

Projet de module POO en C++ :

PICO SPACE

Encadrer par : Prof. EL ACHAK Lotfi



Réaliser par :

CHAABI Youssef
SAMADI Nouha

Sommaire

1. Introduction :

2. Présentation du Jeux :

3. Développement :

4. Conception :

5. Difficultés rencontrer :

Introduction :

La programmation orientée objet (POO), ou programmation par objet, a été élaborée par Alan Kay dans les années 1970. C'est un paradigme de programmation informatique qui consiste en la définition et l'interaction de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait communiquer avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; la communication entre les

objets via leur relation permet de réaliser les fonctionnalités attendues, de résoudre le ou les problèmes.

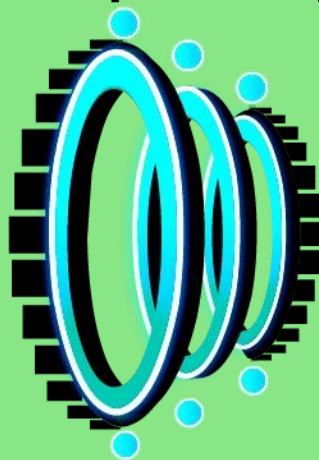
Le Langage C++ est un langage 'MID-LEVEL', Orienté Objet et avec un moteur de jeux ou avec la bibliothèque SDL on peut créer des jeux vidéo, il existe plusieurs moteur de jeux parmi eux Unity, Godot, UnrealEngine, Cocos2d-x, etc.

dans ce projet le moteur de jeux Cocos2d-x qui nous intéresse et qui va nous servir à réaliser notre jeux 2d 'PICO SPACE'.

Cocos2d est un framework libre en Python, permettant de développer des applications ou des jeux vidéo. Des jeux comme FarmVille, Geometry Dash ou Angry Birds Fight ! ont été développés avec Cocos2D.

Présentation du jeux :

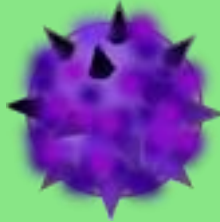
Le jeux contient 3 niveaux, pour qu'un joueur gagne la partie il doit passer tous les obstacles du niveau ainsi se téléporter via la porte magique :



Obstacle Mob :



Obstacle Ball :



Obstacle Hamer :



Obstacle Spiner :



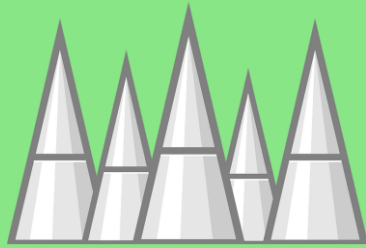
Obstacle Trap_3 :



Obstacle Trap_2 :



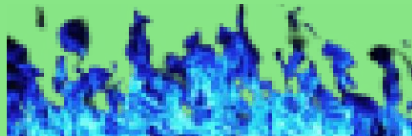
Obstacle PARK_Trap :



Obstacle Bob :



Obstacle Fire :



PICO SPACE :

Le monde de Pico a été attaqué par des extraterrestres, ces derniers ont ce monde vers une autre planète :



Niveau 1 :



Niveau 2 :



Niveau 3 :



