

# compte rendu du jeux pico park

Réalisé par :

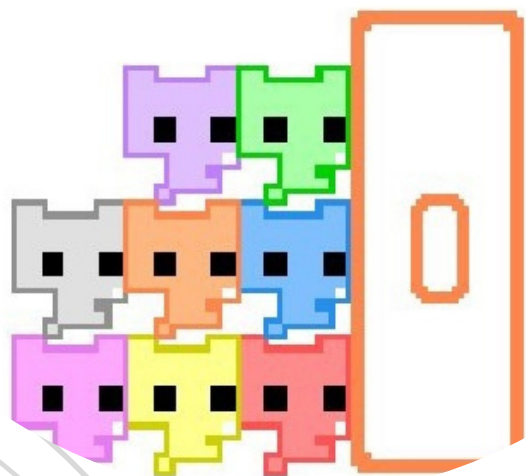
Imane El yahyaouy

group 1

Sara Wardi

group 2

## PICO PARK



# **sommaire :**

**1.introduction**

**2.Conception du jeu:**

**2.1 .pico park**

**2.2 .cocos2d-x**

**2.3 les objectifs**

**3.Développement et le test  
du jeu**

**4. les problèmes rencontrés**

**5.conclusion**



# introduction :

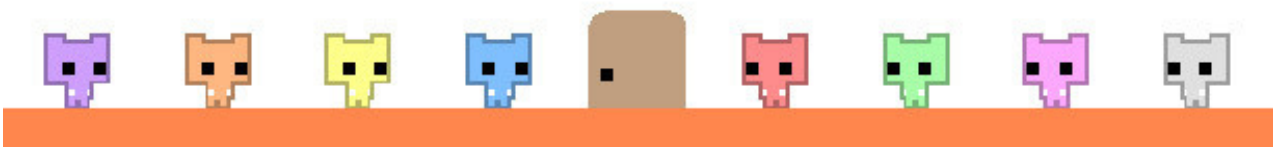
Le développement de jeux est un processus complexe qui implique la création de tous les aspects d'un jeu, y compris les graphismes, le son, l'histoire, la mécanique de jeu et la programmation. Dans ce rapport, nous allons parler du développement du jeu Pico Park avec le moteur de jeu Cocos2d et le langage de programmation C++.

## Conception du jeu:

### pico parck :

Pico Park est un jeu de simulation de parc d'attractions en 2D pour mobile. Il permet aux joueurs de créer et de gérer leur propre parc d'attractions, en sélectionnant les attractions et en gérant les finances de leur parc.

Dans Pico Park, les joueurs peuvent construire différents types d'attractions, comme des montagnes russes, des manèges et des stands de nourriture, et les personnaliser en ajoutant des décorations et en changeant leur apparence. Ils peuvent également gérer les finances de leur parc en vendant des billets et en dépensant de l'argent pour améliorer leur parc.



Le but du jeu est de construire et de gérer un parc d'attractions prospère et de divertir les visiteurs. Les joueurs peuvent s'affronter en ligne pour voir qui peut créer le parc d'attractions le plus réussi.

Pico Park est développé avec le moteur de jeu Cocos2d-x et le langage de programmation C++. Il est disponible sur iOS et Android.

