UNIVERSITE ABDELMALEK ESSAADI
Faculté des Sciences et Techniques de Tanger
Département Génie Informatique
LST GI S5
Algorithmique et POO C++
Annee 2022-2023

Rapport Projet Cocos2d-x avec le concept POO de C++

COMPTE RENDU DE : ***VEZOLO MBETTE ASSEHT ROSAIRE***
GROUPE : ***02***

Table de contenu :

Introduction :

Avant d'entamer la partie saisie du code, il fallait faire quelques modificationsau niveau du code source donné par défaut lors de la création d'un nouveau projet sur 'Cocos2d-x'.

Ce code par défaut permet au débutant et ou aux personnes qui vient d'installer cocos2d-x de se rassurer s'il est bien installé en testant la compilation.

Processus :

La 1er chose on ouvre le fichier EpicGame.sln contenu dans win.32 puis modifier la résolution et l'adaptation le code en fonction d'objet utiliser comme le téléphone, portable ou la résolution du pc.

Personnellement étant débutant j'ai opté pour la résolution automatique :

```
#include "AppDelegate.h"
#include "HelloWorldScene.h"

// #define USE_AUDIO_ENGINE 1

#if USE_AUDIO_ENGINE
#include "audio/include/AudioEngine.h"
using namespace cocos2d::experimental;
#endif

USING_NS_CC;

static cocos2d::Size designResolutionSize = cocos2d::Size(480, 320);
static cocos2d::Size smallResolutionSize = cocos2d::Size(480, 320);
static cocos2d::Size mediumResolutionSize = cocos2d::Size(1024, 768);
 static cocos2d::Size largeResolutionSize = cocos2d::Size(2048, 1536);
```

Ce fichier est disponible dans AppDelegate.cpp

❖*Référence et outils utiliser*

Les et outils sont :

Visual code Edition 2019 qui offre la possibilité orienter objet, Cocos2d-x disponible sur avec tous les possibles disponibles sur :

http://ps://docs.cocos.com/cocos2d-x/manual/en/

http://ps://github.com/cocos2d/cocos2d-x-samples

http://ps://gamefromscratch.com/cocos2d-x-c-game-programming-tutorial-series/

❖*Création Menu :*

1-Fond Menu

Le Menu est créé dans le fichier nommer HelloWord.cpp et HelloWord.h

Ses deux fichiers permettent d'appeler les classes et l'autre d'appeler les fonctions et faire la liaison entre les autres fichiers

L'appel du fond Menu est fait dans le fichier Helloword.cpp avec la commande :

Appel du fond

```
auto sprite = Sprite::create("myprojet/back/image-6.png");
```

Condition d'appel et limitation d'image dans mon cas j'ai opter pour la petite taille

```
if (sprite == nullptr)
{
    problemLoading("'myprojet/back/image-6.png'");
```

```
    }
    else
    {
        // position the sprite on the center of the screen
        sprite->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height
/ 2 + origin.y));

        // add the sprite as a child to this layer
        this->addChild(sprite, 0);
```

2-Bouton Play

Le bouton Play permet de créer du texte de déclarer, mettre en place un bouton permettant Louverture du Game :

Déclaration du bouton Play, positon souhaiter et caractère :

```
        auto playItem = MenuItemFont::create("play",
CC_CALLBACK_1(HelloWorld::GoToNewScene, this));

        auto* menu = Menu::create(playItem, NULL);

        //postion of menu play and visibility;

        menu->alignItemsVertically();
        this->addChild(menu);
    }
    return true;
}
```

3-Bouton exit

Permet la sortie de dans le jeu depuis le menu avec un Icon indiquant la sortie :

```
    // call image for exit to Menu
    auto closeItem = MenuItemImage::create(
                                           "exit.png",
                                           "exit.png",

CC_CALLBACK_1(HelloWorld::menuCloseCallback, this));
    //size of images and position for my image


    if (closeItem == nullptr ||
        closeItem->getContentSize().width <= 0 ||
        closeItem->getContentSize().height <= 0)
    {
        problemLoading("'exit.png' and 'exit.png'");
    }
    else
    {
        float x = origin.x + visibleSize.width - closeItem->getContentSize().width/2;
        float y = origin.y + closeItem->getContentSize().height/2;
        closeItem->setPosition(Vec2(x,y));
    }
```

Je tiens à rappeler que les appels des fonction vod() utiliser sont présent dans le fichier pour l'appel exit, background

❖Création de la Scène :

1-Fond

J'ai fait la création deux fichiers intituler NewScene.cpp et .h

Dans NewScene.cpp j'ai fait appel a mon back ground et la résolution.

