

Présentation d'un Manic'shooter

Nicolas.A - David.R - Théo.B

7 mars 2018

Table des matières

1	Manic Shooter	1
1.1	Présentation-Objectif	1
1.2	diagramme	2
2	Fonctionnalité du jeu	2
3	Cahier des charges	2
3.1	Architecture du projet	2
3.2	Une carte	2
3.3	Un personnage	3
3.4	Des ennemis	3
3.5	Déplacement	3
3.6	Tir	4
3.7	collisions	4
4	Elements technique	4
4.1	Choix des utilitaires utiliser	4
4.2	Tkinter	4
4.3	Pygame	4
5	Expérimentation et Usage	4
6	Conclusion	4

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.

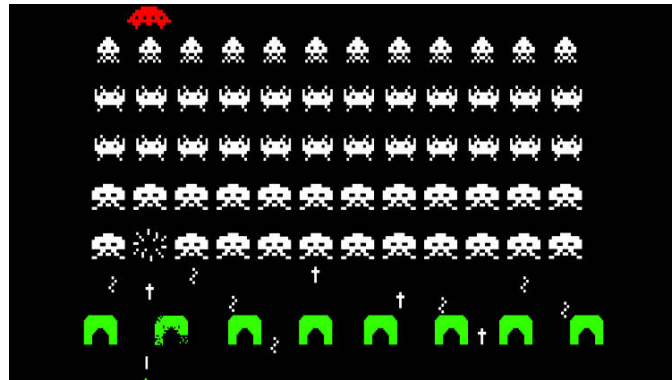


FIGURE 1 – Space invaders

1.2 diagramme

diagramme

2 Fonctionnalité du jeu

3 Cahier des charges

3.1 Architecture du projet

Le cahier des charges est l'une des chose essentiel pour bien débiter un projet sans cela la conception de notre logiciel serait devenue impossible. Tout d'abord nous voulions définir les normes logique du manic shooter une carte / un personnage / des ennemies / ce déplacer / tirer etc ...

3.2 Une 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



FIGURE 2 – Notre carte

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.

3.3 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



FIGURE 3 – Notre personnage trouver sur internet "open source"

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

3.4 Des ennemis

Le héro n'est pas le seul personnage à géré il nous faut également gérer les ennemies

3.5 Déplacement

Une fois le contexte poser pour les personnages / la carte, c'est maintenant qu'intervient la notion de déplacement évidemment le héro doit se déplacer pour éviter les ennemies et avant dans sont épopée, mais les ennemies doivent se déplacer également, il se déplace aléatoirement pour atteindre la

héro pour cela nous allons utiliser la librairie random qui génère de chiffre aléatoire ce qui nous permet d'associer cela à des déplacement. Au début les déplacement était de la "téléportation" c'est à dire que la position du héro était traité avec des rafraichissement d'images c'est à dire, quand le joueur appuie sur la touche déplacer le héro change de position et donc une nouvelle image et créer sur cette position et l'autre est supprimer, mais cela pose un problème, quand la vitesse du héro est trop grande on voit les différente images des différentes position du héro, suite

3.6 Tir

Généralement dans un manic shooter le héro doit posséder un tir qui va détruire les ennemies

3.7 collisions

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

4 Elements technique

4.1 Choix des utilitaires utiliser

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

4.2 Tkinter

tkinter est une interface graphique qui permet la conception de logiciel relativement pousser mais d'un rendu graphique faible, dans notre cas nous l'utilisons pour notre menu de notre jeu ce qui nous à rendement faciliter la tache car sur tkinter permet de créer des menu relativement facilement.

4.3 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.

5 Expérimentation et Usage

capture d'écran rendu final / perfomance

6 Conclusion