## cocos2d-x :

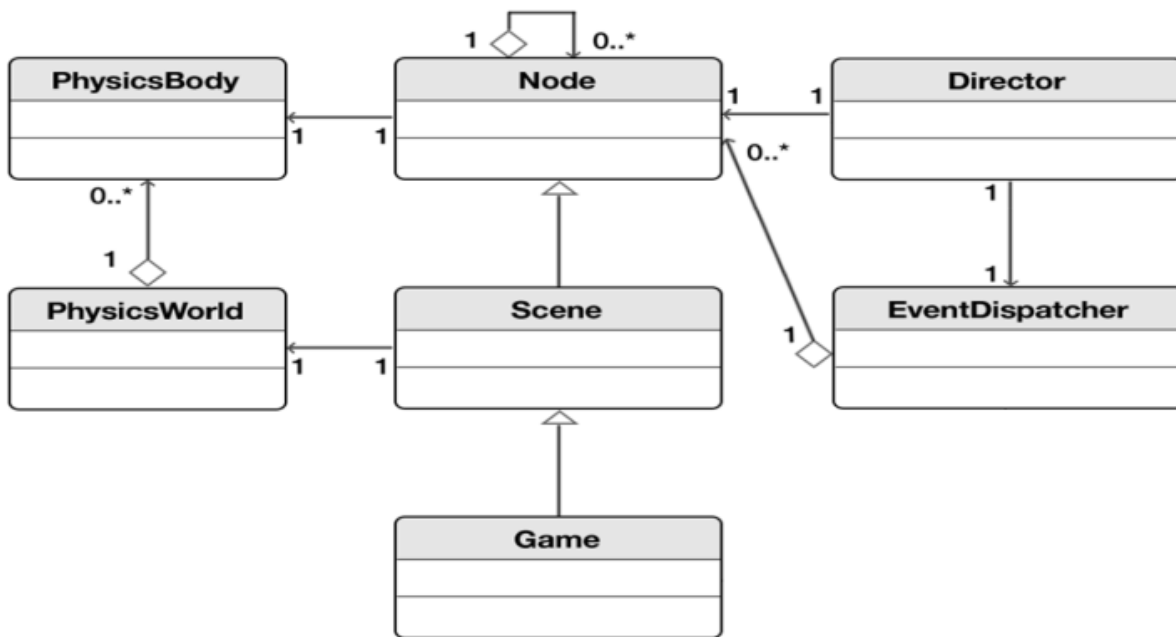


Cocos2d-x est un moteur de jeu open source qui permet de créer des jeux 2D et 3D pour diverses plateformes, telles que iOS, Android et PC. Il est basé sur la bibliothèque Cocos2d et a été conçu pour être facile à utiliser et à développer pour les développeurs de jeux.

Cocos2d-x offre une variété de fonctionnalités pour aider les développeurs à créer rapidement et facilement , telles que des graphiques de haute qualité, une gestion efficace des ressources, un système de physique intégré et de nombreux autres outils utiles. Il prend en charge plusieurs langages de programmation, tels que C++, C# ...

Cocos2d-x est très populaire parmi les développeurs de jeux en raison de sa flexibilité, de sa performance et de sa facilité d'utilisation. Il est souvent utilisé pour développer des jeux mobiles, mais peut également être utilisé pour créer des jeux pour PC et d'autres plateformes.

Cocos2d-x offre Scene, Transition, Sprite, Menu, Sprite3D, Audio et plus encore :



## Scene :

Scenes définissent les différents modes de votre jeu. Si vous avez un 'main menu', quelque niveaux... vous pouvez les organiser et les parcourir facilement avec Scene. Chaque cocos2d-x application nécessite au moins une Scene. Scenes sont des nœuds invisibles dans Cocos2d-X. Vous ne pouvez pas en fait voir une scène, mais vous pouvez voir la plupart de ses enfants, comme les Sprites, boutons, arrière-plans, etc.

## Layer :

Layer est une sous-classe de Node qui implémente le protocole TouchEventsDelegate ,Vous pouvez avoir plusieurs couches dans une scène, mais pour la plupart, vous n'aurez besoin que d'une seule couche. L'ordre z le plus élevé est en haut.



## **Node :**

Layers, ainsi que Scenes, Sprites et tout le reste dans Cocos2d-X, dérivent de Node. Nodes donnent à chaque élément de Cocos2d-X une commune fonctionnalité. Tous les nodes ont une position, par exemple, qui peut être définie afin de déplacer le Node. Nodes ont également des anchor points, cameras, z order, rotation, scaling, flipping, skewing et on/off visibility. Un Node a un parent possible et de nombreux enfants possibles. Cela organise tous les éléments de Cocos2d-X dans une hiérarchie. La scène est au sommet de la hiérarchie, puis vient layer, puis les Sprites, les backgrounds

## **Sprite:**

Un sprite est une image utilisée dans un jeu qui peut être animée et déplacée indépendamment dans le monde du jeu. Dans Cocos2d, les sprites sont généralement utilisés pour représenter des personnages, des objets et d'autres éléments du monde du jeu.

## **EventDispatcher :**

Event Dispatch est un mécanisme permettant de répondre aux événements des utilisateurs.

- Event Listeners encapsulent votre code de traitement d'événement - Event Dispatcher informe les auditeurs des événements des utilisateurs.
- Event Object contiennent des informations sur l'événement

5 types of event listeners :

EventListenerTouch - répond aux événements de contact EventListenerKeyboard -

répond aux événements du clavier EventListenerAcceleration - répond aux

événements d'accéléromètre EventListenMouse - répond aux événements de la souris

EventListenerCustom - répond aux événements personnalisés

## **PyhiscsBody :**

PhysicsBody est une classe de Cocos2d-x qui permet de gérer la physique des objets dans un jeu. Elle permet aux objets de se déplacer, de rebondir et de réagir aux forces et aux collisions.

Pour utiliser PhysicsBody, vous devez d'abord créer un objet PhysicsBody et lui assigner une forme physique (par exemple, une boîte, un cercle ou un polygone). Vous pouvez ensuite configurer les propriétés de l'objet, telles que sa masse, sa friction et sa restitution, et ajouter des forces pour le déplacer.

