Rapport de projet

Projet: Fancy Fencing
Auteur: Comlan Amouwotor
No Etudiant: 22221465

Installation du jeu

Ce jeu nécessite les modules pynput pour traiter les entrées clavier. Ce module peut être installé par la commande:

python3.9 –m pip install pynput

Lancement du jeu:

Le jeu se lance dans le terminal ou dans la ligne de commande, depuis le repertoire du projet, par une commande de la forme:

```
python3.9 main.py --fps 24 --ms 9 8 --as 12 13 --ar 8 15 --bt 7 9 --scene default.ffscene
```

Dans cette commande:
--fps 24 : lance le jeu à 24 frames par seconds
--ms 9 8 : le movement speed des joueurs (celui du joueur 1 est 9 et 8 pour le joueur 2)
--as 12 13: les valeurs du attacking_speed pour les deux joueurs respectivement
--ar 8 15 : les valeurs du attacking_range pour les deux joueurs respectivement
--bt 7 9 : les valeurs du blocking_time pour les deux joueurs respectivement
--scene filename: pour préciser le fichier scene utilisé (sans les quotes).

Chacun de ces parametres est requis. La commande suivante,

```
python3.9 main.py –help
```

renvoie un message d'aide qui rexplique les parametres.

La liste complète des modules utilisés dans ce projet est la suivante:

```
import sys
import os
from time import sleep
from pynput import keyboard
from collections import deque
from enum import Enum
import numpy as np
import re
from math import sqrt
import pickle
```

En particulier, l'affichage du jeu s'est fait entièrement grâce à des np.ndarray, pas de curses ni blessed.

Un autre point est que le projet n'utilise par de librairie pour le threading. La simultanéité des movements des joueurs est implémenté grâce à des structures de file

```
self.pending_motions = deque()
self.pending_states = deque()
```

dans la classe HumanPlayer. Pour faire un movement complexe comme le saut par exemple qui est censé durer plusieurs frames et contenir plusieurs movement elementaire (bouger gauche, droite, haut, bas), le joueur concerné met dans sa file d'execution la liste de touts les prochains movements qu'il est censé faire. Puis à chaque frame, il execute le movement elementaire actuel et attend le prochain frame pour le prochain.

Avec cette technique on peut rigoureusement garantir que les mouvements prennent le nombre de frame qu'on souhaite qu'ils durent. Aussi on peut par exemple saisir une longue combinaison de commandes de mouvements que le joueur enregistrera et executera progressivement à la vitesse du frame_per_second du jeu.

Par contre quand par exemple plusieurs mouvements vers la gauche ont été mis dans la file d'execution du joueur, et qu'on souhaite les enlever un à un il suffit d'entrer des commandes de mouvements vers la droite et la file sera dépilé du haut (à la manière d'une pile, Last in First out).

Les joueurs changent leur orientation quand ils se croisent afin de toujours se faire face l'un et l'autre.

Description des fonctionnalités

- Les preréquis ont été implémentés
- Pour aller en pause on utilise la touche Espace. Pour revenir au jeu après la pause on utilise la meme touche.
- Pour enregistrer l'etat actuel du jeu on va en mode Pause, puis il suffit de faire <Entrée>. Le jeu sera enregistré dans un fichier cache nommé .game.pickle du repertoire courant. Pour récupérer l'etat enregistré on retourne en mode pause et on fait <Shift>.
- Un mini son est joué quand les joueur echouent ou reussissent une attaque.
- Au lancement du jeu on peut preciser l'adresse d'un fichier valide .ffscene qui spécifie la scène du jeu.
- Le projet est mis sur Github à l'adresse: https://github.com/realComlan/fancy-fencing

Happy Fencing!