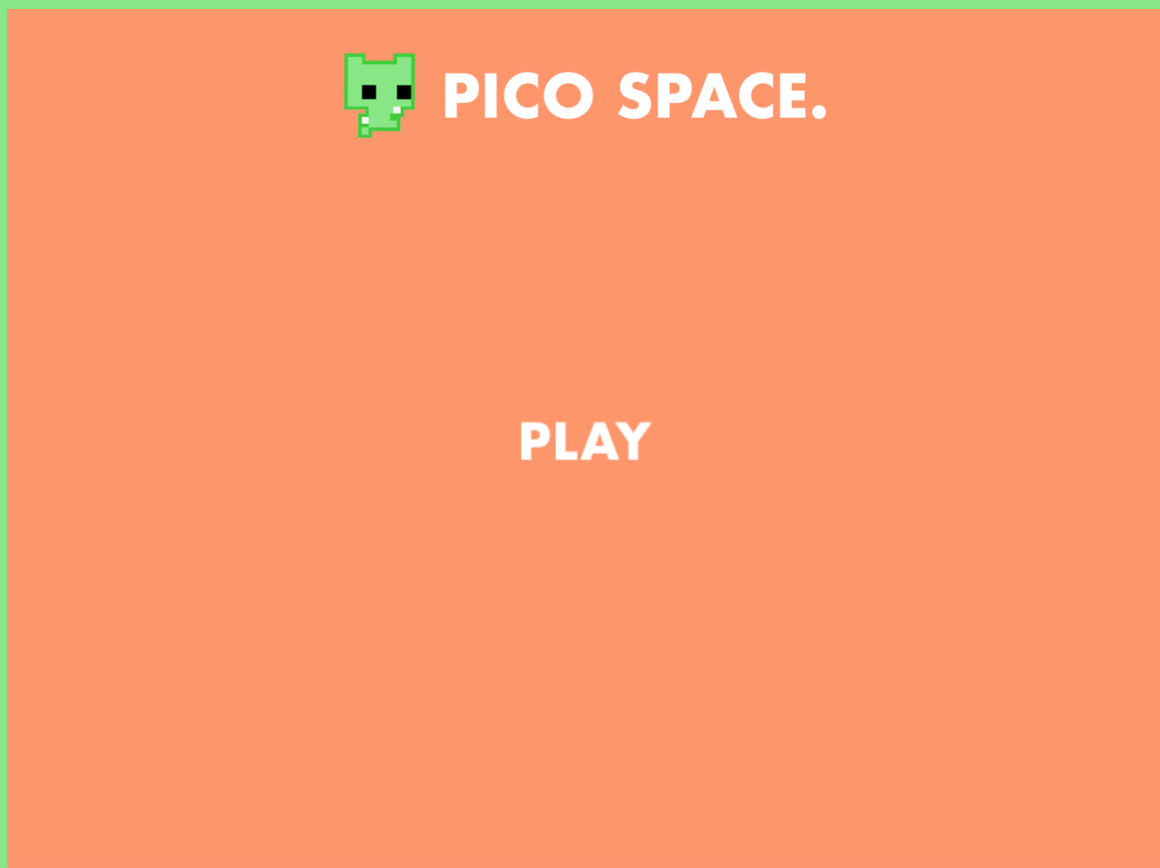
Pico vers la porte magique :



Loading Screen (première scène) :



Scène de Menu :



Scène des Niveau :



Développement :

Dans Pico Space, il y a six scènes différentes, qui sont Loadingscreen scene (splash scene), Helloworld, Levels scene, Level1 scene, Level2 scene et Level3 scene. Ils sont utilisés pour contenir différentes layers (couches) et d'autres CCNodes à l'intérieur. Le processus de création de chacune des scènes seront abordées brièvement dans les chapitres suivants.

Loading scene :

Une scène de démarrage (splash scene) est ce que l'on voit la plupart du temps lors de l'ouverture d'un programme/jeu. Il affiche les crédits, le nom de l'entreprise, etc. par exemple lors du démarrage de Windows, il affiche l'écran d'accueil Windows, qui mène à l'écran de connexion.

L'écran de démarrage a été le premier élément constitutif du projet Pico Space. Tout ce qui devrait arriver dans cette scène, c'est qu'elle chargerait un logo, afficherait et après un certain temps, appelez le directeur (Director) et changez les scènes en scène de menu (HelloWorld scene).

