



Enjoy smart

ROLLER SPLAT

RAPPORT



AM - MARA KHAOULA
BOUSAID Chaïmae



Enjoy smart

Sommaire

- ◀ Les outiles
- ◀ Scene Home
- ◀ Scene Menu
- ◀ Scene Level One
- ◀ Scene Level Two
- ◀ Scene level Three
- ◀ Scene level Next
- ◀ Scene Game over

Enjoy smart

Les Outiles

On a utiliser le site "Canva"
pour tout ce qu'est design du
jeux:

- Image du fand
- Obstacele
- Desing des buttons
- Desing de boules



Enjoy smart

Scene Home



Enjoy smart

Scene Home

Pour crée le fond de la scene :

On a utiliser < sprite :: creation > pour cree le fond

Et < setposition > pour bien precise la position d'image

```
22 // declaration et position du fond
23 auto sprite = Sprite::create("Roller.png");
24 sprite->setScale(2);
25 sprite->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
26 this->addChild(sprite, 0);
```



Pour crée le bouton play :

On a utiliser < MenuItemImage :: creation > pour créé image du Button

Et < CC_CALLBACK_1 > pour lien la fonction 'play ' avec le button

Et < setposition > pour bien precise la position d'image

```
28 auto playItem = MenuItemImage::create("play_avant.png", "play_apres.png", CC_CALLBACK_1(Home::play, this));
29 playItem->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
30 playItem->setScale(2);
31 auto* menu = Menu::create(playItem, NULL);
32 menu->setPosition(Point::ZERO);
33 this->addChild(menu);
```

La fonction pour passer au menu

```
38 void Home::play(cocos2d::Ref* psender)
39 {
40     auto scene = MainMenuScene::createScene();
41     Director::getInstance()->pushScene(scene);
42 }
```

