
SUJET : Jeux Vidéo 2D (PICO PARK)

Programmation orientée objet en C++



Réalisé par :

- BERRADI Ibrahim
- SAHLI Mustapha

Encadré par :

Mme. BEN ABDEL WAHAB Ikram

Année universitaire : 2022/2023

Dédicace :

Au nom de notre binôme nous tenons de prime abord à remercier très profondément Mme. BEN ABDEL WAHAB Ikram notre professeure chargée du module programmation orientée objet en C++ pour sa patience , son dévouement et son soutien lors de la réalisation de ce modeste projet , sans oublier également de remercier toutes celles et ceux qui nous ont aidé de près ou de loin pour donner naissance à cet ouvrage .

Introduction :

Il va sans dire que la réalisation d'un projet informatique nécessite l'acquisition de plusieurs informations dans différents langages utilisés pour pouvoir en donner naissance. Notre modeste projet représente un travail acharné qui porte sur la réalisation d'un jeu 2D intitulé PIKO PARK en utilisant le moteur de jeux Cocos 2D avec trois niveaux. Notre travail comprend.

Objectifs de notre travail :

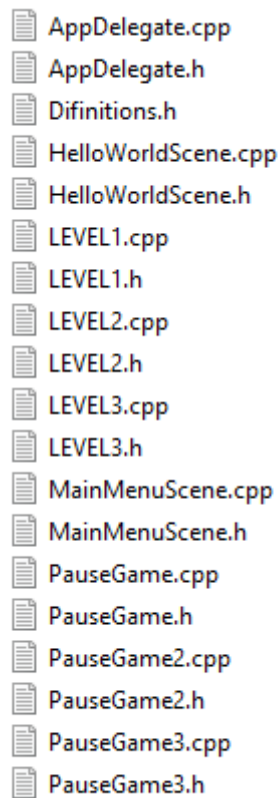
L'objectif principal de ce projet est de maîtriser la programmation orientée objet par la mise en place d'un jeu vidéo 2D, le jeu proposé s'appelle PICO PARK, un jeu de type Platformer Puzzle Game.

Le travail demandé :

- Développer le jeu 2D via le moteur de jeu Cocos2D « C++ » avec le respect du paradigme POO.
- Réaliser au moins 3 niveaux de jeu.
- Le jeu doit être en mode mono player.

Outils : C++ , Cocos2D, Photoshop/Tiled.

Commençons par une vue globale sur notre projet :



