

School Fighter

Par Nadim DOUHANE et Geoffrey PIERRE

1. Présentation du jeu

School Fighter est un jeu de combat en **2D** inspiré de *Street Fighter*.

Les personnages sont des étudiants représentés par des sprites, et les combats se déroulent dans des décors scolaires (salle de classe, cour de récréation, couloirs, cafétéria).

Le jeu est codé en **Python** avec la bibliothèque **Pygame**.

2. Fonctionnalités principales

Modes de jeu

- **Single Player** → le joueur combat un bot avec une IA faite à la main (pas statistique).
- **Two Players** → deux joueurs humains s'affrontent sur le même clavier.
- **Menu principal** → permet de choisir le mode, lire les instructions, accéder aux paramètres audio, ou quitter.

Audio

- Tous les sons sont stockés dans le dossier **sound** :
 - **punch.wav** → coup de poing
 - **kick.wav** → coup de pied
 - **special.wav** → attaque spéciale
 - **block.wav** → défense
- Une musique de fond (**background_music.mp3**) joue en continu.

- Dans le menu **Settings**, le joueur peut :
 - Couper la musique
 - Régler le volume

Contrôles (AZERTY)

Player 1

- Gauche : Q
- Droite : D
- Saut : Z
- Défense : S
- Coup de poing : F
- Coup de pied : G
- Attaque spéciale : H (lance une boule de feu)

Player 2

- Gauche : ←
- Droite : →
- Saut : ↑
- Défense : ↓
- Coup de poing : K
- Coup de pied : L
- Attaque spéciale : M (lance un éclair)

Combat

- Chaque attaque dure **~1 seconde**, ce qui permet de voir l'animation.
- Les attaques spéciales sont des **projectiles** :
 - **Player 1** : boule de feu (`fireball.png`)
 - **Player 2** : éclair (`lightning.png`)
- Les projectiles se déplacent à l'écran, infligent des dégâts en cas de collision, puis disparaissent.
- Les personnages ne peuvent pas sortir de l'écran (déplacements limités aux bords).
- Les sprites sont redimensionnés pour une meilleure jouabilité.

Graphismes

- Les décors de fond (`classroom.jpg`, `playground.jpg`, `corridors.jpg`, `cafeteria.jpg`) sont **automatiquement redimensionnés** à la taille de la fenêtre.
 - Chaque personnage possède des animations (idle, walk, jump, punch, kick, special, block).
 - Les mêmes sprites sont utilisés pour les deux joueurs.
-

3. Comment lancer le jeu

Prérequis

- Installer **Python 3.10+**
- Installer **Pygame** :

```
pip install pygame
```

Lancement

1. Place tous les fichiers dans un dossier :

- `main.py` (le fichier principal du jeu)
- `img/` → tous les sprites et backgrounds
- `sound/` → tous les fichiers sons et musique

2. Dans un terminal, lance le jeu avec :

```
python School_fighter1.py
```

4. Ce que nous avons appris

Au cours de la réalisation de ce projet, nous avons eu l'opportunité d'explorer et de maîtriser une technologie innovante : l'intelligence artificielle Copilot. Cette expérience nous a permis de comprendre le potentiel d'un agent IA capable d'interagir directement avec l'environnement de développement, notamment en créant, organisant et modifiant des fichiers de manière autonome lorsque les permissions lui sont accordées.

Nous avons également approfondi nos connaissances dans le domaine du prompt engineering, une compétence essentielle pour communiquer efficacement avec une IA générative. À la suite de recherches et d'expérimentations, nous avons découvert qu'il était possible de structurer nos instructions sous la forme de fichiers JSON, permettant de définir avec précision nos attentes vis-à-vis du modèle. Cette approche a rendu le processus de développement plus rigoureux et reproductible, en précisant notamment les technologies requises, les dépendances du projet ainsi que les ressources nécessaires au bon fonctionnement du jeu (sons, sprites, images d'arrière-plan, etc.).

5. Prompts

Prompt 1 :

```
{
  "requirements" : {
    "code_quality" : "use clean code architecture",
    "programming_language" : "python",
    "environement" : "python venv",
    "libs" : ["pygame"],
    "identity": "senior python expert"
  },
  "instructions" : "The goal is to create a fighting game. The playable characters are stick figure students. The game can be played locally with up to 2 players, and there is also a single player mode where the player fights against a bot controlled by a handcrafted AI (not based on statistical methods). The battles take place in a school-themed environment, with backgrounds such as classrooms, gymnasiums, schoolyards, and hallways, providing a fun and playful atmosphere for the ninja fights."
}
```

