

## **Rapport de Projet :**

Jeu en Python inspiré de Mario Bros : Contrôle  
du personnage par la caméra et publication sur  
YouTube

## **Rapport final**

Ben ammar Iliass

Professeur :SALEH IMAD

Doctorant : MEKOUAR YUCEF



## Table des matières

Préambule .....	2
Les technologies.....	<b>Erreur ! Signet non défini.</b>
Architecture du jeu .....	5
L'upload de la vidéo .....	6
Le pseudo .....	7
Le Menu.....	8
Le Score et la base de données .....	9
Le Jeu .....	10
La victoire .....	12
Conclusion .....	13

## Préambule

Dans ce projet, j'ai utilisé différentes librairies telles que arcade, tkinter, pyscreenrec, et opencv2 pour créer un jeu en python inspiré de Mario Bros. Le jeu consiste à éviter des obstacles tout en courant contre la montre pour atteindre un coffre final.

J'ai utilisé tkinter pour demander le pseudo du joueur, pyscreenrec pour enregistrer la partie, et l'API Youtube pour uploader la vidéo. Pour gérer les scores, j'ai utilisé une base de données MySQL.

J'ai également utilisé Tiled pour créer les maps et les tuiles de mon jeu. Grâce à la librairie opencv2, j'ai ajouté une fonctionnalité de contrôle de personnage avec la tête, qui permet au joueur de contrôler les déplacements du personnage en suivant les mouvements de sa tête.

Cela permet un niveau d'immersion supplémentaire pour le joueur et rend l'expérience de jeu plus immersif et interactive. Ce projet a été réalisé dans le cadre du cours d'I.O.T, sous la direction du professeur Imad Saleh.

## Les technologies

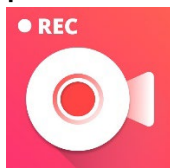
Pour ce projet, j'ai choisi d'utiliser la librairie Arcade pour la création du jeu en lui-même. Arcade est une librairie open-source pour Python qui facilite la création de jeux 2D en fournissant un ensemble complet de fonctionnalités pour dessiner des formes, gérer les animations, et gérer les entrées utilisateur. Elle est facile à utiliser et permet de créer des jeux rapidement et efficacement.



J'ai également choisi d'utiliser la librairie tkinter pour demander le pseudo du joueur. Tkinter est une librairie intégrée à Python qui permet de créer des interfaces graphiques pour les applications. Elle est simple à utiliser et permet de créer des fenêtres, des boutons, des champs de saisie et d'autres éléments graphiques pour les applications.



Pour enregistrer la partie, j'ai utilisé la librairie pyscreenrec. Cette librairie permet d'enregistrer l'écran de l'ordinateur en utilisant la webcam, elle est facile à utiliser et facilite l'enregistrement de la partie pour la sauvegarde ou pour la partager avec d'autres personnes.



J'ai utilisé l'API Youtube pour uploader la vidéo de la partie enregistrée. Cette API permet de connecter l'application à YouTube et de télécharger des vidéos sur le compte du joueur.



Enfin, pour gérer les scores, j'ai utilisé une base de données MySQL. MySQL est un système de gestion de base de données relationnelle populaire qui permet de stocker et de gérer les données de manière efficace. Cela permet de stocker les scores et de les afficher pour que les joueurs puissent les comparer et s'améliorer.



Enfin, pour gérer les maps et les tuiles de mon jeu, j'ai utilisé Tiled. Tiled est un éditeur de maps pour jeux vidéo qui permet de créer des maps en utilisant des tuiles. Il est facile à utiliser et permet de créer des maps rapidement et efficacement au format .tmx.



Enfin pour contrôler le personnage avec la tête j'ai utilisé opencv2, cette librairie permet de détecter les mouvements et la reconnaissance faciale pour suivre les mouvements de la tête du joueur et les utiliser pour contrôler les déplacements du personnage dans le jeu, ce qui ajoute un niveau d'immersion supplémentaire pour le joueur et rend l'expérience de jeu plus immersif et interactive.



## Architecture du jeu

L'architecture de mon projet est basée sur l'utilisation de différentes "vues" qui sont gérées par la librairie Arcade. Chaque vue correspond à une partie spécifique de mon jeu, comme le menu principal, la vue de jeu, la vue des scores, etc.

La classe `JeuView` gère la partie de jeu elle-même. Elle utilise la librairie `pyscreenrec` pour enregistrer la partie en cours et utilise la fonction `datetime` pour nommer la vidéo enregistrée en fonction de la date et de l'heure actuelles. Elle stocke également le nom de la vidéo dans un fichier texte pour utilisation ultérieure.

