Université Abdelmalek Essaâdi

Faculté Des Sciences Et Techniques – Tanger

LST GENIE INFORMATIQUE GRP1 2021/2022

PICO PARK GAME

REALISÉ PAR : KIBLOU NOUSSAIBA

ENCADRÉ PAR : Pr. EL AACHAK LOTFI

INTRODUCTION :

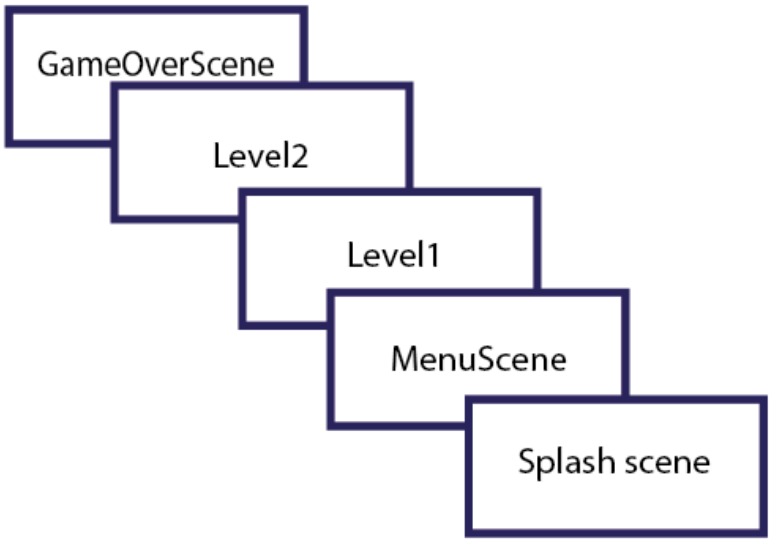
Objectif :

L'objectif principal de ce projet est de maitriser 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.

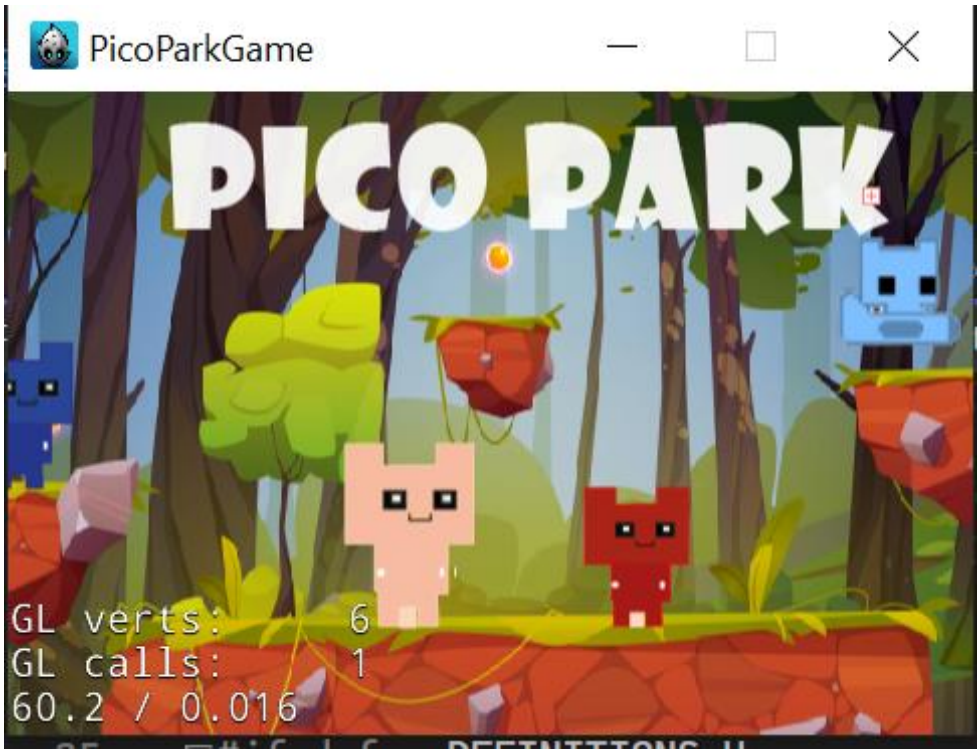
- Développer le jeu 2D via le moteur de jeu Cocos2D « C++ » avec le respect du paradigme POO.
- Il faut au moins développer 3 niveaux de jeu.
- Le jeu doit être en mode mono player.