PhysicsBody est un élément essentiel de la plupart des jeux qui impliquent des objets en mouvement et des interactions physiques. Il est très utile pour créer des expériences de jeu immersives et réalistes.

## **PyhiscsWorld :**

PhysicsWorld est une classe de Cocos2d-x qui permet de gérer l'univers physique d'un jeu. Elle contient tous les objets PhysicsBody du jeu et gère les interactions et les collisions entre eux.

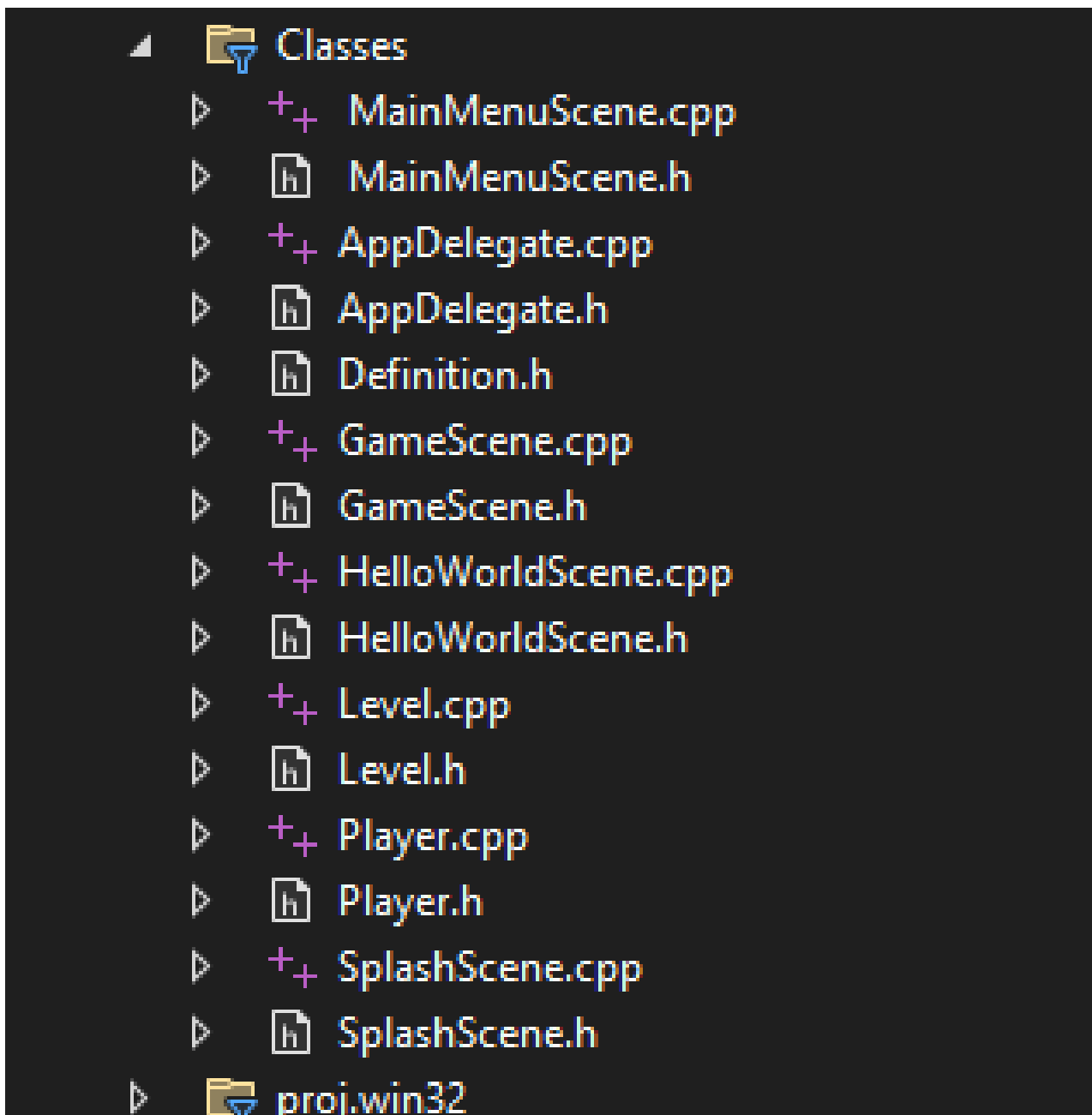
Pour utiliser PhysicsWorld, vous devez d'abord créer un objet PhysicsWorld et lui ajouter des objets PhysicsBody. Vous pouvez ensuite utiliser les méthodes de PhysicsWorld pour appliquer des forces aux objets, déterminer les collisions et résoudre les problèmes de physique.

PhysicsWorld est un élément essentiel de tout jeu qui implique des interactions physiques complexes. Il est très utile pour créer des expériences de jeu immersives et réalistes.

