UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

RÉALISATION DE PROJET N°1

Par ANTON NELCON Steve

1

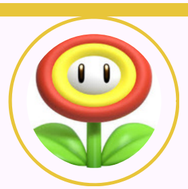
Licence 3 ISEI

8 JANVIER 2024

SOMMAIRE :

PRÉSENTATION DU PROJET

→ *En quoi consiste mon projet ?*



PRÉSENTATION DU JEUX

L'objectif du jeu
Les personnages
Les mondes
Les objets

FONCTIONNEMENT DU JEU

- Se déplacer
- Lancer une carapac
- Lancer une boule de feu
- Ramasser des objets

FONCTIONNALITÉS DU JEU

- Mode sombre/ claire
- Changer de langue
- La caméra
- Les sons

LES CODES

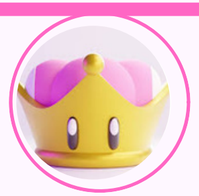
Voyons le code

DEMONSTRATION DU JEU

Démonstration du jeu

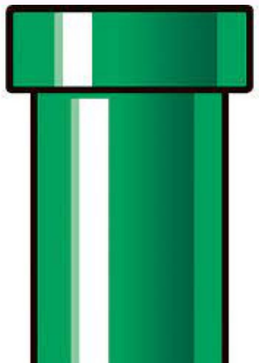
MES IMPRESSIONS PERSONNELLES

- Ce que j'ai apprécié
- Ce que je n'ai pas apprécié
- Mes facilités
- Mes difficultés



CONCLUSION

- Pourquoi ce projet ?



PRÉSENTATION DU PROJET

Par ANTON NELCON Steve

PRÉSENTATION DU PROJET

→ *En quoi consiste mon projet ?*

PRÉSENTATION DU PROJET

Objectif du projet :

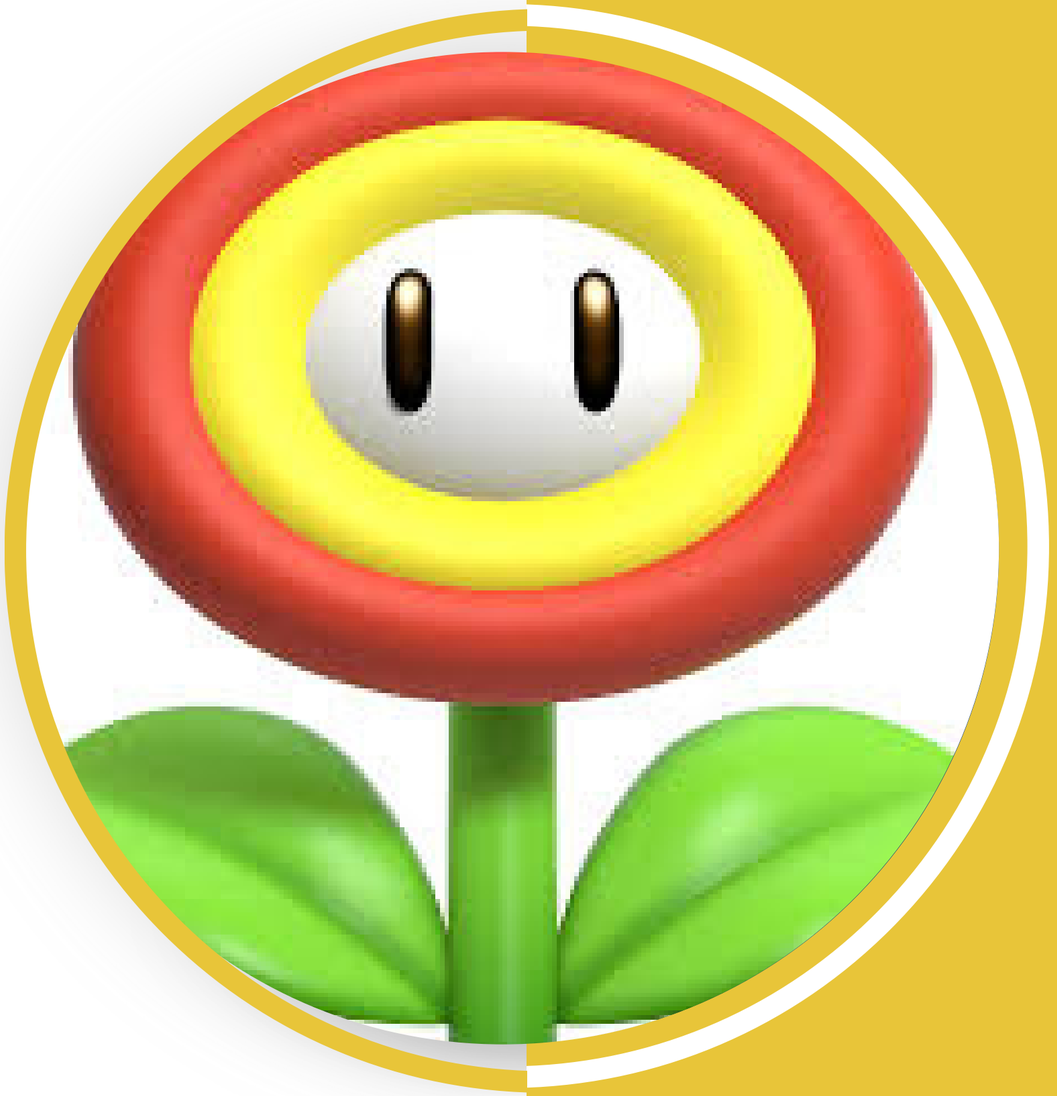
Créer une application web, un logiciel ou un module embarqué pour le module "réalisation de projet".

Les outils utiliser :

- Python
- GitHub, Visual Studio Code
- Moteur graphique tel que : Ursina

o

PRÉSENTATION DU JEU



Par ANTON NELCON Steve

PRÉSENTATION DU JEU

→ *En quoi consiste mon Jeu ?*

Description du jeu :

- 1 plateforme en 2D
- 3 personnages (Mario, BowserJr, ennemi)
- Des objets (bananes, pièces , champignon etc)

Objectif du jeu :

- Le joueur peut se déplacer dans tout les sens
- Le joueur doit lancer des carapaces pour éliminer BowserJr
- Le joueur doit esquiver les boule de feu lancé par Bowse Jr et les bananes
- Le joueur doit ramasser des pièces et des champignons

