

POTOI POTOI POTOI

Emanuel et Isabella

Camino, 1G3

1.INTRODUCTION

Remarque : Certains codes ont des mots en espagnol pour une meilleure compréhension entre nous.

Titre du projet: Potoi Potoi Potoi

Présentation:

Création d'un jeu interactif en utilisant Tkinter.

Notre projet de NSI consiste en un jeu en Python avec Tkinter. Dans ce jeu, le joueur contrôle un professeur dans une salle de classe où les élèves parlent sans arrêt. Son objectif est de « manger » tous les élèves qui parlent trop. Ce jeu est divisé en trois niveaux: le niveau facile, le niveau normal, et le niveau difficile. Dans un niveau facile, le professeur doit « manger » les élèves sans vraiment aucune difficulté. Cependant, dans le niveau normal, un bébé se déplace constamment dans la salle poursuivant le professeur, et s'il l'attrape, la partie est perdue. Finalement, dans le niveau difficile, un obstacle est ajouté : un inspecteur apparaît chaque x temps pour surveiller la classe, et s'il a le prof en flagrant délit, le joueur perd, c'est-à-dire, si le prof mange un élève au moment où l'inspecteur apparaît, il perd. Une fois que tous les élèves ont été attrapés, le joueur gagne la partie.

Ce projet nous permet d'apprendre à programmer un jeu en Python en utilisant Tkinter pour créer l'interface. Nous travaillons aussi sur le déplacement des personnages, la détection des collisions (quand le professeur touche un élève ou le bébé), et la gestion des règles du jeu (conditions pour gagner ou perdre). Cela nous aide à mieux comprendre comment structurer un programme et organiser des interactions entre différents éléments du jeu.

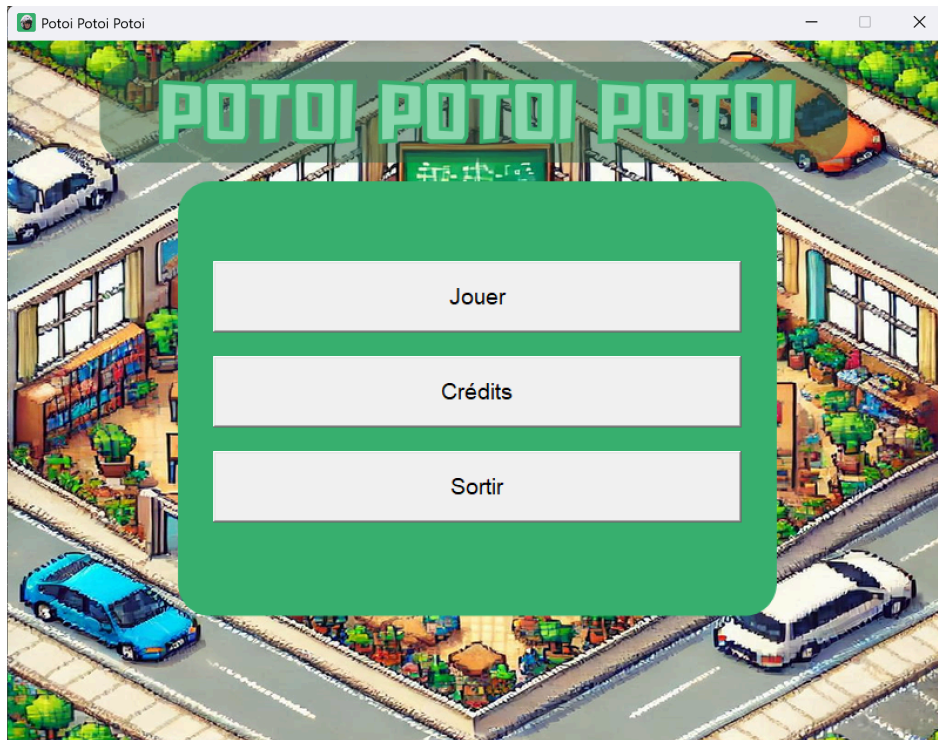
L'idée finale est de créer les personnages en fonction des élèves de la classe de NSI, M. Genty comme le professeur, et le bébé comme celui qui est en classe de NSI, pour faire une sorte de parodie qui amuse le joueur. Plus tard, en discutant avec le prof d'histoire, on a eu l'idée que M. Tornow soit l'inspecteur, qui d'abord allait être une simple animation.