Outils utilisés : C++ , Cocos2D, Photoshop/ Adobe Illustrator.

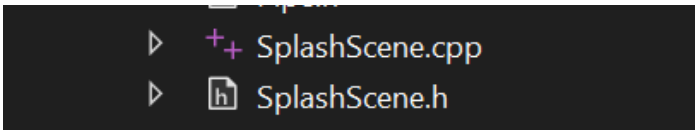
SCENES DE JEUX :



Splash Scene :



- **Creation du classe splashscene.cpp et splashscene.h :**




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

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

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

        // return the scene
           return scene;
}
```

- **Ajout de la photo du background :**

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

this->addChild(backgroundSprite);
```

- **Passage du Splashscene a Menuscene (transition):**

- Creation du fonction GoToMainMenu :

```
private:
    void GoToMainMenuScene(float dt);
};
```

```
this->scheduleOnce(CC_SCHEDULE_SELECTOR(SplashScene::GoToMainMenuScene), DISPLAY_TIME_SPLASH_SCENE);
```

```
void SplashScene::GoToMainMenuScene(float dt)
{
    auto scene = MainMenuScene::createScene();

    Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME, scene));
}
```

Définition du temps d'affichage et de transition dans la classe Definitions.h

```
#ifndef __DEFINITIONS_H__
#define __DEFINITIONS_H__

#define DISPLAY_TIME_SPLASH_SCENE 3
#define TRANSITION_TIME 0.5
```

MainMenuScene:

Creation du background et de bouton play :

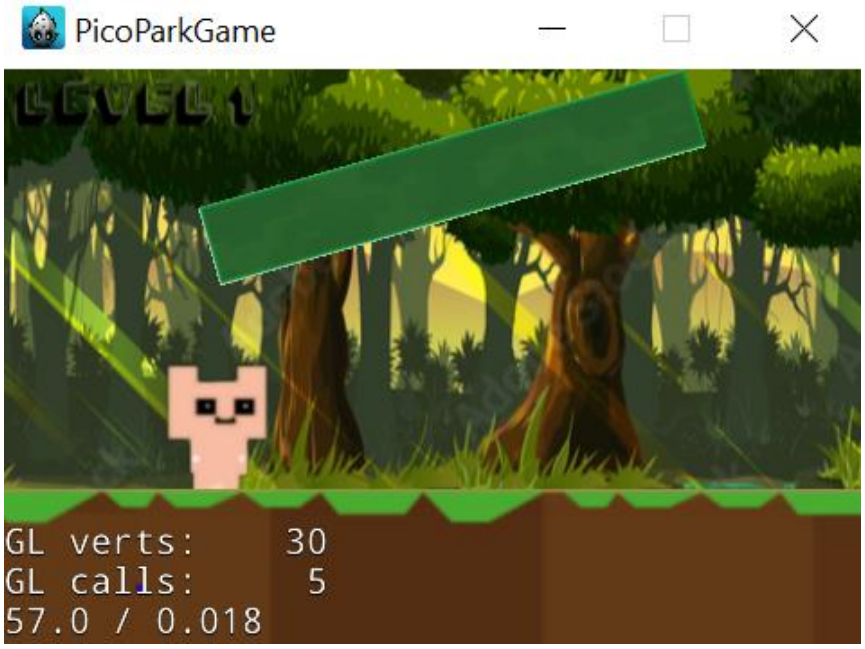
```
auto backgroundSprite = Sprite::create("1.png");
backgroundSprite->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
this->addChild(backgroundSprite);

auto playItem = MenuItemImage::create("Button.png", "button.clicked.png", CC_CALLBACK_1(MainMenuScene::GoToGameScene, this));
playItem->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));

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

this->addChild(menu);
```