PRÉSENTATION DU JEU

→ *Les personnages*

→*Les 6 personnages disponibles:*

Mario

Luigi

Peach

Toad

Daisy

Harmonie

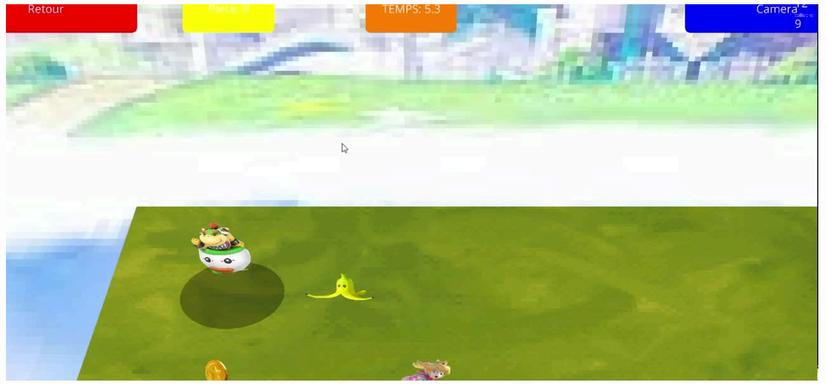
PRÉSENTATION DU JEU

→ *Les pouvoirs de chaque personnages*

→ *Chaque personnages possèdent un pouvoir:*

Luigi : ***se téléporter***

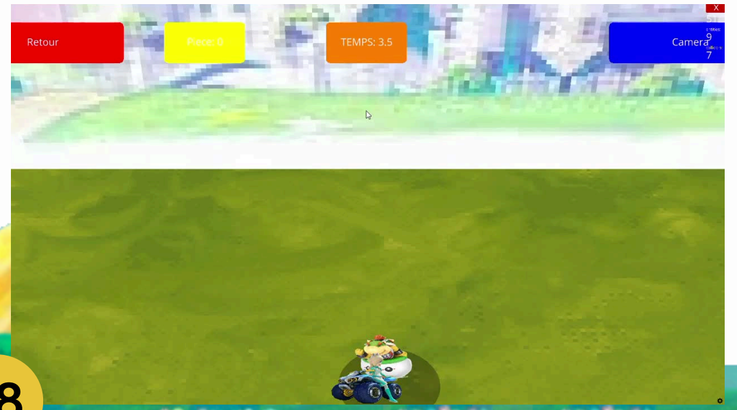
Peach: ***onde de choc***



Toad: ***super vitesse***

Daisy : ***double saut***

Harmonie: ***invincible***



Mario: *pas de pouvoir*

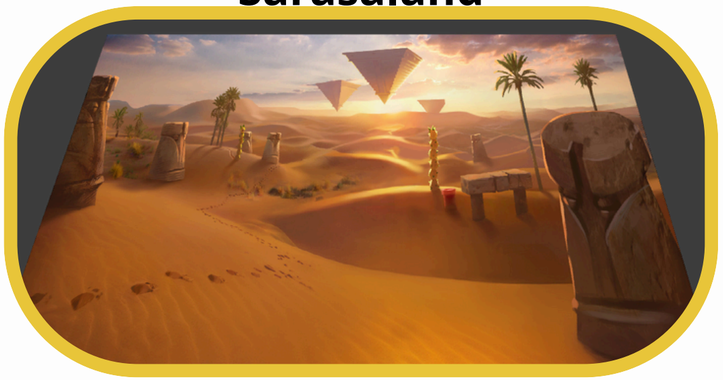
PRÉSENTATION DU JEU

→ *Les Mondes*

→ *Les 4 mondes à découvrir*

Le Royaume Champignon

Sarasaland



L’observatoire de la Comète

Le chateau de Bowser

PRÉSENTATION DU JEU

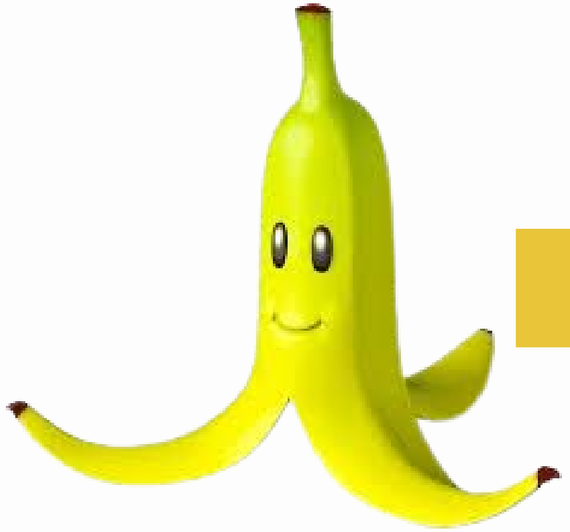
→ *Les ennemis*

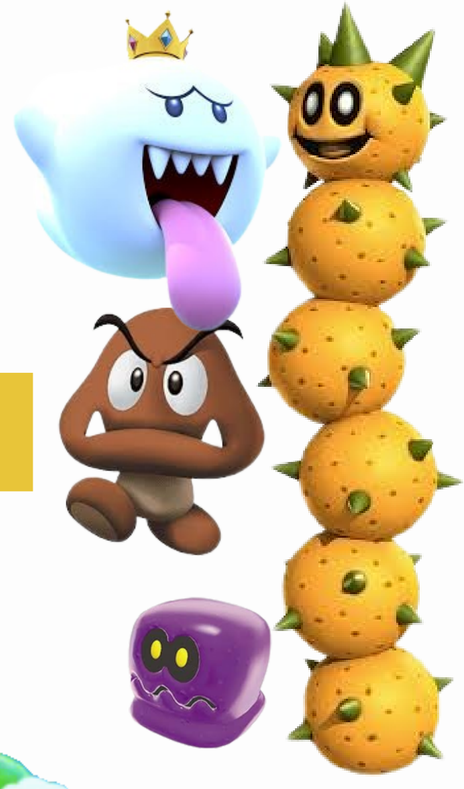
→ *Les ennemies:*

Bowser Jr

boule de feu

des bananes





PRÉSENTATION DU JEU

→ *Les objets à collecter*

Les 2 objets à collecter

les pièces

La carapaces rouge

Les champignons

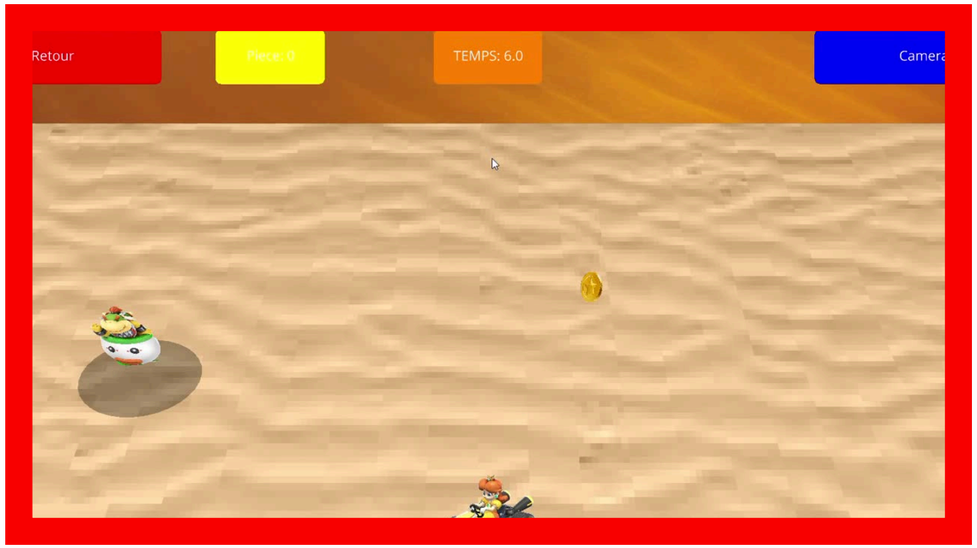
FONCTIONNEMENT DU JEU

Par ANTON NELCON Steve

FONCTIONNEMENT DU JEU

→ *se déplacer*

- POUR SE DÉPLACER, LE JOUEUR DOIT UTILISER LES **TOUCHES DIRECTIONNELLES** DU CLAVIER.



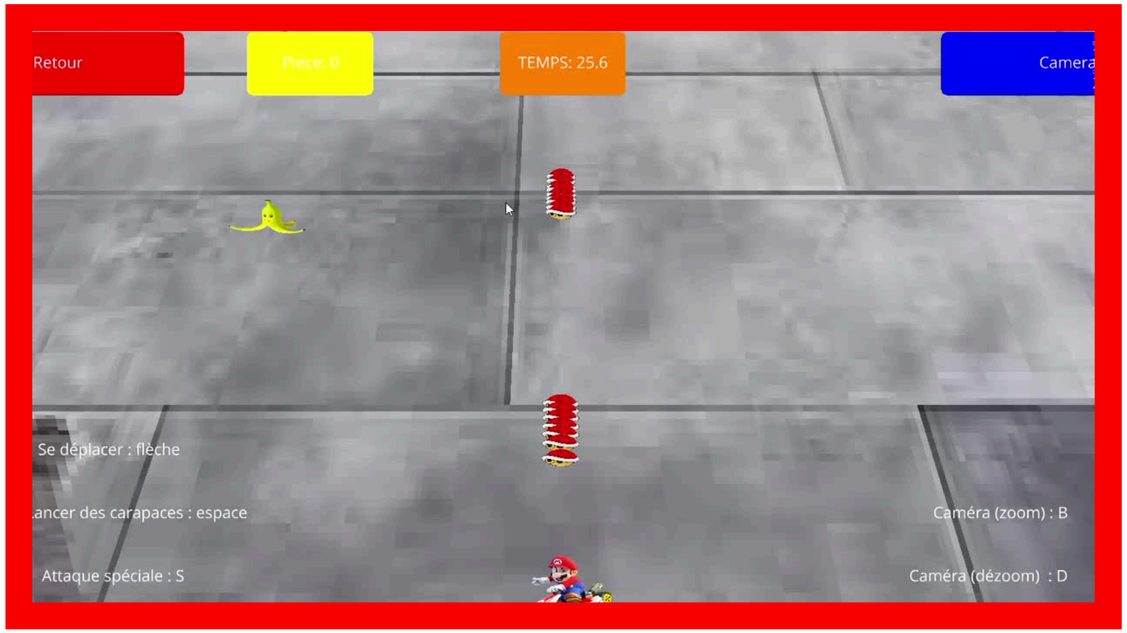

```
def update() :
  if player :
    # Déplacements avec les flèches directionnelles
    if held_keys['up arrow']:
        player.position += player.forward * speed
    if held_keys['down arrow']:
        player.position -= player.forward * speed
    if held_keys['left arrow']:
        player.position -= player.right * speed
    if held_keys['right arrow']:
        player.position += player.right * speed
```

FONCTIONNEMENT DU JEU

→ *Lancement de la carapaces*

- POUR LANCER UNE CARAPACE, LE JOUEUR DOIT APPUYER SUR LA TOUCHE **ESPACE** DU CLAVIER.

Espace

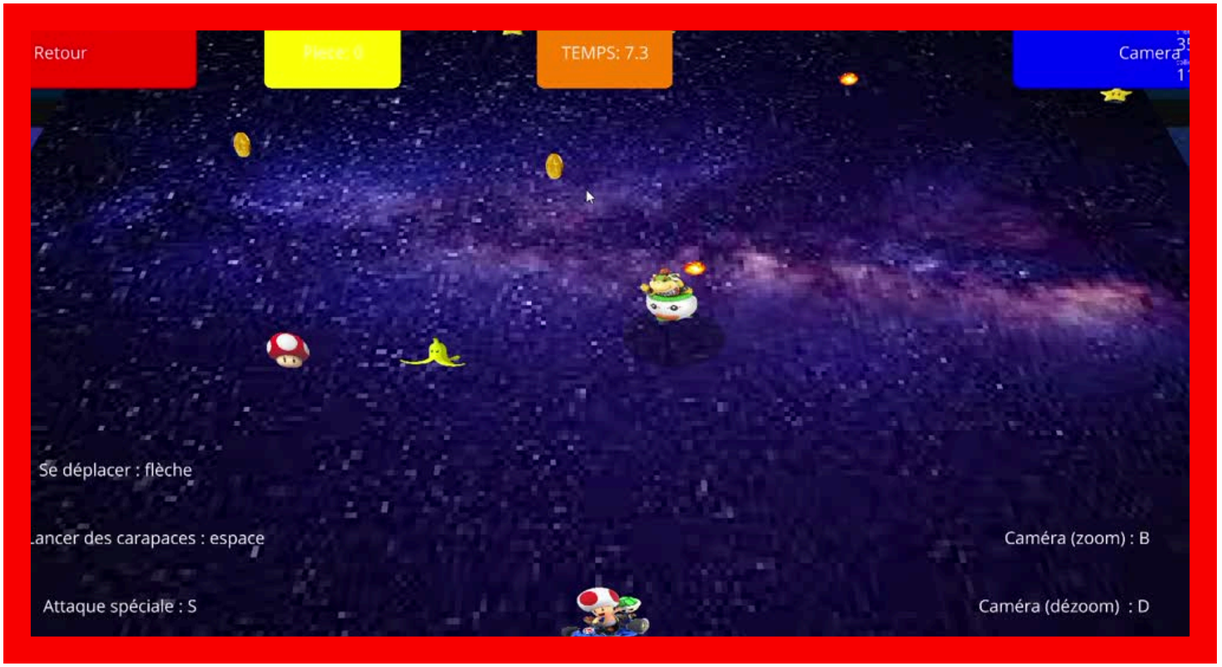



```
def update() :
  if player :
      # Lancer une carapace avec la touche espace
      if held_keys['space']:
          lancer_carapace()
```

