

PICO PARK EN C++

epicGame



PICO PARK

PLAY

GL verts: 18
GL calls: 3
58.6 / 0.013



- ❖ **RÉALISER PAR** : EL MOUSTAOU FADOUA ET DOUKKAR SALMA
- ❖ **GROUPE**: 2
- ❖ **ENCADRER PAR** : Mme BEN ABDEL OUAHAB IKRAM & Mr EL AACHAK LOTFI
- ❖ **LIEN DE GITHUB** : <https://github.com/FadouaSalma/Fadoua-Salma-JEUX>

➤ INTRODUCTION:

1- L'IDÉE DU JEUX:

Le jeux contient trois niveaux principaux, en mode mono palyer , c'est un joueur qui parcourt un trajet à chaque niveau afin d'atteindre la clé et de revenir dans le même trajet pour arriver à la porte qui va par laquelle passer à un autre niveau .

2- COCOS2D-x V4 :

Cocos2d-x est l'une des bibliothèques logicielles de référence. En plus de permettre de créer des jeux en 2D pour les appareils mobiles Android, iOS et Windows Phone, elle compile sur Windows, Mac et Linux. La bibliothèque peut être utilisée pour le développement en C++, Javascript et Lua.

3- ADOBE PHOTOSHOP:

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur, édité par Adobe, il est principalement utilisé pour le traitement des photographies numériques, Il travaille essentiellement sur images matricielles car les images sont constituées d'une grille de points appelés pixels. L'intérêt de ces images est de reproduire des gradations subtiles de couleurs.

4- MICROSOFT VISUAL STUDIO:

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.

➤ CREATION DE SCENE :

Déclaration de scène dans la classe principale (Dans level1.h) :

```
static cocos2d::Scene* createScene();
```

Après la déclaration, dans level1.cpp on fait la création en utilisant

```
Scene* Lvl1::createScene()  
{  
    auto scene = Scene::createWithPhysics();  
    scene->getPhysicsWorld()->setDebugDrawMask(PhysicsWorld::DEBUGDRAW_ALL);  
    scene->getPhysicsWorld()->setGravity(Vec2(0,-400));  
    auto layer = Lvl1::create();  
    scene->addChild(layer);  
    return scene;  
}
```

Scene* Lvl1::createScene() .

➤ CRÉATION DU MENU:

Dans le fichier menu.h on utilise une classe MainMenu

```
class MainMenu : public cocos2d::Layer  
{  
public:  
    // there's no 'id' in cpp, so we recommend returning the class instance pointer  
    static cocos2d::Scene* createScene();  
  
    // Here's a difference. Method 'init' in cocos2d-x returns bool, instead of returning 'id'  
    virtual bool init();  
    // void update(float dt);  
    void moveright(float dt);  
    void moveleft(float dt);  
  
    // implement the "static create()" method manually  
    CREATE_FUNC(MainMenu);  
    OurPlayer* player;  
  
    void GoToHelloWorldScene(cocos2d::Ref* pSender);  
}
```

Dans le fichier menu.cpp

```
Scene* MainMenu::createScene()
{
    // 'scene' is an autorelease object
    auto scene = Scene::create();

    // 'layer' is an autorelease object
    auto layer = MainMenu::create();

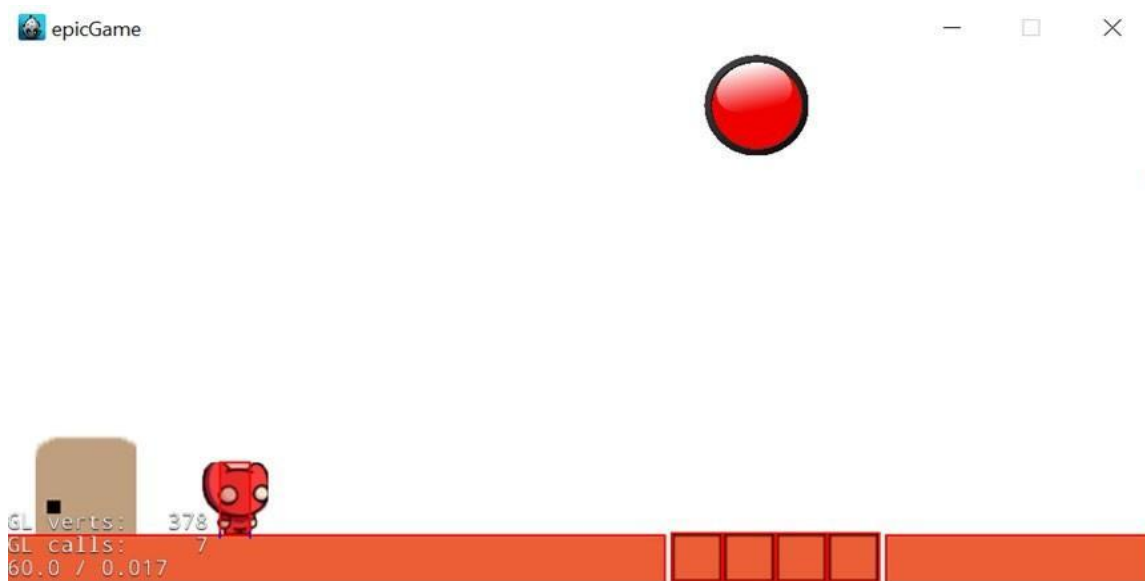
    // add layer as a child to scene
    scene->addChild(layer);

    // return the scene
    return scene;
}

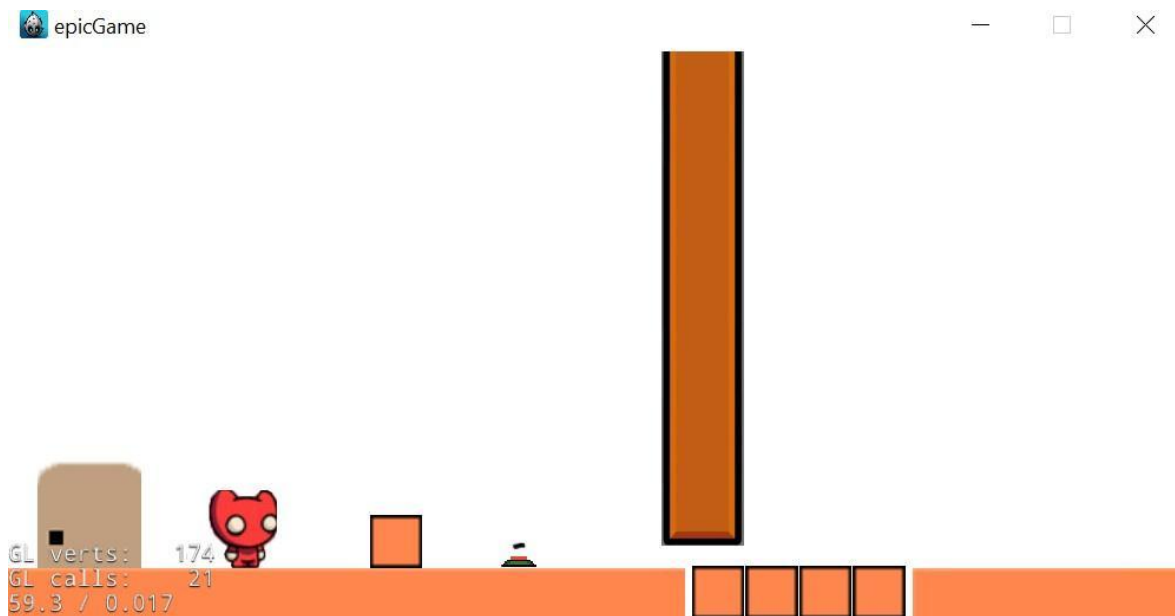
// on "init" you need to initialize your instance
bool MainMenu::init()
{
    // 1. super init first
    if (!Layer::init())
    {
        return false;
    }
}
```

