

REALISATION D'UN JEU D'ATTENTION

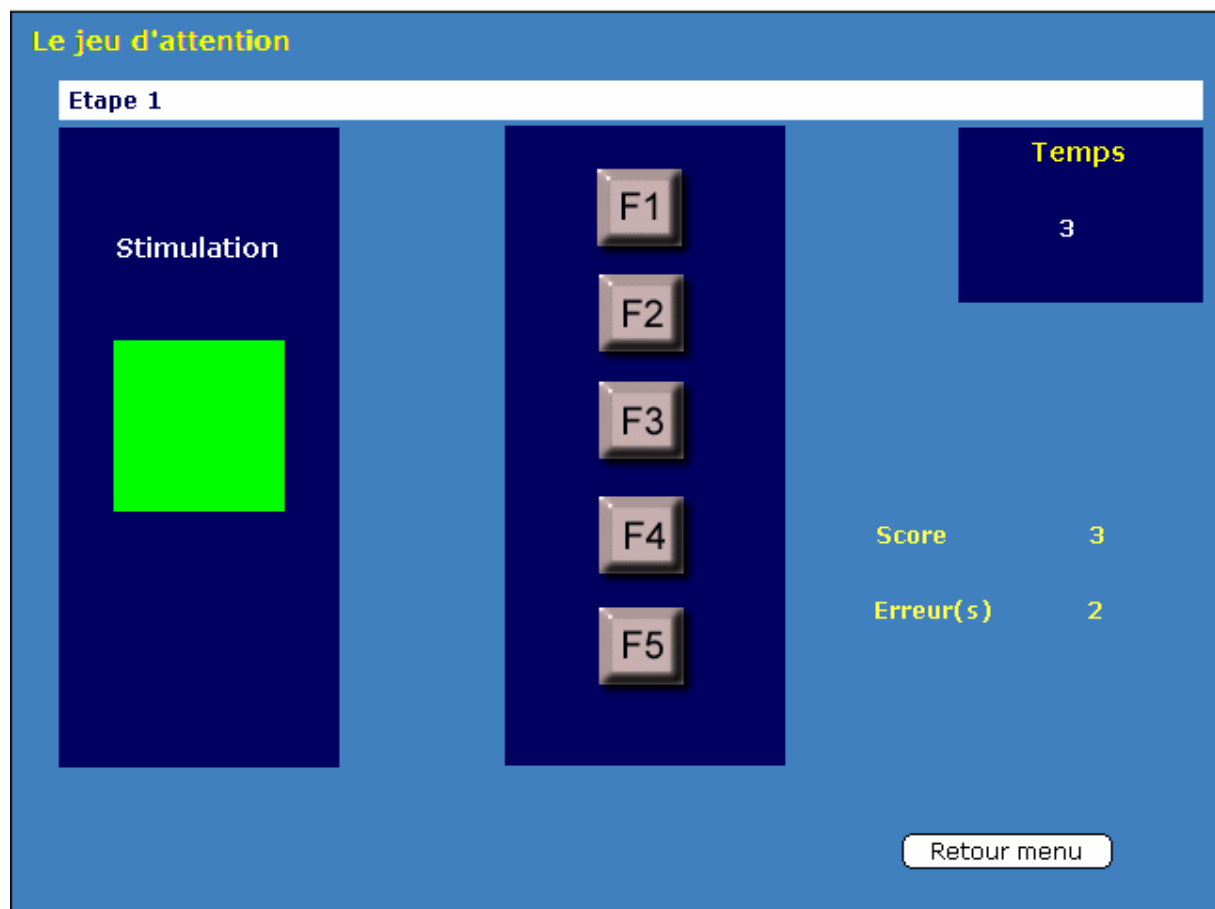


TABLE DES MATIERES

1. Cahier des charges.....	3
2. Etapes de la réalisation.....	4
2.1. Interface du jeu.....	4
2.2. Scripts d'image.....	5
2.3. Scripts de frames.....	6
3. Problèmes rencontrés.....	9
3.1. Différentes versions de Director.....	9
3.2. Définition du sujet.....	9
3.3. Prise en compte des touches.....	9
3.4. Passage entre les étapes.....	10
4. Conclusion.....	11