Je tiens à rappeler que Cocos2d-x utilise deux axes (abscisse X et Ordonne Y) cela permet de diriger l'objet sur la position vertical et horizontal en suivant ses deux axes de repère

```
    auto label = Label::createWithTTF("", "fonts/Marker Felt.ttf", 24);
    if (label == nullptr)
    {
        problemLoading("'fonts/Marker Felt.ttf'");
    }
    else
    {
        // position the label on the center of the screen
        label->setPosition(Vec2(origin.x + visibleSize.width/2,
                                origin.y + visibleSize.height - label-
>getContentSize().height));

        // add the label as a child to this layer
        this->addChild(label, 1);
    }

    // add back ground
    auto mysprite = Sprite::create("myprojet/background/fond2.png");

    mysprite->setAnchorPoint(Vec2::ZERO);
    mysprite->setPosition(0, 0);
    this->addChild(mysprite);
```

2-Playeur et Deplanement

J'ai fait appel au la commande auto qui me permet de créer automatique les objets afin d'éviter les oublis de la création de la commande

`mysprite = Sprite::create(mysprite) ;` dans le fichier .h qui accompagne le fichier .cpp

Dans ce code playeur Jai aussi défini la résolution de l'image, sa taille le fond avec l'outils Grimp

```
    // insertion playeur
    auto mysprite2 = Sprite::create("myprojet-2/conejo.png");
    mysprite2->setPosition(Point((visibleSize.width / 2) + origin.x,
(visibleSize.height / 2) + origin.y));
    addChild(mysprite2, 1);
// Mouve to my playeur
    auto action = MoveBy::create(8, Point(100, 0));
    mysprite2->runAction(action);
```

3-Bouton exit

Cette commande est la mise en place d'un bouton de sorti permettant la sortie en plaine scène

```
bool NewScene::init()
{
    ////////////////////////////
    // 1. super init first
    if ( !Scene::init() )
    {
        return false;
    }

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

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

    auto closeItem = MenuItemImage::create(
                                           "CloseSelected.png",
                                           "CloseSelected.png",
                                           CC_CALLBACK_1(NewScene::menuCloseCallback,
this));

    if (closeItem == nullptr ||
        closeItem->getContentSize().width <= 0 ||
        closeItem->getContentSize().height <= 0)
    {
        problemLoading("'CloseSelected.png' and 'CloseSelected.png'");
    }
    else
    {
        float x = origin.x + visibleSize.width - closeItem->getContentSize().width/2;
        float y = origin.y + closeItem->getContentSize().height/2;
        closeItem->setPosition(Vec2(x,y));
    }

    // create menu, it's an autorelease object
    auto menu = Menu::create(closeItem, NULL);
    menu->setPosition(Vec2::ZERO);
    this->addChild(menu, 1);

    /////////////////////////////
    // 3. add your codes below...

    // add a label shows "Hello World"
    // create and initialize a label

    auto label = Label::createWithTTF("", "fonts/Marker Felt.ttf", 24);
    if (label == nullptr)
    {
```

```
        problemLoading("'fonts/Marker Felt.ttf'");
    }
    else
    {
        // position the label on the center of the screen
        label->setPosition(Vec2(origin.x + visibleSize.width/2,
                                origin.y + visibleSize.height - label-
>getContentSize().height));

        // add the label as a child to this layer
        this->addChild(label, 1);
    }
```

5-Lien entre scène et Menu

Pour faire une liaison entre les deux j'ai fait appel à la commande

```
//call Scene
void HelloWorld::GoToNewScene(cocos2d::Ref* pSender)
{
    auto scene = NewScene::createScene();
    Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME,
scene));
}
```

Dans le fichier helloworld.cpp et fait la créer d'un nouveau fichier intituler Definition.h

```
#ifndef __DEFINITIONS_H__
#define __DEFINITIONS_H__

#include "cocos2d.h"

#define DISPLAY_TIME_NEW_SCENE 2
#define TRANSITION_TIME 1
```

```
#endif // __DEFINITIONS_H__
```

Ce fichier assure la transition de menu vers la 1ère scène

Derrière J'ai aussi fait une déclaration dans le fichier Newscene.cpp permettant la liaison des deux fichiers.

❖*Difficultés rencontrer*

1-Organisation du travail :

Etant seul dans le projet sans binôme et ayant fournie un travail de mon dur labeur, organiser le travail a été pour moi une grande tache avec la course entre les documents, teste de code, faire une connexion entre le mode scolaire et professionnel sur la manipulation des objets, tout ça étant nouveau pour moi cela m'a pris beaucoup du temps que prévue

2-Faire le Rapport entre POO et C++

Le rapport entre le POO et C++ a été aussi un grand travail car il y des concepts à maitriser, une bibliothèque à comprendre y compris celui de Cocos tout ce cela se joint dans le retard et la difficulté rencontre.

❖ *Conclusion :*

En dépit de tout ce qui précède ce travail certes étant incomplet (compte faire bouger les playeur avec le clavier, constituer les trois étapes, etc..) m'a permis de comprendre le concept POO et faire la relation entre le C++ et l'orientation des objets.

Ce travail a aussi réveillé en moi l'envie de faire des recherches a approfondit afin de bien comprendre et maitriser la bibliothèque cocos2d-x et l'importance de travailler seul avec des projets aussi approfondis.