➤ LES RÉSULTATS DES TROIS NIVEAUX

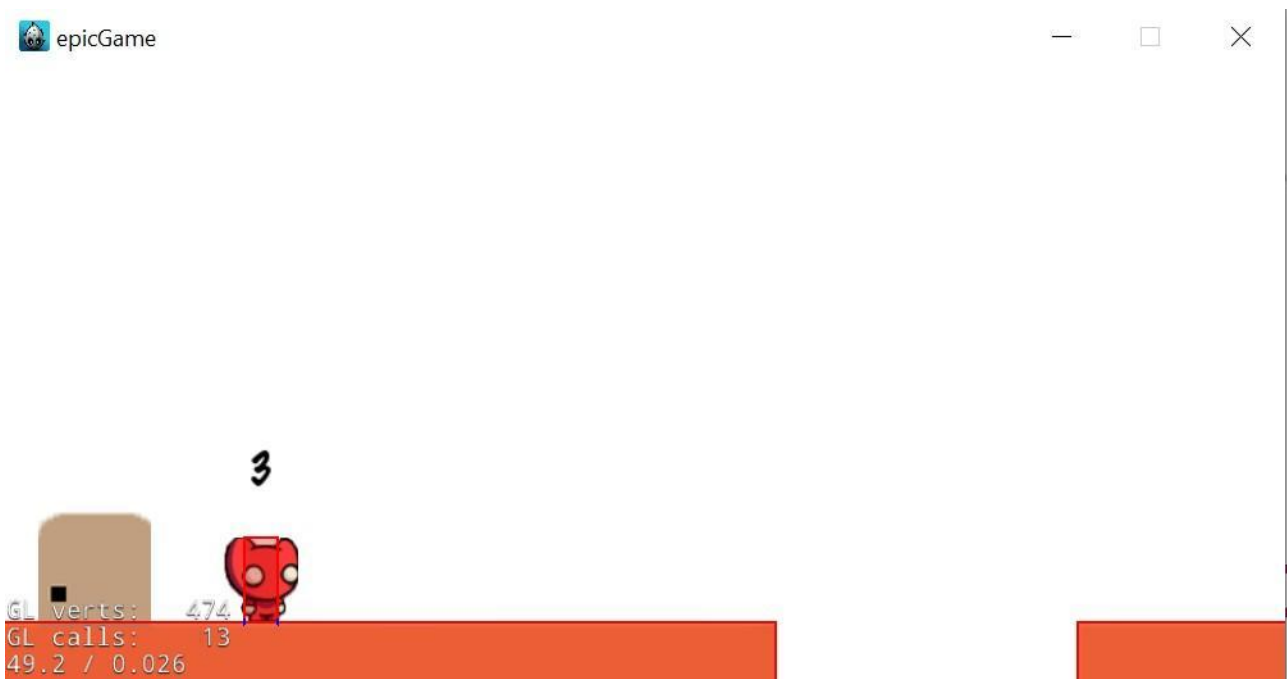
→ Le premier niveau :



→ Le deuxième niveau :



→ Le troisième niveau :



➤ CONCLUSION:

Au cours de ce projet nous avons développé certaines compétences tel que la prise d'initiative ,et la résolution les problèmes comme le mouvement du joueur , Le passage entre les niveaux,ainsi le travail en équipe qui seront des aspects essentiels de notre futur métier.Enfin ce projet nous a permis d'appliquer nos connaissances qu'on a acquis à travers le module de la programmation orienté objet en C++.

