

PICO PARK



Réaliser par :

**Aachabi
Mohammed
Daghmoumi
Marouan**

Encadrer par :

Mr . lotfi aachak

Mrs.Ikram ben abdel ouahab

Step 1

Creation de projet

```
D:\>cocos new picopark -l cpp
```

Creation de projet picopark

```
C:\Users\iyadd>cd C:\Users\iyadd\tutogame\proj.win32
```

```
C:\Users\iyadd\tutogame\proj.win32>
```

Acceder au dossier puis à (proj.win32)

```
C:\Users\iyadd\tutogame\proj.win32>cmake .. -G "Visual Studio 17 2022" -Awin32
```

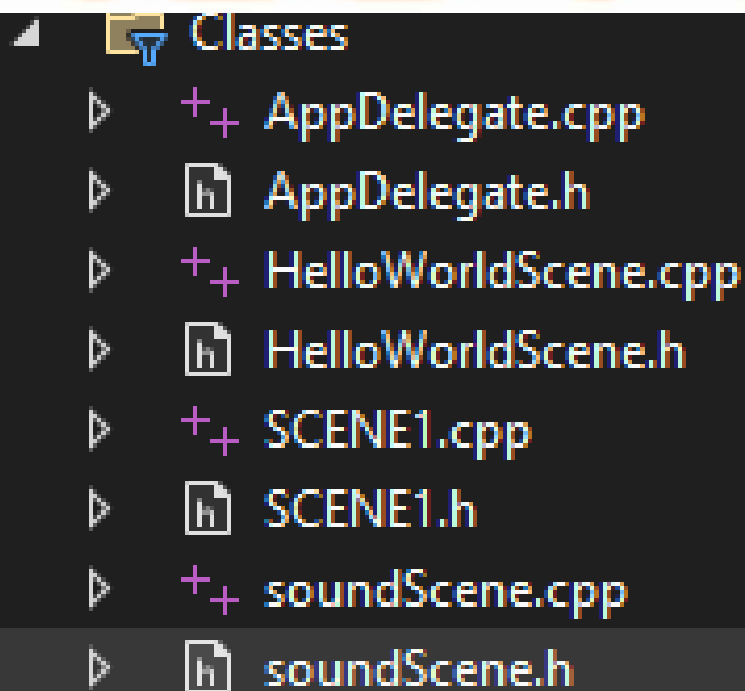
Cmake

Step2

Les scenes

On a scenes principal :

PICO PARK



Step3

Démarrage de la jeux

On click sur

▶ Débogueur Windows local ▾

Résultat



On a 3 Button cliquable :

NEW GAME

OPTIONS

EXIT

Réaliser dans le code source «HelloWorldScene.cpp »

Step 4

THE GAME

HelloWorldScene.h

```
class HelloWorld : public cocos2d::Layer
{
public:
    static cocos2d::Scene* createScene();
    virtual bool init();
    //a selector one
    void menuCloseCallback(cocos2d::Ref* pSender);

    CREATE_FUNC(HelloWorld);
    cocos2d::Sprite* menuback;
    cocos2d::Menu* menu1;
    //menu
    void play(Ref* pSender);
    void exit(Ref* pSender);
    void sound(Ref* pSender);
    void imgbutton(Ref* psender);
};
```

Dans cette code source on declare notre class «HelloWorld »

, après creation de class on cree notre menu «menuCloseCallback »

Et aussi les fonction des buttom (play, exit et sound)

**ALORS QUE DANS EN GENERALE LES CODE SOURCE QUE CE TERMINE
AVEC { .h }**

ON CREE LA CLASS ET LES ELEMENT DE SCENE

HelloWorldScene.cpp

```
#include "HelloWorldScene.h"
#include "cocos2d.h"
#include "SCENE1.h" //pour switcher vers
#include "AudioEngine.h" //biblio pour importer les audioa
#include "soundScene.h"//pour pouvoir switcher vers sound
USING_NS_CC;
```

On déclare les biblio pour l'utiliser

```
menuback = Sprite::create("menubackground.png");
menuback->setPosition(Point((visibleSize.width / 2) + origin.x, (visibleSize.height / 2) + origin.y));
menuback->setScale(0.125);
```