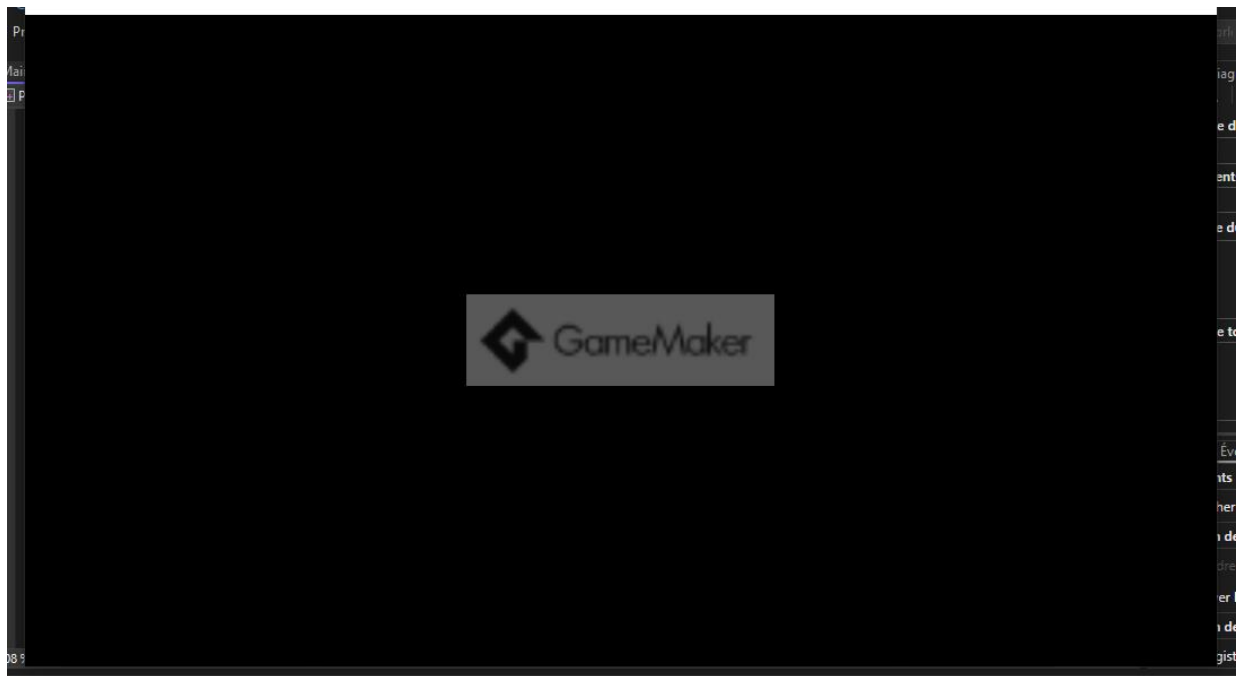
- AppDelegate.cpp
- AppDelegate.h
- Difinitions.h
- HelloWorldScene.cpp
- HelloWorldScene.h
- LEVEL1.cpp
- LEVEL1.h
- LEVEL2.cpp
- LEVEL2.h
- LEVEL3.cpp
- LEVEL3.h
- MainMenuScene.cpp
- MainMenuScene.h
- PauseGame.cpp
- PauseGame.h
- PauseGame2.cpp
- PauseGame2.h
- PauseGame3.cpp
- PauseGame3.h

Comme vous voyez il y'a plusieurs fichiers de code source C++ avec leurs fichier `header` chaque fichier contient une scène de jeu par exemple le fichier LEVEL1.cpp c'est le level 1 de jeu

On également le fichier PauseGame.cpp et les autres qui ont presque le même nom , ces fichiers contiennent les codes des boutons **Pause** , **Restart** , **Home** et **Next** .

[Passons maintenant aux détails :](#)

Une fois accéder au jeu on trouve un logo s'affiche :



Cet affichage a réalisé par le code C++ suivant :

```
this->scheduleOnce(CC_SCHEDULE_SELECTOR(HelloWorld::GoToMainMenu),  
DISPLAY_TIME_SPLASH_SCENE );  
  
    auto backgroundSprite = Sprite::create("background/images.png");  
    backgroundSprite->setPosition(Point(visibleSize.width / 2 + origin.x,  
visibleSize.height / 2 + origin.y));  
    this->addChild(backgroundSprite);
```

qui se trouve dans la classe **HelloWorldScene** .

quelque secondes après on se trouve face au **menu** de notre jeu qui contient un background sur lequel il y'a les buttons **Play** et **Exit** .



Le bouton **Play** permet de commencer le jeu , ce bouton a été réalisé par le code suivant :

```
auto playItem = MenuItemFont::create("Play",
CC_CALLBACK_1(MainMenuScene::GoToGameScene, this));
playItem->setColor(cocos2d::Color3B(255, 134, 77));
playItem->setFontName("fonts/Marker Felt.ttf");
playItem->setFontSize(30);
playItem->setPosition(Point(visibleSize.width / 2 + origin.x,
visibleSize.height * 0.6));

auto menu = Menu::create(playItem, NULL);
menu->setPosition(Point::ZERO);

this->addChild(menu);
```

à l'aide de la fonction suivante :

```
void MainMenuScene::GoToGameScene(Ref* sender)
{
    auto scene = LEVEL1::createScene();
    Director::getInstance()-
>replaceScene(TransitionFade::create(TRANSITION_TIME, scene));
}
```