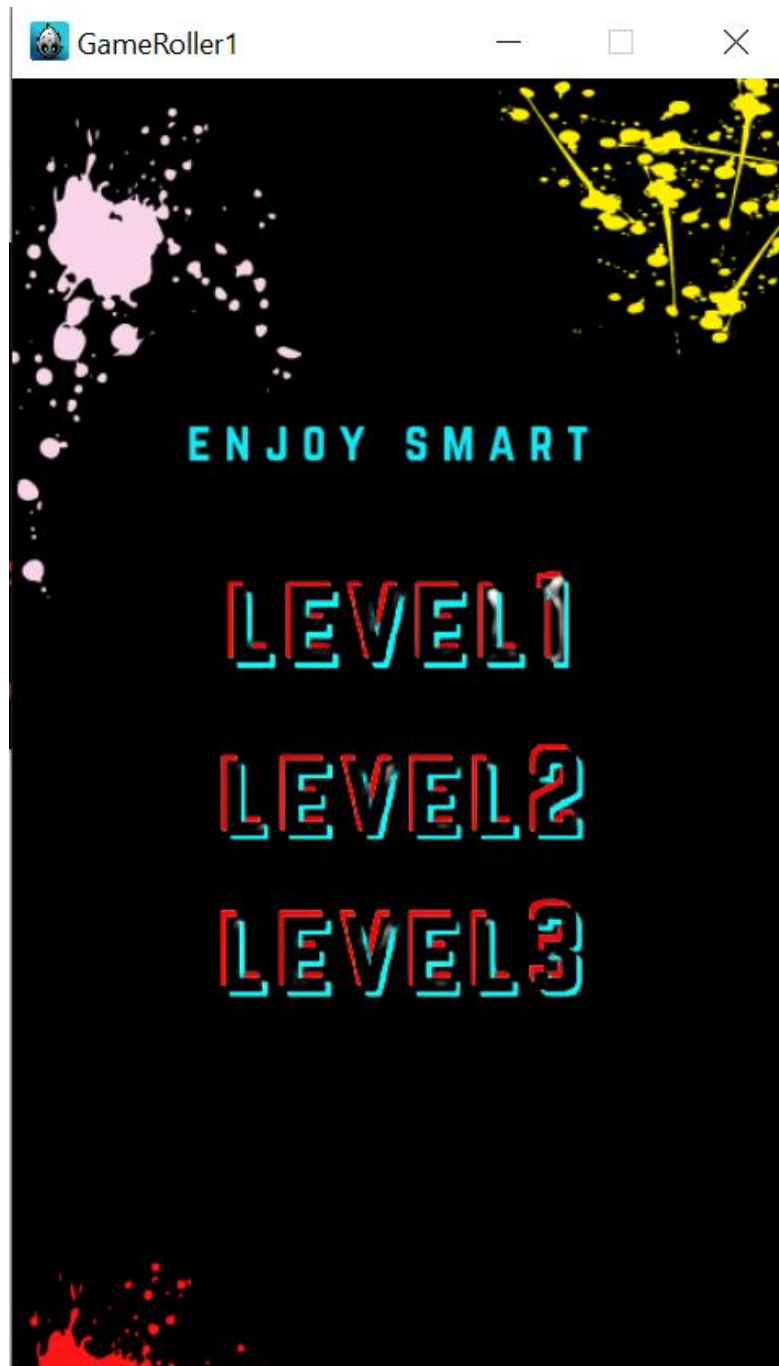


la Deffinition du classe Home :

```
2  #define __Home_SCENE_H__
3  #include "cocos2d.h"
4  using namespace cocos2d;
5
6  class Home : public Scene
7  {
8  public:
9      static Scene* createScene();
10
11      virtual bool init();
12
13      // implement the "static create()" method manually
14      CREATE_FUNC(Home);
15
16      cocos2d::Sprite* mySprite;
17      // declaration de la methode 'play'
18      void play(Ref* psender);
19
20 };
21 #endif // __Home_SCENE_H__
22
```

Enjoy smart

Scene Menu



Enjoy smart

Scene Menu



Pour crée le fand de la scene :

On a utiliser < sprite :: creation > pour cree le fond
Et < setposition > pour bien precise la position d'image

```
30 auto sprite = Sprite::create("menu.png");
31 sprite->setScale(2);
32 sprite->setPosition(Vec2(visibleSize.width / 2 + origin.x,
33 visibleSize.height / 2.3 + origin.y));
34 this->addChild(sprite, 0);
35
```



Pour crée le bouton 'levelOne' :

On a utiliser < MenuItemImage :: creation > pour créé image du
Button 'levelOne'
Et < CC_CALLBACK_1 > pour lien la fonction 'one' avec le button
Et < setposition > pour bien precise la position d'image

```
37 auto menu_item_1 = MenuItemImage::create("level1.png", "level1.png",
38 CC_CALLBACK_1(MainMenuScene::one, this));
39 menu_item_1->setPosition(Point(visibleSize.width / 2 + origin.x,
40 visibleSize.height / 2 + origin.y));
41 menu_item_1->setScale(2);
42 this->addChild(menu_item_1);
43
```

La fonction pour passer alevelOne

```
61 void MainMenuScene::one(cocos2d::Ref* psender)
62 {
63     CCLOG("level1.png");
64     auto scene = levelOne::createScene();
65     Director::getInstance()->replaceScene(scene);
66 }
```


Enjoy smart Scene Menu

Pour declaree le simages :



On a utiliser < sprite :: creation > pour crée les images de level2
Level3

```
45 auto menu_item_2 = Sprite::create("level2.png");
46 menu_item_2->setScale(2);
47 menu_item_2->setPosition(Vec2(visibleSize.width / 2 + origin.x,
48     visibleSize.height / 2 + origin.y));
49 this->addChild(menu_item_2, 0);
50
51 // declaration et position du button level3
52 auto menu_item_3 = Sprite::create("level3.png");
53 menu_item_3->setScale(2);
54 menu_item_3->setPosition(Vec2(visibleSize.width / 2 + origin.x,
55     visibleSize.height / 2 + origin.y));
56 this->addChild(menu_item_3, 0);
```