HelloWorld Scene(Menu scene) :

La scène du menu de ce jeu est très simple, contrairement à d'autres jeux qui ont systèmes de menus complexes. La scène du menu comporte un bouton qui change la scène du menu à la scène des niveaux. On a rencontré un problème lors la création de la scène de menu de la même manière que la scène LoadinScreen. Le problème était que certains des CCNodes (childs) enfants tels que les effets de particules et le logo se redondone d'une manière non désirée. Le problème a été résolu en ajoutant tous les composants du menu en tant que nœuds enfants (child nodes) au Scène de menu. De cette façon, l'ordre d'affichage des enfants pourrait être réorganisé.

Level1, Level2, Level3 :

Ce sont les scènes principales où tout le jeu se déroule. La création est semblable aux trois scènes précédentes. Contrairement aux écrans de démarrage et de menu précédents, les scènes du jeu faillaient avoir beaucoup de nœuds enfants, et gérer cela avec une seule couche (layer) n'était pas une option. À long terme, cela rendrait le code très désordonné, difficile à comprendre et surtout il va très difficile à maintenir.

Levels scene :

La scène des niveaux comporte trois boutons qui change la scène des Levels à la scène d'un niveau choisis. La création est semblable aux deux scènes précédentes.

Conception :

La conception de la couche des niveaux (Level Layer) était l'une des parties les plus difficiles de ce projet. Même bien que tout se joue à l'intérieur des scènes Level1, Level2 et Level3, la couche des niveaux devait s'occuper de toute la logique du jeu, le déplacement du personnage, le sol et les effets du jeu.

Les fonctionnalités suivantes devaient être présentes dans la couche :

La boucle du jeu et l'initialisation de tous les objets du jeu et de certaines couches HUD objets. Il doit s'occuper du chargement du sol et gérer les scènes coupées. Affichez et mettez à jour tous les nœuds enfants, tels que le joueur, les obstacles. Réorganiser toutes les positions des sprites d'acteurs enfants et mettre à jour la vue du joueur. Ainsi, la couche des niveaux peut être considérée comme le cœur de ce jeu. Il contient tous ensemble et traite les interactions entre les objets.

Heureusement, le nœud CCLayer prend en charge le scheduleUpdate, qui ajoute un mettre à jour la couche. En surchargeant la fonction de mise à jour (deltaTime) qu'il fournira, on peut créer une boucle de jeu.

Difficultés rencontrer :

En réalisant ce projet on a passé énormément de temps à apprendre comment fonctionne le moteur Cocos2d-x. Durant la création de ce jeu on a rencontré beaucoup de difficultés, parce que c'était quelque chose qu'aucun de nous n'avaient travaillé auparavant on avaient donc aucune expérience. Par exemple

voici quelques-unes des difficultés qu'on a rencontré en développant ce jeux vidéo :

- **on souhaitais utiliser Physics Body Editor (éditeur de polygones de collision pour les moteurs physiques) pour personnaliser notre joueur et nos obstacles, mais l'objet ne fonctionnait pas après l'avoir ajouté au code, nous avons donc décidé d'utiliser des formes qui existaient déjà dans la bibliothèque cocos2d-x qui sont de format circulaire et rectangulaire.**
- **après avoir mis en place la scène du niveau 1(level1), nous avons essayé de déplacer notre joueur en utilisant le clavier, mais nous avons eu beaucoup de mal à trouver une solution à ce problème, grâce à notre collègue ABDERRAHMAN qui a publié une solution sur stackoverflow on a pu résoudre ce problème mais au lieu de déplacer notre joueur nous avons décidé de déplacer la scène, cette solution se base sur la fonction update qui sera appelé à chaque "dt".**

Conclusion :

Tout au long de la préparation de notre jeu, nous avons essayé de pratiquer les connaissances requises durant notre cours de programmation orienté objet, et aussi notre connaissance sur ce qu'on a trouvé concernant cocos2d, ce projet nous a permis d'entrer dans le monde de développement et acquérir une expérience concernant le mode de développement des Jeux vidéo 2D.