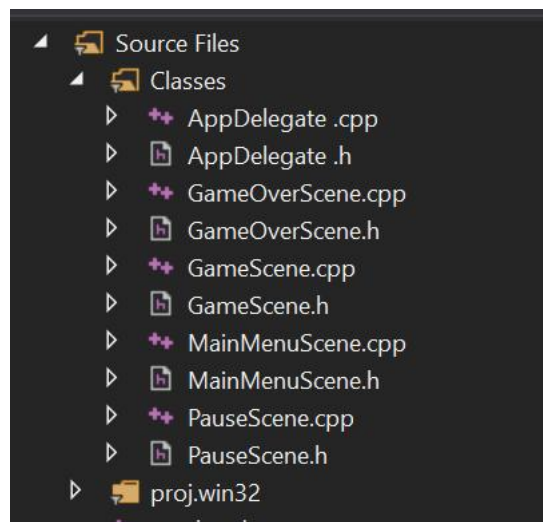


RAPPORT DE JEUX :

- Dans le jeu **PICO PARK** , nous avons utilisé : langage **C++** ; **Cocos2d** fait partie des bibliothèques logicielles de référence , en plus de permettre la création de jeux 2D pour Android, iOS et Windows Phone ;
- et **Photoshop/Canva** pour la préparation des images.
- D'abord, nous avons créé un ensemble de fichiers dans Classes :



- On déclare la fonction dans **MainMenuScene.h** pour qui sera appelée lorsque le joueur cliquer sur le bouton de lecture de la scène du menu principal pour la remplacer par la scène du jeu.

```
CREATE_FUNC(MainMenu);  
  
void GoToGameScene(Ref* pSender);  
};
```

- Nous ajoutons le code suivant au fichier **MainMenuScene.cpp**, la première ligne est utilisée pour inclure la scène du jeu afin que la scène soit accessible. La fonction **GoToGameScene** crée une instance **GameScene** locale.

```

void MainMenu::GoToGameScene(Ref* pSender)
{
    auto scene = GameScreen::createScene();

    Director::getInstance()->replaceScene(scene);
}

```

- Dans **GameScene.h**, la première déclaration de fonction sera appelée lorsque le joueur clique sur le bouton pause dans la scène Jeu pour pousser la scène Pause sur la pile.
- La deuxième déclaration de fonction sera appelée lorsque le joueur mourra pour remplacer la **Game Scene** avec la **Game Over Scene**.

```

CREATE_FUNC(GameScreen);

void GoToPauseScene(Ref* pSender);
void GoToGameOverScene(Ref* pSender);

};

```

- Nous ajoutons le code suivant au fichier **GameScene.cpp**, Les deux premières lignes de l'extrait de code précédent sont utilisées pour inclure la **Pause** et **Game Over scenes** afin que les scènes soient accessibles.
- La fonction **GoToPauseScene** crée d'abord une instance de scène locale de la **Pause scene**, puis la pousse sur la empiler.
- La fonction **GoToGameOverScene** crée d'abord une instance de scène locale du **Game Over scene**, puis remplace la **Game scene** par celle-ci.

```

6
7
8 void GameScreen::GoToPauseScene(cocos2d::Ref* pSender)
9 {
10     auto scene = PauseMenu::createScene();
11     Director::getInstance()->pushScene(scene);
12 }
13
14
15 void GameScreen::GoToGameOverScene(cocos2d::Ref* pSender)
16 {
17     auto scene = GameOver::createScene();
18     Director::getInstance()->replaceScene(scene);
19 }
20
21
22

```

- On déclare la fonction dans **GameOverScene.h**, Dans l'extrait de code précédent, la première déclaration de fonction sera appelée lorsque le joueur clique sur le bouton réessayer de la scène **Game Over** pour remplacer l'actuel scène avec la scène du jeu. La deuxième déclaration de fonction sera appelée lorsque le joueur clique sur le bouton du menu principal de la scène **Game Over** pour remplacer le scène actuelle avec la scène du menu principal.

```
CREATE_FUNC(GameOver);

void GoToGameScene(Ref* pSender);
void GoToMainMenuScene(Ref* pSender);
```

- Nous ajoutons le code suivant au fichier **GameOverScene.cpp**, Les deux premières lignes de l'extrait de code précédent sont utilisées pour inclure les **Game et Main Menu scenes** afin que les scènes soient accessibles.
- La fonction **GoToGameScene** crée d'abord une instance de scène locale de la **Game scene**, puis remplace la **Game Over Scene** avec elle.
- La fonction **GoToMainMenuScene** crée d'abord une scène locale instance de la **Main Menu Scene**, puis remplace la **Game Over Scene** par celle-ci.

```
void GameOver::GoToGameScene(cocos2d::Ref* pSender)
{
    auto scene = GameScreen::createScene();

    Director::getInstance()->replaceScene(scene);
}

void GameOver::GoToMainMenuScene(cocos2d::Ref* pSender)
{
    auto scene = MainMenu::createScene();

    Director::getInstance()->replaceScene(scene);
}
```