# les objectifs :

L'objectif de ce projet est de maîtriser la programmation orientée objet en C++ par la mise en place d'un jeu vidéo 2D en utilisant le platform COCOS2D-X et photoshope ou un autres application de designer les images

## Développement et le test du jeu :





## menu :

Donc, après l'exécution du code, le jeu est lancé. Il commence par un fond d'écran qui disparaît après 3 secondes et est remplacé par le menu permettant de lancer le premier niveau.



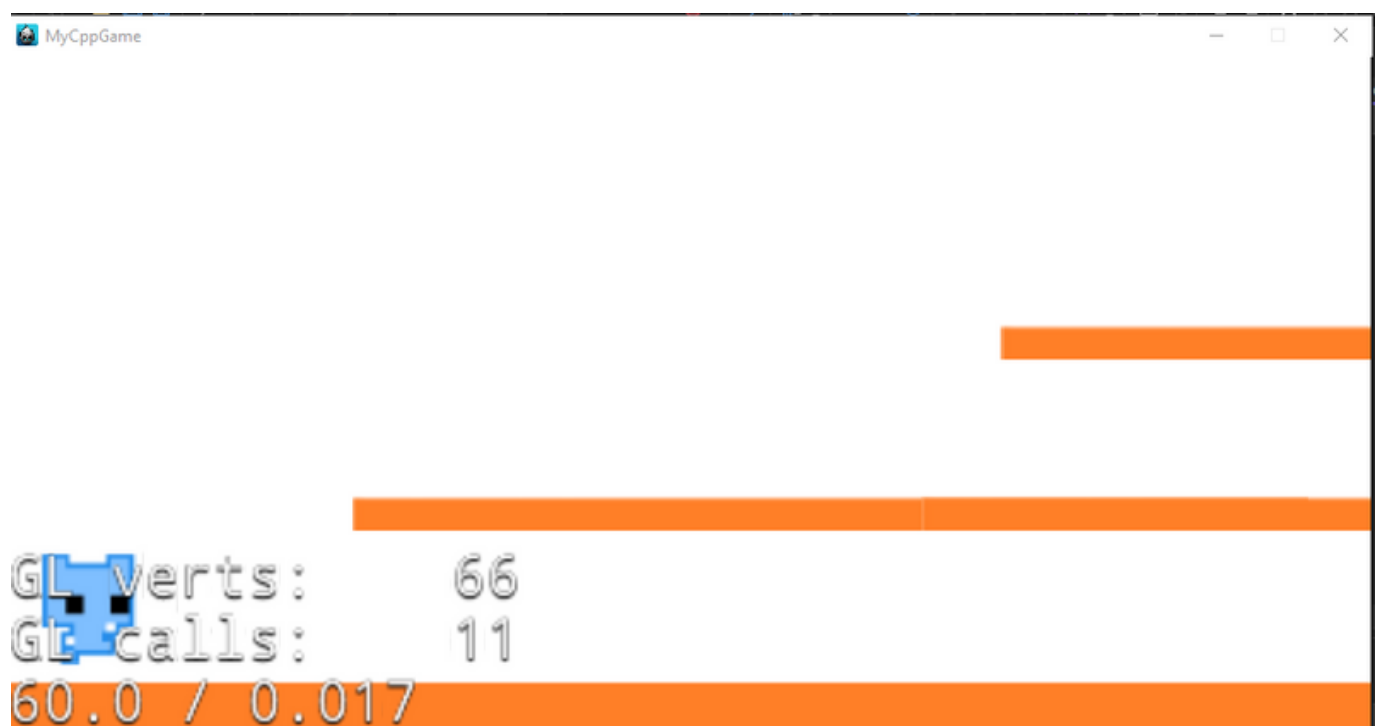
# LEVEL 1-1



GL verts: 12  
GL calls: 2  
60.0 / 0.017

# level 1:

Dans ce niveau 'level 1', le joueur doit sauter par-dessus des etages et apprendre la clé qui lui permet d'ouvrir la porte pour gagner. Dans ce niveau, il n'y a pas d'échec





GL verts: 84  
GL calls: 14  
60.0 / 0.017



## level 2:

Dans ce niveau, le joueur doit aussi sauter par-dessus des obstacles pour apprendre la clé qui lui permet d'ouvrir la porte pour gagner. Cependant, dans ce niveau, il y a une limite d'échecs : si le joueur a un contact avec les obstacles 3 fois, il aura une "game over" (fin de la partie).

