]:= oui Anum[i]:=2 Deta[2] := i
    ELSE AChoix[i]:= non Anum[i] := 0
  ENDIF

```

```
ENDDO
ENDIF
```

4. Troisième phase : lancement des descriptions détaillées

1. Une message paramétré indique à l'enquêté (et à l'enquêteur) le nombre de logements qu'il va maintenant avoir à décrire en détail.

```
FIELDS
```

```
Intro "Nous allons décrire en détail ^tt vous venez d'évoquer. ": STRING[1]
```

```
RULES
```

```
IF NbDet = Mlo THEN
  IF Mlo = 1 THEN tt := ' le logement que'
  ELSE tt := ' les '+str(NbDet)+' logements que'
  ENDIF
ELSE
  IF NbDet = 1 THEN tt := ' un des '+str(Mlo)+' logements que'
  ELSE tt := ' '+str(NbDet)+' des '+str(Mlo)+' logements que'
  ENDIF
ENDIF
Intro
```

2. Remarque : Il est absolument nécessaire de «verrouiller» le choix des logements au début de cette deuxième partie. En effet, si après avoir commencé la partie détaillée on modifie dans la première partie ne serait-ce qu'une caractéristique des logements décrits, les paramètres du tirage des logements vont être modifiés, et éventuellement le choix des logements tirés. Il y a donc un risque que deux variables d'un même block détaillé se réfèrent en fait à deux logements différents ! L'utilisation de l'ordre KEEP et de filtres du type « IF Intro = Empty THEN » fait en sorte que le choix des logements à décrire n'est plus remis en cause une fois la deuxième partie entamée.

3. Avant de poser les questions du block BDetaill (questions beaucoup plus nombreuses que dans le block BIsol: environ 20 minutes d'entretien contre moins de 5), il faut récupérer et noter dans ce block les informations recueillies dans la première partie concernant ce logement.

La difficulté de cette opération vient de ce qu'il n'y a aucun lien systématique entre les rangs des blocks décrivant un même logement entre les parties : le premier logement décrit en détail peut être le second, ou le troisième, décrit dans la première partie. Ce peut être un logement isolé, un appartement situé dans un petit immeuble, ou encore situé dans un grand immeuble.

Pour chaque indice j (j étant égal à 1, puis éventuellement à 2), on va donc d'abord parcourir les indices i (i allant de 1 à Nelig) pour trouver celui qui, à l'issue du tirage, a été choisi pour être décrit en jème (chercher i tel que Achoix[i] = oui et ANum[i] = j).

Si l'indice i concerné est inférieur ou égal au nombre d'isolés éligibles ($i < Mis$), on va aller chercher dans la première partie, parmi les logements isolés, celui qui a été décrit en nème position, n tenant compte des logements non éligibles éventuellement décrits avant ce logement. Ce

sont les caractéristiques de ce logement isolé qui seront affectées au block détaillé j. En particulier, son adresse et sa taille seront indiquées au tout début du block détaillé, lorsqu'il faudra indiquer au ménage sur quel logement portera la description (variable Detail[j].tex).

Si $i > \text{Mis}$, c'est que le logement choisi est issu d'un immeuble. On va alors parcourir les immeubles (z allant de 1 à Nim) pour repérer celui qui contient le logement en question. Une fois l'immeuble repéré, il faut trouver à l'intérieur de l'immeuble l'appartement concerné (indice k). Quand l'immeuble est petit, on est ramené à la situation des logements isolés et la récupération des caractéristiques ne pose pas de difficultés. Quand l'immeuble est grand, on détermine la taille de l'appartement à décrire en fonction de la répartition par tailles donnée dans la première partie et on génère un message du type « Nous allons maintenant décrire le 2ème appartement de 4 pièces de l'immeuble situé.... ».

Une fois ces opérations réalisées, le lancement du block détaillé lui-même est à la portée d'un programmeur BLAISE débutant !

FIELDS

Detail : ARRAY[1..2] Of BDetail

RULES

FOR j:= 1 TO Nbdet DO

{ ** récupération des données de la première partie ** }

FOR i:= 1 TO nelig DO

IF AChoix[i] = oui AND ANum[i] = j THEN

IF $i \leq \text{Mis}$ THEN { données issues de logements isolés }

n := (i+ ANbCorr[i])

Detail[j].num := j

Detail[j].numlo := n

Detail[j].tex := 'Nous allons maintenant décrire '+Isole[n].tex+

Isole[n].ladres+ 'à '+Isole[n].Commune+Isole[n].LCom+'.'

Detail[j].app := Isole[n].Appma

Detail[j].Inopi := Isole[n].Nopi

ELSEIF $i > \text{Mis}$ THEN { données issues d'immeubles }

FOR z := 1 TO nim DO

IF ($i > \text{Immeub}[z].\text{debut}$) AND ($i < (\text{Immeub}[z].\text{debut} + \text{Immeub}[z].\text{Nbelig} + 1)$)

THEN

Detail[j].num := j

Detail[j].numlo := (Mis+z)

k := (i - immeub[z].debut)

IF $\text{Immeub}[z].\text{Inolo} < 10$ THEN { petits immeubles }

FOR y := 1 TO 10 DO

IF $\text{Immeub}[z].\text{Appart}[y].\text{ClassEli} = k$

THEN v := y

ENDIF

ENDDO

IF $\text{Immeub}[z].\text{Appart}[v].\text{INop} = 1$ THEN tx := 'le studio'

ELSE tx := 'l'appartement de+STR(Immeub[z].Appart[v].INop)+' pièces'

ENDIF

Detail[j].tex := 'Nous allons maintenant décrire '+tx+' situé '+

Immeub[z].appart[v].lbatesc + ' de l'immeuble situé '+ Immeub[z].MAdres+ 'à '+

Immeub[z].Commune+

Immeub[z].LCom+'.'

Detail[j].LNopi := Immeub[z].Appart[v].INop

ELSE { grands immeubles }

y := INT(k/Immeub[z].max)

IF $y \leq \text{Immeub}[z].\text{Inop1}$ THEN v:= y tx:= 'studio' Detail[j].LNopi := 1

ELSEIF $y \leq \text{Immeub}[z].\text{Inop22}$ THEN

v:=(y-Immeub[z].Inop1) tx := 'appartement de 2 pièces' Detail[j].LNopi := 2