Le bouton **Exit** permet de quitter le jeu , ce bouton a été réalisé par le code suivant :

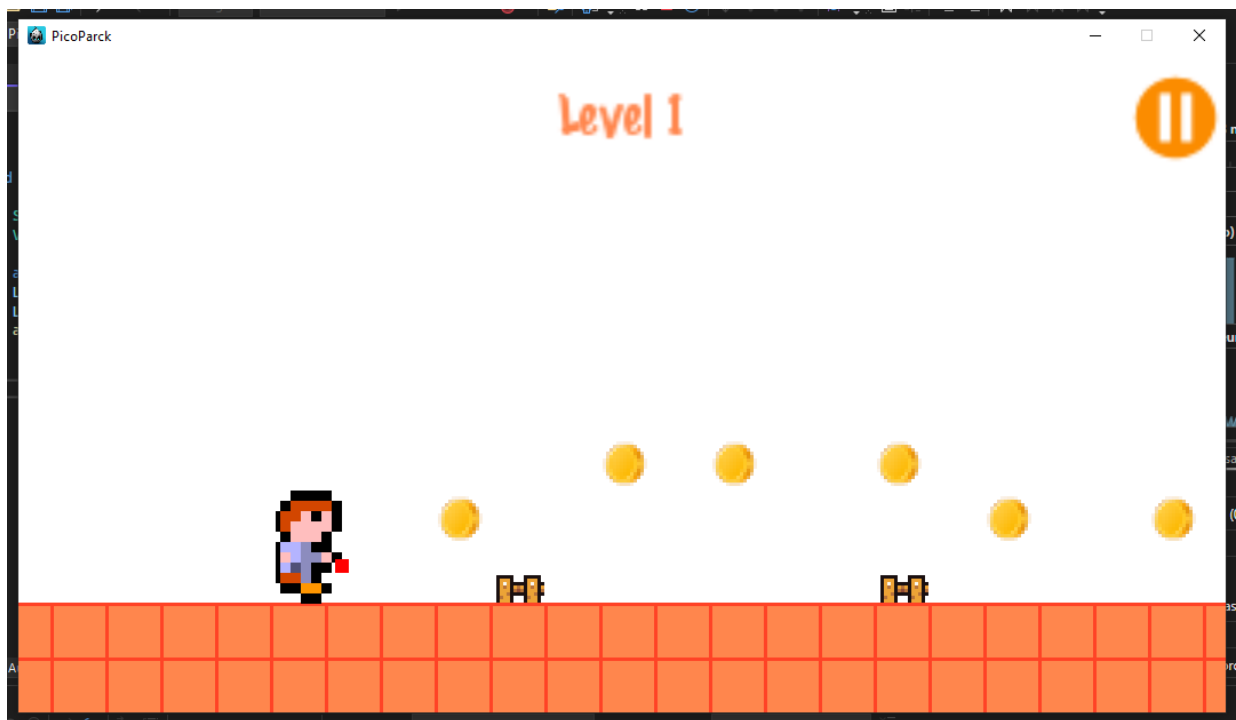
```
auto closeItem = MenuItemFont::create("Exit",
CC_CALLBACK_1(MainMenuScene::menuCloseCallback, this));
closeItem->setColor(cocos2d::Color3B(255, 134, 77));
closeItem->setFontName("fonts/Marker Felt.ttf");
closeItem->setFontSize(30);
if (closeItem == nullptr ||
    closeItem->getContentSize().width <= 0 ||
    closeItem->getContentSize().height <= 0)
{
    problemLoading("'Exit' and 'Exit'");
}
else
{
    float x = origin.x + visibleSize.width / 2;
    float y = origin.y + visibleSize.height * 0.4;
    closeItem->setPosition(Vec2(x, y));
}
```

à l'aide de la fonction suivante :

```
void MainMenuScene::menuCloseCallback(Ref* pSender)
{
    CCLOG("Exit");

    Director::getInstance()->end();
}
```

Après avoir cliqué une seule fois sur le bouton **Play** on passe à **Level 1** , comme le montre l'image suivante :



Premièrement on a fait la scène de notre jeu par le programme **Tiled** qui nous a donné l'opportunité de mettre la position des objets plus précisément .