LEVEL 1-2

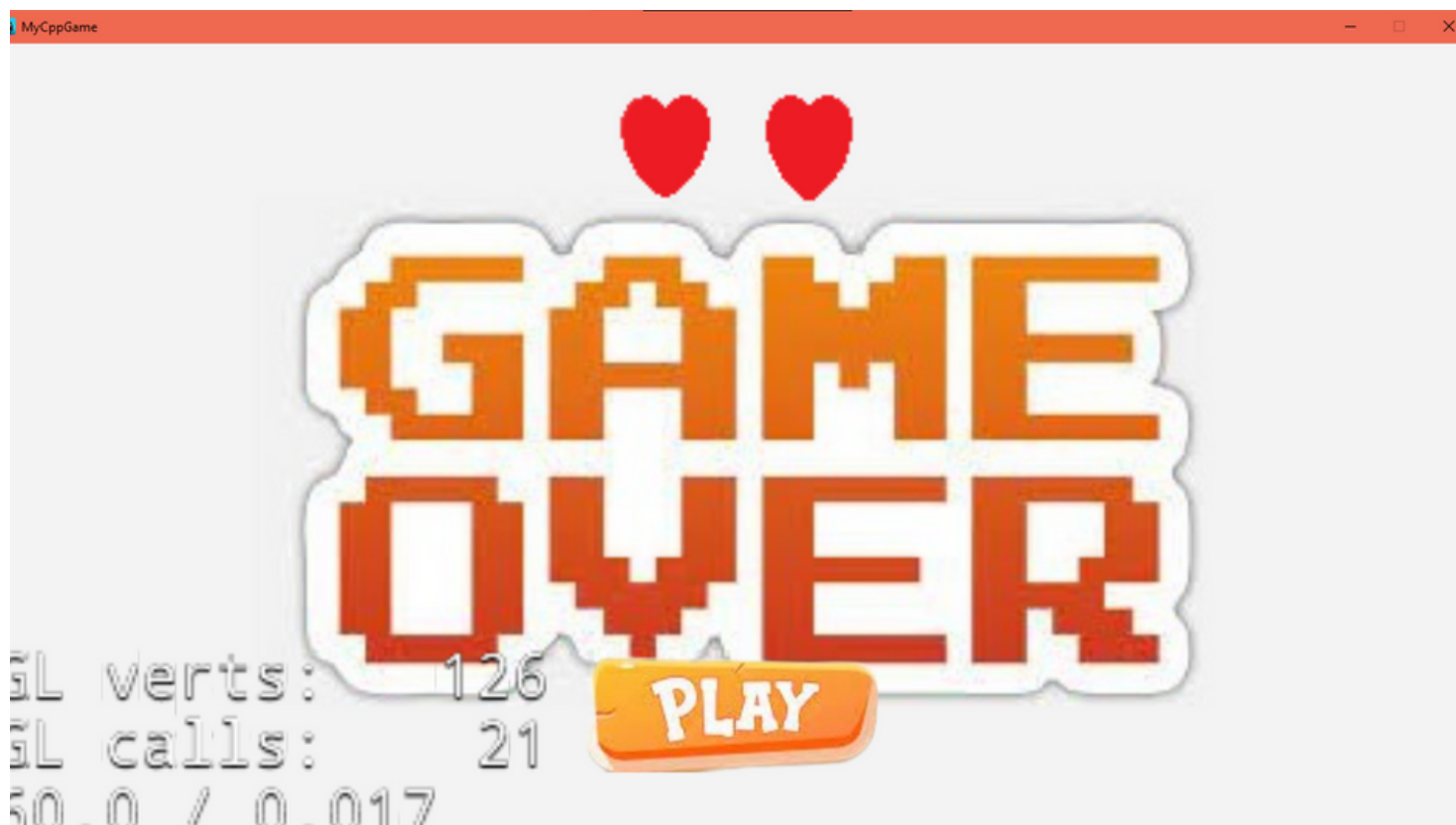


PLAY

GL verts: 84  
GL calls: 14  
60.0 / 0.017

