

## *PICO PARK EN C++*



**RÉALISÉ PAR :** EL MOUSTAOU FADOUA ET DOUKKAR SALMA

**GROUPE:** 2

**ENCADRE PAR :** Mme BEN ABDEL OUAHAB IKRAM & Mr EL AACHAK LOTFI

**LIEN DE GITHUB :**

# **INTRODUCTION:**

## **1- L'IDEE DU JEUX:**

Le jeux contient trois niveaux principaux, en mode mono palyer , c'est un joueur qui parcouru 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 la quelle 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 :

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

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

2-Après la déclaration, dans level1.cpp on fait la création en utilisant la fonction Scene\* Lvl1::createScene() .

```
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;  
}
```

# CREATION DU MENU:

Dans le fichier menu.h :

```
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' in cocos2d-iphone
    virtual bool init();
    // void update(float dt);

    // implement the "static create()" method manually
    CREATE_FUNC(MainMenu);
    OurPlayer* player;

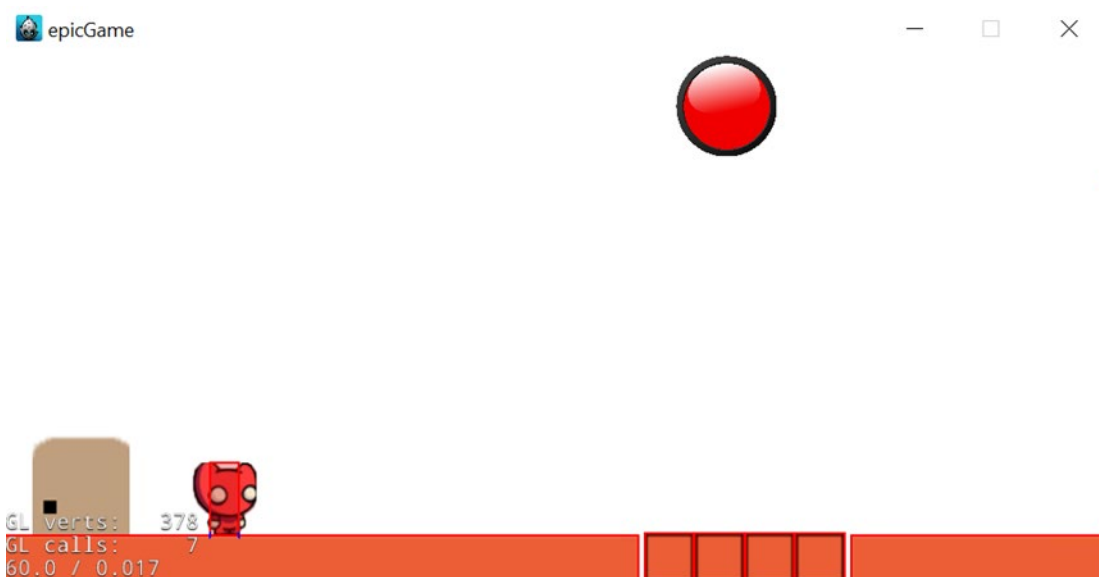
    void GoToHelloWorldScene(cocos2d::Ref* pSender);
    void Exit(cocos2d::Ref* pSender);

};
```

Dans le fichier menu.cpp :

```
auto playItem = MenuItemImage::create("button2.png", "pressed.png", CC_CALLBACK_1(MainMenu::GoToHelloWorldScene, this));
auto menu = Menu::create(playItem, NULL);
menu->setPosition(visibleSize.width, visibleSize.height - 50 );
menu->setScale(2);
this->addChild(menu);
```

**A-Premier niveau :**



## Dans Level1.h :

```
class Lvl1 : public cocos2d::Layer
{
public:
    void update(float dt);
    void moveright(float dt);
    void moveleft(float dt);
    void movetop(float dt);
    void switchLights(float dt);
    void signal(float dt);
    void fallfirstbridge(float dt);
    void fallsecondbridge(float dt);
    void fallfourthbridge(float dt);
    void fallthirdbridge(float dt);
    static cocos2d::Scene* createScene();
    float time;
    bool isUpPressed = false;
    bool isRightPressed = false;
    bool isLeftPressed = false;
    bool isSpacePressed = false;
    bool isDoorOpened = false;
    bool isPlayerOnGround = true;
    bool isLightGreen = false;
    cocos2d::Label *timeTxt;
    cocos2d::Label* inst;
    virtual bool init();
    bool onContactBegin(cocos2d::PhysicsContact& contact);
```

## Dans level1.cpp :

```
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;
}

// on "init" you need to initialize your instance
bool Lvl1::init()
{
    if (!Layer::init())
    {
        return false;
    }

    //-----create backgroud color-----

    LayerColor* _bgColor = LayerColor::create(Color4B::WHITE);
    this->addChild(_bgColor, -10);

    //-----create variables for positioning our instance-----

    auto visibleSize = Director::getInstance()->getWinSize();
    Vec2 origin = Director::getInstance()->getVisibleOrigin();
}
```