FONCTIONNEMENT DU JEU

→ *Lancement de la boule de feu*

- LA BOULE DE FEU APPARAIT **ALEATOIREMENT** A PARTIR DE BOWSERJR

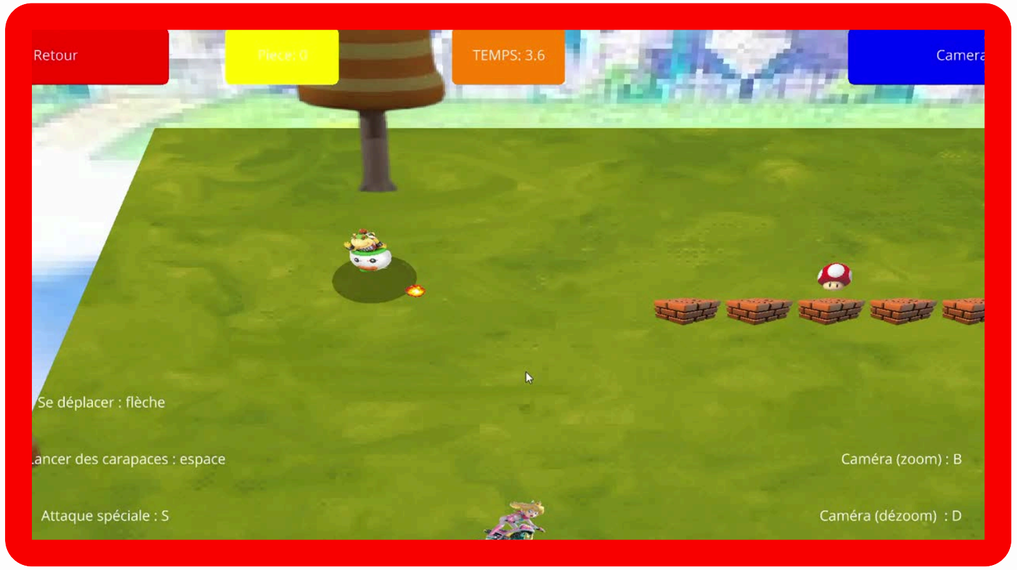



```python
boules_de_feu = []
# Fonction pour lancer une boule de feu
def lancer_boule_feu():
    sound_fire.play()
    global bowserjr
    if bowserjr:
        boule_feu = Entity(model='quad',texture='boulefeu.png',scale=(0.5, 0.5, 0.5),position=bowserjr.position + Vec3(0, 0, 0),  collider='box'
        )
        # Choix aléatoire de direction de la boule de feu
        direction_principale = choice([
            Vec3(0, 0, -1),  # Devant
            Vec3(0, 0, 1),   # Derrière
            Vec3(-1, 0, 0),  # Gauche
            Vec3(1, 0, 0)    # Droite
        ])
        variation = Vec3(uniform(1, 1), uniform(-0.5, 0.5), uniform(-0.5, 0.5))
        boule_feu.direction = (direction_principale + variation).normalized()
        boules_de_feu.append(boule_feu)

        invoke(supprimer_boule_feu, boule_feu, delay=10) # La boule de feu disparaît après 10 secondes


def supprimer_boule_feu(boule_feu):
    if boule_feu in boules_de_feu:
        boules_de_feu.remove(boule_feu)
    destroy(boule_feu)
```

FONCTIONNEMENT DU JEU

→ *Collusion de la boule de feu*

- SI LA DISTANCE ENTRE LE PERSONNAGE ET LA BOULE DE FEU EST INFÉREIUR A 1 , IL Y A UNE COLLISION



- COLLISION ENTRE LA BOULE DE FEU ET LES PERSONNAGE

```
# Vérifier si une boule de feu touche Mario
    for boule_feu in boules_de_feu[:]:
        if boule_feu and hasattr(boule_feu, 'position') and boule_feu.enabled:
            if (player.position - boule_feu.position).length() < 1:
                print("Game Over! Mario a été touché par une boule de feu.")
                sound_explosion.play()
                gameover()
```

- APPARITION DES BOULE DE FEU TOUTE LES 5 SECONDES

```
# Lancer une boule de feu toutes les 5 secondes
    if time.time() - dernier_lancement_boule_feu >= 5:
        lancer_boule_feu()
        dernier_lancement_boule_feu = time.time()
```

FONCTIONNEMENT DU JEU

→ *Apparition et Ramassage des pièces*

- **CREATION DES PIECES DE FAÇON ALEATOIRE | DISPARITION AU BOUT DE 10 SEC**

```
# Fonction pour créer une pièce à une position aléatoire
def creer_piece():
    x = uniform(-10, 10)
    z = uniform(-10, 10)
    piece = Entity(model='quad', texture='piece.png', scale=(1, 1, 1), position=(x, 1, z), collider='box')
    piece.collected = False  # Propriété pour marquer si la pièce a été collectée
    pieces.append(piece)
    invoke(disparaitre_piece, piece, delay=10) # disparaître la pièce après 10 secondes

# Fonction pour faire disparaître la pièce
def disparaitre_piece(piece):
    destroy(piece)
```

- **FONCTION PERMETTANT D'AJOUTER +1 AU SCORE**

```
for piece in pieces:
        if piece and (player.position - piece.position).length() < 0.5 and not piece.collected:
            print("Mario a touché une pièce!")
            son_piece.play()  # Joue le son de la pièce
            piece.collected = True  #
            disparaitre_piece(piece)
            score_piece += 1
            score_pieces_text.text = f"Piece: {score_piece}"  # Mettre à jour le texte du bouton pour afficher le score
```

- **APPARITION DES PIÈCES TOUTE LES 5 SECONDES**

```
# Créer une pièce toutes les 5 secondes
if int(elapsed_time) % 5 == 0 and elapsed_time % 5 < time.dt:
    creer_piece()
```

FONCTIONNEMENT DU JEU

→ *Apparition et Ramassage des champignons*

▪ APPARITION ET DISPARITION DES CHAMPIGNONS

```python
# Fonction pour créer un champignon à une position aléatoire
def creer_champi():
    global champignon, point_C, point_D
    x = uniform(-10, 10)
    z = uniform(-10, 10)
    champignon = Entity(model='quad', texture='champignon.png', scale=(1, 2, 1), position=(x, 1, z), rotation_y=0)

    # Générer un point de départ (A) et un point d'arrivée (B)
    point_C = Vec3(uniform(-10, 10), 1, uniform(-10, 10))
    point_D = Vec3(uniform(-10, 10), 1, uniform(-10, 10))

    invoke(disparaitre_piece, champignon, delay=10) #

# Fonction pour faire disparaître le champignon
def disparaitre_champi(champignon):
    champignon.disable()
```

▪ COLLISION AVEC LE CHAMPIGNON

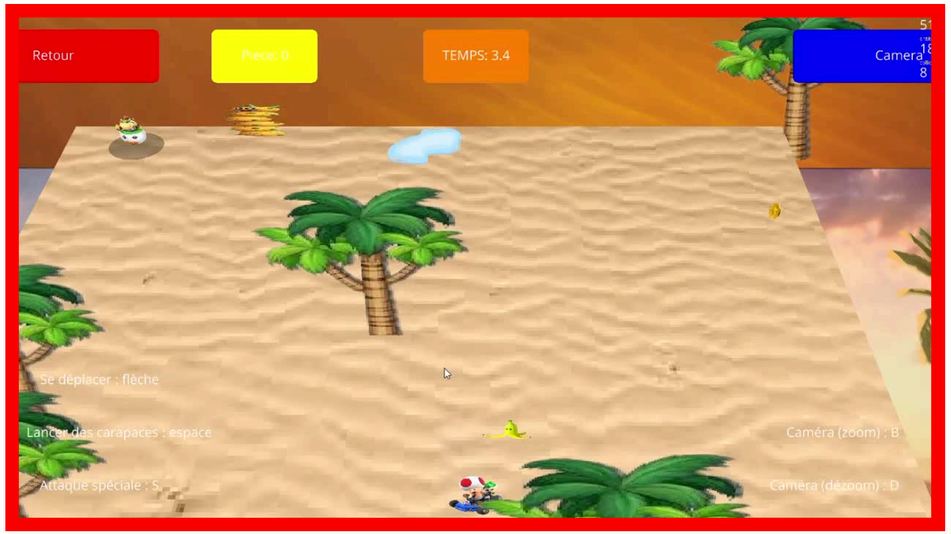
```python
# Vérification de collision entre le joueur et le champignon
        if champignon and (player.position - champignon.position).length() < 1:
            son_acc.play()
            speed = 10
            boost_duration = 5  # Durée du boost en secondes
            boost_time = time.dt  # Enregistrer le temps lorsque le boost est activé
            print("Boost accéléré.")
            destroy(champignon)
```