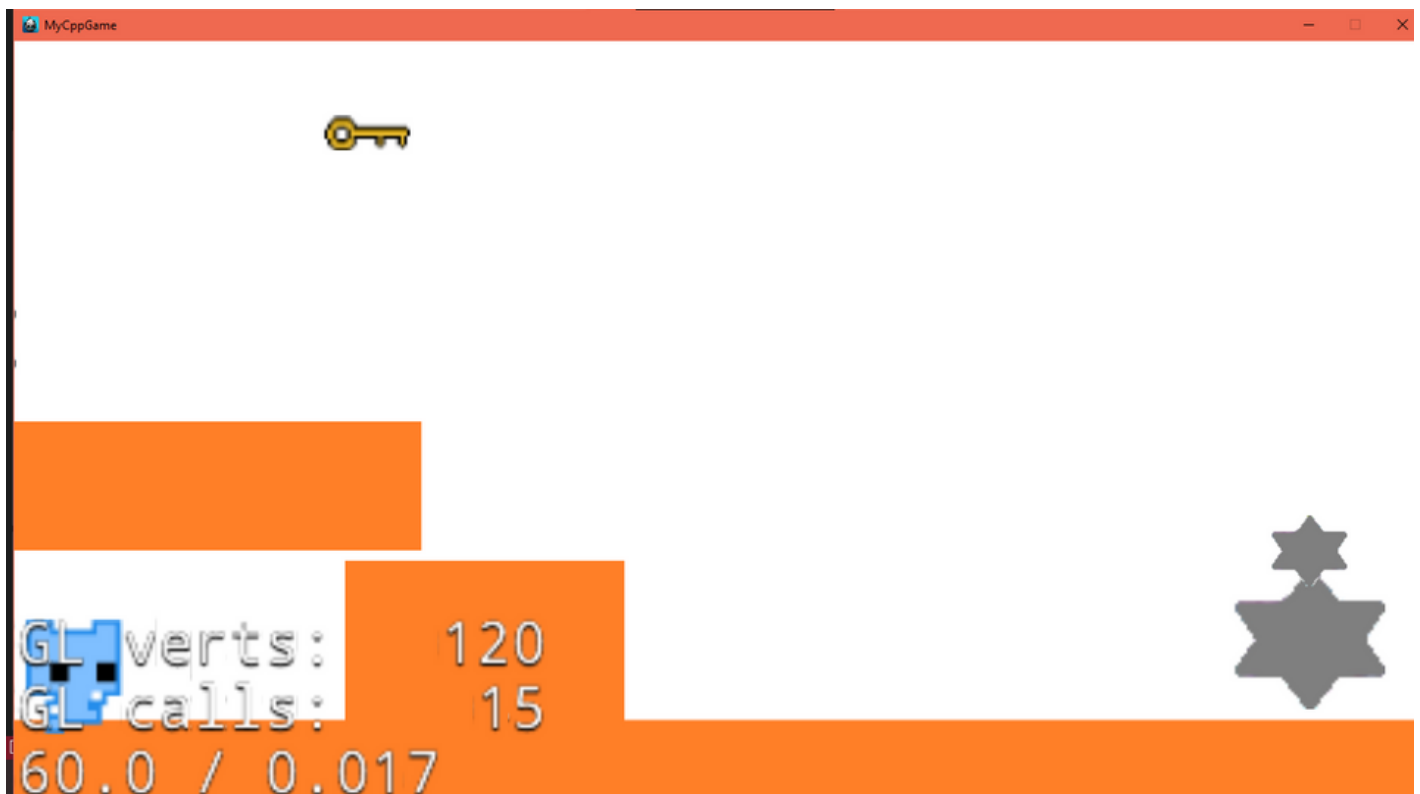
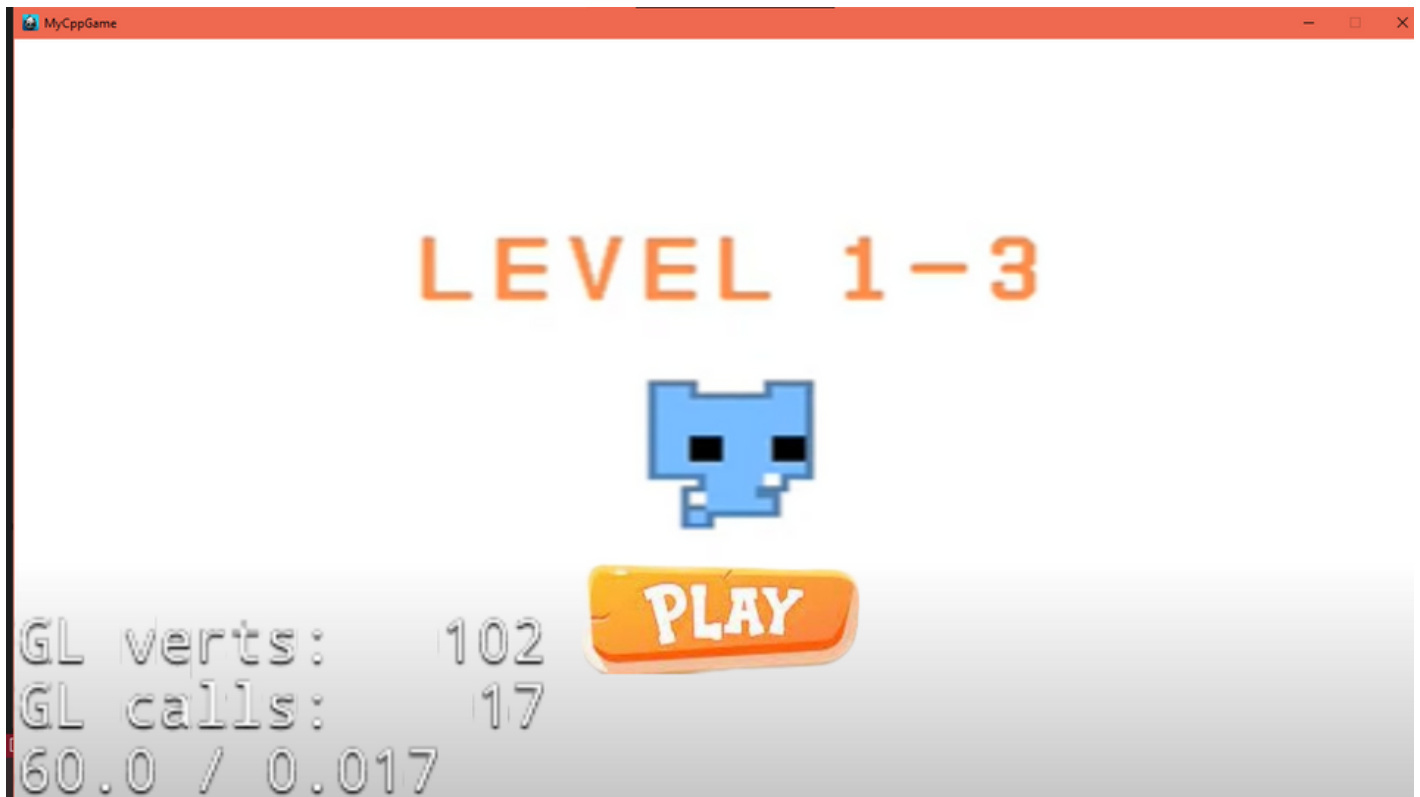
GL verts: 90  
GL calls: 15  
60.0 / 0.017