Objectifs :

- Développer un jeu en Python avec Tkinter
- Programmer le déplacement du professeur, des élèves et du bébé
- Gérer les collisions et les conditions de victoire/défaite (le professeur touche un élève → L'élève disparaît, le professeur touche le bébé → Le joueur perd la partie)

2.ORGANISATION DU JEU :

Présentation :

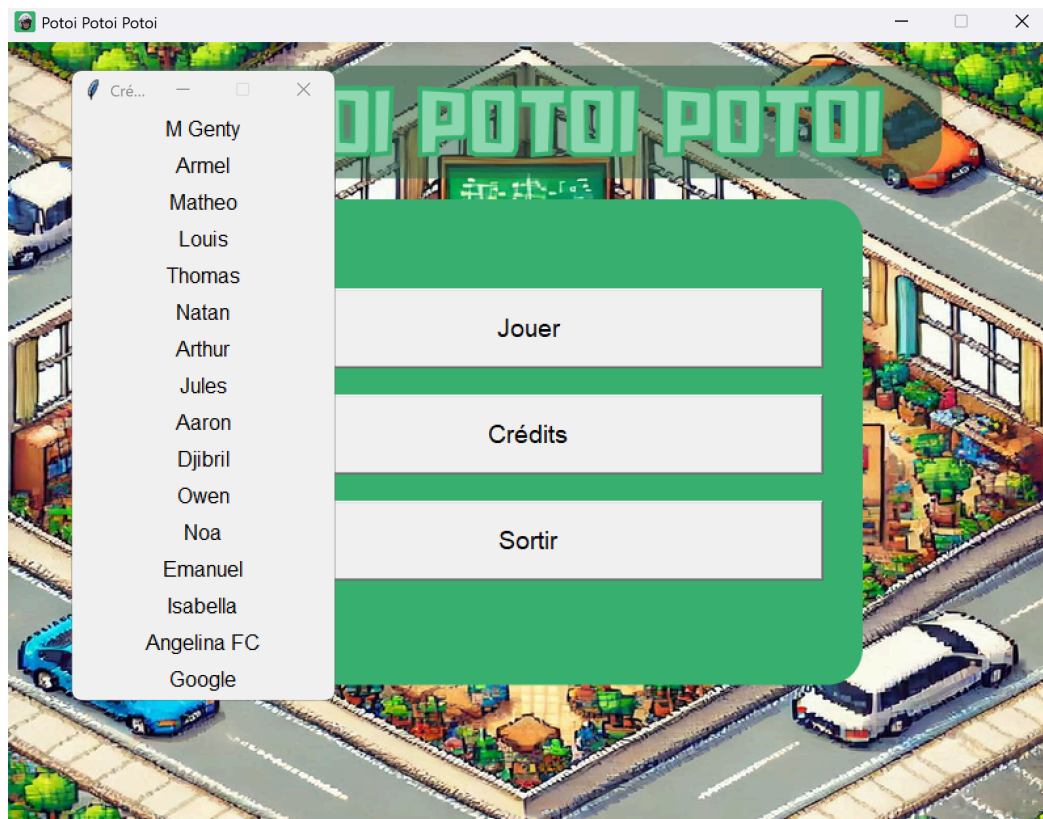
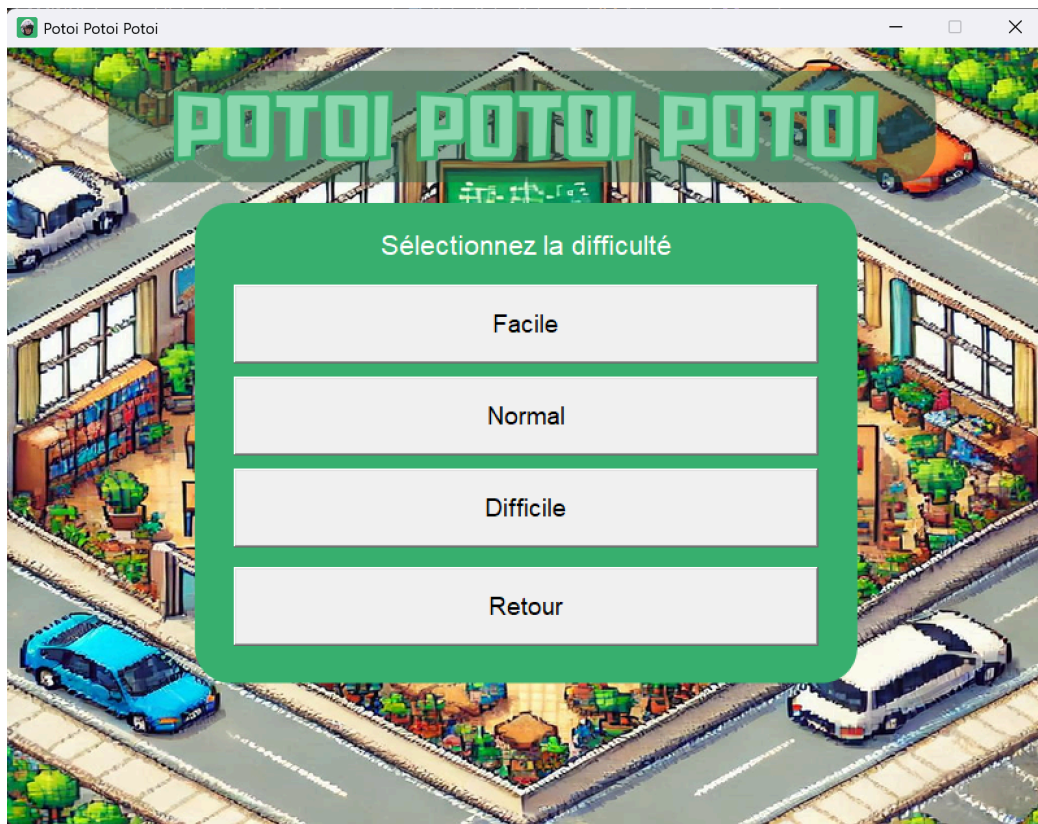