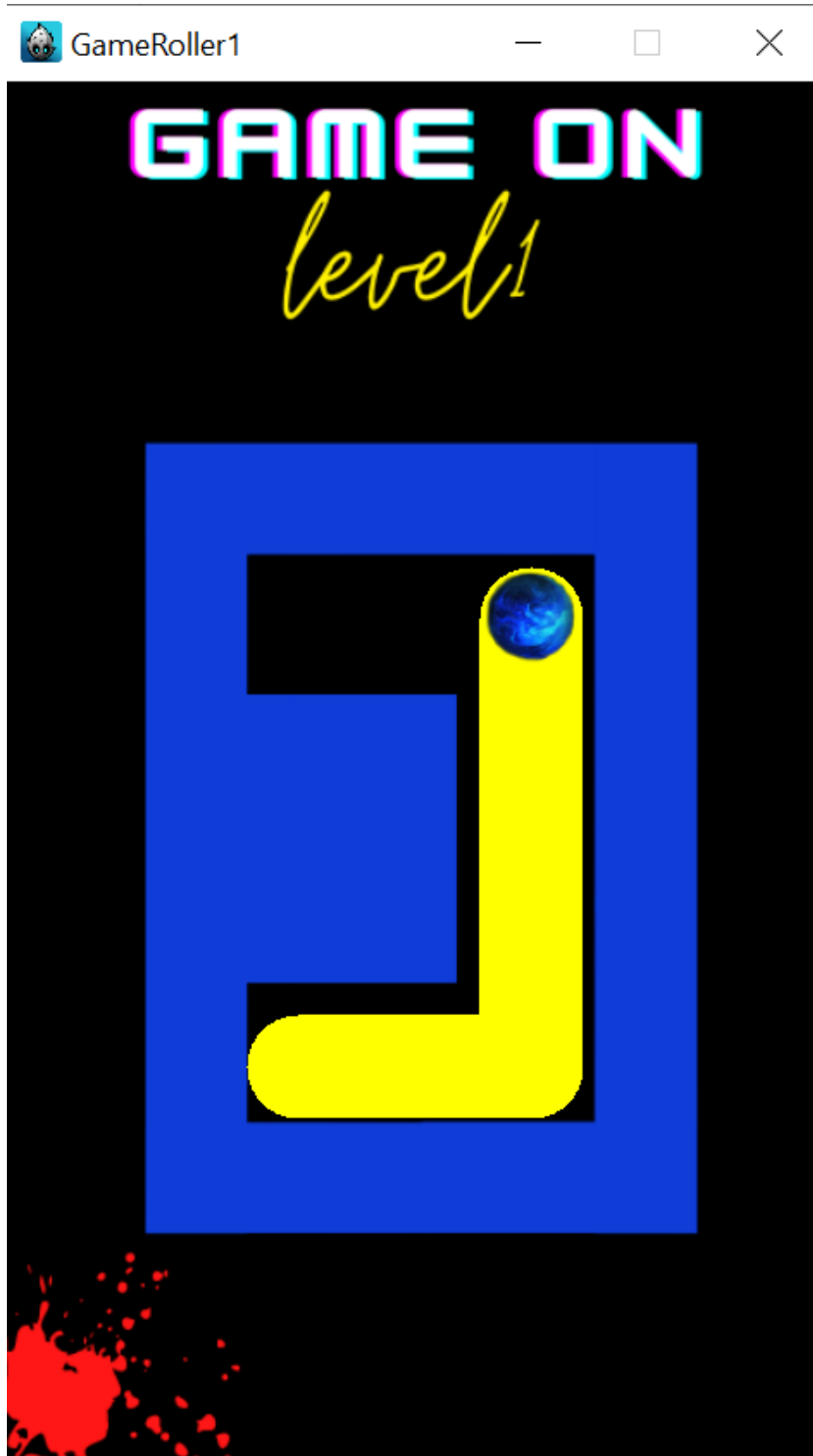
la Deffinition du classe Menu :



```
1  #ifndef __MAIN_MENU_SCENE_H__
2  #define __MAIN_MENU_SCENE_H__
3
4  #include "cocos2d.h"
5  using namespace cocos2d;
6
7  class MainMenuScene : public cocos2d::Scene
8  {
9  public:
10     static cocos2d::Scene* createScene();
11
12     virtual bool init();
13
14     // implement the "static create()" method manually
15
16     CREATE_FUNC(MainMenuScene);
17
18     cocos2d::Sprite* mySprite;
19     // declaration de la methode 'one'
20     void one(Ref* psender);
21 };
22 #endif // __MAIN_MENU_SCENE_H__
```

Enjoy smart

Scene level One



Scene level One

Pour crée le fond de la scene :

On a utiliser < sprite :: creation > pour cree le fond
Et < setposition > pour bien precise la position d'image



```
25 auto background_1 = Sprite::create("Roller.1.png");  
26 background_1->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2.1 + origin.y));  
27 this->addChild(background_1, 0);  
28 background_1->setScale(2);
```

Pour crée l'obstacle :

On a utiliser < sprite :: creation > pour cree l'obstacle
et < setposition > pour bien precise la position d'image



```
31 auto obstacle1 = Sprite::create("obstacle.png");  
32 obstacle1->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));  
33 this->addChild(obstacle1, 0);  
34 obstacle1->setScale(2);
```

Pour crée la boule :