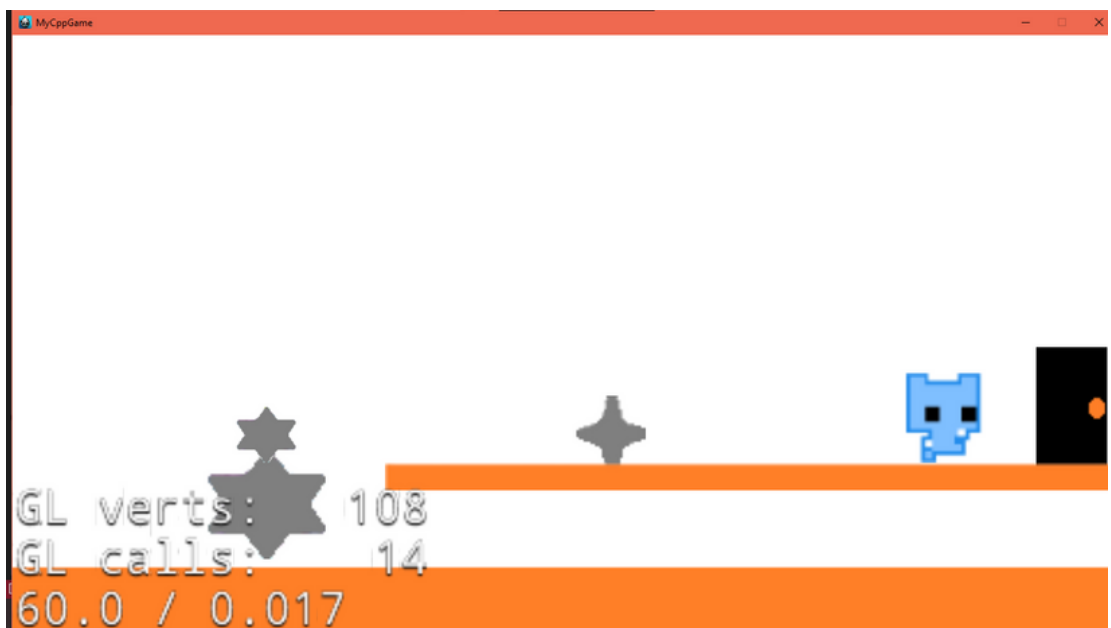
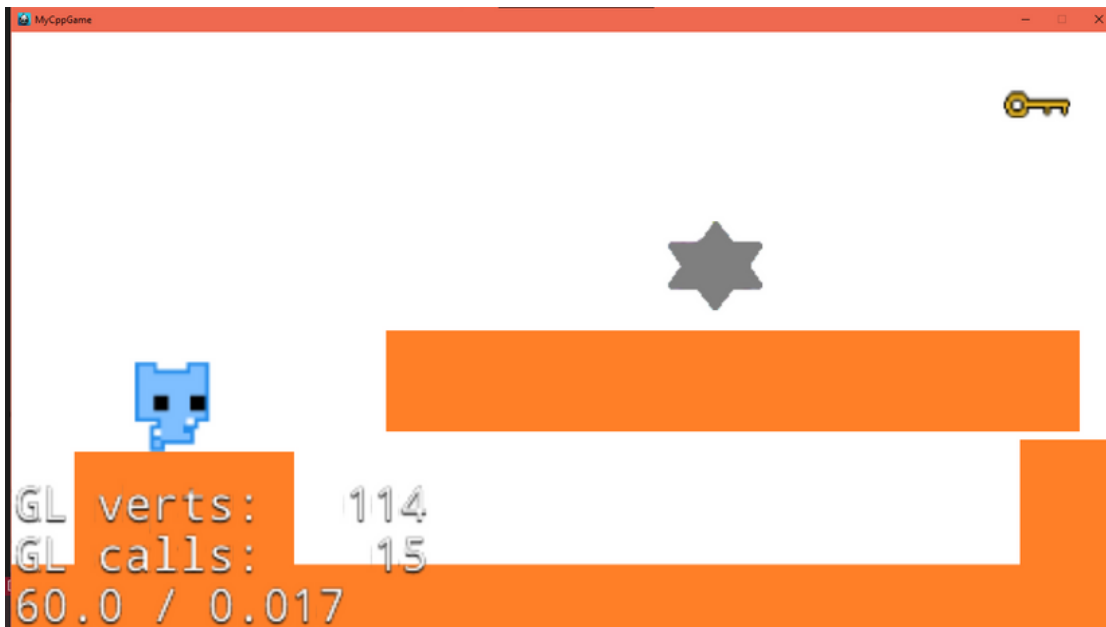
GL verts: 96  
GL calls: 16  
59.6 / 0.017