On voit L'objet qui bouge donne le but de collecter des pièces de monnaie sans toucher les obstacles pour ne pas perdre le jeu.

On voit également en haut de la scène du jeu le nom de le premier level , on a fait ce nom en utilisant la fonction suivante :

```
void LEVEL1::NomLEVEL1(float dt)
{
```

```

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

auto Level1Label = Label::createWithTTF("Level 1", "fonts/Marker Felt.ttf",
15);
Level1Label->setPosition(Point(visibleSize.width / 2 + origin.x,
visibleSize.height * 0.9));
Level1Label->setColor(cocos2d::Color3B(255, 134, 77));
addChild(Level1Label);
}

```

On a un bouton qui permet d'arrêter le jeu , qui a réalisé par le code suivant :

```

auto pause = MenuItemImage::create("menu/pause.png", "menu/pause.png",
CC_CALLBACK_1(LEVEL1::MenuPause, this));
pause->setPosition(Point(visibleSize.width * 0.96, visibleSize.height *
0.9));
auto menupause = Menu::create(pause, NULL);
menupause->setPosition(Point::ZERO);
this->addChild(menupause);

```

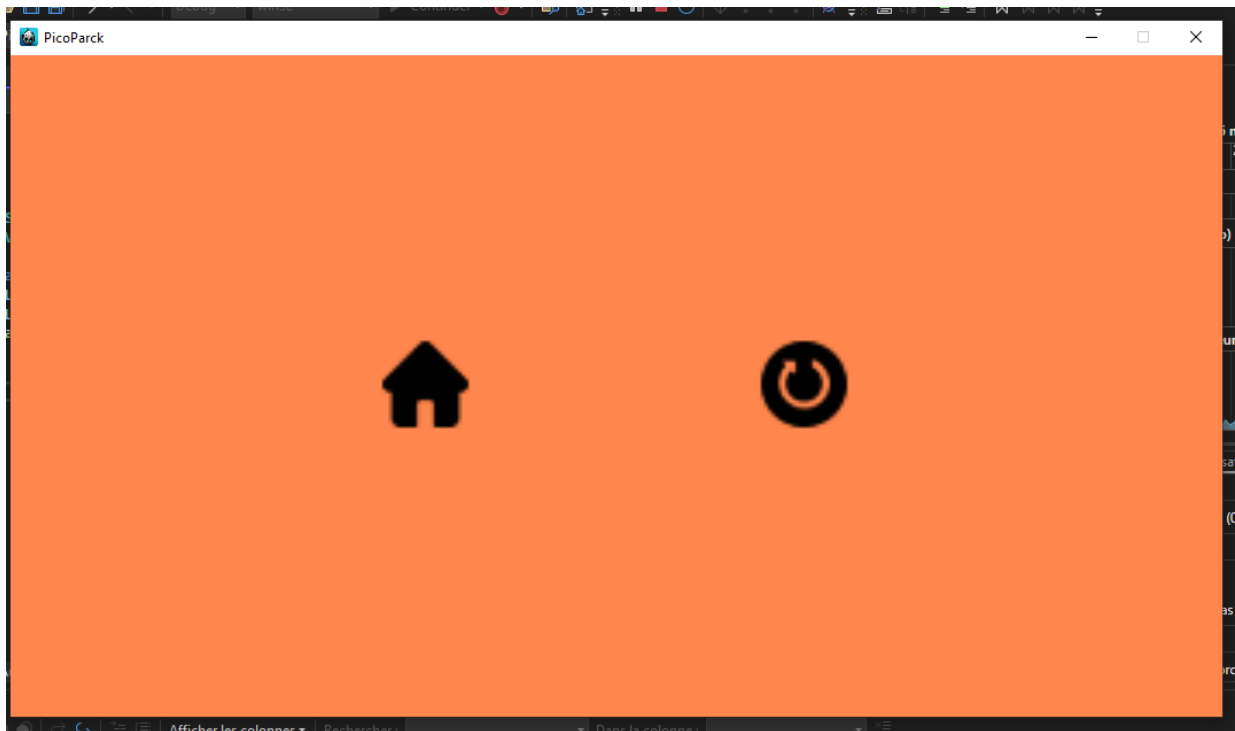
à l'aide de la fonction suivant :

```

void LEVEL1::MenuPause(Ref* sender)
{
    auto scene = PauseGame::createScene();
    Director::getInstance()-
>replaceScene(TransitionFade::create(TRANSITION_TIME_2, scene));
}