▪ APPARITION DES CHAMPIGNONS TOUTE LES 5 SECONDES

```python
##Apparaition des champignons toutes les 5 secondes
if time.time() % 5 < time.dt:  # Créer un champi toutes les 5 secondes
    creer_champi()
```

FONCTIONNEMENT DU JEU

→ *Collision avec la bananes*



▪ APPARITION DES BANANES

```
##Apparaition des bananes toutes les 5 secondes
if time.time() % 5 < time.dt:  # Créer une banane toutes les 5 secondes
    creer_banane()
```

▪ COLLISION ENTRE LES PERSONNAGES ET LA BANANES

```
    # Vérifier si une bananes  touche Mario
    for banane in bananes[:]:
        if banane and hasattr(banane, 'position') and banane.enabled:
            if (player.position - banane.position).length() < 1:
                print("Game Over! Mario a été touché par une bananes.")
                sound_explosion.play()
                gameover()
```

▪ CREATION DE LA BANANES

```
# Fonction pour créer une banane à une position aléatoire
def creer_banane():
    x = uniform(-10, 10)
    z = uniform(-10, 10)
    banane = Entity(model='quad', texture='banane.png', scale=(1, 1, 1), position=(x, 0.5, z), collider='box')
    banane.apparue = time.time()
    bananes.append(banane)
    invoke(disparaitre_banane, banane, delay=10)  # Faire disparaître la banane après 10 secondes
```

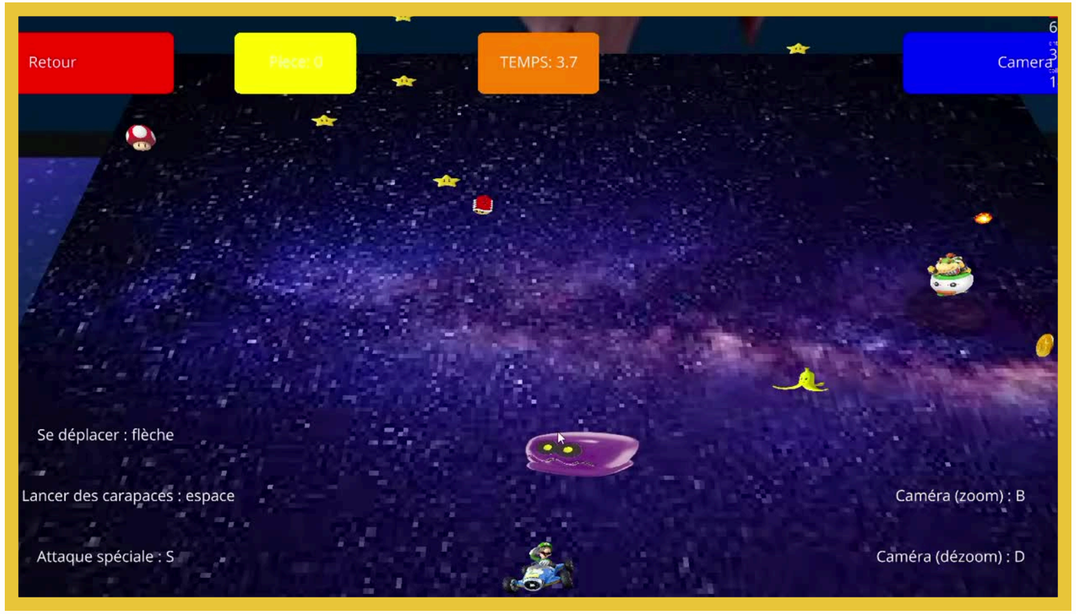
▪ DISPARITION DES BANANES

```
# Fonction pour faire disparaître
def disparaitre_banane(banane):
    if banane in bananes:
        bananes.remove(banane)
        destroy(banane)
```

FONCTIONNEMENT DU JEU

→ *Collision carapace+ ennemi*

- SI LA DISTANCE ENTRE LA CARAPACE ET L'ENNEMI EST INFERIEUR A 1 = L'ENNEMI DISPARAIT PUIS APPARAIT AU BOUT DE 5 SECONDES

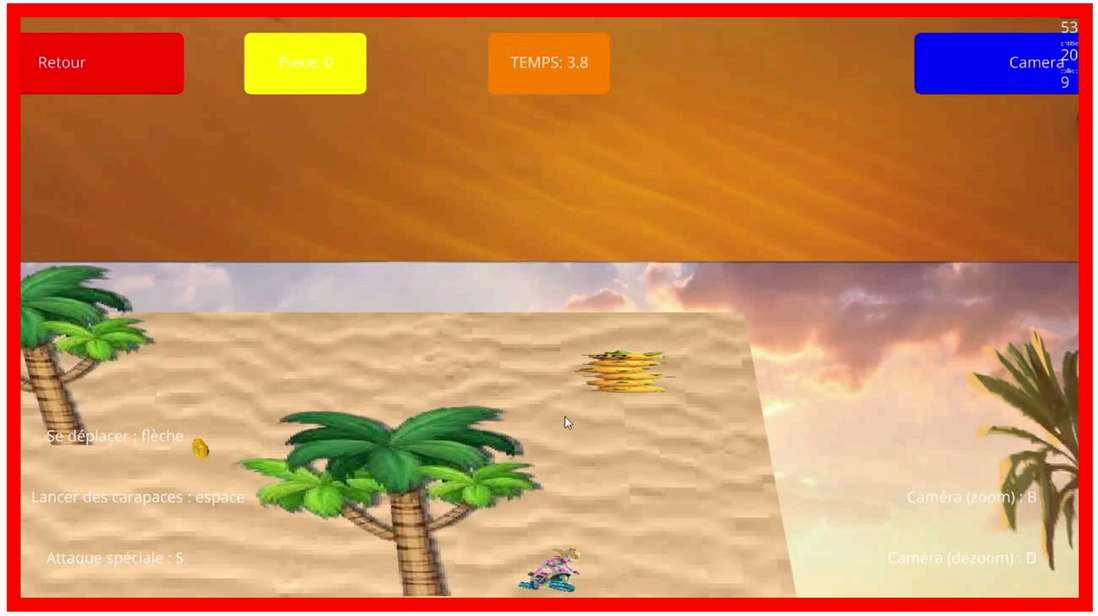



```python
# Vérifier collision entre carapaces et l'ennemi
    for carapace in carapaces[:]:
        if carapace and carapace.enabled:
            if (carapace.position - ennemi.position).length() < 1:
                print("L'ennemi a été touché par une carapace.")
                invoke(detruire_ennemi, ennemi, delay=0.5)
                carapace.disable()

# Fonction pour gérer la disparaition de l'ennemi
def detruire_ennemi(ennemi):
    print("L'ennemi est détruit!")
    sound_explosion.play()
    ennemi.disable()  # Désactive l'ennemi
    invoke(reapparition_ennemi, ennemi, delay=5)
# Fonction pour gérer la réapparition de l'ennemi
def reapparition_ennemi(ennemi):
    ennemi.position = (random.uniform(-10, 10), 1, random.uniform(-10, 10))
    ennemi.enable()
```

FONCTIONNEMENT DU JEU

→ *Collision avec les ennemis*

- **LA DISTANCE ENTRE LES PERSONNAGES ET LES ENNEMIS EST INFERIEUR A 1.**



- **CREATION DE L'ENNEMI**

```
ennemi = Entity(model='quad', texture='goomba.png', scale=1, position=(0,
deplacement_ennemi(ennemi)
```

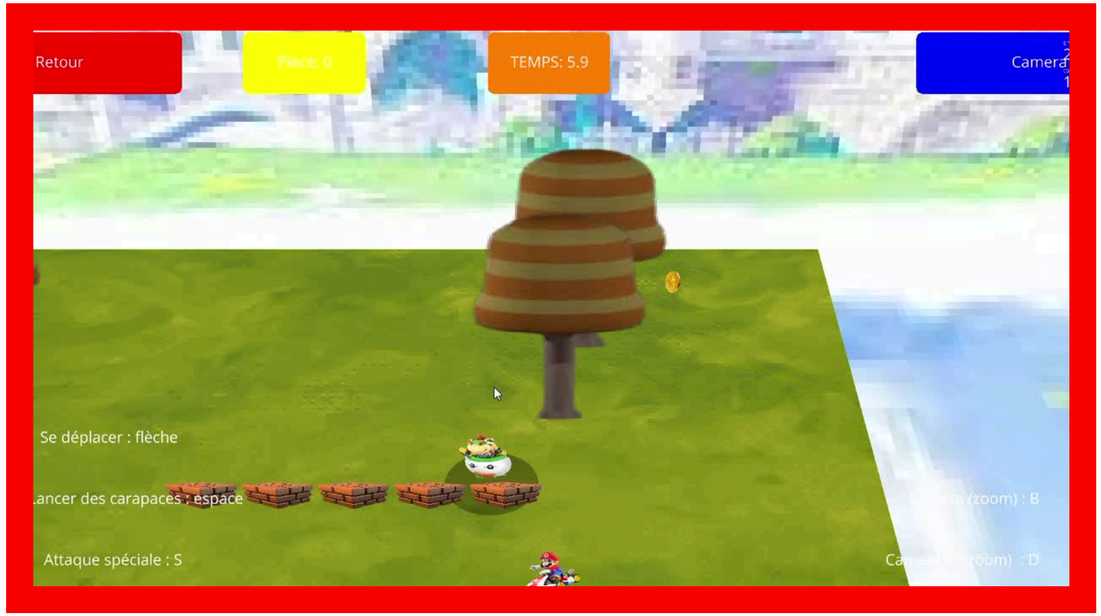
- **COLLISION ENTRE PERSONNAGE ET L'ENNEMI**

```
# Vérification de collision entre Mario et l'ennemi
if ennemi and (player.position - ennemi.position).length() < 1:
    if selected_player in ["Mario", "Luigi", "Peach", "Daisy","Toad"]
        print("Game Over! Mario a été touché par un ennemi.")
        sound_explosion.play()
        gameover()
```

