

Présentation d'un Manic'shooter

Nicolas.A - David.R - Théo.B

15 mars 2018

Table des matières

1	Manic Shooter	1
1.1	Présentation-Objectif	1
2	Architecture	2
3	Fonctionnalité du jeu	3
3.1	Une carte	3
3.2	Un personnage	3
3.3	Des ennemis	4
3.4	Déplacement	4
3.5	Tir	4
3.6	collisions	4
4	Elements technique	4
4.1	Python	4
4.2	Choix de l'interface graphique	5
4.3	Tkinter	5
4.4	Pygame	5
5	Expérimentation et Usage	5
5.1	Expérimentation	6
6	Conclusion	6

1 Manic Shooter

1.1 Présentation-Objectif

Présentation d'un Manic Shooter : Manic shooter ou Shoot them up ou Shmup qui signifie littéralement "Descendez-les tous". Un Manic shooter est un jeu où le joueur doit diriger un personnage ou véhicule devant tuer un grand nombre d'ennemis à l'aide d'armes de plus en plus puissantes au fur

et à mesure des niveaux, le personnages doit esquiver les tirs ou projectile ennemis. Ce système de jeu est sorti en 1978 avec Space invaders present dans les salles d'arcades de base c'est un jeu 2D il trouve son succès enfin des années 80 début 90 dès que le graphisme tri-dimensionnelle apparut, son succès disparut.

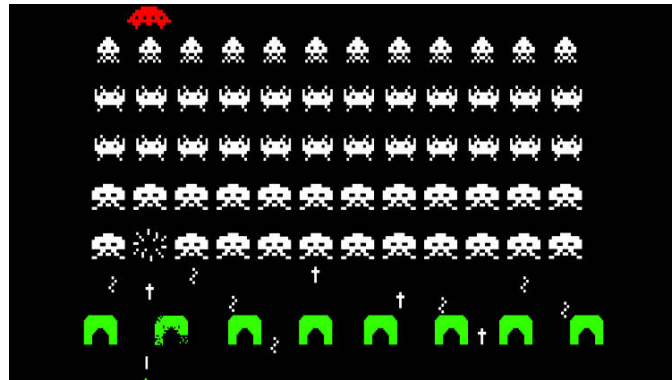


FIGURE 1 – Space invaders

2 Architecture

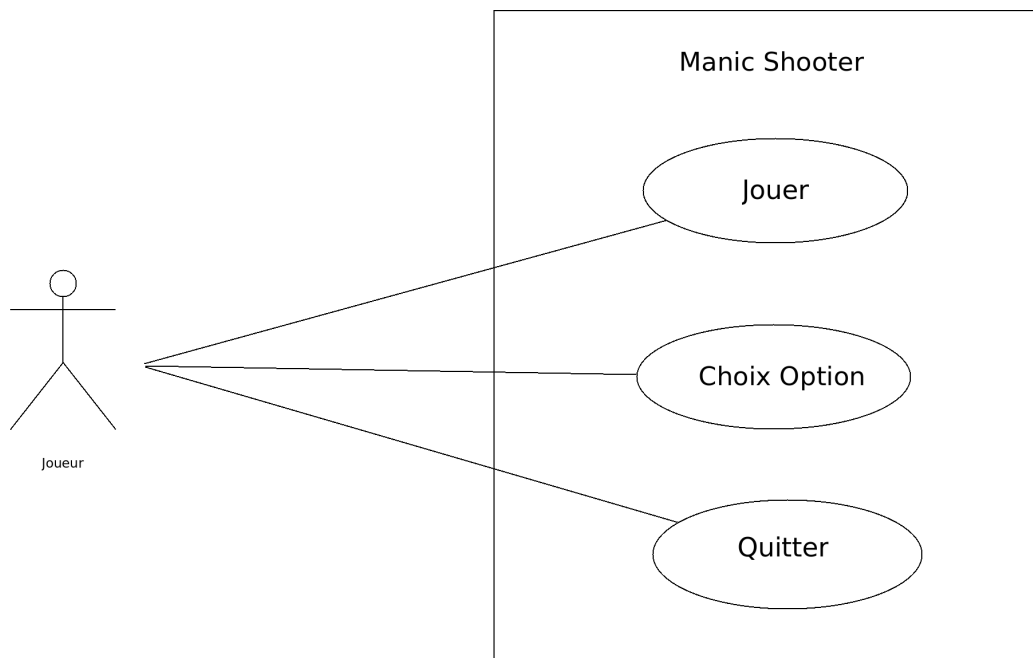


FIGURE 2 – diagramme cas d'utilisation

Ici nous avons a faire au premier menu qui va proposer de jouer , Quitter et de choisir la résolution de l'écran souhaitée. Jouer lance pygame et donne accès au menu final du jeu ou lancer le jeu est possible ou de changer des paramètres comme les préférences de jeu du joueur.

3 Fonctionnalité du jeu

3.1 Une carte



FIGURE 3 – Notre carte

dans un premier temps il nous fallait une carte, c'est à dire un espace de jeu permettant a un héro d'être livrer à une épopée contre des ennemies, notre premier réflexe à était de créer une grille et de créer un héro en l'initialisant en 1 et en initialisant des ennemies à 0 le reste de la grille étant que des "-", menant notre objectif jusqu'au bout nous nous sommes rendu compte qu'en associant l'interface graphique avec notre modèle de jeu posait problème et surtout qu'il existait une manière "plus simple" de faire la carte et surtout plus efficace, car en définissant la fenêtre avec Pygame cela aller être notre grille c'est à dire que la grille est la taille de fenêtre fois le nombre de pixels ce qui nous à simplifié la vie pour la suite.