A l'aide de nôtre sœur, on a réussi à faire apparaître la fenêtre où se trouve le menu principal, et on a mis le titre du jeu, puis trois boutons principaux : le bouton 'Jouer', le bouton 'Crédits' et le bouton 'Sortir'.

En appuyant sur le bouton 'Jouer', on retrouve les trois niveaux à choisir : Facile, Normal et Difficile et un bouton Retour pour retourner au menu principal.

Avec le bouton Crédits, apparaît une nouvelle fenêtre où on nomme tous les participants et les personnes qui ont aidé à faire le résultat final.

Finalement, en appuyant sur Sortir, la fenêtre se ferme et on sort du jeu.



3.ANALYSE DES BESOINS

- **Fonctionnalités attendues :**

- Mouvement des personnages (M. Genty et les élèves)
- Compteur pour suivre le le nombre d'élèves du joueur
- Génération d'ennemis (les élèves)
- Compteur d'élèves restants à manger
- Fin du jeu
- Sortir/crédits
- Création du bébé
- Que les élèves fuient le professeur au lieu de rester immobiles

- **Contraintes et limites :**

- Les librairies (pygame pour le son, Pillow pour la qualité des images)
- Vitesse de rendu
- Issus de collisions (que les élèves et le professeur ne passent pas au-dessus des tables)
- Des Bugs
- Manque de connaissances

4. RÉPARTITION TÂCHES :

Isabella :

- L'interface (skins, fond, etc.) :

Isabella a créé l'apparence du jeu. Elle a choisi et intégré les photos des élèves. Elle a aussi mis en place le fond d'écran, en veillant à ce que ce soit agréable à regarder et cohérent avec l'univers du projet. Elle a fait attention aux couleurs, aux images utilisées, et à la disposition des éléments sur l'écran.

- Le compteur :

Elle a programmé un compteur qui sert à afficher le nombre d'élèves restant. Ce compteur est visible à l'écran et se met à jour automatiquement selon ce que fait l'utilisateur. Détecte si le professeur touche un élève. Si oui, se réduit le compteur.