FONCTIONNEMENT DU JEU

→ *Collision avec BowserJr*

- **LA DISTANCE ENTRE LES PERSONNAGES ET BOWSER JR EST INFERIEUR A 1.**

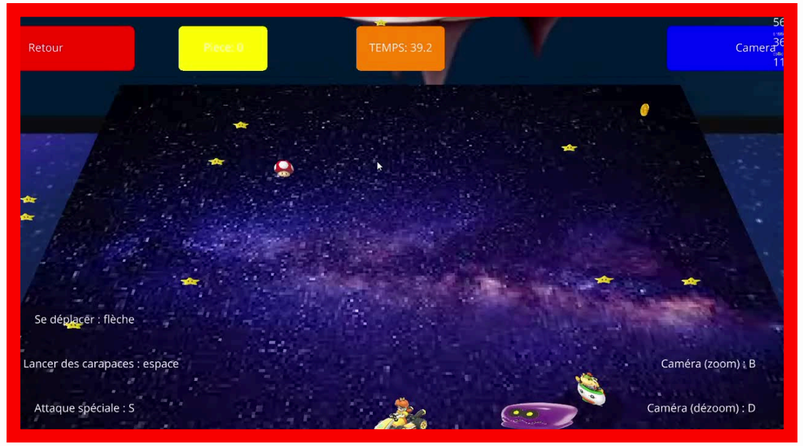


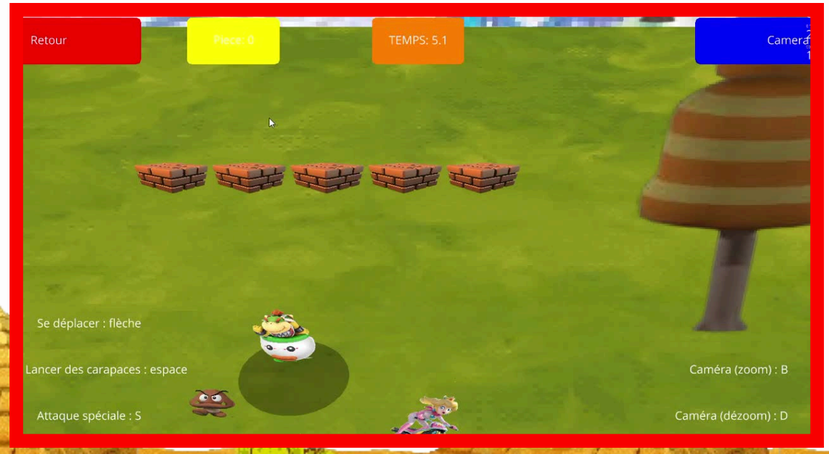

```python
# Vérification de collision entre Mario , Luigi, Peach, Daisy Toad et BowserJr
        if bowserjr and (player.position - bowserjr.position).length() < 1:
            if selected_player in ["Mario", "Luigi", "Peach", "Daisy","Toad"]:
                print("Game Over! Mario a touché BowserJr.")
                sound_explosion.play()
                gameover()
```

FONCTIONNEMENT DU JEU

→ *Pouvoir de chaque personnage*

- **POUR ACTIVER LEUR POUVOIR , LE JOUEUR DOIT APPUYER SUR LA TOUCHE S DU CLAVIER.**






```
def update():
    ### Caractérique des personnages##
    if player:
        speed = 5 * time.dt

        if selected_player == "Toad":
            if held_keys['s']:
                speed *= 2  # Toad roule plus vite lorsqu'on maintient la touche 'S'

            ###Harmonie devient invincible lorsque elle touche Bowser JR (ligne 334)##

                # Peach flotte temporairement lorsqu'on appuie sur 'S'
        if selected_player == "Daisy":
            if held_keys['s'] and not is_jumping:  # Appui sur 'S' pour activer le flotement
                is_jumping = True
                velocity_y = 1  # Vitesse réduite pour flotter
            if is_jumping:
                player.y += velocity_y  # Appliquer la vitesse de flottement
                velocity_y += -0.05  # Appliquer une faible gravité pour rester en l'air plus longtemps
            if player.y <= ground_level:  # Quand Daisy touche le sol
                player.y = ground_level
                is_jumping = False

        # Peach crée une vague de choc avec la touche 'S'
        if selected_player == "Peach":
            if held_keys['s']:  # Lorsque la touche 'S' est maintenue
                create_shockwave(player.position)  # Créer une onde de choc qui repousse BowserJr

        ##Luigi peut se téléporter##
        if selected_player == "Luigi":
            if held_keys['s']:
                teleport_x = player.position.x + random.randint(-10, 10)
                teleport_y = player.position.y + random.randint(-10, 10)
                teleport_y = ground_level
                player.position = Vec3(teleport_x, teleport_y, ground_level)

                S POUVOIRS DES PERSONNAGES##
```