On a utiliser < sprite :: creation > pour cree la boule
Et < setposition > pour bien precise la position d'image



```
35 boule = Sprite::create("boule.png");  
38 boule->setPosition(Vec2(238, 56));  
39 this->addChild(boule, 2);  
40 boule->setScale(2);
```



Scene level One

Pour faire la boule sauter :

On a utiliser < JumpBy :: creation > comme une methode pour Sauter la boule



```
42 auto action = JumpBy::create(2, Point(0, 0), 15, 1);  
43 boule->runAction(action);
```

Pour declare l'événement keyboard :



```
// declaration de l'événement keyboard  
auto eventListener = EventListenerKeyboard::create();  
eventListener->onKeyPressed = [=](EventKeyboard::KeyCode keycode, Event* event)
```

Pour declare l'événement keyboard :



Avec cette condition on donne la prochaine position de la boule avec Des pas precise

```
68 EventKeyboard::KeyCode::KEY_LEFT_ARROW // pour la click gauche  
69 x = event->getCurrentTarget()->getPositionX();  
70 y = event->getCurrentTarget()->getPositionY();  
71 if (x == 292 && y == 160)  
72 {  
73     offsetX = -54;  
74     // condition pour passer automatiquement de levelOne au page 'Nextlevel'  
75     auto action = JumpBy::create(1, Point(0, 0), 18, 2);  
76     boule->runAction(action);  
77     this->scheduleOnce(CC_SCHEDULE_SELECTOR(levelOne::next), DISPLAY_TIME_SPLASH_SCENE);  
78 }
```

La fonction pour passer de levelOne au next level

```
76     boule->fadeIn(1.0f);  
77     this->scheduleOnce(CC_SCHEDULE_SELECTOR(levelOne::next), DISPLAY_TIME, PLASH_SCENE);  
78 }
```

```
122 // declaration de la fonction pour passer au 'Nextlevel'  
123 void levelOne::next(float dt)  
124 {  
125     auto scene = levelNext::createScene();  
126     Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME, scene));  
127 }  
128 }
```

La fonction pour déplacer la boule:

```
106  
107 // fonction pour déplacer la boule  
108 auto Move = MoveTo::create(0.1, Vec2(event->getCurrentTarget()->getPositionX() + offsetX,  
109                                     event->getCurrentTarget()->getPositionY() + offsetY));  
110 event->getCurrentTarget()->runAction(Move);
```

La fonction pour colorer le chemin du boule :

```
111 //fonction de colorer le chemin du boule  
112 auto color = DrawNode::create();  
113 color->drawSegment(Vec2(x - 25, y + 35), Vec2(x - 25 + offsetX, y + offsetY + 35), 12.0f, Color4F::YELLOW);  
114 this->addChild(color, 1);
```

La fonction pour lier l'événement de keyboard avec la boule:

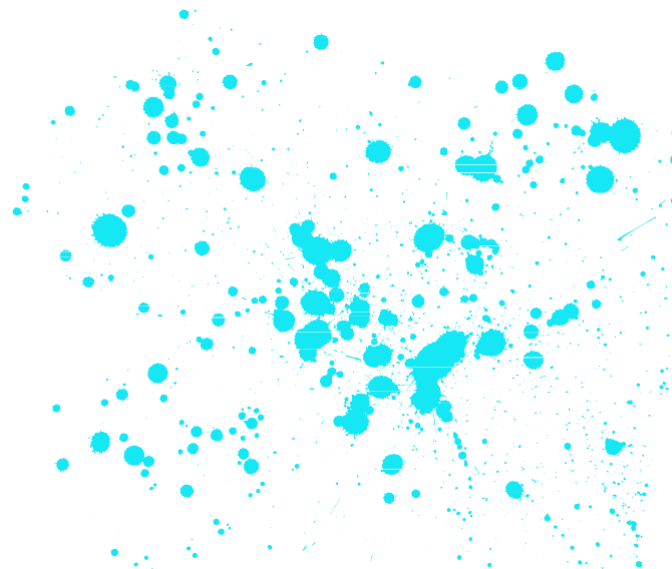
```
119 this->_eventDispatcher->addEventListenerWithSceneGraphPriority(eventListener, boule);  
120 }
```