1. Cahier des charges

L'objectif du projet est de créer un jeu d'attention afin éventuellement de l'intégrer dans une suite complète de jeux sur CD-ROM tels que le jeu des images, le jeu de mémoire, le jeu des bacs et le jeu d'observation.

Le jeu d'attention consiste à faire associer par l'utilisateur des couleurs tirées au hasard à des touches définies préalablement. Ce jeu se déroule sur plusieurs étapes avec un score récapitulatif à la fin du jeu. Ce score pourra être visualisé de manière partielle par l'utilisateur (détail de chaque étape à la fin) et de manière globale (score total).

La correspondance entre les couleurs stimulées et les touches est la suivante :

Stimulations	Réponse/touche
Bleu	F1
Vert	F2
Rouge	F3
Jaune	F4
Blanc	F5

Ce jeu comporte un total de 8 étapes, chacune comportant elle-même 8 tests. Au départ l'utilisateur possède 10 secondes pour répondre correctement ; au-delà une erreur est comptabilisée. A chaque étape, donc au bout de 8 tests réussis, le temps imparti pour répondre diminue d'une seconde, pour finir à 3 secondes à la fin du jeu, le rendant ainsi plus difficile au fur et à mesure de la progression.

Mais l'utilisateur ne possède le droit qu'à 3 erreurs consécutives par étape ; à la troisième, le jeu est arrêté et les résultats sont affichés. Si l'utilisateur réussit les 8 tests de chaque étape sans faire plus de deux erreurs pendant l'étape, il parvient au bout du jeu, à savoir des 8 étapes.

Au tout début du jeu, l'utilisateur peut visualiser ce quelques règles à travers une animation pas à pas pour lui expliquer le fonctionnement du jeu.

Le jeu sera réalisé avec le logiciel Macromedia Director, donc avec des scripts en langage Lingo.

2. Etapes de la réalisation

La réalisation du jeu comporte plusieurs étapes ; l'interface graphique, les scripts d'image et scripts de comportement. Concernant les étapes, nous en avons d'abord créée une qui fonctionnait correctement, ensuite on l'a copiée 8 fois et enfin nous avons adaptés les scripts pour les variables propres à chaque étape et assurer l'enchaînement correct des étapes entre elles au cours du jeu.

2.1. Interface du jeu

Elle est conforme au cahier des charges, avec la même charte graphique et la même disposition. Elle fut rapide à réaliser car nous avions déjà un modèle ; cependant la prise en main du logiciel nous posa quelques problèmes, du moins au début car ce logiciel est moins intuitif que Macromedia Flash, par exemple. Les images correspondant aux touches F1, F2, F3, F4 et F5 qui permettent au joueur de faire son choix étant déjà réalisées, nous n'avions plus qu'à les importer dans Director en tant que bitmaps. L'interface se présente donc de la manière suivante :