FONCTIONNALITÉS DU JEU

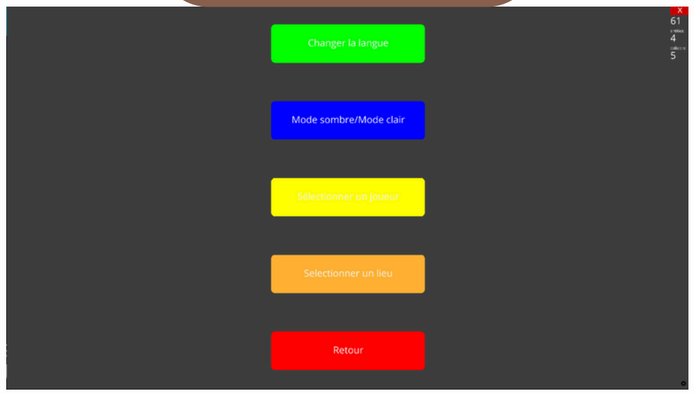
Par ANTON NELCON Steve

FONCTIONNALITÉS DU JEU

→ *Les pages du Jeu*

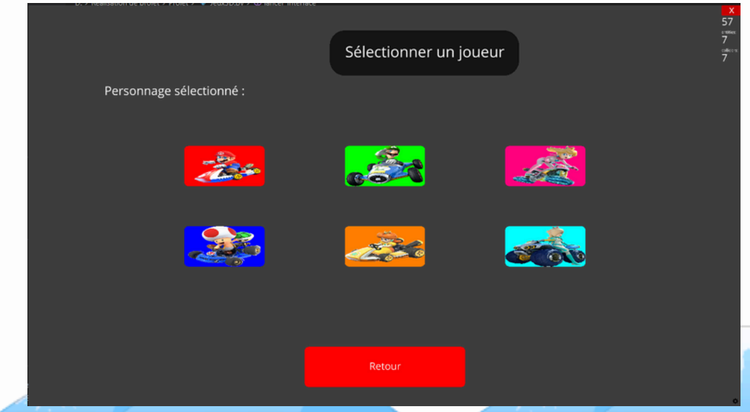
LA PAGE D'ACCUEIL

LA PAGE DE PARAMÈTRE

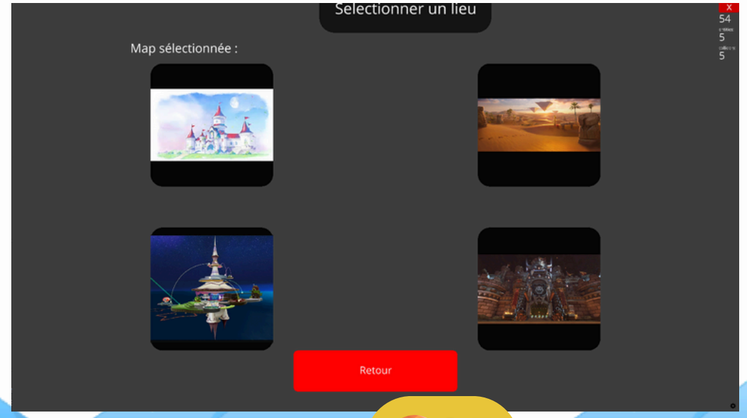


LA PAGE DE JEU

LA SÉLECTION DE PERSONNAGES



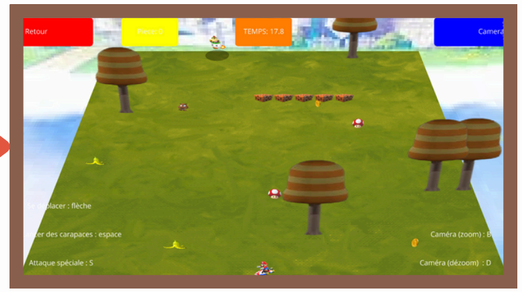
LA SÉLECTION DE MAPS

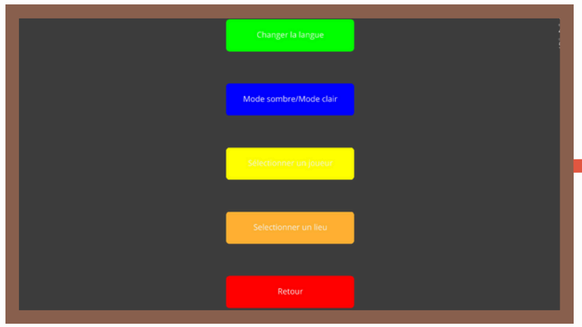


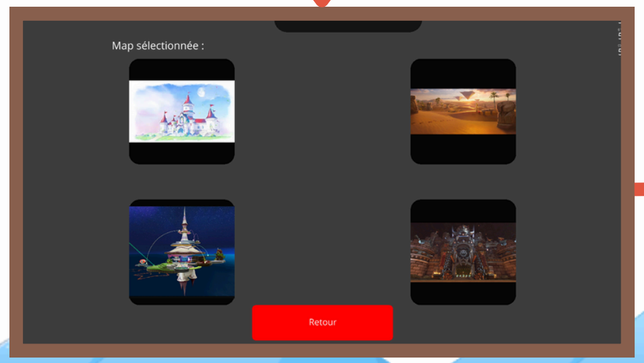
LA PAGE DE FIN

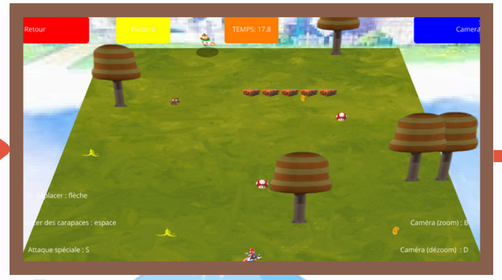
FONCTIONNALITÉS DU JEU

→ *Fonctionnement du Jeu*









FONCTIONNALITÉS DU JEU

→ *Le mode clair/sombre*

FONCTION POUR LE CHANGEMENT ENTRE LE MODE SOMBRE/CLAIR

```
# Fonction mode sombre/clair
def toggle_theme():
    global is_dark_mode
    is_dark_mode = not is_dark_mode
    if is_dark_mode:
        window.color = color.black
        Button.default_color = color.gray
        Button.default_text_color = color.white
    else:
        window.color = color.white
        Button.default_color = color.azure
        Button.default_text_color = color.black
    afficher_parametres()
```

BOUTON POUR ACTIVER LE MODE SOMBRE/CLAIR

```
Button(text=translations[current_language]["toggle_theme"], scale=(0.4, 0.1), position=(0, 0.2),  color=color.blue, on_click=toggle_theme)
```

FONCTIONNALITÉS DU JEU

→ *le changement de langue*

- **LISTE DE TOUTES LES TRADUCTION FRANCE/ANGLAIS**

```
##TRADUCTIONS##
translations = {
    "fr": {"play": "Jouer", "settings": "Paramètres", "quit": "Quitter", "change_language": "Changer la langue","select_map":"Selectionner un lieu",
    "en": {"play": "Play", "settings": "Settings", "quit": "Quit", "change_language": "Change Language","select_map":"Select a map", "toggle_theme":
}
```

- **LE CHANGELENT DE LANGUE**

```
##FONCTION POUR LES PARAMETRES##
def afficher_parametres():
    scene.clear()

    Text("Mario Kart", position=(-0.9, 0), scale=2, color=color.red, origin=(0.5, 0.5))

    Button(text=translations[current_language]["change_language"], scale=(0.4, 0.1), position=(0, 0.4),  color=color.green, on_click=changer_la_lang
    Button(text=translations[current_language]["toggle_theme"], scale=(0.4, 0.1), position=(0, 0.2),  color=color.blue, on_click=toggle_theme)  # Ch
    Button(text=translations[current_language]["select_player"], scale=(0.4, 0.1), position=(0, 0),  color=color.yellow, on_click=afficher_selection
    Button(text=translations[current_language]["select_map"], scale=(0.4, 0.1), position=(0, -0.2),  color=color.orange, on_click=afficher_selection
    Button(text=translations[current_language]["back"], scale=(0.4, 0.1), position=(0, -0.4),  color=color.red, on_click=afficher_menu_principal)  #
```

FONCTIONNALITÉS DU JEU

→ *le changement de personnage*

- **BOUTON PERSONNAGE SELECTIONNÉ DANS LES PARAMETRES DE SELECTION DES PERSONNAGE**

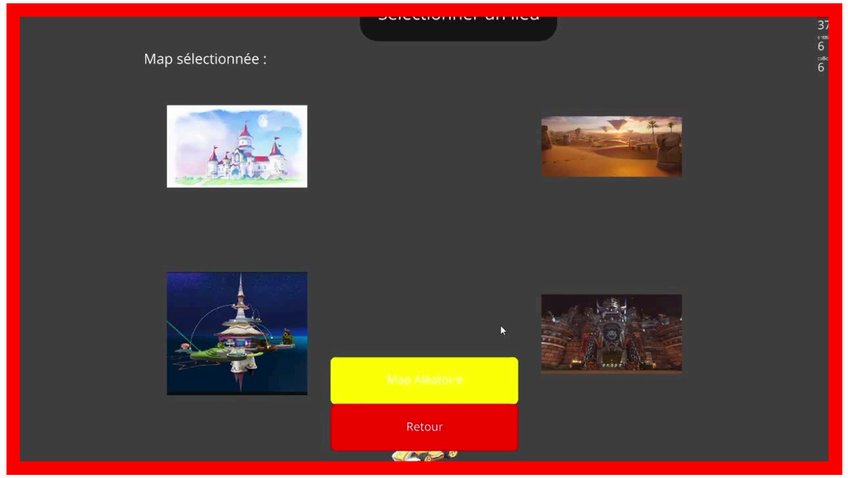
```
def afficher_selection_joueur():
    # Boutons pour choisir les personnages avec événements de survol
    Button(scale=(0.2, 0.1), position=(-0.4, 0.1), color=color.red, icon='mario.png',
           on_click=lambda: choisir_personnage("Mario"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Mario"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))

    Button(scale=(0.2, 0.1), position=(0.0, 0.1), color=color.green, icon='luigi.png',
           on_click=lambda: choisir_personnage("Luigi"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Luigi"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))

    Button(scale=(0.2, 0.1), position=(0.4, 0.1), color=color.pink, icon='peach.png',
           on_click=lambda: choisir_personnage("Peach"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Peach"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))

    Button(scale=(0.2, 0.1), position=(-0.4, -0.1), color=color.blue, icon='toad.png',
           on_click=lambda: choisir_personnage("Toad"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Toad"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))

    Button(scale=(0.2, 0.1), position=(0.0, -0.1), color=color.orange, icon='daisy.png',
           on_click=lambda: choisir_personnage("Daisy"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Daisy"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))

    Button(scale=(0.2, 0.1), position=(0.4, -0.1), color=color.cyan, icon='Harmonie.png',
           on_click=lambda: choisir_personnage("Harmonie"),
           on_mouse_enter=lambda: mettre_a_jour_texte_survol("Harmonie"),
           on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))
```

PERSONNAGE SELECTIONNÉ INSÉRER DANS LE JEU

```
###CHOIX DES PERSONNAGES##
if selected_player == "Mario":
    player = Entity(model='quad', texture='mario.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Mario
elif selected_player == "Luigi":
    player = Entity(model='quad', texture='luigi.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Luigi
elif selected_player == "Peach":
    player = Entity(model='quad', texture='peach.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Peach
elif selected_player == "Toad":
    player = Entity(model='quad', texture='toad.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Toad
elif selected_player == "Daisy":
    player = Entity(model='quad', texture='daisy.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Daisy
elif selected_player == "Harmonie":
    player = Entity(model='quad', texture='harmonie.png', scale=(1, 2, 1), collider='box', position=(0, 1, -10))  # Harmonie
```

FONCTIONNALITÉS DU JEU

→ *le changement de map*



- **INFORMATIONS DU CONTENU DANS CHAQUE MAPS**

```
### CHOIX DE LA MAP#####
if selected_map == 1:  # Le royaume Champignon (Map 1)
    sky = Entity(model='cube', texture='paysage.png', scale=100, double_sided=True)  # Ciel pour la Map 1
    ground = Entity(model='plane', texture='grass', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 1
    trees = []
    for i in range(5):
        tree = Entity(model='quad',texture='tree.png',scale=10, position=(random.uniform(-10, 10), 4, random.uniform(-10, 10)))
    trees.append(tree)

    briques = []
    spacing = 0.02
    briqueespace = 5
    for i in range(5):
        brique = Entity(model='quad',texture='brique.png',scale=1,position=(i * (1 + spacing),briqueespace , 0)  )
    briques.append(brique)

    ennemi = Entity(model='quad', texture='goomba.png', scale=1, position=(0, 0.5, 0))
    deplacement_ennemi(ennemi)


elif selected_map == 2:  # Sarasaland (Map 2)
    sky = Entity(model='cube', texture='dessert.png', scale=100, double_sided=True)  # Ciel pour la Map 2
    ground = Entity(model='plane', texture='sand.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 2
    lacs = []
    for i in range(2):
        lac= Entity(model='plane',texture='lac.png',scale=2, position=(random.uniform(-10, 10), 0.1, random.uniform(-10, 10)))
    lacs.append(lac)

    tropics=[]
    for i in range(5):
        tropic= Entity(model='quad',texture='tropic.png',scale=10, position=(random.uniform(-10, 10), 3, random.uniform(-10, 10)))
    tropics.append(tropic)

    ennemi = Entity(model='quad', texture='pokey.png', scale=2, position=(0, 1, 4))
    deplacement_ennemi(ennemi)
```

```
elif selected_map == 3:  # Univers (Map 3)
    sky = Entity(model='cube', texture='galaxie.png', scale=100, double_sided=True)  # Ciel pour la Map 3
    ground = Entity(model='plane', texture='sol.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 3
    stars = []
    for i in range(20):
        star = Entity(model='quad',texture='starts.png',scale=0.5,  position=(random.uniform(-10, 10), 10, random.uniform(-10, 10)))
    stars.append(star)
    ennemi = Entity(model='quad', texture='gloobe.png', scale=2, position=(0, 1, 4))
    deplacement_ennemi(ennemi)


elif selected_map == 4:  # Chateau Bowser(Map 4)
    sky = Entity(model='cube', texture='chateau.png', scale=100, double_sided=True)  # Ciel pour la Map 4
    ground = Entity(model='plane', texture='dea.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 4
    tours=[]
    for i in range(5):
        tour= Entity(model='quad',texture='tour.png',scale=15, position=(random.uniform(-10, 10), 3, random.uniform(-10, 10)))
    tours.append(tour)
    ennemi = Entity(model='quad', texture='Roi BOO.png', scale=3, position=(5, 5, 5))
    deplacement_ennemi(ennemi)
```