Notre background et leur position

```
auto start_button = Sprite::create("newgame.png");
auto start_button_selected = Sprite::create("newgame.png");
MenuItemImage* menu_play = MenuItemImage::create();
menu_play->setNormalImage(start_button);
menu_play->setSelectedImage(start_button_selected);
menu_play->setCallback(CC_CALLBACK_1(HelloWorld::play, this));
menu->addChild(menu_play, 4);
```

Création de buttom play

La même chose pour les autre élément

```
menu_play->setPosition(Point(visibleSize.width / 2, (visibleSize.height / 2)+50 ));
sound_play->setPosition(Point(visibleSize.width / 2, (visibleSize.height / 2)));
menu_exit->setPosition(Point(visibleSize.width / 2, (visibleSize.height / 2)-50));
```

Leurs position dans la menu

```
// music
cocos2d::AudioEngine::preload("music.mp3"); //upload de notre music mp3
cocos2d::AudioEngine::play2d("music.mp3", true); //PLAY MUSIC
```

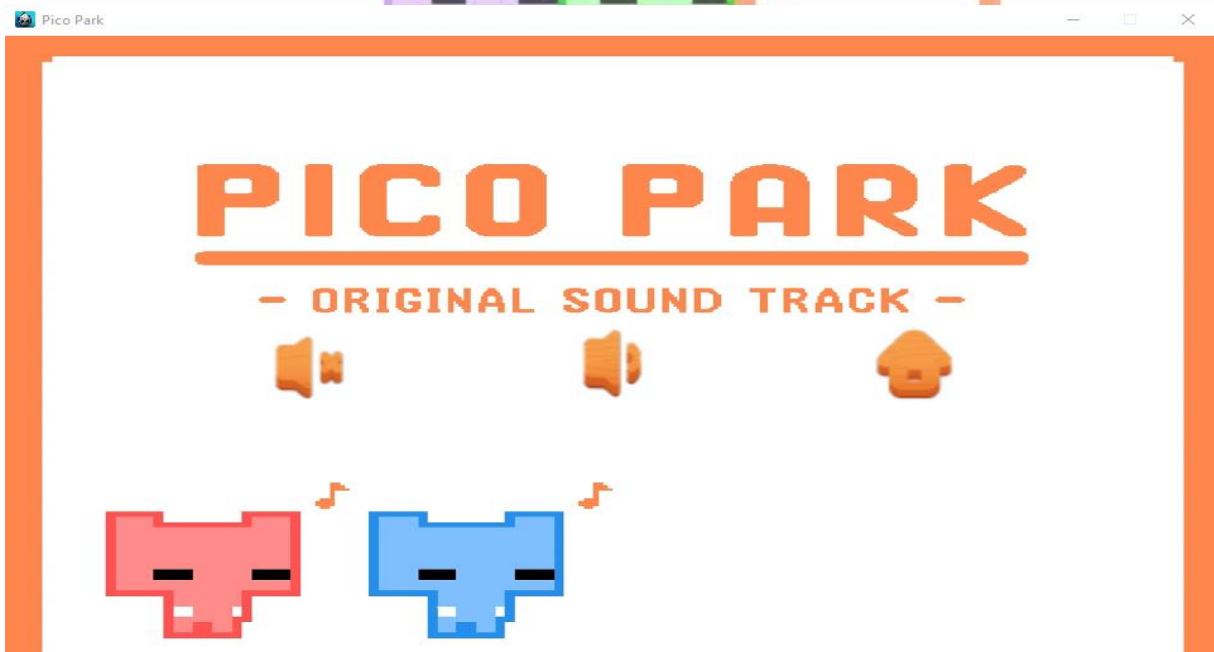
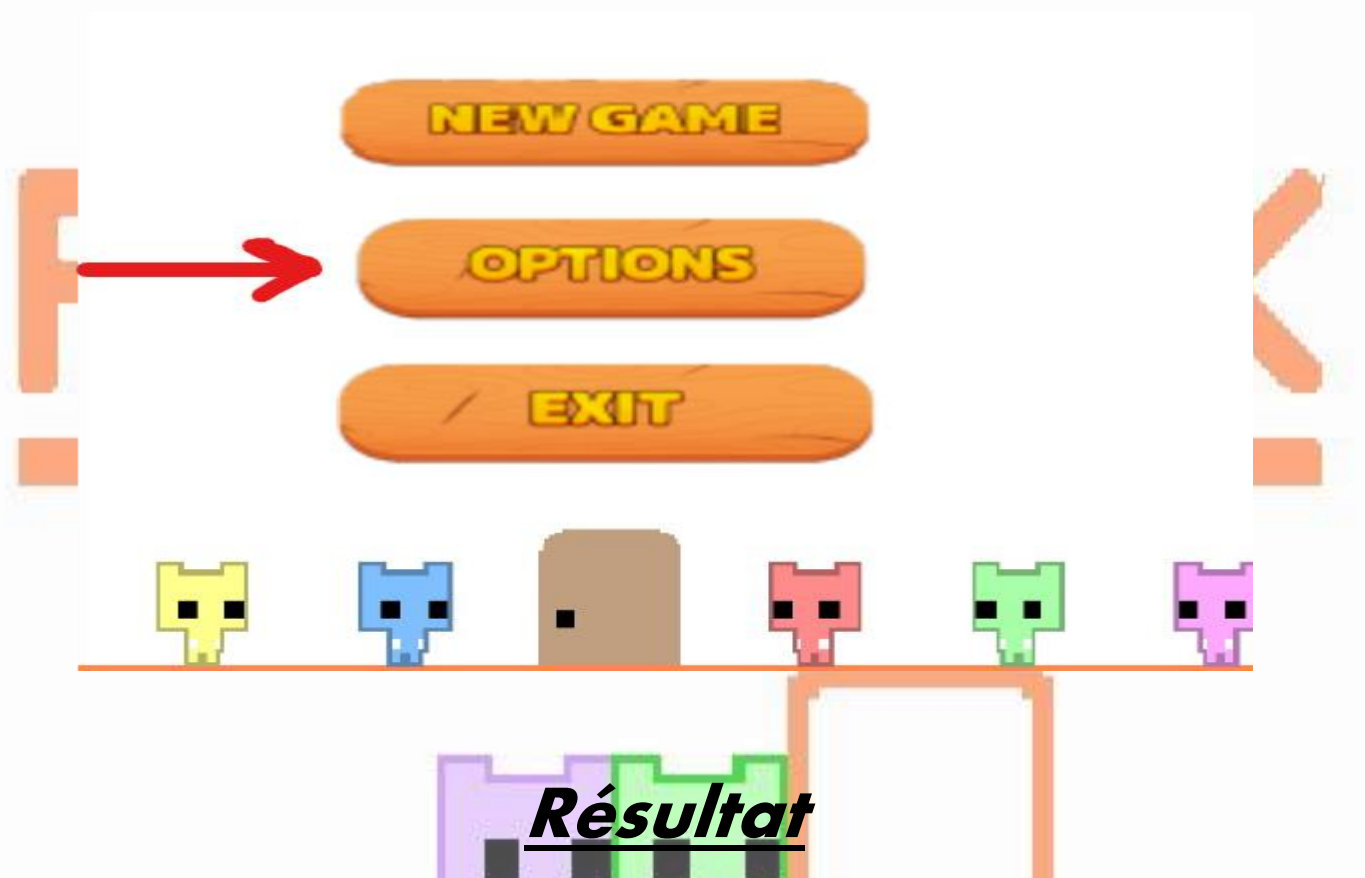
Création de la music

```
//impletation des fcts creees :
void HelloWorld::play(cocos2d::Ref* PSender) {
    CCLOG("PLAY"); //affichage de play
    //fct en cliquant sur play, on switch vers scene 1
    auto scene1 = SCENE1::createScene();
    Director::getInstance()->pushScene(TransitionSplitCols::create(1, scene1)); //la scene qu on veut y switch vers
}
```