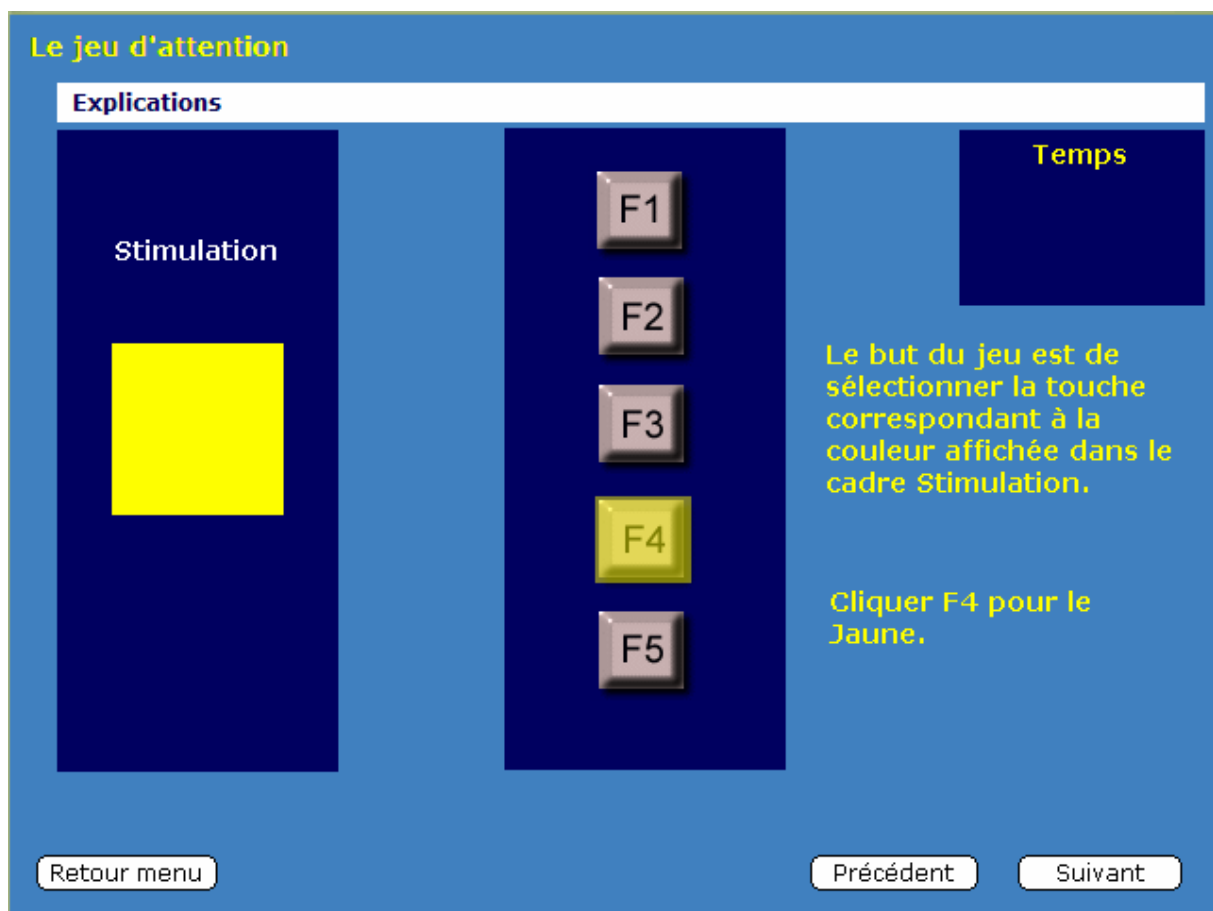


Figure 1 : Phase de démonstration.

2.2. Scripts d'image

Ce type de script regroupe les scripts qui concernent les boutons et images, tels que les boutons « Commencer », « Explications » ou « Retour menu » et les images F1...F5.