Prompt 2 :

```
{
  "game_title": "School Fighter",
  "code_quality": "use clean code architecture",
  "programming_language": "python",
  "libs": ["pygame"],
  "inspiration": "Street Fighter",
  "instruction": "Continue developing the fighting game 'School Fighter'. Add sound effects for attacks, kicks, specials, and blocks. All sounds are located in a 'sound' folder. Include background music that plays continuously throughout the game. The player must have options to mute the music and adjust its volume. Characters cannot leave the screen boundaries, so their size should be reduced accordingly to fit gameplay comfortably. All background images should be scaled to fit the game window. Maintain clean code architecture, separating rendering, logic, input handling, and audio management.",
  "audio": {
    "sound_effects_folder": "sound",
    "effects": {
      "punch": "punch.wav",
      "kick": "kick.wav",
      "special": "special.wav",
      "block": "block.wav"
    }
  },
  "background_music": "sound/background_music.mp3",
  "controls": {
    "mute_music": true,
    "adjust_volume": true
  }
},
"character_constraints": {
```

```

"stay_on_screen": true,
"size_adjustment": "reduce sprites to fit gameplay area"
},
"menu": {
  "options": [
    "Single Player",
    "Two Players",
    "Instructions",
    "Settings",
    "Exit"
  ],
  "settings_options": [
    "Mute Music",
    "Adjust Music Volume"
  ],
  "controls": {
    "player1": {
      "move_left": "Q",
      "move_right": "D",
      "jump": "Z",
      "punch": "F",
      "kick": "G",
      "special": "H",
      "block": "S"
    },
    "player2": {
      "move_left": "Left Arrow",
      "move_right": "Right Arrow",
      "jump": "Up Arrow",
      "punch": "K",
      "kick": "L",
      "special": "M",
      "block": "Down Arrow"
    }
  }
},
"assets": {
  "backgrounds": [
    "classroom.jpg",
    "playground.jpg",
    "corridors.jpg",
    "cafeteria.jpg"
  ],
  "sprites": {
    "player1": {
      "idle": ["player1_idle.png"],
      "walk": ["player1_walk0.png", "player1_walk1.png", "player1_walk2.png"],
      "jump": ["player1_jump.png"],
      "punch": ["player1_punch0.png", "player1_punch1.png", "player1_punch2.png"],
      "kick": ["player1_kick0.png", "player1_kick1.png", "player1_kick2.png"],
      "special": ["player1_special0.png", "player1_special1.png", "player1_special2.png"],
      "block": ["player1_block.png"]
    }
  }
},

```

```

"player2": {
  "idle": ["player1_idle.png"],
  "walk": ["player1_walk0.png", "player1_walk1.png", "player1_walk2.png"],
  "jump": ["player1_jump.png"],
  "punch": ["player1_punch0.png", "player1_punch1.png", "player1_punch2.png"],
  "kick": ["player1_kick0.png", "player1_kick1.png", "player1_kick2.png"],
  "special": ["player1_special0.png", "player1_special1.png", "player1_special2.png"],
  "block": ["player1_block.png"]
}
}
}
}

```

Prompt 3 :

réduit la vitesse d'attaque pour que la réalisation d'une attaque soit étalée sur approximativement 1 seconde, Ajuste la hitbox pour que la hitbox en largeur soit 2 fois moins large (même si cela rogne sur l'image car le personnage est au centre d'un carré et la hitbox prend en compte l'image et non le personnage), enfin ajoute une touche permettant à chaque joueur d'utiliser une attaque spéciale (attaque à distance, le 1er joueur lancera une boule de feu et le second joueur un éclair avec des pistes sonores adéquates (présentes dans sound))

Prompt 4 :

réduit la vitesse de réalisation d'une attaque pour qu'elle dure 0.3 seconde puis le personnage revient à sa position de départ. ensuite, décompose l'exécution d'une attaque spéciale pour chaque joueur en la succession des images special0 special1 special2 et special3 (respectivement pour chacun des joueurs) et enfin multiplie par 4 la taille des sprite relatifs aux attaques spéciales : (boule de feu et éclair) et fais en sorte qu'ils se lancent 0.5 seconde plus tard que actuellement. Ultime modification, permet le double saut pour les personnages récupérable uniquement lorsqu'il touche le sol comme un double saut classique

Prompt 5 :

réduit la hitbox verticale et horizontale des sprite des attaques spéciales en la divisant par 3 sans réduire leurs taille et lorsqu'un joueur utilise son attaque spéciale, réinitialise son sprite au sprite de départ une fois le projectile lancé et enfin augmente la vitesse de déplacement des joueurs de 40%. Lorsqu'un joueur effectue une attaque (kick, punch ou block et special) après l'attaque, réinitialise le sprite à la sprite de départ. Ensuite, il faut que la music background_music ne soit jouée que lorsque l'utilisateur est dans le menu de sélection, lorsqu'il lance un combat la musique s'arrête et reprend à la fin d'un combat lorsqu'on retourne sur le menu