GameScene LEVEL1 :



-

- **Création du sprite “PICI” :**

```
mySprite1 = Sprite::create("PICI.png");
mySprite1->setPosition(Point(120, 120));
mySprite1->setScale(0.3);
auto spriteBody = PhysicsBody::createBox(mySprite1->getContentSize(), PhysicsMaterial(0, 1, 0));
mySprite1->setPhysicsBody(spriteBody);
this->addChild(mySprite1, 1);
```

- **Création du plan et du background et la pipe :**

```
mySprite2 = Sprite::create("PLAN.png");
mySprite2->setPosition(Point(0, 0));
this->addChild(mySprite2, 1);

mySprite3 = Sprite::create("LEVEL1.png");
mySprite3->setPosition(Point(75, 300));
this->addChild(mySprite3, 1);

auto backgroundSprite = Sprite::create("bgd.png");
backgroundSprite->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + or
this->addChild(backgroundSprite);

mySprite4 = Sprite::create("PIPE.png");
mySprite4->setPosition(Point(250, 260));
this->addChild(mySprite4);
auto rotateAction = RotateBy::create(9, 360);
auto action = RepeatForever::create(rotateAction);
mySprite4->runAction(action);
```

Les mouvements liés au Keyboard pour PICI :

```
// keyboard
auto eventListener = EventListenerKeyboard::create();
eventListener->onKeyPressed = [=](EventKeyboard::KeyCode keyCode, Event* event)
{
    switch (keyCode)
    {

    case EventKeyboard::KeyCode::KEY_A:
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
        DirX -= 3.0f;
        break;

    case EventKeyboard::KeyCode::KEY_D:
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
        DirX += 3.0f;
        break;
    }
};
```

```
eventListener->onKeyReleased = [=](EventKeyboard::KeyCode keyCode, Event* event)
{
    switch (keyCode)
    {

    case EventKeyboard::KeyCode::KEY_A:
    case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
        DirX += 3.0f;
        break;

    case EventKeyboard::KeyCode::KEY_D:
    case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
        DirX -= 3.0f;
        break;
    }
};
```

```
    this->_eventDispatcher->addEventListenerWithSceneGraphPriority(eventListener, this);

    this->scheduleUpdate();


    return true;
}

void GameScene::update(float dt) {

    float newPosX = mySprite1->getPositionX() + (DirX);
    float newPosY = mySprite1->getPositionY() + (DirY);

    mySprite1->setPosition(newPosX, newPosY);

}
```

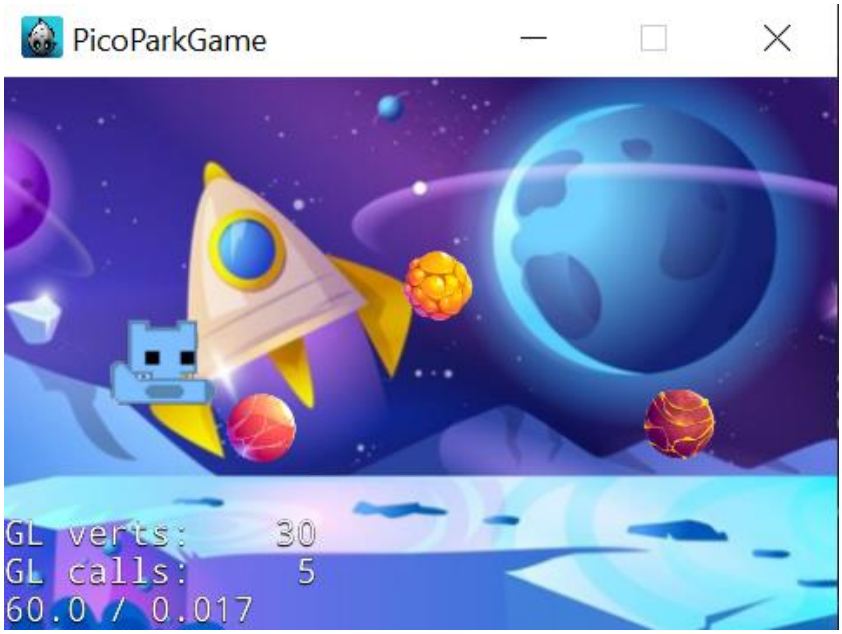
Passage du level 1 au level 2: (Transition)

```
this->scheduleOnce(CC_SCHEDULE_SELECTOR(GameScene::GoToNextGameScene), DISPLAY_TIME_GAME_SCENE);
```

```
void GameScene::GoToNextGameScene(float dt)
{
    auto scene = NextGameScene::createScene();

    Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME, scene));
}
```