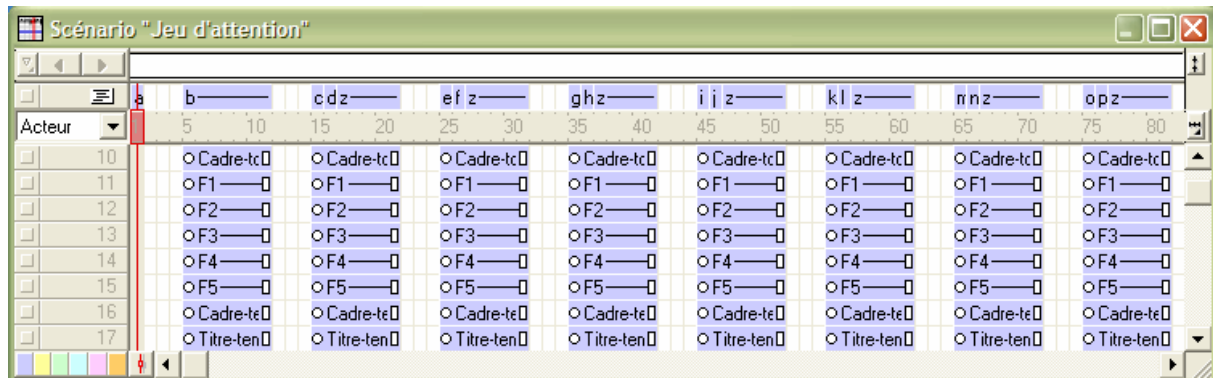


Figure 2 : Scénario du jeu avec les références aux scripts.

Concernant les scripts des touches F1 à F5, ils se ressemblent structurellement donc on ne va en expliquer qu'un, celui de la touche F1.

```
global etape          --correspond au score de l'étape en cours
global erreur        --comptabilise le nombre d'erreurs de l'étape en cours
global debut         --numéro d'une frame
on mouseDown me
  if the frame = debut+1 then
    etape = etape+1
    go to debut
  else
    alert ("Mauvaise Réponse.")
    erreur = erreur+1
    go to debut
  end if
end
```

Ce script vérifie si la touche cliquée correspond à la bonne image (en l'occurrence bleu = F1) ; si c'est le cas alors, on incrémente le score de l'étape en cours de 1 ; sinon on incrémente le nombre d'erreurs de 1. Dans tous les cas le script renvoie l'animation à la frame « debut », soit la frame de départ de l'étape en cours (ex : pour l'étape 1 la frame de départ correspond à 16). Pour les autres scripts des touches F1 à F5, seules la condition change, puisque par exemple pour F2 on aura « if the frame = debut+2 ... ».

Concernant le bouton « retour menu » ; il permet d'arrêter le jeu et de retourner au menu de départ, son script associé est donc simple :

```
on mouseDown me
  go to 1
```

end

De même, pour le bouton « commencer le niveau », on obtient ceci :

```
global erreur      --comptabilise le nombre d'erreurs de l'étape en cours
global test        --comptabilise le nombre de tests réalisés
global etape       --comptabilise le score de l'étape en cours
global debut       --numéro d'une frame

on mouseDown me
  erreur = 0
  etape = 0
  test = 1
  go to random(5)+debut
end
```

On réinitialise les variables globales à chaque début d'étape ; on laisse ainsi l'utilisateur choisir de commencer afin qu'il se concentre correctement.

2.3. Scripts de frames

Ce type de script regroupe les scripts qui concernent les frames, soit les scripts « a » à « z » et « 0 ». Les scripts « a » et « b » font la même chose ; ils attendent dans la frame en cours de lecture jusqu'à ce que l'utilisateur clique sur un bouton pour exécuter un script d'image :

```
on exitFrame me
  go to the frame
end
```

Les scripts « c, e, g, i, k, m, o, q » permettent d'attendre une action sur un bouton de l'utilisateur ainsi que de fixer les variables globales « seconde » et « debut » qui définissent à chaque étape le temps alloué pour la réponse et la frame de départ de l'étape. Par exemple le script « k » commence l'étape 5 :

```
global seconde     --définition du temps de réponse
global debut       --définition de la frame de départ

on exitFrame me
  seconde = 6       --le joueur possède 6 secondes pour chaque réponse
  debut = 56        --la frame de départ de l'étape 5 est la #56
  go to the frame
end
```