- **BOUTON POUR SELECTIONNER LA MAPS**

```
# Boutons pour choisir les maps
    Button(scale=(0.3, 0.3), position=(-0.4, 0.2), icon='paysage.png', on_click=lambda: choisir_map(1),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Le royaume Champignon"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(0.4, 0.2), icon='dessert.png', on_click=lambda: choisir_map(2),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Sarasaland"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(-0.4, -0.2), icon='galaxie.png', on_click=lambda: choisir_map(3),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Observatoire de la Comète"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(0.4, -0.2), icon='chateau.png', on_click=lambda: choisir_map(4),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Chateau de Bowser"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))
```

FONCTIONNALITÉS DU JEU

→ *Aléatoire*

- **INFORMATIONS DU CONTENU DANS CHAQUE MAPS**

```
### CHOIX DE LA MAP#####
if selected_map == 1:  # Le royaume Champignon (Map 1)
    sky = Entity(model='cube', texture='paysage.png', scale=100, double_sided=True)  # Ciel pour la Map 1
    ground = Entity(model='plane', texture='grass', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 1
    trees = []
    for i in range(5):
        tree = Entity(model='quad',texture='tree.png',scale=10, position=(random.uniform(-10, 10), 4, random.uniform(-10, 10)))
    trees.append(tree)

    briques = []
    spacing = 0.02
    briqueespace = 5
    for i in range(5):
        brique = Entity(model='quad',texture='brique.png',scale=1,position=(i * (1 + spacing),briqueespace , 0)  )
    briques.append(brique)

    ennemi = Entity(model='quad', texture='goomba.png', scale=1, position=(0, 0.5, 0))
    deplacement_ennemi(ennemi)


elif selected_map == 2:  # Sarasaland (Map 2)
    sky = Entity(model='cube', texture='dessert.png', scale=100, double_sided=True)  # Ciel pour la Map 2
    ground = Entity(model='plane', texture='sand.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 2
    lacs = []
    for i in range(2):
        lac= Entity(model='plane',texture='lac.png',scale=2, position=(random.uniform(-10, 10), 0.1, random.uniform(-10, 10)))
    lacs.append(lac)

    tropics=[]
    for i in range(5):
        tropic= Entity(model='quad',texture='tropic.png',scale=10, position=(random.uniform(-10, 10), 3, random.uniform(-10, 10)))
    tropics.append(tropic)

    ennemi = Entity(model='quad', texture='pokey.png', scale=2, position=(0, 1, 4))
    deplacement_ennemi(ennemi)
```

```
elif selected_map == 3:  # Univers (Map 3)
    sky = Entity(model='cube', texture='galaxie.png', scale=100, double_sided=True)  # Ciel pour la Map 3
    ground = Entity(model='plane', texture='sol.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 3
    stars = []
    for i in range(20):
        star = Entity(model='quad',texture='starts.png',scale=0.5,  position=(random.uniform(-10, 10), 10, random.uniform(-10, 10)))
    stars.append(star)
    ennemi = Entity(model='quad', texture='gloobe.png', scale=2, position=(0, 1, 4))
    deplacement_ennemi(ennemi)


elif selected_map == 4:  # Chateau Bowser(Map 4)
    sky = Entity(model='cube', texture='chateau.png', scale=100, double_sided=True)  # Ciel pour la Map 4
    ground = Entity(model='plane', texture='dea.png', collider='mesh', scale=(20, 20, 20))  # Sol pour la Map 4
    tours=[]
    for i in range(5):
        tour= Entity(model='quad',texture='tour.png',scale=15, position=(random.uniform(-10, 10), 3, random.uniform(-10, 10)))
    tours.append(tour)
    ennemi = Entity(model='quad', texture='Roi BOO.png', scale=3, position=(5, 5, 5))
    deplacement_ennemi(ennemi)
```

- **BOUTON POUR SELECTIONNER LA MAPS**

```
# Boutons pour choisir les maps
    Button(scale=(0.3, 0.3), position=(-0.4, 0.2), icon='paysage.png', on_click=lambda: choisir_map(1),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Le royaume Champignon"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(0.4, 0.2), icon='dessert.png', on_click=lambda: choisir_map(2),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Sarasaland"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(-0.4, -0.2), icon='galaxie.png', on_click=lambda: choisir_map(3),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Observatoire de la Comète"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol("")),

    Button(scale=(0.3, 0.3), position=(0.4, -0.2), icon='chateau.png', on_click=lambda: choisir_map(4),
        on_mouse_enter=lambda: mettre_a_jour_texte_survol("Chateau de Bowser"),
        on_mouse_exit=lambda: mettre_a_jour_texte_survol(""))
```

FONCTIONNALITÉS DU JEU

→ *La temps*

- **TEXTE POUR AFFICHER LE TGEMPS**

```
Text(text=f"Temps écoulé: {elapsed_time} secondes", origin=(0, 0), scale=1.5, color=color.green, position=(0, 0))
```

- **AFFICHE LE TIMER**

```
# Afficher le timer
elapsed_time = time.time() - start_time
timer_text.text = f"{'TEMPS' if current_language == 'fr' else 'TIME'}: {elapsed_time:.1f}"
```

FONCTIONNALITÉS DU JEU

→ *La caméra*

- **BOUTON DE LA CAMERA**

```
camera_button = Button(text="Camera", scale=(0.4, 0.1), position=(0.8, 0.4), color=color.blue, on_click=toggle_camera_view) # Bouton pour change
```

- **FONCTION DE LA CAMERA**

```
##Fonction pour la caméra
def toggle_camera_view():
    global is_first_person
    is_first_person = not is_first_person

    if is_first_person:
        # Vue subjective : la caméra suit la position de Mario
        camera.parent = player
        camera.position = (0, 2, 0)  # Position au niveau des yeux de Mario
        camera.rotation = (0, 0, 0)  # Orientée vers l'avant
    else:
        # Vue classique : caméra dézoomée
        camera.parent = None
        camera.position = (0, 10, -10)
        camera.rotation_x = 35
```

- **TOUCHE CLAVIER**

```
    # Contrôle du zoom avec B et D
    if held_keys['b']:
        camera.position += Vec3(0, -0.2, 0.2)
    if held_keys['d']:
        camera.position += Vec3(0, 0.2, -0.2)
```

- **SE DEPLACER AVEC LA CAMERA**

```
# Déplacer la caméra pour dézoomer
camera.position = (0, 10, -10)
camera.rotation_x = 35
camera.parent = player
global is_first_person    # Initialisation de la vue
is_first_person = False
```

FONCTIONNALITÉS DU JEU

→ *Les autre pages*

- **LA PAGE D'ACCUEIL**

```
def afficher_menu_principal():
    scene.clear()
    musique_fond.play()

    Text("Mario Kart", position=(0.1, 0.4), scale=2, color=color.red, origin=(0.5, 0.5))
    Text("Jeu d'arcade", position=(0.1, 0.35), scale=1.5, color=color.yellow, origin=(0.5, 0.5))

    image = Entity(model='quad', texture='fond.png', scale=(4, 4), position=(-5, 0))
    image_bowser = Entity(model='quad', texture='fond_bowserjr.png', scale=(4, 4), position=(5, 0))

    texture_toggle_image = True
    texture_toggle_bowser = True

    # Fonction pour changer l'image de manière alternée
    def changer_image_survol():
        nonlocal texture_toggle_image
        if texture_toggle_image:
            image.texture = 'fond.png'
        else:
            image.texture = 'peachmoto.png'

        texture_toggle_image = not texture_toggle_image
        invoke(changer_image_survol, delay=3)

    def changer_image():
        nonlocal texture_toggle_bowser
        if texture_toggle_bowser:
            image_bowser.texture = 'fond_bowserjr.png'
        else:
            image_bowser.texture = 'bowserboxe.png'

        texture_toggle_bowser = not texture_toggle_bowser
        invoke(changer_image, delay=3)
    changer_image_survol()
    changer_image()
```