La classe `MenuView` gère l'affichage du menu principal. Elle utilise la méthode `load_texture` de Arcade pour charger les images des boutons play, score et quit, puis utilise la méthode `on_draw` pour les afficher à l'écran. Elle utilise également la méthode `on_mouse_press` pour gérer les clics de souris et rediriger l'utilisateur vers les vues correspondantes.

La classe `VictoireView` gère la vue de la victoire dans mon jeu. Elle s'active lorsque le joueur atteint le coffre à la fin de la map. Elle affiche les informations relatives à la victoire, comme le temps final obtenu, le score et le message de félicitations.

Les autres classes de vue fonctionnent de manière similaire, en utilisant les fonctionnalités de la librairie Arcade pour gérer l'affichage et la gestion des événements utilisateur pour naviguer entre les différentes parties du jeu.

## L'upload de la vidéo

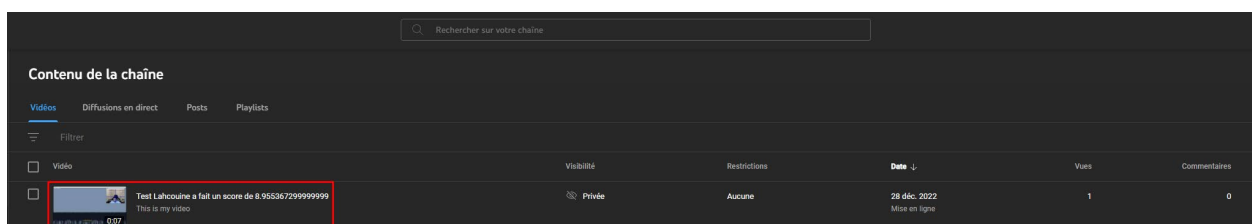
La fonction `upload_video` que je vais utiliser dans mon projet permet de télécharger la vidéo de la partie enregistrée sur YouTube. Elle prend en entrée le nom de fichier de la vidéo, le pseudo du joueur et le score obtenu.

La première étape de cette fonction est de définir les jetons d'accès, qui sont utilisés pour authentifier l'application auprès de YouTube. Ces jetons d'accès sont ensuite utilisés pour créer un objet `Credentials` qui est utilisé pour créer une chaîne de découverte de service YouTube.

Ensuite, j'ai défini les propriétés de la vidéo que je souhaite uploader, telles que le titre, la description et les tags. Le titre de la vidéo est généré automatiquement en utilisant le pseudo du joueur et le score obtenu.

Ensuite, j'ai chargé le fichier vidéo en utilisant la classe `MediaFileUpload` de la librairie `google-auth-oauthlib`, qui permet de charger le fichier vidéo en utilisant des chunks de taille définie, ce qui permet de reprendre l'upload en cas de coupure de connexion.

Enfin, j'ai créé une requête de création de vidéo en utilisant l'objet `youtube` créé précédemment et en spécifiant les propriétés de la vidéo, et j'ai envoyé la requête pour uploader la vidéo sur YouTube. En retour, la fonction retourne l'ID de la vidéo uploadée pour pouvoir la récupérer et la partager par la suite.



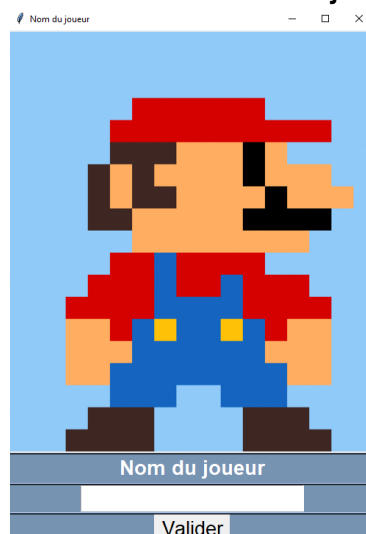
## Le pseudo

La fonction `popup()` que j'ai créée permet de demander le nom du joueur avant de démarrer la partie. Elle utilise la librairie `tkinter` pour créer une fenêtre qui contient une image de Mario, un titre, une entrée pour le nom du joueur et un bouton de validation.