Les scripts « d, f, h, j, l, n, p, r » permettent d'une part de vérifier que le nombre d'erreurs consécutives dans une même étape reste inférieur à 3 (sinon le jeu s'arrête et on passe au récapitulatif des scores) et d'autre part que le nombre de tests effectués est

inférieur à 8 (sinon on passe à l'étape suivante). Cette structure de script se charge aussi du tirage au hasard des couleurs à stimulées dans le cas où le nombre d'erreurs consécutives est inférieur à 3 et le nombre de tests effectués inférieur à 8. Par exemple le script « l » se charge de l'étape 5 :

```
global erreur      --nombre d'erreurs consécutives
global test        --nombre de tests effectués
global debut       --frame de départ
global etape5       --score de l'étape 5
global etape        --score de l'étape en cours

on exitFrame me
  delay 60
  if (erreur > 2) then
    etape5 = etape      --le score de l'étape 5 est celui en cours
    go to 95            --si on a plus de 2 erreurs on va à la frame 95 (scores)
  else
    if test < 8 then
      go to random (5)+debut --tirage de la couleur
      test = test +1        --incréméntation du nombre de tests
    else
      etape5 = etape      --le score de l'étape 5 est celui en cours
      go to debut +9      --on va à l'étape suivante
    endif
  end if
end
```

Le script « z » est identique pour toutes les étapes ; il permet notamment l'affichage du score de l'étape en cours, du nombre d'erreurs consécutives et du temps restant. Il gère aussi le décompte du temps, sachant qu'à la fin de celui-ci il est compté une erreur pour le joueur qui n'a pas répondu à temps :

```
global erreur      --nombre d'erreurs dans l'étape en cours
global test        --nombre de tests effectués dans l'étape en cours
global debut       --frame de départ de l'étape en cours
global seconde      --temps imparti pour répondre
global etape        --score de l'étape en cours

-- se lance quand on arrive pour la 1ere fois sur le frame
on beginSprite
  member("temps").text = string(seconde) --affiche le temps restant en sec.
  StartTimer              --lance le chronomètre
  member("score").text = string(etape)   --affiche le score courant
  member("mauvais").text = string(erreur) --affiche le nombre d'erreurs
end

on exitFrame me
  member("temps").text = string(seconde -(the timer / 60))
  if (the timer < 60 * seconde) then
    go to the frame      --lorsque le temps est écoulé, on stoppe
  else
    alert "Absence de réponse." --on alerte le joueur
    go to debut          --on revient au début de l'étape
    erreur = erreur +1   --on incrémente le nombre d'erreurs
  end if
end
```

Pour le dernier script, le script “s”, c’est très simple, on affiche les scores partiels de chaque étape ainsi que le score total, somme des autres afin que le joueur puisse l’évaluer par rapport à la moyenne qui est 32 :

```
global etape1      --score de l'étape 1
global etape2      --score de l'étape 2
global etape3      --score de l'étape 3
global etape4      --score de l'étape 4
global etape5      --score de l'étape 5
global etape6      --score de l'étape 6
global etape7      --score de l'étape 7
global etape8      --score de l'étape 8

on exitFrame me
  member("et1").text = string(etape1)
  member("et2").text = string(etape2)
  member("et3").text = string(etape3)
  member("et4").text = string(etape4)
  member("et5").text = string(etape5)
  member("et6").text = string(etape6)
  member("et7").text = string(etape7)
  member("et8").text = string(etape8)
  member("tot").text =
    string(etape1+etape2+etape3+etape4+etape5+etape6+etape7+etape8)
  go to the frame
end
```

Ensuite on attend que le joueur décide de revenir au menu principal.

3. Problèmes rencontrés

3.1. Différentes versions de Director

La majorité du projet ayant été développée sur nos ordinateurs personnels, il fallait que nous cherchions une version d'essai de Macromedia Director. Nous avons utilisé la version 8.5, la version 8F installée à l'UTBM n'étant pas téléchargeable. Il en résulte un problème de compatibilité majeur, à savoir que le fichier créé sous Director 8.5 (chez nous) ne peut être ouvert pas Director 8F (UTBM). Cela nous a obligé de travailler exclusivement chez nous et à apporter un ordinateur portable pour soutenir le projet.