Les fonction par exemple si on click sur play on switch vers scene 1

SoundScene

Si on click sur options on switch vers SoundScene



On a 3 Botton MUTE , MUSIC ET BACKHOME

SoundScene.h

```
class soundScene : public cocos2d::Layer
{
public:
    static cocos2d::Scene* createScene();
    virtual bool init();
    //a selector one
    void menuCloseCallback(cocos2d::Ref* pSender);

    CREATE_FUNC(soundScene);
    cocos2d::Sprite* menuback;
    cocos2d::Menu* menu1;
    //menu items
    void sound2(Ref* pSender);
    void music(Ref* pSender);
    void goback(Ref* pSender);
};
```

***On cree class «SoundScene »
Avec declaration des fonction suivantes***

Sound2 (MUTE)

Music (START LA MUSIC)

Goback (BACKHOME)

SoundScene.cpp

```
menuback = Sprite::create("soundbackground.png");  
menuback->setPosition(Point((visibleSize.width / 2) + origin.x, (visibleSize.height / 2) + origin.y));  
menuback->setScale(0.125);
```

Background est ça position

```
auto sound_button = Sprite::create("soundon.png");  
auto sound_button_selected = Sprite::create("soundon.png");  
MenuItemImage* menu_sound = MenuItemImage::create();  
menu_sound->setNormalImage(sound_button);  
menu_sound->setSelectedImage(sound_button_selected);  
menu_sound->setCallback(CC_CALLBACK_1(soundScene::sound2, this));  
  
menu->addChild(menu_sound, 4);  
  
menu_sound->setPosition(Point(visibleSize.width / 2, (visibleSize.height / 2)));
```

***Création de Buttom Sound est ça position.
La même chose pour les autres éléments***

```
void soundScene::music(cocos2d::Ref* pSender)  
{  
    CCLOG("MUSIC");  
    cocos2d::AudioEngine::stopAll();  
}
```

La fonction qui arrête la music

```
void soundScene::goback(cocos2d::Ref* pSender)  
{  
    CCLOG("GOBACK");  
    //fct en cliquant sur play, on switch vers  
    Director::getInstance()->popScene();  
}
```

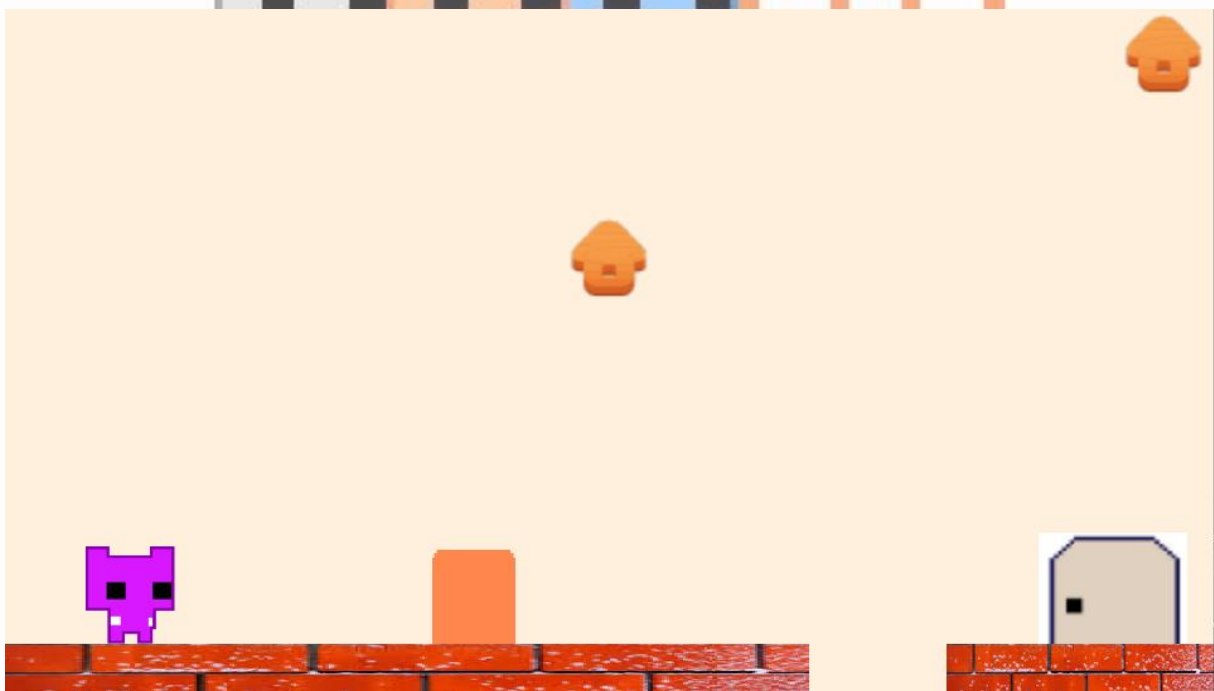
***La fonction qui nous permet switch vers la
scène dernier «HelloWorldscene »***