```

quand on clique sur ce bouton on trouve l'affichage suivant :



Dans cet affichage on trouve deux boutons , l'un permet de revenir au menu et l'autre permet de recommencer le jeu .

Le premier bouton a réalisé en utilisant le code suivant :

```
auto home = MenuItemImage::create("menu/home.png", "menu/homeSelected.png",  
CC_CALLBACK_1(LEVEL1::GoToGameHome, this));
```

à l'aide de la fonction suivant :

```
void LEVEL1::GoToGameHome(Ref* sender)  
{  
    auto scene = MainMenuScene::createScene();  
    Director::getInstance()-  
>replaceScene(TransitionFade::create(TRANSITION_TIME, scene));  
}
```

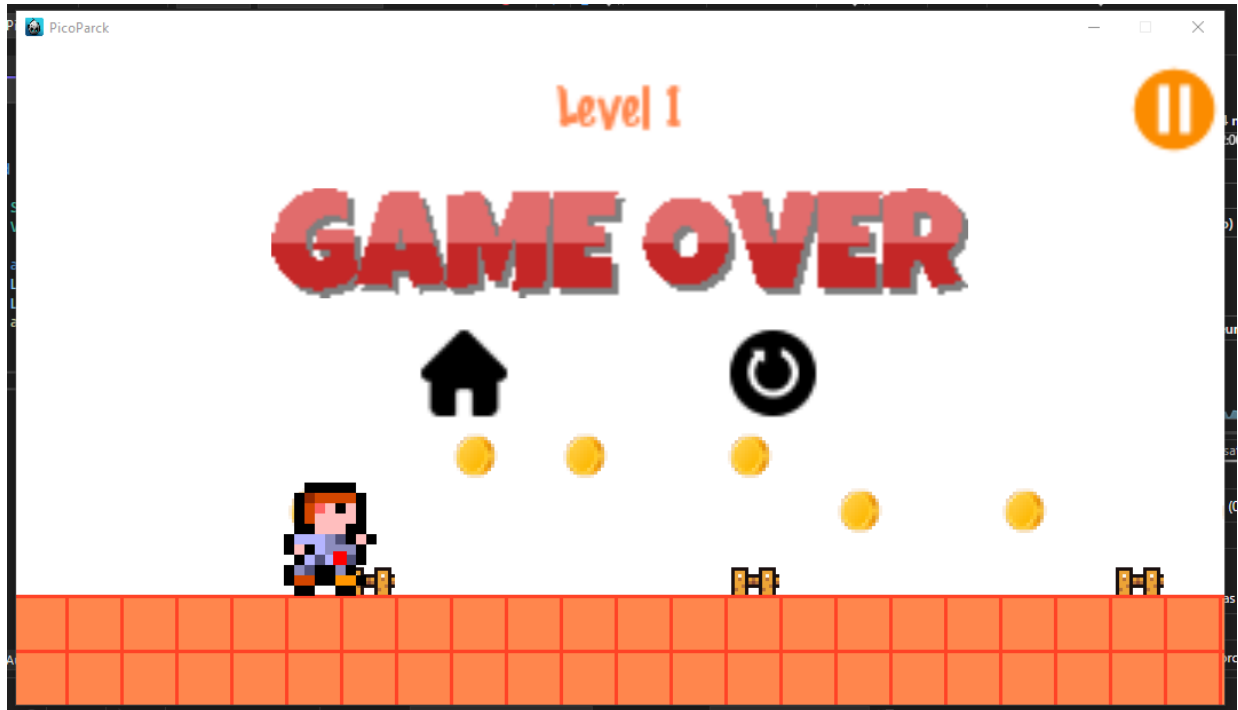
L'autre bouton a réalisé en utilisant le code suivant :

```
auto restart = MenuItemImage::create("menu/restart.png",  
"menu/restartSelected.png", CC_CALLBACK_1(LEVEL1::GoToGameLEVEL1, this));
```