Scene level One

la Definition du classe levelOne :



```
1  #ifndef __LEVEL_ONE_SCENE_H__
2  #define __LEVEL_ONE_SCENE_H__
3  #include "cocos2d.h"
4  using namespace cocos2d;
5  class levelOne : public Scene
6  {
7  public:
8      static Scene* createScene();
9
10     virtual bool init();
11     cocos2d::Sprite* boule;
12
13     // implement the "static create()" method manually
14     CREATE_FUNC(levelOne);
15     // declaration de la methode 'next'
16     void next(float dt);
17     // declaration de la methode de coloration
18     DrawNode* color;
19 };
20 #endif // LEVEL_ONE_SCENE_H__
```

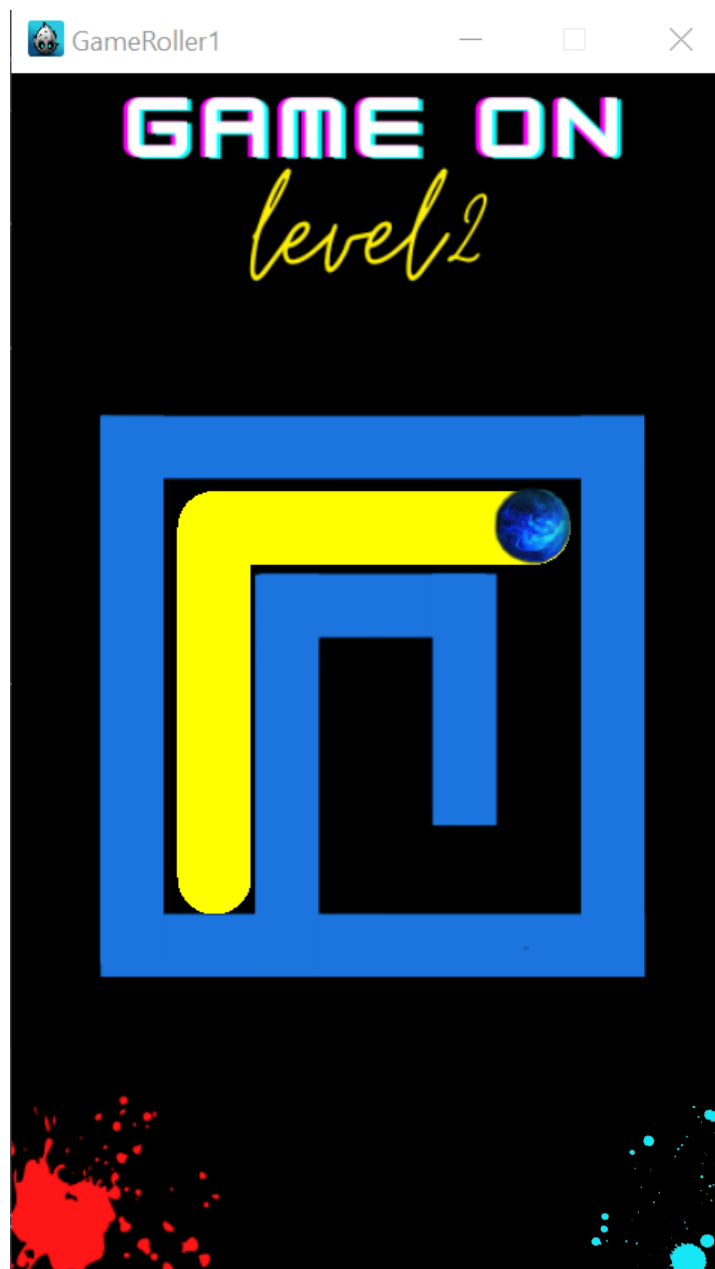


Enjoy smart

Scene level TWO



Pour la Scene Two on a fais les memes étapes que la premiere scene

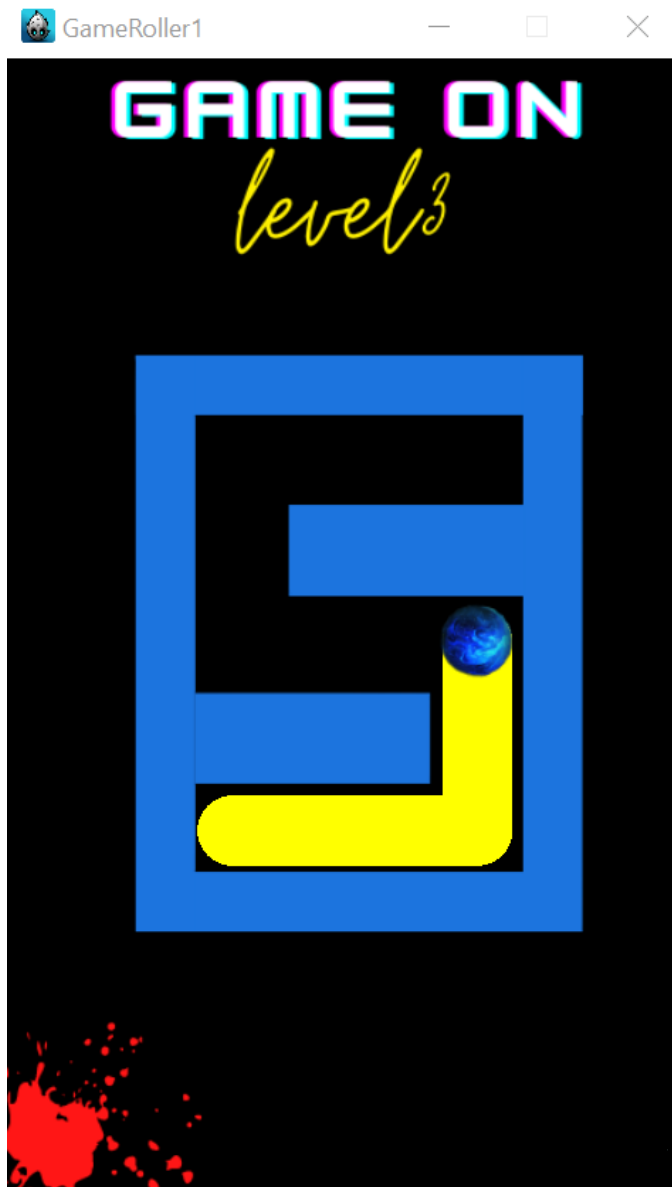


Enjoy smart

Scene level Three

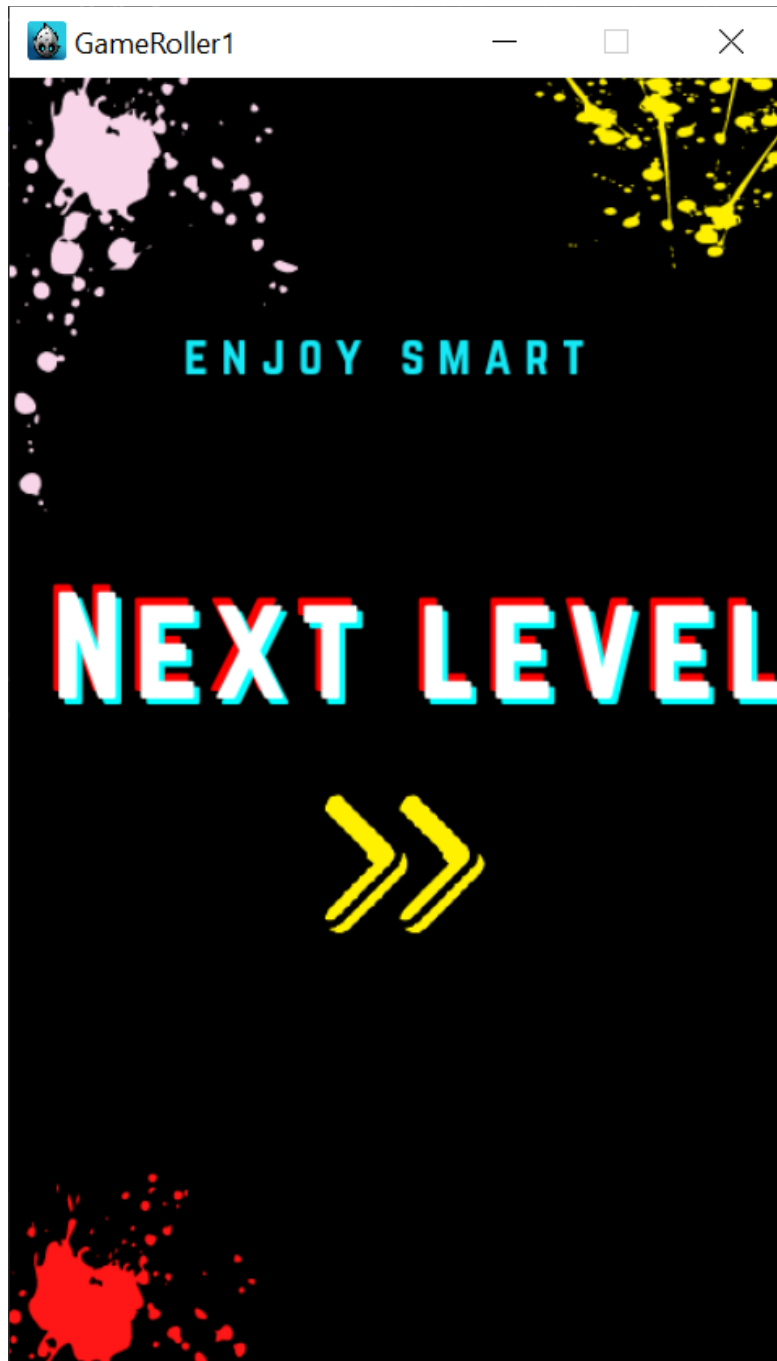


Pour la Scene Two on a fais les memes étapes que la premiere et la deuxieme scene



Enjoy smart

Scene level Next



Enjoy smart

Scene levelNext



Pour crée le fand de la scene :

On a utiliser < sprite :: creation > pour cree le fond
Et < setposition > pour bien precise la position d'image

```
24 auto background_Next = Sprite::create("next.png");  
25 background_Next->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));  
26 this->addChild(background_Next, 0);  
27 background_Next->setScale(2);
```



Pour crée le bouton 'levelOne' :

On a utiliser < MenuItemImage :: creation > pour créé image du
Button 'levelOne'
Et < CC_CALLBACK_1 > pour lien la fonction 'one' avec le button
Et < setposition > pour bien precise la position d'image

```
28  
29 auto playItem = MenuItemImage::create("bouton_next.png", "bouton_next.png",  
30 CC_CALLBACK_1(levelNext::next, this));  
31 playItem->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));  
32 playItem->setScale(2);  
33 auto* menu = Menu::create(playItem, NULL);  
34 menu->setPosition(Point::ZERO);  
35 this->addChild(menu);
```

La fonction pour passer alevelTwo

```
38 void levelNext::next(cocos2d::Ref* psender)  
39 {  
40     auto scene = levelTwo::createScene();  
41     Director::getInstance()->pushScene(scene);  
42 }
```