## Dans level2.h :

```

class Lvl2 : public cocos2d::Layer
{
public:
    void update(float dt);
    void moveright(float dt);
    void moveleft(float dt);
    void movetop(float dt);
    void movebot(float dt);
    static cocos2d::Scene* createScene();
    float size=0.4;
    float boxSize = 1.2;
    bool isUpPressed = false;
    bool isRightPressed = false;
    bool isLeftPressed = false;
    bool isSpacePressed = false;
    bool isDoorOpened = false;
    bool isPlayerOnGround = true;

    cocos2d::Label *timeTxt;
    cocos2d::Label* inst;
    virtual bool init();
    bool onContactBegin(cocos2d::PhysicsContact& contact);

    // implement the "static create()" method manually
    CREATE_FUNC(Lvl2);
    OurPlayer* player;
    OurKey* key;
    OurDoor* door;
    Sprite* ground;
}

```

Dans level2.cpp :

```

Scene* Lvl2::createScene()
{
    auto scene = Scene::createWithPhysics();
    scene->getPhysicsWorld()->setGravity(Vec2(0, -400));
    auto layer = Lvl2::create();
    scene->addChild(layer);
    return scene;
}

// Print useful error message instead of segfaulting when files are not there
// on "init" you need to initialize your instance
bool Lvl2::init()
{
    if (!Layer::init())
    {
        return false;
    }

    //-----create backgroud color-----

    LayerColor* _bgColor = LayerColor::create(Color4B::WHITE);
    this->addChild(_bgColor, -10);

    //-----create variables for positioning our instance-----

    auto visibleSize = Director::getInstance()->getWinSize();

    Vec2 origin = Director::getInstance()->getVisibleOrigin();

```

Dans level3.h :



```

class Lvl3 : public cocos2d::Layer
{
public:
    void update(float dt);
    void moveright(float dt);
    void moveleft(float dt);
    void movetop(float dt);
    void movebot(float dt);
    static cocos2d::Scene* createScene();
    float time;
    bool isUpPressed = false;
    bool isRightPressed = false;
    bool isLeftPressed = false;
    bool isSpacePressed = false;
    bool isDoorOpened = false;
    bool isPlayerOnGround = true;
    cocos2d::Label *timeTxt;
    cocos2d::Label* inst;
    virtual bool init();
    bool onContactBegin(cocos2d::PhysicsContact& contact);

    // implement the "static create()" method manually
    CREATE_FUNC(Lvl3);
    OurPlayer* player;

```

Dans level3.cpp

```
USING_NS_CC;
```

```
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;  
    }  
    LayerColor* _bgColor = LayerColor::create(Color4B(255, 255, 255, 255));  
    this->addChild(_bgColor, -10);  
    Size visibleSize = Director::getInstance()->getVisibleSize();  
    Point origin = Director::getInstance()->getVisibleOrigin();
```

## Les difficultés rencontrer :

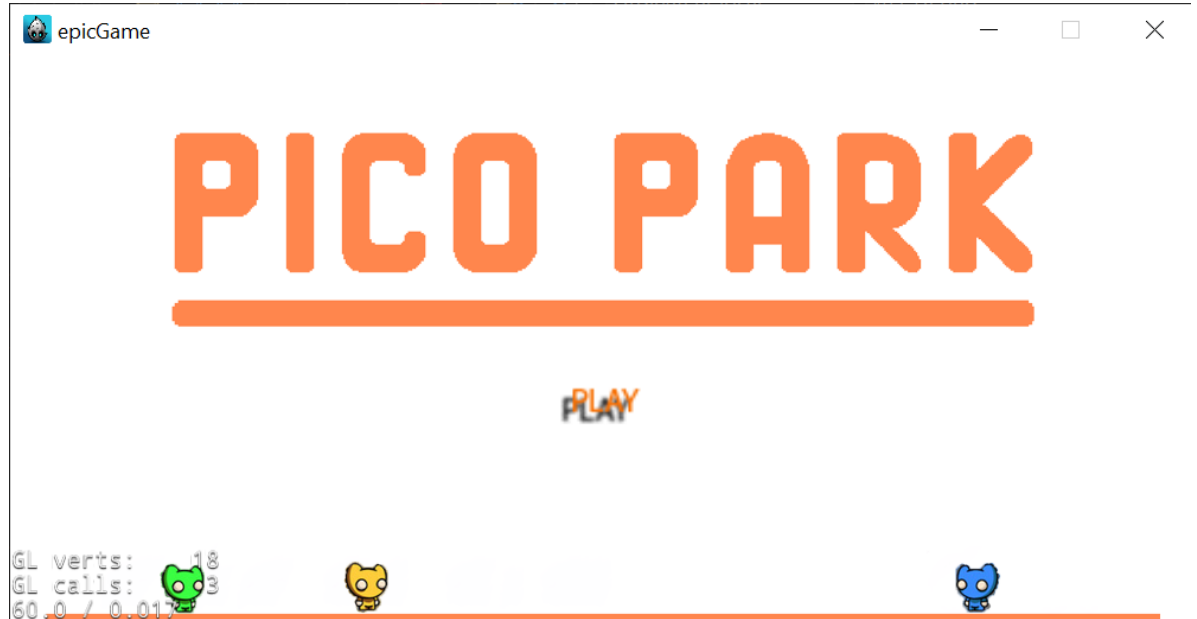
- La gravitation et les mouvements du joueur
- Le passage entre les niveaux
- Détection des collisions