SCENE₁

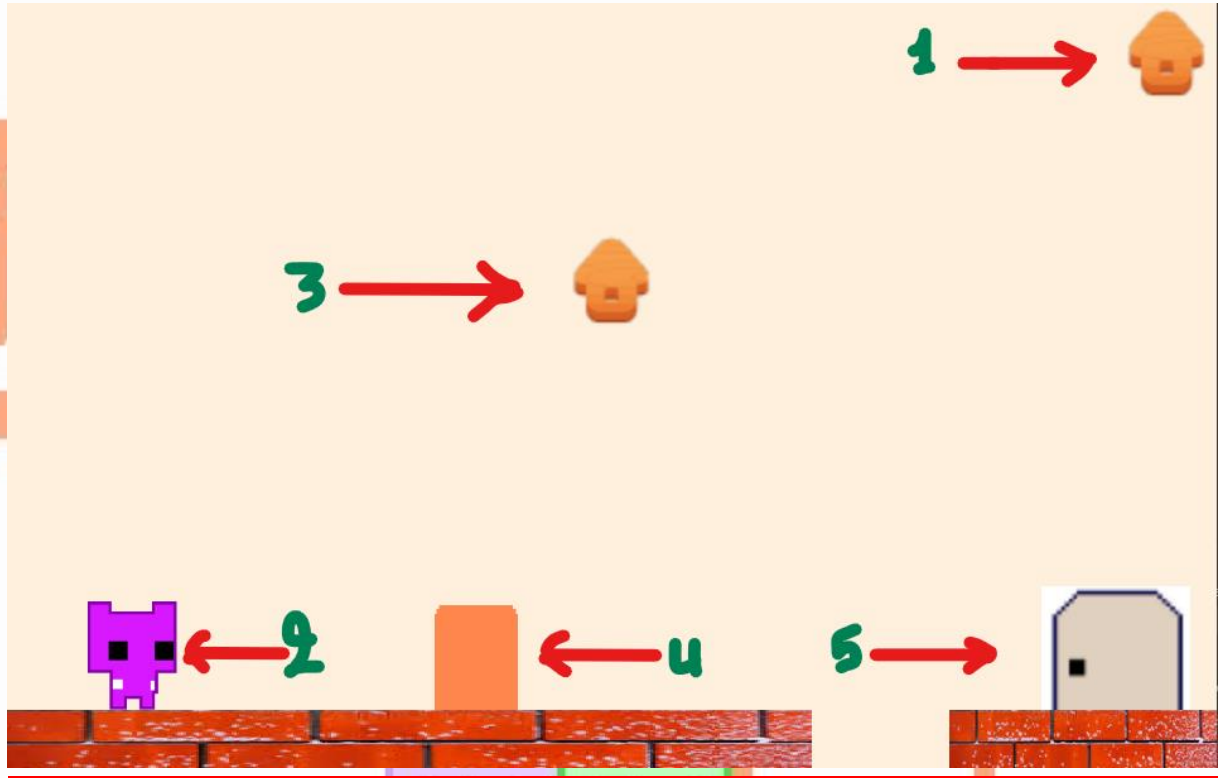
On switch vers SCENE₁ si on click sur le botton « NEW GAME »



Resultat :



SCENE₁



1. Goback :

Permet retour on arrier

2. Player :

Notre joueur

3. Obstacle 1

4. Obstacle 2

5. The door :

Pour passer a level 2

SCENE1.cpp

```
////////// add Sprite

auto pinfo = AutoPolygon::generatePolygon("Pico_Player_blue.png");
auto player = Sprite::create(pinfo);
player->setAnchorPoint(Vec2(0.5f, 0.5f));
player->setPosition(Vec2(50, 60));
player->setScale(0.3); //scale d'yal player
player->setName("player");
this->addChild(player, 2);
// creating physique for player

auto physicsBody1 = PhysicsBody::createBox(player->getContentSize(), PhysicsMaterial(100.0f, 0.5f, 0.5f));
physicsBody1->setGravityEnable(true);
physicsBody1->setDynamic(true);

physicsBody1->setCollisionBitmask(1);
physicsBody1->setCategoryBitmask(1);

player->setPhysicsBody(physicsBody1);
```

✓ Aucun problème détecté

Creation de player avec son physicsBody

```
auto background = Sprite::create("background.png");
background->setAnchorPoint(Vec2(0, 0));
background->setPosition(Vec2(0, 0));
background->setScale(0.125);
this->addChild(background, 0);
////////// creating the floor1(THE LAND)

auto floor = Sprite::create("l2ard1.png");
floor->setAnchorPoint(Vec2(0, 0));
floor->setPosition(Vec2(0, 0));
floor->setScale(0.125);
this->addChild(floor, 1);

//creating physique for THE FLOOR

auto physicsBody_floor = PhysicsBody::createBox(floor->getContentSize(), PhysicsMaterial(1500.0f, 0.1f, 0.9f));
physicsBody_floor->setDynamic(false);
physicsBody_floor->setCollisionBitmask(1);
physicsBody_floor->setCategoryBitmask(1);
```

Creation de FLOOR avec sa physics

LA MEME CHAUSE POUR (DOOR) AND (ABSTACLE).

```
// Create a keyboard event listener
auto keyboardListener = EventListenerKeyboard::create();
keyboardListener->onKeyPressed = CC_CALLBACK_2(SCENE1::onKeyPressed, this);
keyboardListener->onKeyReleased = CC_CALLBACK_2(SCENE1::onKeyPressed, this);

Director::getInstance()->getEventDispatcher()->addEventListenerWithSceneGraphPriority(keyboardListener, this);

keyboardListener->onKeyPressed = [player](EventKeyboard::KeyCode KeyCode, Event* event)
{
    if (KeyCode == EventKeyboard::KeyCode::KEY_UP_ARROW || KeyCode == EventKeyboard::KeyCode::KEY_SPACE) {
        auto action1 = JumpBy::create(0.7f, Vec2(80, 50), 60.0f, 1);
        player->runAction(action1);
    }
    if (KeyCode == EventKeyboard::KeyCode::KEY_RIGHT_ARROW || KeyCode == EventKeyboard::KeyCode::KEY_D) {
        auto jump = JumpBy::create(0.5f, Vec2(50, 50), 50.0f, 1);
        MoveBy* moveAction = MoveBy::create(2, Vec2(70, 0));
        RepeatForever* repeatAction = RepeatForever::create(moveAction);
        player->runAction(repeatAction);
    }
    if (KeyCode == EventKeyboard::KeyCode::KEY_LEFT_ARROW || KeyCode == EventKeyboard::KeyCode::KEY_A) {
        auto jump = JumpBy::create(0.5f, Vec2(50, 50), 50.0f, 1);
        MoveBy* moveAction = MoveBy::create(2, Vec2(-70, 0));
        RepeatForever* repeatAction = RepeatForever::create(moveAction);
        player->runAction(repeatAction);
    }
}
```

Les bouton ou le joueur peut marcher, marcher en arrière, jump

```
auto contactListener = EventListenerPhysicsContact::create();
contactListener->onContactBegin = [player](PhysicsContact& contact) {

    PhysicsBody* x = contact.getShapeA()->getBody();
    PhysicsBody* y = contact.getShapeB()->getBody();

    if (1 == x->getCollisionBitmask() && 3 == y->getCollisionBitmask() || 3 == x->getCollisionBitmask() && 1 == y->getCollisionBitmask()) {
        auto scene = SCENE2::createScene();
        Director::getInstance()->pushScene(TransitionFade::create(0.5, scene));
    }

    if (1 == x->getCollisionBitmask() && 5 == y->getCollisionBitmask() || 5 == x->getCollisionBitmask() && 1 == y->getCollisionBitmask()) {
        auto scene = Gameover::createScene();
        Director::getInstance()->pushScene(TransitionFade::create(0.5, scene));
    }
}
```

***Création des event comme si
Le player touch the door (go to the next scene)
Si le player tombe (scene game over)***

SCENE1.h

```
class SCENE1 : public cocos2d::Layer
{
public:
    static cocos2d::Scene* createScene();

    virtual bool init();

    CREATE_FUNC(SCENE1);

    void go(Ref* pSender);
    void goback(Ref* pSender);
    bool onContactBegin(cocos2d::PhysicsContact& contact);
    void door(Ref* pSender);
    void reload(Ref* pSender); //fct de remplacement de scene lors d echec

    //physics
    cocos2d::PhysicsWorld* sceneWorld;
    void setPhysicsWorld(cocos2d::PhysicsWorld* world) { sceneWorld = world; };
}
```