Enjoy smart

Scene levelNext

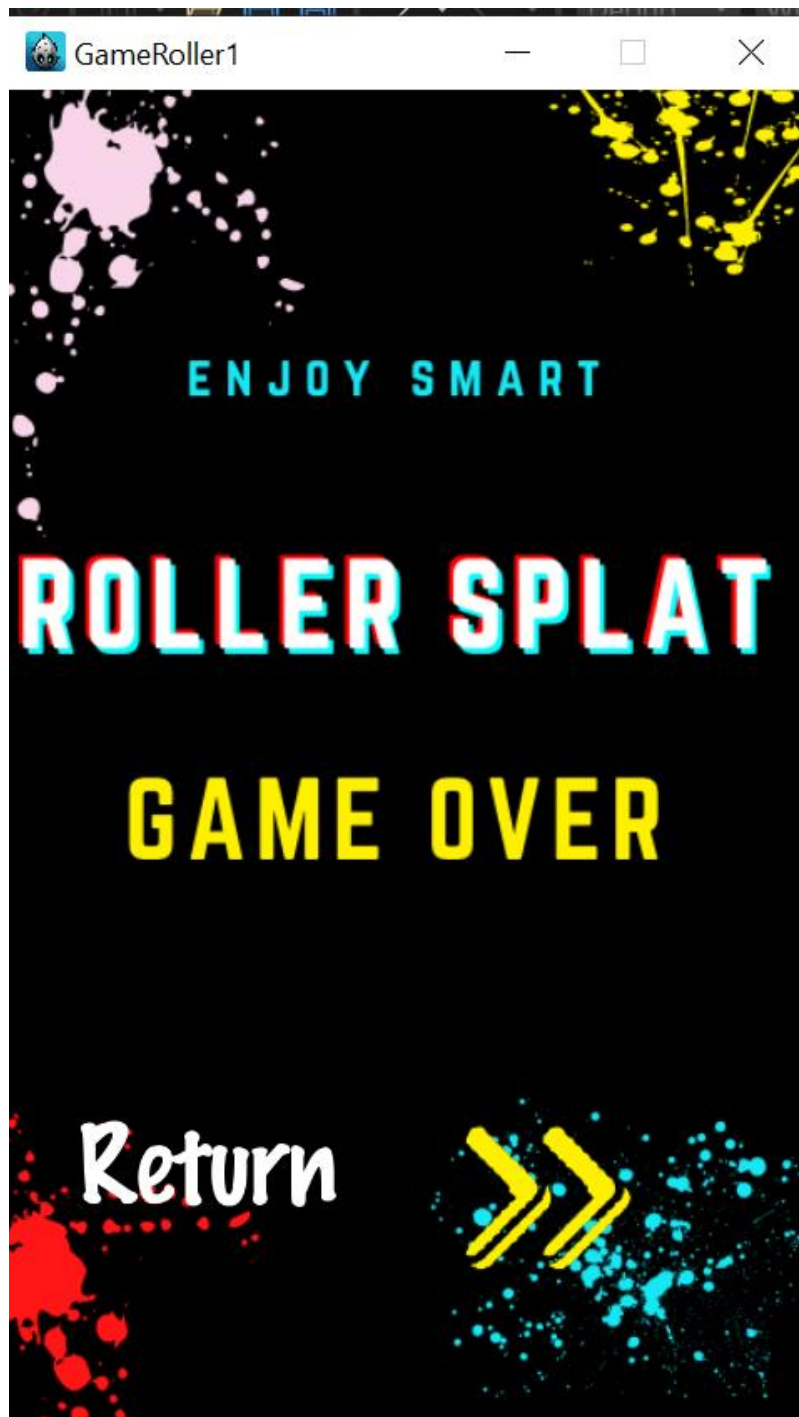


la Deffinition du classe levelNext :

```
1  #ifndef __LEVEL_NEXT_SCENE_H__
2  #define __LEVEL_NEXT_SCENE_H__
3  #include "cocos2d.h"
4  using namespace cocos2d;
5  class levelNext : public Scene
6  {
7  public:
8      static Scene* createScene();
9      virtual bool init();
10     cocos2d::Sprite* boule;
11     // a selector callback
12     void menuCloseCallback(Ref* pSender);
13     // implement the "static create()" method manually
14     CREATE_FUNC(levelNext);
15     cocos2d::Sprite* mySprite;
16     void next(Ref* psender);
17 };
18 #endif // LEVEL_NEXT_SCENE_H__
```

Enjoy smart

Scene Game over



Enjoy smart

Scene Game over



Pour crée le fand de la scene :

On a utiliser < sprite :: creation > pour cree le fond
Et < setposition > pour bien precise la position d'image

```
23 // creation et position du background
24 auto background = Sprite::create("GameOver.png");
25 background->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
26 this->addChild(background);
27 background->setScale(2);
28
```



Pour crée le bouton :

On a utiliser < MenuItemImage :: creation > pour créé image du
Button
Et < CC_CALLBACK_1 > pour lien la fonction 'one' avec le button
Et < setposition > pour bien precise la position d'image

```
33 auto playItem = MenuItemImage::create("bouton_next.png", "bouton_next.png",
34                                     CC_CALLBACK_1(GameOver::next, this));