La conclus

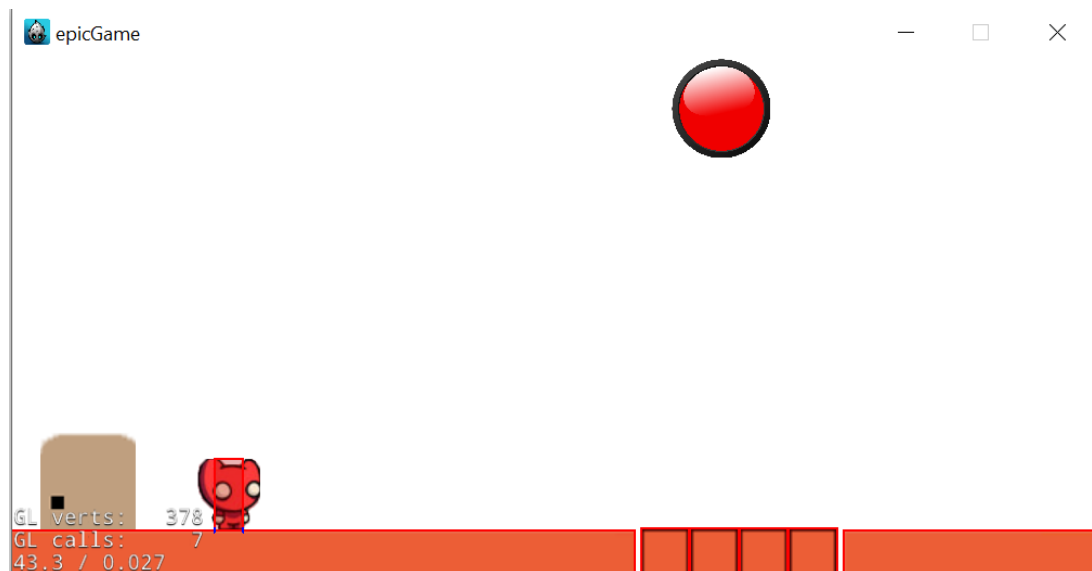


# RESULTAS:

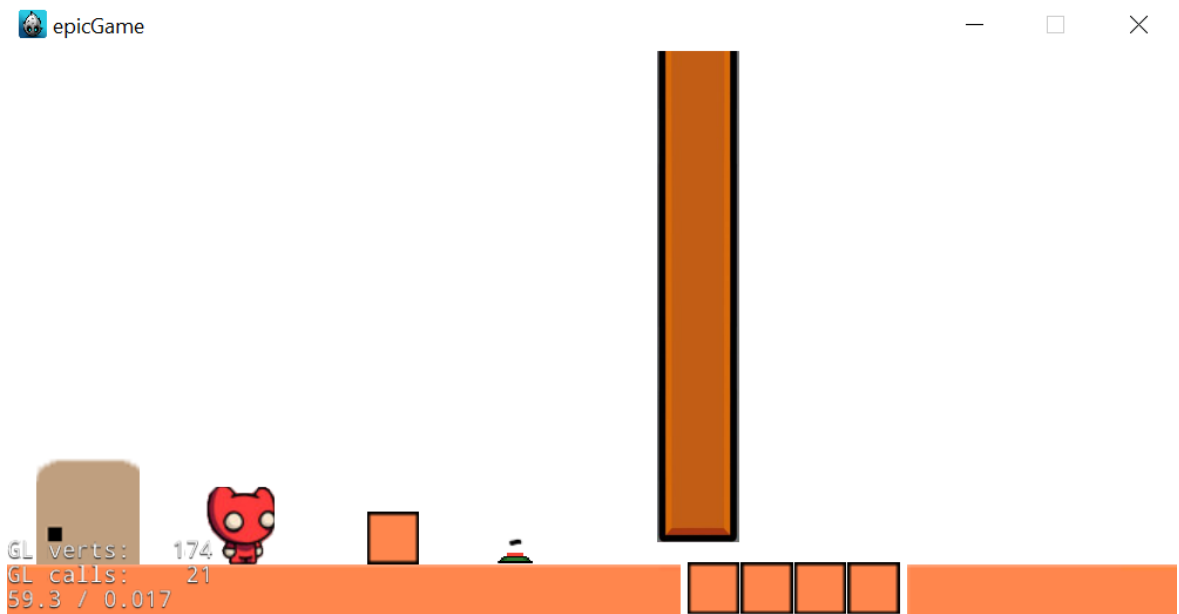
## 1- Le menu principal:



## 2- Le 1er niveau:



### 3- Le 2em niveau:



### Le 3em niveau:



