

JEU D'ESCRIME

Novembre 2022

1 Informations

L'objectif de ce projet est de réaliser un programme permettant de jouer à un jeu d'escrime avec deux joueurs en local, sur un écran de terminal. Le nombre d'images par seconde est modifiable avec une option.

Le jeu se déroule dans une scène qui peut être modifiée par le joueur par l'intermédiaire de fichiers. (La scène peut être uniquement lue avec un fichier d'extension ".ffscene".) Le fichier doit contenir une seule ligne avec les caractères suivants : "1" (représentant le premier joueur), "2" (représentant le deuxième joueur), "_" (représentant le sol) et "x" (représentant un obstacle).

*Exemple. Un fichier avec le nom "scene.ffscene" avec le contenu
"_____1____x____2_____" est un fichier valide.*

Chaque joueur possède plusieurs attributs différents permettant de réaliser différentes actions :

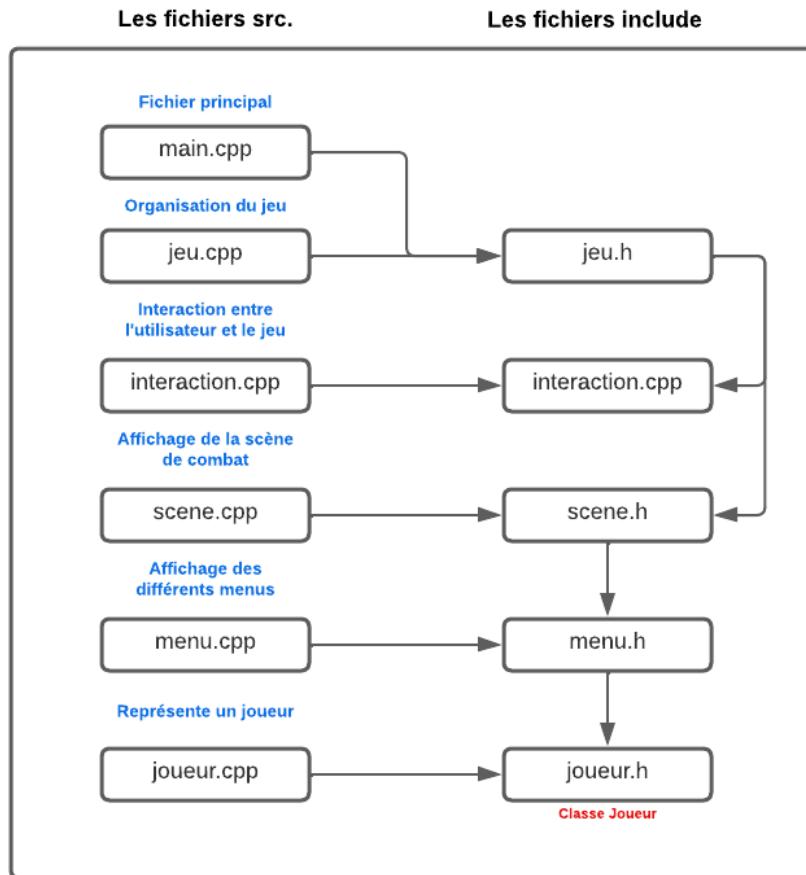
- **movement_speed** : la vitesse de mouvement.
- **attacking_range** : la portée d'attaque.
- **attacking_speed** : la durée d'attaque..
- **defending_range** : la portée de défense.
- **block_time** : la durée du blocaque.

Durant le combat, le joueur peut effectuer les mouvements suivants : se déplacer vers la gauche, se déplacer vers la droite, sauter par dessus un obstacle vers la droite, sauter par dessus un obstacle vers la gauche, attaquer et

bloquer une attaque. Les mouvements mettront **movement_speed** images pour s'activer et les attaques (resp. le blocage des coups) durera pendant **attacking_speed** (resp. **block_time**).

Le programme à réaliser devra être capable de répondre à toutes ces indications.

2 Présentation du projet



Le fichier principal de ce projet est le fichier **main.cpp** qui exécute le jeu. Voici un rappel des commandes principales (que vous pouvez également trouver dans le README.md) :

- Pour compiler : make
- Pour exécuter : ./main [ARG1]

ARG1 est la seule option, c'est le **nombre d'image par seconde**.

2.1 Description des fichiers

- **conio.cpp** : le code source de ce fichier est sur internet. Il implémente une seule fonction **kbhit** qui vérifie si une touche du clavier a été pressée. Cette fonction est très utile : dans notre jeu, l'utilisateur va communiquer avec le terminal avec l'aide du clavier. Il est donc impératif d'avoir une telle fonction.

- **interaction.cpp** : ce fichier représente l'interaction entre les joueurs et le jeu. Ses fonctions peuvent détecter et comprendre la touche du clavier qui a été pressée (à l'aide de **conio.h**).