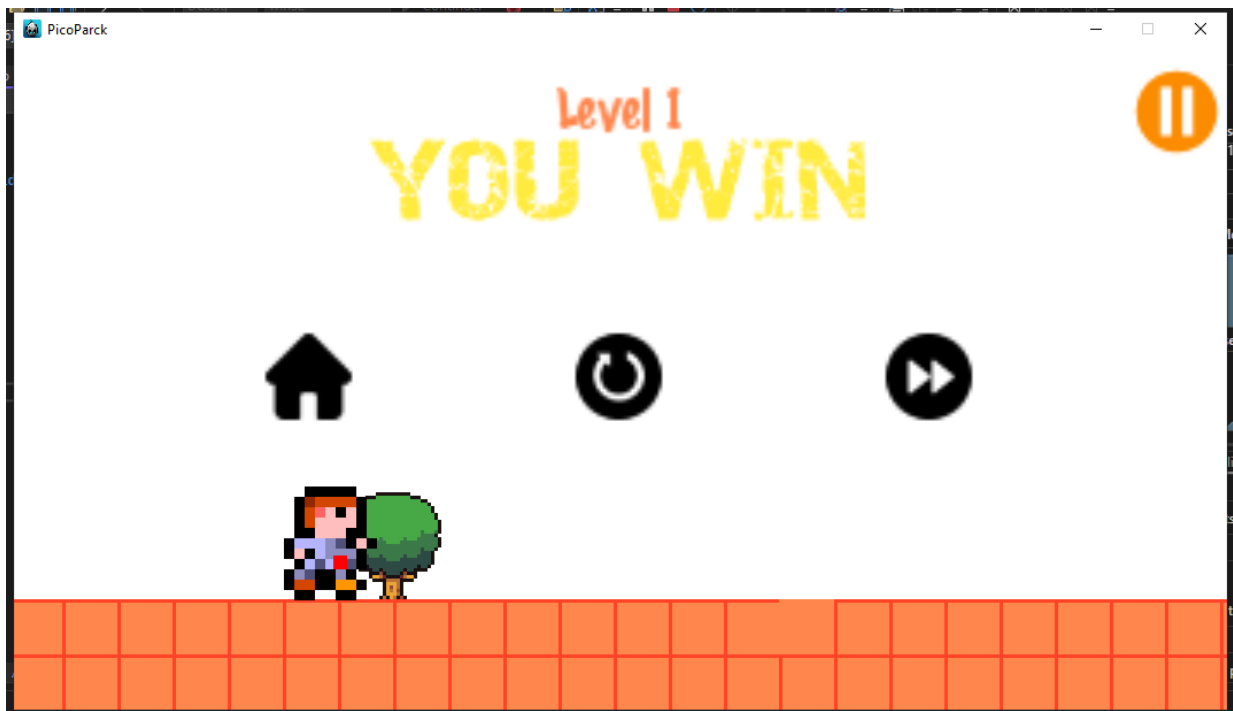
à l'aide de la fonction suivant :

```
void LEVEL1::GoToGameLEVEL1(Ref* sender)  
{  
    auto scene = LEVEL1::createScene();  
    Director::getInstance()-  
>replaceScene(TransitionFade::create(TRANSITION_TIME, scene));}
```