Scène LEVEL2 :



Création du Background :

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

Création du sprite Piicii :

```
mySprite5 = Sprite::create("Piicii.png");
mySprite5->setPosition(Point(100, 150));
mySprite5->setScale(0.3);
this->addChild(mySprite5, 1);
```

--L'ajout de mouvements liés au Keyboard comme précédent en ajoutant le touche up et down.

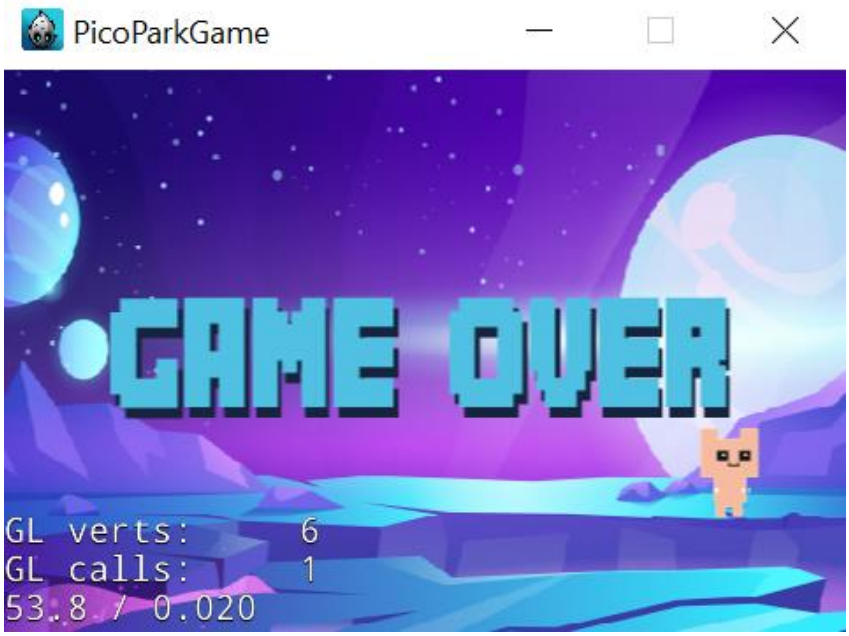
-- Création du sprite des barrières :

```
mySprite6 = Sprite::create("A.png");
mySprite6->setPosition(Point(150, 120));
mySprite6->setScale(0.3);
this->addChild(mySprite6, 1);

mySprite7 = Sprite::create("B.png");
mySprite7->setPosition(Point(250, 200));
mySprite7->setScale(0.3);
this->addChild(mySprite7, 1);

mySprite8= Sprite::create("C.png");
mySprite8->setPosition(Point(390, 120));
mySprite8->setScale(0.3);
this->addChild(mySprite8, 1);
```

GameOverscene :



- Ajout du Background :

```
auto backgroundSprite = Sprite::create("GO.png");
backgroundSprite->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y))

this->addChild(backgroundSprite);
```