- **LES PAGES (GAME OVER-GAME END)**

```
758 def gameover():
762     Text("Projet réalisé par ", position=(-0.50, 0.5), scale=1, color=color.white)
763     Text("ANTON NELCON Steve", position=(-0.29, 0.5), scale=1, color=color.white, bold=True)
764     Text(" , étudiant en L3 ISEI à l'Université de Paris 8!", position=(-0.020,0.5), scale=1, color=color.white)
765
766     Button(scale=(0.6, 0.6), position=(-0.6, -0.1), color=color.clear, icon='gameo.png', border=False)
767     Button(scale=(0.6, 0.6), position=(0.6, -0.1), color=color.clear, icon='bowserfin.png', border=False)
768
769     Text(text="Game Over !", position=(0, 0.3), scale=2.5, origin=(0, 0), color=color.red)
770     Text(text="Bowser Jr vous a vaincu !", position=(0,0.7 - 0.5), scale=1.5, origin=(0, 0), color=color.white)
771     Text(text=f"Pièces collectées: {score_piece}", origin=(0, 0), scale=1.5, color=color.yellow, position=(0, 0.1))
772     Text(text=f"Temps écoulé: {elapsed_time} secondes", origin=(0, 0), scale=1.5, color=color.green, position=(0, 0.05))
773     Button(text="Rejouer", scale=(0.3, 0.1), position=(0, -0.2), color=color.green, on_click=rejouer)
774     Button(text="Quitter", scale=(0.3, 0.1), position=(0, -0.4), color=color.red, on_click=quitter)
775
776
777
778 def game_end():
779     elapsed_time = int(time.time() - start_time)
780     scene.clear()
781
782     Text("Projet réalisé par ", position=(-0.50, 0.5), scale=1, color=color.white)
783     Text("ANTON NELCON Steve", position=(-0.29, 0.5), scale=1, color=color.white, bold=True)
784     Text(" , étudiant en L3 ISEI à l'Université de Paris 8!", position=(-0.020,0.5), scale=1, color=color.white)
785
786     Button(scale=(0.6, 0.6), position=(-0.6, -0.1), color=color.clear, icon='ami.png', border=False)
787     Button(scale=(0.6, 0.6), position=(0.6, -0.1), color=color.clear, icon='bowserpleure.png', border=False)
788
789     Text(text="Bien Joué !", position=(0, 0.3), scale=2.5, origin=(0, 0), color=color.gold)
790     Text(text="Vous avez tué Bowser Jr !", position=(0, 0.25), scale=1.5, origin=(0, 0), color=color.red)
791     Text(text=f"Pièces collectées: {score_piece}", origin=(0, 0), scale=1.5, color=color.yellow, position=(0, 0.05))
792     Text(text=f"Temps écoulé: {elapsed_time} secondes", origin=(0, 0), scale=1.5, color=color.green, position=(0, 0))
793
794     Button(text="Rejouer", scale=(0.3, 0.1), position=(0, -0.2), color=color.azure, text_color=color.black, on_click=rejouer)
795     Button(text="Quitter", scale=(0.3, 0.1), position=(0, -0.4), color=color.red, text_color=color.white, on_click=quitter)
796
```

FONCTIONNALITÉS DU JEU

→ *Les sons*

- **CREATION DE CHAQUE SONS**

```
##MUSQIUE##
musique_fond = Audio('voiture.wav',autoplay=True,volume=0.5)
sound_effect = Audio('carapace.wav', loop=False, autoplay=False)
sound_fire = Audio('fire.wav', loop=False, autoplay=False)
son_piece = Audio('piece.wav', loop=False, autoplay=False)
sound_explosion= Audio('explosion.wav', loop=False, autoplay=False)
son_acc= Audio('acceleration.mp3', loop=False, autoplay=False)
```

- **ACTIVER LE SON**

```
def afficher_menu_principal():
    global image, quitter_butt
    scene.clear()
    musique_fond.play()
```

- **DESACTIVER LE SON**

```
def lancer_interface():
    global player, selec
    global sky, ground,

    scene.clear()
    musique_fond.stop()
```

LE CODE

Par ANTON NELCON Steve

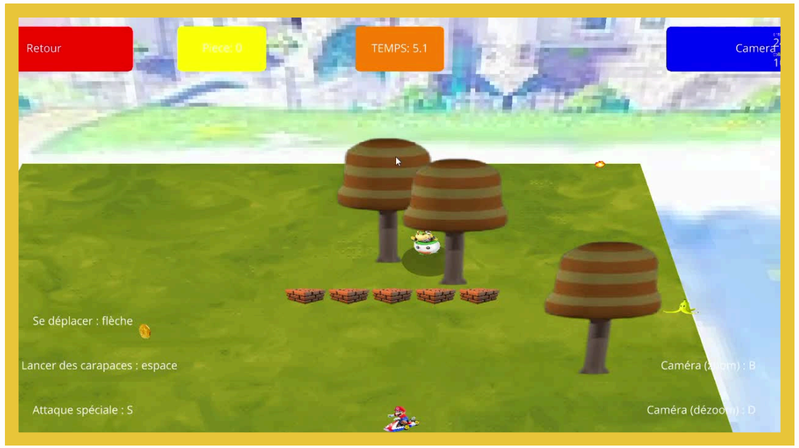
LE CODE
→ *Les variables*

- **LES VARIABLES**

```
##VARIABLE##
current_language = "fr"
is_dark_mode = True
dernier_lancement_boule_feu = 0
dernier_champignon = 0
player = None
bowserjr = None
point_A = None
point_B = None
point_C = None
point_D = None
champignon = None
pieces = []
carapaces = []
bananes = []
score_piece=0
selected_player = "Mario"
is_first_person = False  # Commence par la vue classique
selected_map = 1
is_jumping = False
velocity_y = 0
space_press_count = 0
last_press_time = 0
ground_level = 1
boost_time = 0  # Le temps depuis lequel le boost a commencé
boost_duration = 3  # Durée du boost en secondes
boules_de_feu = []
dernier_lancement_boule_feu = time.time()
```

LE CODE

→ *le déplacement de Bowser JR*

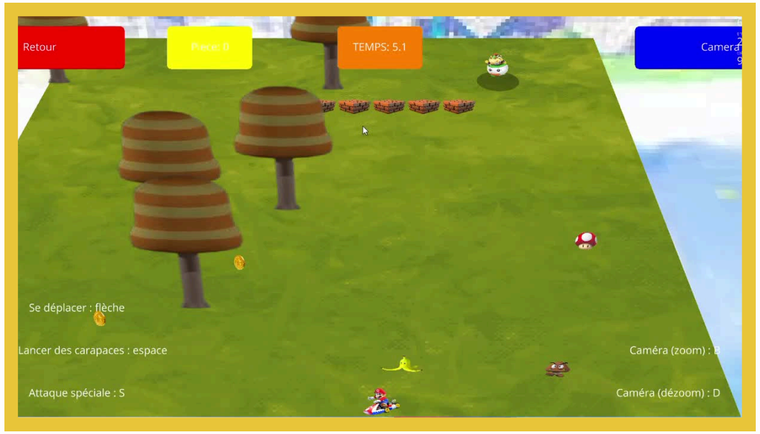


- **BOWSERJR SE DEPLACE ALEATOIREMENT**

```
# Déplacement de BowserJr entre deux points
if bowserjr:
    direction = (point_B - point_A).normalized()
    speed_bowserjr = 1
    bowserjr.position += direction * speed_bowserjr * time.dt

    if (bowserjr.position - point_A).length() >= (point_B - point_A).length():
        point_A = point_B
        point_B = Vec3(uniform(-10, 10), 1, uniform(-10, 10))

    bowserjr.position = Vec3(
        clamp(bowserjr.position.x, -10, 10),
        clamp(bowserjr.position.y, 0.5, 2),
        clamp(bowserjr.position.z, -10, 10)
    )
    bowserjr.position += Vec3(0, math.sin(time.time() * 4) * 0.05, 0)
    bowserjr.rotation_y += 1 * time.dt
```

LE CODE

→ *le déplacement des ennemis*



- **ENNEMI SE DEPLACE ALEATOIREMENT**

```
def deplacement_ennemi(ennemi):
    nouveau_point = Vec3(random.uniform(-10, 10), ennemi.y, random.uniform(-10, 10))
    ennemi.animate_position(nouveau_point, duration=0.5, curve=curve.linear)
    invoke(deplacement_ennemi, ennemi, delay=2.5)
```

LE CODE
→ *les animations*

- **LE CHANGEMENT D'IMAGE**

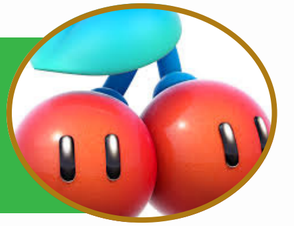
```
# Fonction pour changer l'image de manière alternée
def changer_image_survol():
    nonlocal texture_toggle_image
    if texture_toggle_image:
        image.texture = 'fond.png'
    else:
        image.texture = 'peachmoto.png'

    texture_toggle_image = not texture_toggle_image
    invoke(changer_image_survol, delay=3)

def changer_image():
    nonlocal texture_toggle_bowser
    if texture_toggle_bowser:
        image_bowser.texture = 'fond_bowserjr.png'
    else:
        image_bowser.texture = 'bowserboxe.png'

    texture_toggle_bowser = not texture_toggle_bowser
    invoke(changer_image, delay=3)
changer_image_survol()
changer_image()
```

DÉMONSTRATION DU JEU

Par ANTON NELCON Steve

DÉMONSTRATION DU JEU

→ En Vidéo



Démonstration sur Visual Studio Code

MES RESSENTIE PERSONNEELES

Par ANTON NELCON Steve

MES IMPRESSIONS PERSONNELLES

→ *Ce que j'ai apprécié*

Meilleur projet

Choix du projet

Développer ce jeu

Appendre un nouveau langage

Utiliser Githib

MES IMPRESSIONS PERSONNELLES

→ *Ce que je n'ai pas apprécié*

La durée du Projet

Travailler en Groupe

MES IMPRESSIONS PERSONNELLES

→ *Mes facilités*

Être autonome

Chercher sur Internet

Utiliser les interface

Avoir de l'imagination

MES IMPRESSIONS PERSONNELLES

→ *Mes diffultés*

Travailler en Groupe

Utiliser un nouveau langage

Respecter le temps

Utiliser Githib

CONCLUSION

concluui

Pourquoi j'ai choisi de faire CE PROJET ?

RÉALISATION DE PROJET N°1

MERCI DE VOTRE ECOUTE

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS

Avez-vous des questions ?

50

Licence 3 ISEI

Par ANTON NELCON Steve

8 JANVIER 2024