Au cas où l'objet touche l'un de ces obstacles on finit par avoir une arrière-plan sur laquelle il est écrit **GAME OVER** avec deux boutons l'un pour revenir au menu et l'autre pour recommencer le jeu , comme le montre l'image suivante :



Au cas où l'objet termine le **Level 1** , on finit par avoir une arrière-plan sur laquelle il est écrit **YOU WIN** avec trois boutons , le premier pour revenir au menu , la deuxième pour recommencer le jeu , et troisième pour passer à **Level 2** , comme le montre l'image suivante :



On a réalisé l’affichage de `YOU WIN` en utilisant le code suivant :

```
auto win = Sprite::create("win.png");
win->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height *
0.8));
addChild(win);
```

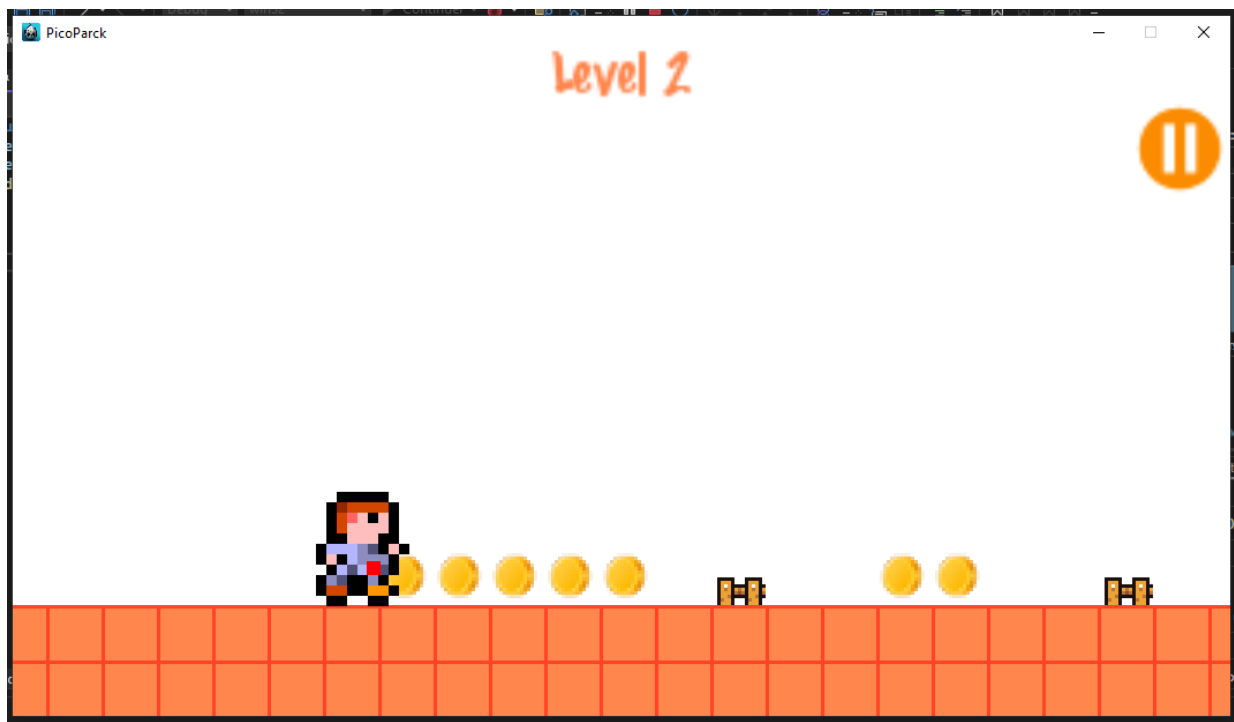
on a réalisé le bouton qui nous permet de passer à **Level 2** en utilisant le code suivant :

```
auto next = MenuItemImage::create("menu/next.png", "menu/nextSelected.png",
CC_CALLBACK_1(LEVEL1::GoToGameWin, this));
```