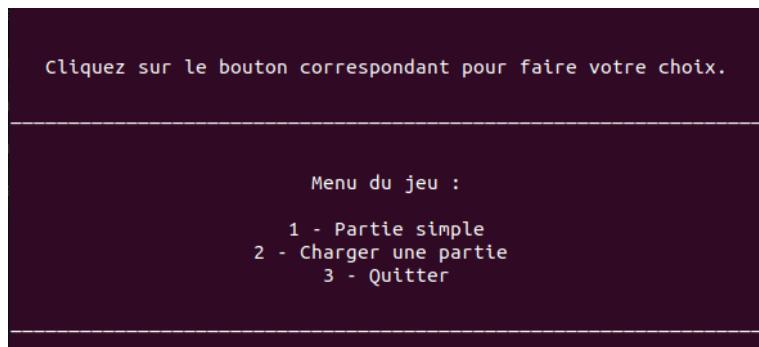
- **joueur.cpp** : un jeu est composé de plusieurs joueurs : ce fichier est donc la représentation d'un joueur sur le terminal (*classe Joueur*). Cette classe possède plusieurs attributs qui décrivent l'état d'un joueur en particulier comme sa vitesse de mouvement, sa position dans la scène de combat, la direction de ses mouvements... Il peut aussi être dans trois positions différentes : attaquer son adversaire, bloquer les coups et en repos.



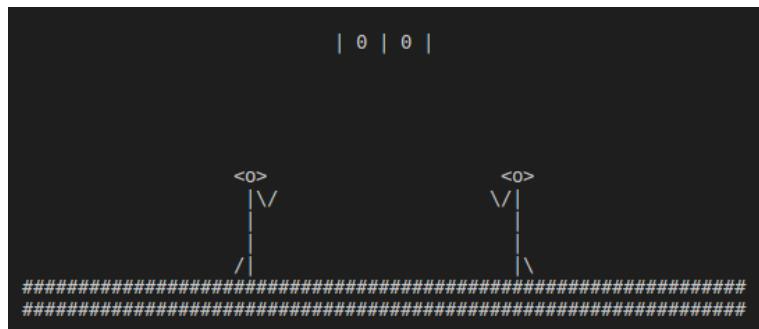
En attaque - Bloque - Repos

Les fonctions implémentées dans cette classe sont propres au joueur : elles peuvent changer sa position de combat, sa position dans la scène de combat, compter ses points s'il réussit une attaque...

- **menu.cpp** : Avant de pouvoir jouer, les joueurs peuvent choisir des paramètres comme la scène de combat (implémentée dans le fichier **scene.cpp** que nous verrons au point suivant). Pour choisir ses paramètres facilement, un affichage d'un menu facile d'utilisation a été implémenté. **menu.cpp** représente toutes les pages principales et intermédiaires du jeu (comme le menu au début du jeu ou le menu de pause pendant un combat).



- **scene.cpp** : Comme son précurseur, ce fichier possède des fonctions pour afficher un écran sur le terminal. Cependant, il ne prend en compte que la scène de combat : ses fonctions peuvent charger une scène d'un fichier d'extension **.ffscene**, reprendre les données d'une partie sauvegardée et afficher une scène de combat avec mes paramètres qu'un joueur a choisi lors du menu (implémentée dans **menu.cpp**).



- **jeu.cpp** : ce fichier représente l'organisation du jeu. Il lance les différents menus visibles, gère les liaisons entre les interactions entre l'utilisateur et le terminal et gère les lignes principales du jeu. Il utilise principalement toutes les fonctions des différents fichiers ci-dessus.

2.2 Liste des fonctionnalités

Comme demandé (voir partie 1), ce programme est capable de faire jouer deux joueurs locaux sur l'écran de terminal et de reconnaître le joueur qui gagne des points en attaquant son adversaire (s'il n'est pas en position d'attaque ou de blocage).

De plus, l'utilisateur peut donner le nombre d'images par seconde qui permettra d'actualiser l'affichage de l'écran, à l'aide d'un argument supplémentaire à la commande d'exécution.

Fonctionnalités supplémentaires :

1. Un menu simple d'accès a été ajouté pour pouvoir choisir les paramètres de son jeu. Les joueurs peuvent choisir les paramètres de jeu. Un mode de jeu simple a été rajouté pour jouer rapidement. (Le menu est implémenté dans le fichier **menu.cpp**.)
2. Durant une partie, le joueur a la possibilité de mettre en pause la partie en cours en cliquant sur " ". Une page intermédiaire s'affichera et proposera au joueur trois choix différents :
 - Reprendre la partie.
 - Sauvegarder et quitter le jeu.
 - Revenir au menu.
3. Une partie s'arrête au moment où l'un des joueurs a atteint trois points.
4. À la fin du jeu, les joueurs peuvent décider de recommencer une partie avec les mêmes paramètres ou de revenir sur le menu.
5. Les joueurs ont plusieurs choix possibles pour choisir leur scène de combat : soit avec le transfert de données avec un fichier, soit avec le menu.