Lorsque la fonction est appelée, elle crée une fenêtre avec le titre "Nom du joueur" et la taille 500x700 pixels. Il est possible de redimensionner la fenêtre. Ensuite, j'ai chargé une image en mémoire en utilisant la classe `PhotoImage` de `tkinter`, cette image est utilisée pour afficher un aperçu de l'image de Mario dans la fenêtre. J'ai ensuite créé un label pour contenir l'image, un frame pour contenir le titre, un label pour afficher le titre, un frame pour contenir l'entrée du nom, une entrée pour le nom du joueur.

Lorsque l'utilisateur clique sur le bouton "Valider", la fonction `getNom()` est appelée. Cette fonction utilise la méthode `get()` de l'objet `tkinter Entry` pour récupérer le nom entré par le joueur et l'enregistrer dans une variable globale `nomJoueur`. Enfin, la fenêtre est fermée en utilisant la méthode `destroy()` de l'objet `tkinter Tk`.

Avec cette fonction, le nom du joueur est récupéré de manière interactive et est ensuite utilisé pour personnaliser le titre de la vidéo uploadée, pour enregistrer le score dans la base de données et pour d'autres utilisations dans le jeu.





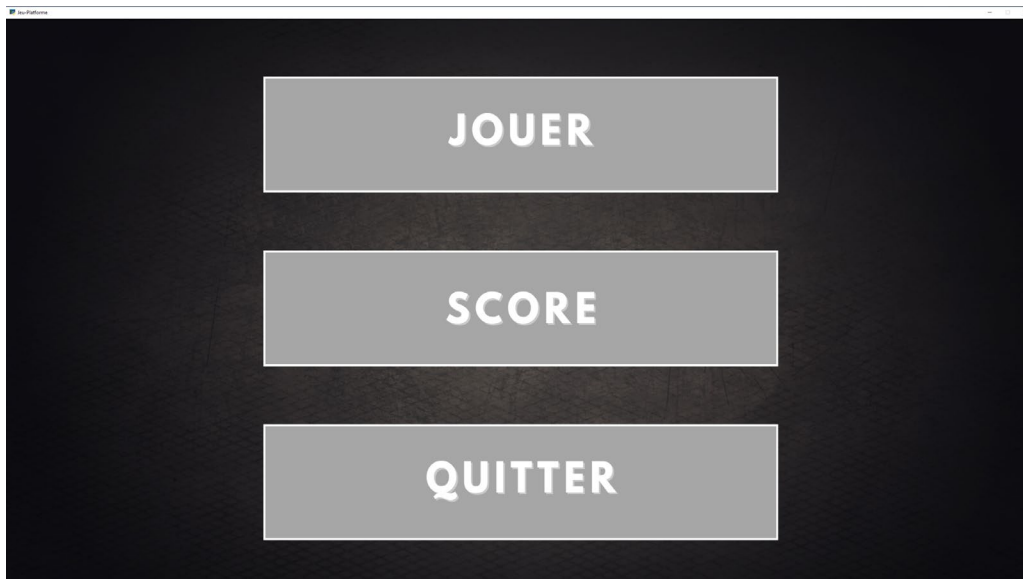
## Le Menu

En utilisant la librairie Arcade, j'ai créé une classe MenuView pour gérer l'affichage de mon menu principal. Dans la méthode setup, j'ai chargé les images nécessaires pour l'affichage de mon menu, comme l'arrière-plan, l'image du bouton de jeu, l'image du bouton de score et l'image du bouton de quitter.

Dans la méthode on\_draw, j'ai utilisé les dimensions de ma fenêtre pour afficher ces images à des positions précises, en utilisant la méthode draw\_texture\_rectangle.

J'ai également utilisé la méthode on\_mouse\_press pour gérer les interactions de la souris, en vérifiant si l'utilisateur a cliqué sur un bouton et en changeant de vue en conséquence.

Cela permet à l'utilisateur de naviguer entre les différentes vues de mon jeu, comme le menu principal, la vue de jeu et la vue des scores.



## Le Score et la base de données

Dans mon projet, j'ai utilisé une base de données MySQL pour gérer les scores des joueurs. La classe `ScoreView` permet de visualiser les 10 meilleurs scores enregistrés dans la base de données.

Dans la méthode `setup`, j'ai connecté la base de données et j'ai utilisé un curseur pour exécuter une requête SQL pour récupérer les 10 meilleurs scores.

J'ai ensuite stocké ces résultats dans une variable pour les afficher dans la méthode `on_draw`. Pour afficher les scores, j'ai utilisé des textures pour les noms de colonnes et j'ai utilisé la fonction `draw_text` pour afficher les noms de joueurs, les scores et les dates.

Dans la méthode `on_key_press`, j'ai utilisé la touche `ECHAP` pour retourner au menu principal et dans la méthode `on_mouse_press`, j'ai utilisé la texture de la flèche pour retourner au menu principal en cliquant dessus. Cette partie de mon projet permet aux joueurs de voir les scores des autres joueurs et de se comparer à eux, ce qui ajoute un élément de compétition et de motivation pour améliorer leur propre score.



Nom	Score	Date
jhjhkikoju	4.12308	2022-11-22 15:15:03
tyrurtyu tyru try	4.37019	2022-12-27 23:22:23
Iliass Ben ammar	4.43089	2022-12-27 23:20:30
Iliass Ben ammar	4.64362	2022-12-27 23:20:24
Test2	4.87284	2022-11-22 15:29:31
Test2	5.02696	2022-11-22 15:29:41
tyyertyertyert	5.24538	2022-11-22 11:46:18
jhjhkikoju	5.44426	2022-11-22 15:14:56
reterttreter	5.70143	2022-11-22 14:55:12
Iliass Ben ammar	5.7564	2022-12-27 23:20:11

## Le Jeu

Dans ma classe `JeuView`, j'ai défini la fonction `setup()` qui est appelée lorsque la vue est créée. Dans cette fonction, j'ai défini les différents éléments nécessaires pour que mon jeu fonctionne.

Tout d'abord, j'ai utilisé la bibliothèque `pyscreenrec` pour enregistrer la partie en cours, en utilisant la date et l'heure actuelles pour nommer le fichier vidéo.

J'ai également défini les variables nécessaires pour gérer le défilement de la caméra et le temps total de la partie.

Ensuite, j'ai créé des `Sprite lists` pour les différents éléments de mon jeu, tels que le personnage, les murs, les pièges et les coffres. J'ai également défini la taille et les calques de mon `background`.

J'ai ensuite utilisé la bibliothèque `arcade` pour charger ma map créée avec `Tiled`, en utilisant les calques appropriés pour les murs, les pièges et les coffres.

J'ai également utilisé le moteur physique de la bibliothèque `arcade` pour gérer les collisions entre les éléments de mon jeu.

J'ai implémenté une fonction pour gérer le `respawn` du personnage lorsqu'il tombe trop bas ou touche un pique. Lorsque cela se produit, le personnage est automatiquement réinitialisé à sa position de départ, prêt à poursuivre le jeu.

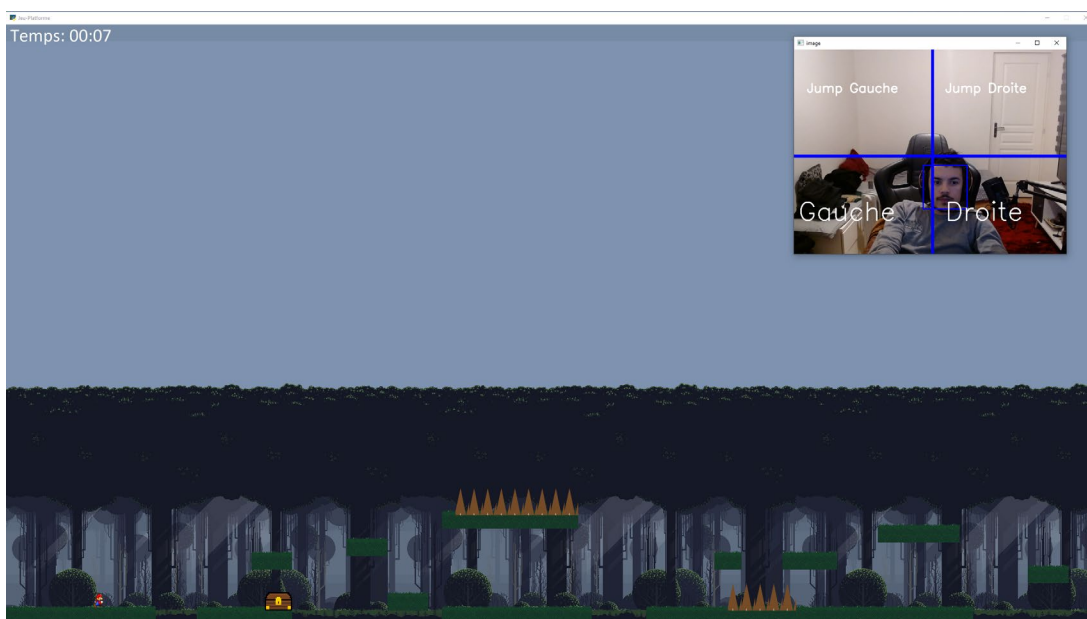
De plus, lorsque le personnage atteint le coffre à la fin de la map, une nouvelle vue "Victoire" est affichée pour féliciter le joueur pour sa performance.

Enfin, j'ai défini les fonctions `on_key_release()` et `on_draw()` pour gérer les interactions avec le joueur et l'affichage de mon jeu.

La fonction `on_key_release()` gère les interactions clavier, en permettant au joueur de quitter le jeu en appuyant sur la touche Échap.

La fonction `on_draw()` est utilisée pour afficher les différents éléments de mon jeu à l'écran.

Voici une image du jeu ou on peut voir le chronomètre en haut gauche ainsi que le background et les différents pièges ainsi que le démarquage de la caméra pour les déplacement :



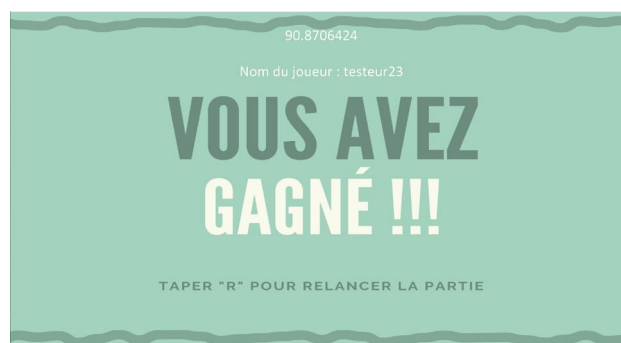
## La victoire

Dans ma classe `VictoireView`, j'ai créé un constructeur qui prend en paramètre le temps total passé dans la partie. J'utilise ce temps pour l'afficher à l'écran et pour l'enregistrer dans une base de données MySQL.

Au début de cette classe, j'utilise la fonction `arcade.load_texture` pour charger l'image de victoire au milieu de l'écran. Je reset également la position de la caméra pour que l'image de victoire soit centrée. Ensuite, j'initialise une variable pour le nom du joueur qui est récupéré depuis une autre classe. Je me connecte ensuite à ma base de données MySQL en utilisant la librairie `mysql.connector` pour créer une connexion et un curseur pour exécuter des requêtes SQL.

Dans la fonction `on_draw`, j'utilise `arcade.start_render()` pour afficher l'image de victoire chargée précédemment. J'affiche également le temps total passé dans la partie en utilisant `arcade.draw_text` et en récupérant la dernière ligne d'un fichier texte. J'affiche également le nom du joueur qui a gagné.

Dans la fonction `on_key_press`, j'ai implémenté une fonctionnalité qui permet de relancer la partie en appuyant sur la touche "R" en appelant la classe `JeuView`. Dans la fonction `on_key_release`, j'ai implémenté une fonctionnalité qui permet de retourner au menu en appuyant sur la touche "Echap" en appelant la classe `MenuView`. Enfin, je fais une requête SQL pour insérer les données du joueur (nom, score et date) dans ma base de données. Je fais un commit pour valider les changements et enregistrer les données dans la base de données.



# Conclusion

Pour conclure sur la création de ce jeu inspiré de Mario Bros, en utilisant la librairie Arcade, j'ai pu créer un jeu 2D facilement et efficacement. La librairie tkinter m'a permis de créer une interface graphique pour demander le pseudo du joueur tandis que la librairie pyscreenrec a facilité l'enregistrement de la partie. L'API Youtube m'a permis de connecter l'application à YouTube et de télécharger des vidéos sur le compte du joueur.

Pour gérer les scores, j'ai utilisé une base de données MySQL qui permet de stocker et de gérer les données de manière efficace. Cela permet de stocker les scores et de les afficher pour que les joueurs puissent les comparer et s'améliorer.

En utilisant Tiled, j'ai pu créer des maps rapidement et efficacement au format .tmx. Opencv2 m'a permis d'ajouter une fonctionnalité de contrôle de personnage avec la tête, qui permet au joueur de contrôler les déplacements du personnage en suivant les mouvements de sa tête, augmentant ainsi le niveau d'immersion pour le joueur.

En somme, ce projet a été l'occasion pour moi de mettre en pratique mes connaissances en programmation en utilisant différentes technologies pour créer un jeu immersif et interactif. Il a été réalisé dans le cadre du cours d'I.O.T, sous la direction du professeur Imad Saleh.