# level 3:

Ce niveau ressemble au deuxième, mais il est un peu plus difficile





# les problèmes rencontrés :

Nous rencontrons beaucoup de problèmes, et parmi eux, les plus importants sont :

1. Moving the Space Pod Using Touch: les fonctions de keypressed , keykeybored...
2. Follow sprite : Camera objects, mouve screen
3. sound : bibliotheque SimpleAudioEngine

## conclusion :

En résumé, le projet "Pico Park" consistait à développer un jeu de plateforme en utilisant le moteur de jeu Cocos2d-x et le langage de programmation C++. Les objectifs du jeu étaient de permettre au joueur de naviguer à travers différents niveaux en sautant par-dessus des obstacles et en résolvant des énigmes pour trouver la clé permettant de déverrouiller la porte menant au niveau suivant. Au cours du développement du jeu, nous avons rencontré plusieurs problèmes, notamment la difficulté à implémenter certaines fonctionnalités telles que la gestion de la collision et la gestion des animations de sprite. Nous avons également dû faire face à des défis liés à l'optimisation du jeu pour différentes plates-formes.

Malgré ces difficultés, nous avons réussi à créer un jeu divertissant qui a été bien reçu par les joueurs. Nous sommes satisfaits du résultat final et sommes impatients de continuer à améliorer "Pico Park" dans le futur.

En conclusion, le développement de "Pico Park" a été une expérience enrichissante qui nous a permis de mettre en pratique nos connaissances en programmation de jeux et de découvrir de nouvelles techniques de développement de jeux. Nous espérons pouvoir continuer à travailler sur ce projet et à le développer davantage dans le futur.



## les sources :

<https://discuss.cocos2d-x.org/t/physicsbody-and-sprite3d/23906>

[https://docs.huihoo.com/doxygen/cocos2d-x/3.3/df/de4/classcocos2d\\_1\\_1\\_layer.html#details](https://docs.huihoo.com/doxygen/cocos2d-x/3.3/df/de4/classcocos2d_1_1_layer.html#details)

[https://cocos2d-x.org/reference/native-cpp/V3.0alpha0/df/de4/classcocos2d\\_1\\_1\\_layer.html#details](https://cocos2d-x.org/reference/native-cpp/V3.0alpha0/df/de4/classcocos2d_1_1_layer.html#details)

<https://www.codeandweb.com/texturepacker/tutorials/animations-and-spritesheets-in-cocos2d-x>

[https://www.youtube.com/watch?v=qXqgSNUf9Cc&list=PLRtjMdoYXLf4od\\_bOKN3WjAPr7snPXzoe&ab\\_channel=SonarSystems](https://www.youtube.com/watch?v=qXqgSNUf9Cc&list=PLRtjMdoYXLf4od_bOKN3WjAPr7snPXzoe&ab_channel=SonarSystems)

<https://stackoverflow.com/search?q=cocos2d>

<https://openai.com/>