

Projet de programmation C

Intervenants : M. FRANÇOIS & E. AHVAR & N. KHODOR & S. SAMAD & A. BRIERE & S. ZOGHLAMI

--- MEMORY SONORE ---

1 Présentation générale

L'objectif de ce projet est de modéliser en C, un jeu de mémoire sonore. Le projet est à réaliser obligatoirement en binôme sous environnement GNU/Linux. L'affichage doit être effectué uniquement sur console. La FIG 1 montre un exemple réalisé avec un affichage sur le terminal.



FIGURE 1 – Exemple de réalisation avec un affichage sur le terminal.

Comme on peut le constater, il y a quatre catégories de sons :

1. **ANIMAUX** : oiseau, cheval, éléphant, dromadaire, etc.
2. **OBJETS** : téléphone, marteau, télé, radio, etc.
3. **SPORT** : football, basketball, natation, boxe, etc.
4. **VOYAGE** : voiture, camion, hélicoptère, avion, etc.

Dans chaque catégorie, on présente les émojis dont les sons sont disponibles. Sous chaque émoji, on trouve le numéro de ce dernier.

La règle du jeu est simple : l'ordinateur va faire écouter au joueur une suite de sons choisis aléatoirement parmi les catégories proposées. Le joueur doit mémoriser l'ordre des sons pendant l'écoute. L'objectif pour le joueur est de proposer un classement de sons, afin de retrouver exactement l'ordre dans lequel il avait écouté ces sons.

Pour choisir le son correspondant à un élément, il suffit de saisir son numéro puis taper la touche entrée pour que l'élément s'ajoute au classement. Vous pouvez aussi modifier votre classement en cours dans le cas où vous souhaiteriez replacer/supprimer le son d'un élément.

Une fois votre classement donné, l'ordinateur indiquera les bonnes et les mauvaises cases (conformément à la FIG 1).

Au lancement du jeu, le joueur a la possibilité de choisir entre les modes suivants :

- **FACILE** : 3 sons à retrouver ;
- **NORMAL** : 5 sons à retrouver ;
- **DIFFICILE** : 10 sons à retrouver.

À la fin d'une partie le joueur doit avoir la possibilité de rejouer ou de quitter définitivement le jeu.

2 Réalisation technique

La première des choses à faire est de dessiner la maquette de votre jeu dans un fichier ".txt". La maquette est chargée depuis ce fichier ".txt" et affichée classiquement sur la sortie standard. Pour cela vous pouvez utiliser l'instruction `system`, permettant de lancer une commande shell depuis votre fichier source. Lors de la construction de votre maquette, vous pouvez utiliser les liens suivants pour récupérer les caractères ascii étendus pour les contours ou même les émojis correspondant à vos éléments de sons :

```
http://www.theasciicode.com.ar/  
https://fr.piliapp.com/twitter-symbols/
```

Il suffit de cliquer sur le symbole souhaité, le copier puis le coller simplement dans votre fichier source. Si le symbole ne s'affiche pas correctement sur le terminal, vous devez installer le package "ttf-ancient-fonts" via la commande :

```
sudo apt install ttf-ancient-fonts
```

Concernant les sons, vous pouvez utiliser la librairie SOX. Pour l'installer saisissez les commandes suivantes :

```
sudo apt update  
sudo apt install libsox*
```

Par exemple, pour lancer le son "dromadaire.mp3", on utilise dans le code source l'instruction :

```
system("play -q 'dromadaire.mp3' &");
```

Le jeu étant axé sur la mémoire sonore, une partie significative du travail sera consacrée à cher-

cher sur internet les fichiers sons des éléments présents dans chaque catégorie.

Lors de votre classement, vous aurez besoin de placer le curseur dans les différentes zones appropriées pour :

- saisir du texte;
- afficher du texte;
- afficher les émojis dans les cadres réservés.

Pour cela vous pouvez utiliser l'instruction suivante :

```
printf("\033[%d;%dH", X, Y); /*Pour placer le curseur à la position (X,Y)
                               sur le terminal*/
```

Après tout ce que vous allez afficher (texte, emoji, etc.) se placera juste après le curseur. Si vous voulez effacer du texte sur le terminal, il suffit de replacer le curseur au début de celui-ci et d'afficher autant d'espaces que nécessaire.

Concernant les éléments, vous pouvez utiliser par exemple des tableaux 2D pour stocker tous les emojis. Comme on a quatre catégories, 4 tableaux 2D suffisent. Des tableaux 1D pour les sons choisis aléatoirement par l'ordinateur et également pour votre classement. Cela sera nécessaire lors de la vérification de votre classement. Vous pouvez aussi faire selon vos envies.

3 Comment rendre le projet

Le projet est à réaliser obligatoirement en binôme de même classe sous environnement GNU/Linux, et doit être déposé sur la plate-forme pédagogique *Moodle* au plus tard le **14/01/2021 à 23h59**, sous la forme d'une archive `.zip` à vos noms et prénoms (*i.e.* `NOM_prenom.zip`), contenant tous les fichiers sources du projet. Un lien sera créé pour chacune des classes.

4 Notation

L'évaluation sera globalement scindée en plusieurs parties :

1. une soutenance individuelle et en binôme;
2. la structuration de votre code source (fonctions, tableaux, code commenté, indenté, etc.);
3. le design de votre jeu (affichage maquette, menu, etc.);
4. test du mode **FACILE**;
5. test du mode **NORMAL**;
6. test du mode **DIFFICILE**.