35 playItem->setPosition(Point(visibleSize.width / 1.5 + origin.x, visibleSize.height / 3.6 + origin.y));
36 playItem->setScale(2);
37 auto* menu = Menu::create(playItem, NULL);
38 menu->setPosition(Point::ZERO);
39 this->addChild(menu);
```

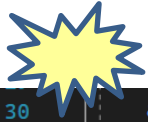
La fonction pour passer a levelTwo

```
42 void GameOver::next(cocos2d::Ref* psender)
43 {
44     auto scene = MainMenuScene::createScene();
45     Director::getInstance()->pushScene(scene);
46 }
47
```

Enjoy smart

Scene Game over

Pour crée le Texte :




```
30 auto label = Label::createWithTTF("Return", "fonts/Marker Felt.ttf", 24);  
31 label->setPosition(Vec2(origin.x + visibleSize.width / 4, origin.y + visibleSize.height / 3.7  
32                          - label->getContentSize().height));  
33 this->addChild(label, 1);
```

Enjoy smart

Scene Game over

la Definition du classe Game Over :



```
1 #ifndef __GAME_OVER_H__
2 #define __GAME_OVER_H__
3 #include "cocos2d.h"
4 using namespace cocos2d;
5 class GameOver : public Scene
6 {
7 public:
8     static Scene* createScene();
9
10    virtual bool init();
11    cocos2d::Sprite* boule;
12    // a selector callback
13    void menuCloseCallback(Ref* pSender);
14
15    // implement the "static create()" method manually
16    CREATE_FUNC(GameOver);
17
18    cocos2d::Sprite* mySprite;
19    void next(Ref* psender);
20 };
21 #endif // __GAME_OVER_H__
22
```