```

ELSEIF y <= Immeub[z].Inop33 THEN
  v:=(y-Immeub[z].Inop22) tx := 'appartement de 3 pièces' Detail[j].Lnopi := 3
ELSEIF y <= Immeub[z].Inop44 THEN
  v:=(y-Immeub[z].Inop33) tx := 'appartement de 4 pièces' Detail[j].Lnopi := 4
ELSEIF y <= Immeub[z].Inop55 THEN
  v:=(y-Immeub[z].Inop44) tx := 'appartement de 5 pièces' Detail[j].Lnopi := 5
ELSE v:= (y-Immeub[z].Inop55) tx := 'appartement de plus de 5 pièces'
ENDIF
IF (v) = 1 THEN ième := 'er '
  ELSE ième := ' ième '
ENDIF
Detail[j].tex := 'Nous allons maintenant décrire le '+str(v)
+ième+tx+' de l'immeuble situé '+Immeub[z].MAdres+' à '
+Immeub[z].Commune+Immeub[z].LCom+'.'
ENDIF { fin du filtre sur la taille de l'immeuble }

Detail[j].Linet := Immeub[z].Inbet
ENDIF { fin du filtre sur z}
Detail[j].imm := Oui
Detail[j].App := Ap
ENDDO {fin de la boucle sur z}

ENDIF { fin du filtre la différence isolés / immeubles }

ENDIF { fin du filtre sur i = j }

ENDDO {fin de la boucle sur i }

{** Lancement du block de description détaillée **}
Detail[j]

ENDDO {fin de la boucle sur j}

```