- Le mouvement du professeur :

Elle a aussi géré les déplacements du personnage du professeur dans l'application. Elle a codé les mouvements pour qu'il puisse bouger selon les actions de l'utilisateur.

```
def crear_profesor(self):
```

```
    x, y = self.profesor_pos
```

```
        profesor_img = self.cargar_imagen(self.profesor_skin,  
(self.profesor_size, self.profesor_size))
```

```
    self.profesor = self.canvas.create_image(x, y, image=profesor_img)
```

```
    self.imagenes_guardadas.append(profesor_img)
```

Début

Définir créer_professeur()

(x, y) ← position_initiale_du_professeur

image_du_professeur ← charger_image(apparence_du_professeur,
(taille_professeur, taille_professeur))

objet_professeur ← canvas.créer_image(x, y, image = image_du_professeur)

ajouter image_du_professeur à la_liste_des_images_enregistrées

Fin

- **Pillow (bibliothèque d'images) :**

Pour charger les images et les adapter, Isabella a utilisé une bibliothèque Python qui s'appelle Pillow. Grâce à cela, elle a pu redimensionner des images, les faire bouger, et même ajouter des effets. Cela lui a permis de mieux adapter les visuels du jeu.

- **Inspecteur :**

Elle a confectionné l'inspecteur pour qu'il apparaisse à chaque fois qu'un randomizer choisit le numéro 7 entre 1 et 10. D'abord il apparaît le texte (J'écoute des pas dans les couloirs...), ensuite, quelques secondes après, on voit l'inspecteur en bas à gauche qui vigile la classe.

Emanuel :

- **Le mouvement des élèves :**

Emanuel a programmé les déplacements des élèves dans le jeu. Il a fait en sorte qu'ils se déplacent dans différentes directions, ce qui rend le jeu plus dynamique. Les élèves

bougent de manière automatique ou aléatoire, ce qui oblige le joueur (le professeur) à les “poursuivre”.

- **Les collisions :**

Il a mis en place un système de collisions pour détecter quand le professeur touche un élève. Lorsqu’une collision est détectée, cela déclenche une action : le prof "mange" l’élève, c’est-à-dire que l’élève disparaît du jeu. Si le professeur touche un élève, il s’efface du jeu, réduit le compteur et produit un son ‘hum’.

```
def verificar_colisiones(self):
    if self.juego_terminado:
        return
    x1, y1, x2, y2 = self.canvas.bbox(self.profesor)
    for alumno, tipo in self.alumnos[:]:
        ax1, ay1, ax2, ay2 = self.canvas.bbox(alumno)
        if (x1 < ax2 and x2 > ax1 and y1 < ay2 and y2 > ay1):
            if self.dificultad == "difficile" and self.imagen_inspector_visible:
                self.terminar_juego(perdiste=True, trahison=True)
                return
            self.canvas.delete(alumno)
            self.alumnos.remove((alumno, tipo))
            self.alumnos_restantes -= 1

            self.canvas.itemconfig(self.marcador, text=f"Élèves qui restent :
{self.alumnos_restantes}", font=("Arial", 30), fill="black")

    if self.sonido_hum:
        self.sonido_hum.play()
```

- **La musique :**

Il a ajouté une bande sonore au jeu pour le rendre plus immersif et des effets de son.

- **Les os:** gère les chemins d'accès aux fichiers.

- **Les touches :**

Emanuel a aussi programmé les contrôles du jeu. Il a défini les flèches du clavier utilisées par le joueur pour déplacer le prof. Il a fait en sorte que ces touches soient simples à utiliser et bien prises en compte pendant la partie.

- **Le bébé :**

Le premier ennemi qui est au niveau normal, mais aussi dans le difficile, est un bébé programmé pour qu'il apparaisse au début de la partie en haut-droit, et son objectif est d'attraper le professeur, s'il le touche, la partie est perdue.

Le résultat final a été vérifié et amélioré par Emanuel et Isabella pour obtenir le meilleur résultat.

- **Planning prévisionnel :**

Développement:

- *Interface technique: avant les vacances*
- *Améliorations et résultat: Pendant les vacances*

5. CRITÈRES DE VALIDATION ET TESTS

- **Critères de réussite :**

Ce n'est pas exactement comme on l'avait imaginé au départ, car on a retrouvé des complications, mais peut être mieux. On aurait voulu que les élèves fuent le professeur et qu'ils suivent des chemins mais on n'a pas su comment le faire. Mais malgré tout, on est très contents et satisfaits du résultat final et fiers de notre jeu, de même que le reste des personnes qui ont participé. On a réussi notre objectif principal: divertir et faire rire comiquement les gens qui le jouent.