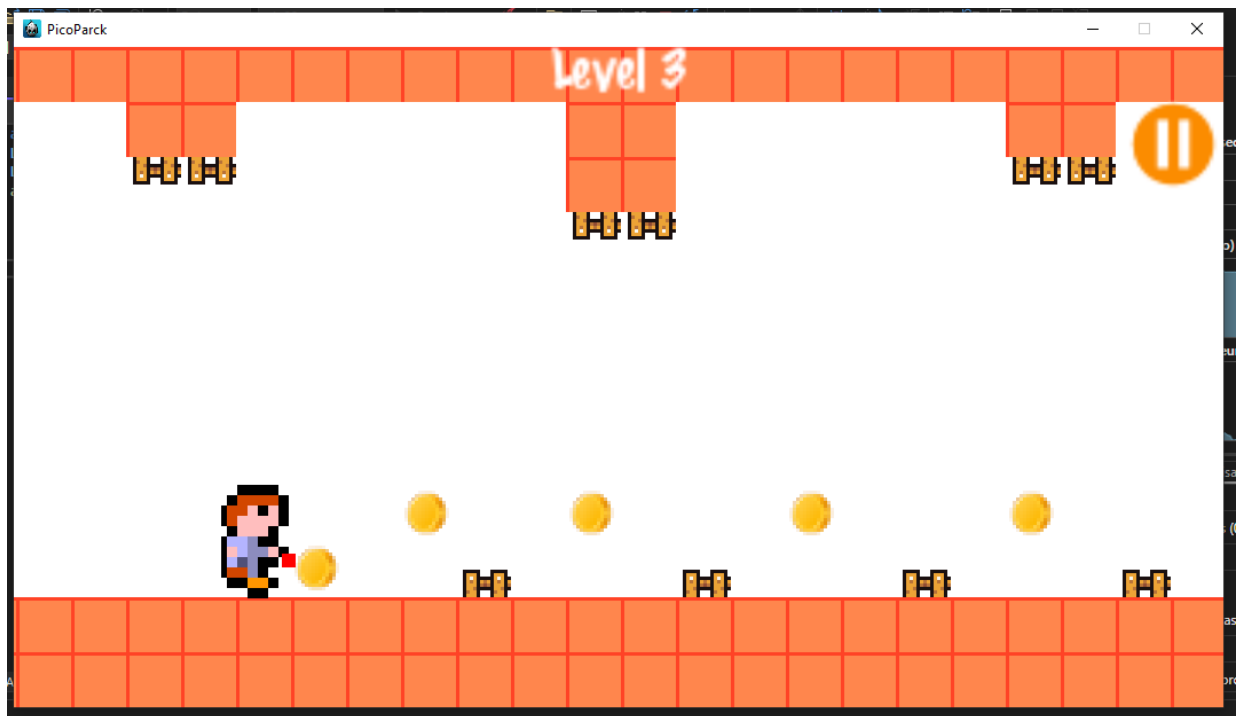
à l’aide de la fonction suivante :

```
void LEVEL1::GoToGameWin(Ref* sender)
{
    auto scene = LEVEL2::createScene();
    Director::getInstance()-
>replaceScene(TransitionFade::create(TRANSITION_TIME_1, scene));
}
```

Maintenant tu peux passer pour jouer le deuxième level qui a réalisé plus difficile que le level précédent , voici l’image :



Et de même méthode on passe au **Level 3** :



Finalement nous nous excusons par ce que on sait que on a oublié plusieurs chose à expliquer ,

On veut indiquer que c'est une bonne idée de faire comme ce projet en **POO** , on a trouvé qu'on bien comprendre et appliquer le cours dans ce jeu et chercher à des informations dans autres sources.

On a utilisé plusieurs plusieurs ressources pour accomplir ce travail comme :

La documentation de cocos2d : [Cocos2d-x: cocos2d-x](https://cocos2d-x.org/docs)

Et plusieurs toturials qui explique comment utiliser les classe , et les fonctions contenu dans le moteur cocos2d .