In **SCENE1.h** on a juste déclarer Les
element

SCENEGAMEOVER

Game Over

NEW GAME

EXIT

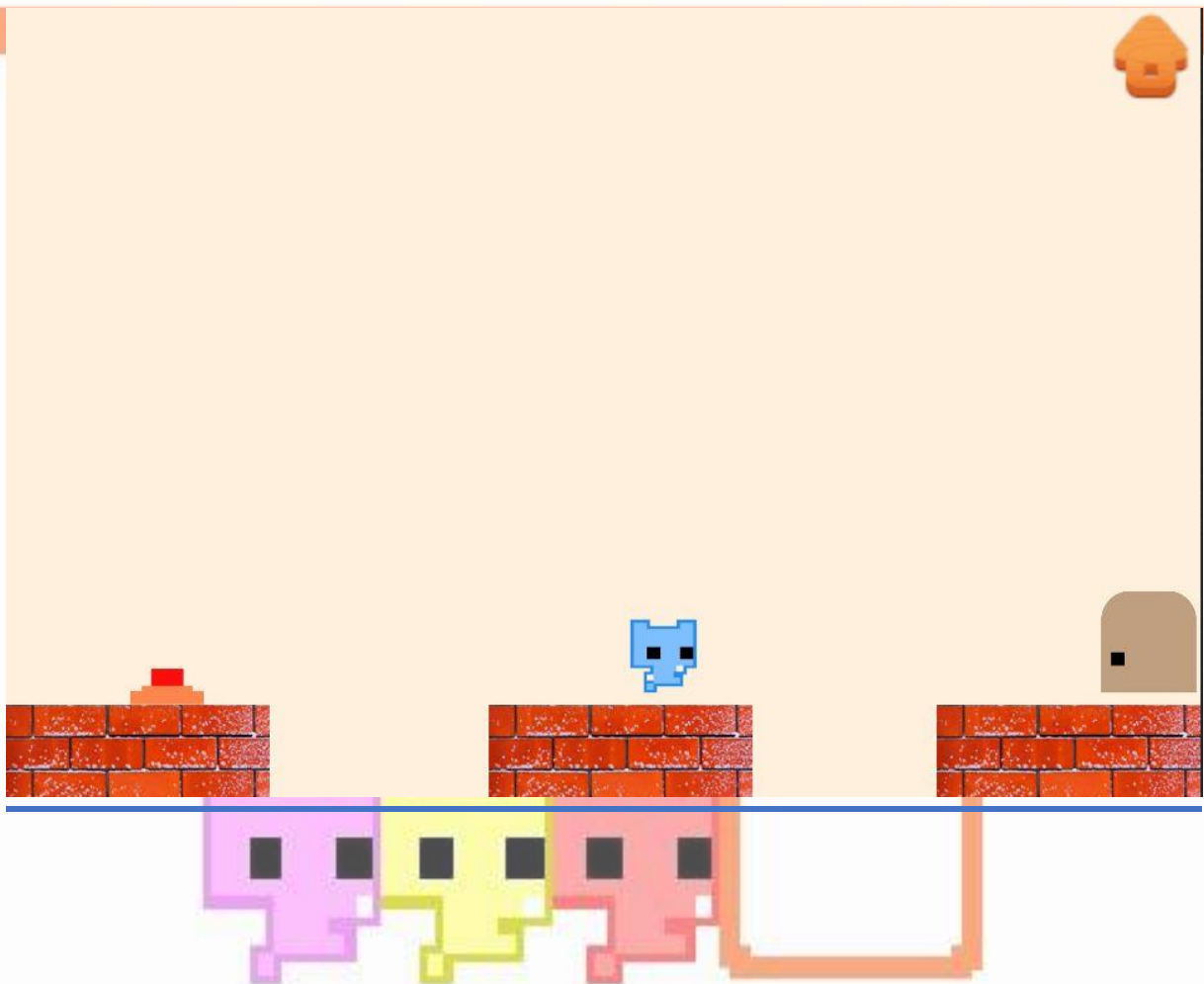


SCENE₂ :

**If the Player touche the door
We go to scene2**

PICO PARK

Résultat



SCENE2.cpp

```
auto visibleSize = Director::getInstance()->getVisibleSize();
auto origin = Director::getInstance()->getVisibleOrigin();
auto winSize = Director::getInstance()->getVisibleSize();

////////// background rectangle
auto edgeBody = PhysicsBody::createEdgeBox(visibleSize, PHYSICSBODY_MATERIAL_DEFAULT, 3);

auto edgeNode = Node::create();
edgeNode->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
edgeNode->setPhysicsBody(edgeBody);

this->addChild(edgeNode);

auto background_rect = DrawNode::create();
background_rect->drawSolidRect(origin, winSize, Color4F(1.0, 0.94, 0.87, 1.0)); //bg orange
this->addChild(background_rect);

////////// add Sprite
```

Physics de cadre pour le joueur nous sort pas .

```
auto pinfo = AutoPolygon::generatePolygon("Pico_Player_blue.png");
auto player = Sprite::create(pinfo);
player->setAnchorPoint(Vec2(0.5f, 0.5f));
player->setPosition(Vec2(20, 60));
player->setScale(0.3); //scale d'yal player
player->setName("player");
this->addChild(player, 2);
// creating physique for player

auto physicsBody1 = PhysicsBody::createBox(player->getContentSize(), PhysicsMaterial(1000.0f, 0.5f, 0.5f));
physicsBody1->setGravityEnable(true);
physicsBody1->setDynamic(true);
physicsBody1->setCollisionBitmask(1);
physicsBody1->setCategoryBitmask(1);
physicsBody1->setContactTestBitmask(1);
physicsBody1->setTag(1);
player->setPhysicsBody(physicsBody1);
```

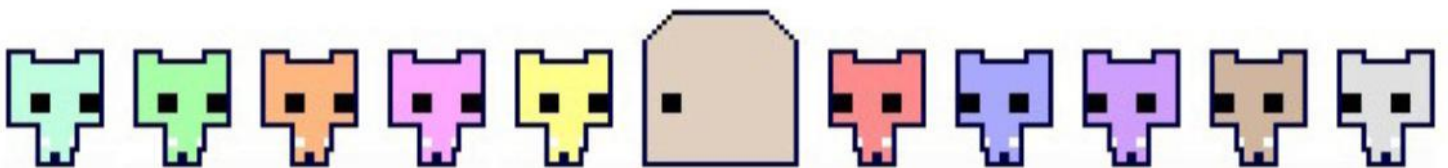
The players and he is physics .

***LA MEME CHOSE FOR THE DOOR AND BOTTOM
AND THE FLOOR***

***IF THE PLAYER TOUCH THE DOOR
GO TO WIN SCENE***

PIRAT POKK

YOU WIN! 



FIN