3.2 Un personnage

un manic shooter à pour habitude d'avoir un héro qui se déplace dans la carte, il nous fallait donc un héro ou personnage qui soit celui que le joueur va contrôler notre personnage : ce personnage est équipé d'un réacteur qui le suit il à fallut intégrer cela au déplacement du héro et de faire en sorte que le réacteur soit collé au héro donc que la position soit identique suite



FIGURE 4 – Notre personnage trouver sur internet "open source"

3.3 Des ennemis

Le héros n'est pas le seul personnage à gérer il nous faut également gérer les ennemis

3.4 Déplacement

Une fois le contexte posé pour les personnages / la carte, c'est maintenant qu'intervient la notion de déplacement évidemment le héros doit se déplacer pour éviter les ennemis et avant dans son épopée, mais les ennemis doivent se déplacer également, il se déplace aléatoirement pour atteindre le héros pour cela nous allons utiliser la librairie random qui génère de chiffres aléatoires ce qui nous permet d'associer cela à des déplacements. Au début les déplacements étaient de la "téléportation" c'est à dire que la position du héros était traitée avec des rafraichissements d'images c'est à dire, quand le joueur appuie sur la touche déplacer le héros change de position et donc une nouvelle image est créée sur cette position et l'autre est supprimée, mais cela pose un problème, quand la vitesse du héros est trop grande on voit les différentes images des différentes positions du héros, suite

3.5 Tir

Généralement dans un jeu de tir le héros doit posséder un tir qui va détruire les ennemis

3.6 collisions

Les personnages sont confrontés à des collisions qui doivent être gérées, c'est à dire quand le tir va aller

4 Elements technique

4.1 Python

pour ce projet nous utilisons python 3 avec 10 librairie externe à python : OS,sys,pygame,tkinter,random,json,time,sympy

4.2 Choix de l'interface graphique

le choix des utilitaires utilisés peut différer le rendu du projet nous avons donc dû tester au fur et à mesure ce qui aller nous convenir le mieux

4.3 Tkinter

tkinter est une interface graphique qui est orienté logiciel elle permet la création de menu beaucoup plus simple dans notre cas cela nous sert pour le premier menu de notre jeu, du faite que les boutons sont directement implémentés dans la librairie de tkinter.

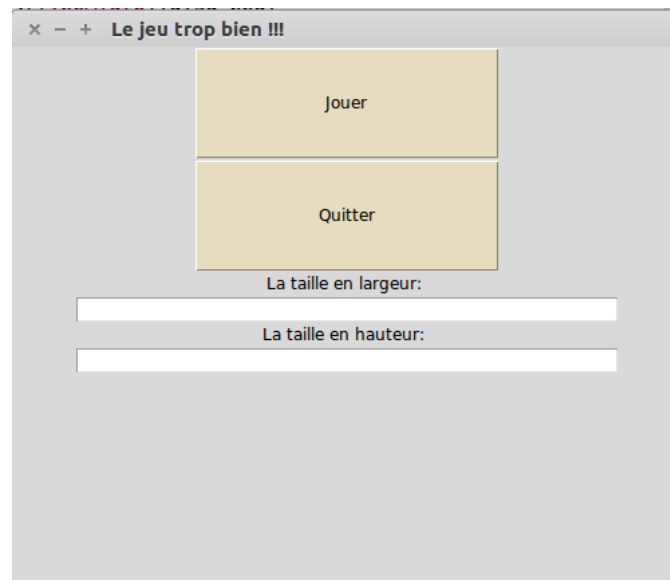


FIGURE 5 – notre premier menu avec tkinter

source tkinter

4.4 Pygame

Pygame est également une interface graphique qui est plus destiné au jeu, dans notre cas nous l'utilisons pour l'intégralité de notre gameplay , les déplacement, les collisions , la carte et le choix de cette interface graphique nous est venu naturellement car elle offre une possibilité de déplacement intéressante.

source pygame

5 Expérimentation et Usage

capture d'écran rendu final / perfomance

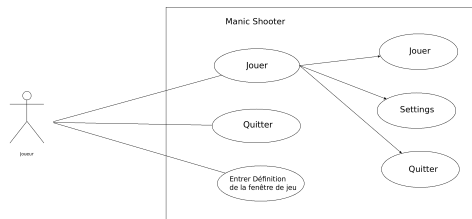


FIGURE 6 – Gameplay avec pygame

5.1 Expérimentation

Une phase d'expérimentation à du être nécessaire pour identifier ce qui peut poser problème. Pour commencer nous avons penser initialiser une grille pour la carte et des personnage étant des numéro pouvait traverser la grille, nous nous sommes vite rendu compte que cela aller poser problème cela ne rendait pas le jeu dynamique. Une fois que nous avons ouvert la fenêtre de pygame nous nous somme rendu compte que la grille aller être cette fenêtre et que les déplacements aller ce faire plus dynamiquement de cette manière nous avons donc oublier la grille. Le choix des interface graphique c'est fait en essayant dans un premier temps tkinter à était solliciter et nous somme rendu compte que le rendu graphique n'était pas qualité, les menu ce faisait facilement donc le premier menu c'est fait avec celui ci mais pour la conception du jeu c'est tkinter qui à était retenu.

6 Conclusion