3.2. Définition du sujet

Dans le sujet il est indiqué que la difficulté doit logiquement augmenter au fur et à mesure que l'on progresse dans le jeu. En revanche, il est précisé au départ que le temps de réponse serait fixé à 5 secondes, ce qui rend impossible la graduation de la difficulté au cours du jeu, celle-ci ne s'effectuant à notre avis que sur la durée du temps de réponse qui doit diminuer. Si on la fixe constante, on ne pourra rendre le jeu plus dur ; c'est pour cette raison que nous avons défini un temps de réponse variable entre 10 et 3 secondes suivant les étapes.

3.3. Prise en compte des touches

Dans le sujet de départ il est précisé que l'utilisateur doit répondre par le biais des touches F1, F2, F3, F4 et F5 en fonction de la couleur de la stimulation. Ceci pose quelques problèmes.

Tout d'abord, la touche F1 est réservée pour un raccourci vers l'Aide de Director. Il est donc difficile (nous n'avons pas trouvé la fonction Lingo adéquate) de désaffecter cette touche pour l'utiliser librement. D'autre part, nous n'avons pas réussi à utiliser la fonction keypressed, même avec l'utilisation de l'aide de Director.

De plus, il paraît plus ergonomique d'avoir à cliquer sur les touches affichées à l'écran (utilisation de la souris plus simple) que d'avoir à appuyer sur des touches voisines les unes des autres (risques d'erreurs).

3.4. Passage entre les étapes

Au moment de tester le jeu, nous avions un problème. En effet, quelque soit la réponse donnée à la première stimulation de la première étape, le jeu passait directement à la seconde étape, sans refaire un test que la réponse soit bonne ou mauvaise. Pour la seconde étape, le jeu marchait correctement et dès la troisième, on revenait à la deuxième étape.

Le problème (qui nous a longtemps occupé !) venait en fait du script de comportement « e » que nous avons mis par erreur sur l'image contenant les touches F1 à F5. Le script réinitialisant la variable globale « debut » à 26, à chaque clic sur une des touches F1...F5, nous cliquions aussi sur l'image de derrière, donc on revenait systématiquement au cadre 26, soit le début de l'étape 2.

4. Conclusion

Ce projet nous a permis de découvrir et mieux comprendre le fonctionnement du logiciel Director (moins répandu que Flash) et de son langage propriétaire, le Lingo. Contrairement à d'autres langages, comme C, C++ ou Java, le Lingo est un langage événementiel, ce qui le rend à la fois simple et complexe à manipuler. D'une part, simple car les scripts générés sont courts (seulement quelques lignes à la fois) et d'autre part, complexes car on a plusieurs types de scripts et il faut les placer aux bons endroits aux bons moments, comme sur certaines images et pas d'autres par exemple.

Flash étant plus connu, il est intéressant de voir les différences entre ces deux logiciels qui servent tous deux à réaliser des animations. Leurs langages, certes différents (Actionscript et Lingo) sont assez proches ; en revanche leur placement sur des objets diffère. Pour ce jeu d'attention, une animation pas trop avancée, on obtient en espace disque 480 Ko pour le code source ce qui est assez réduit. En revanche, lors de l'exportation, le fichier final (un exécutable) pèse près de 3 Mo, ce qui l'élimine pour d'éventuelles applications web mais le destine à des applications CD-ROM du fait de ses possibilités plus étendues que Flash. Le logiciel Flash permet de faire des animations moins poussées que Director, mais plus légères. On peut donc les intégrer sans problème à un site web voire même concevoir entièrement un site web avec Flash, puisqu'il utilise des graphismes vectoriels dans ses animations (redimensionnement automatique, agrandissement...).