- Dans **PauseScene.h** la première déclaration de fonction sera appelée lorsque le joueur clique sur le bouton de reprise de la **Pause Scene** pour faire apparaître la **Pause scene** hors de la pile.
- La deuxième déclaration de fonction sera appelée lorsque le joueur clique sur le bouton du menu principal de la **Pause scene** pour remplacer la scène actuelle avec la **Main Menu Scene** tout en faisant sortir toutes les scènes de la pile.
- Le troisième déclaration de fonction sera appelée lorsque le joueur cliquera sur le bouton réessayer depuis le Mettre la **Pause scene** pour remplacer la scène actuelle par la **Game Scene** tout en faisant apparaître tous les scènes hors de la pile.

```
CREATE_FUNC(PauseMenu);

void Resume(Ref* pSender);
void GoToMainMenuScene(Ref* pSender);
void Retry(Ref* pSender);
```

- Nous ajoutons le code suivant au fichier **PauseScene.cpp**, Les deux premières lignes de l'extrait de code précédent sont utilisées pour inclure le **Game et Main Menu Scenes** afin que les scènes soient accessibles.
- La **GoToMainMenuScene** La fonction crée d'abord une instance de scène locale de la

Main Menu Scene, puis elle apparaît la scène actuelle hors de la pile et la **Game Scene** est remplacée par le **Main Menu Scene**.

- La fonction **Retry** crée d'abord une instance de scène locale de la **Game Scene**, puis il fait sortir la scène actuelle de la pile et la **Game Scene** actuelle est remplacée par la nouvelle **Game Scene** (redémarre le jeu).

```
7
8 void PauseMenu::Resume(cocos2d::Ref* pSender)
9 {
10     Director::getInstance()->popScene();
11 }
12
13 void PauseMenu::GoToMainMenuScene(cocos2d::Ref* pSender)
14 {
15     auto scene = MainMenu::createScene();
16
17     Director::getInstance()->popScene();
18     Director::getInstance()->replaceScene(scene);
19 }
20
21 void PauseMenu::Retry(cocos2d::Ref* pSender)
22 {
23     auto scene = GameScreen::createScene();
24
25     Director::getInstance()->popScene();
26     Director::getInstance()->replaceScene(scene);
27 }
```

- Dans cette étape de la **MainMenuScene.cpp**, les images des éléments de menu seront utilisées pour afficher le titre et l'affichage du jeu Bouton avec lequel l'utilisateur peut interagir pour manœuvrer entre les scènes.

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

auto menuTitle = MenuItemImage::create("Game_Title.png", "Game_Title.png");

auto playItem = MenuItemImage::create("Play_Button.png", "Play_Button(Click).png",
                                     CC_CALLBACK_1(MainMenu::GoToGameScene, this));

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

menu->alignItemsVerticallyWithPadding(visibleSize.height / 4);

this->addChild(menu);

return true;
```

- La **Game Scene** nécessite un système de menu simple pour mettre le **Pause Scene**. Un élément de menu l'image sera utilisée pour afficher le bouton avec lequel

l'utilisateur peut interagir afin de mettre le jeu en pause et charger la **Pause Scene**.

```
auto pauseItem = MenuItemImage::create("Pause_Button.png", "Pause_Button(Click).png", CC_CALLBACK_1(GameScreen::GoToPauseScene, this));
pauseItem->setPosition(Point(pauseItem->getContentSize().width - (pauseItem->getContentSize().width / 2) + origin.x,
    visibleSize.height - pauseItem->getContentSize().height +
    (pauseItem->getContentSize().width / 2) + origin.y));
```

- La **Pause Scene** nécessite un système de menus pour reprendre le jeu, redémarrer le jeu et revenir au menu principal. Les images des éléments de menu seront utilisées pour afficher les boutons que l'utilisateur peut interagir avec afin d'effectuer les actions mentionnées.

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

auto resumeItem = MenuItemImage::create("Resume_Button.png", "Resume_Button(Click).png",
    CC_CALLBACK_1(PauseMenu::Resume, this));

auto retryItem = MenuItemImage::create("Retry_Button.png", "Retry_Button(Click).png",
    CC_CALLBACK_1(PauseMenu::Retry, this));

auto mainMenuItem = MenuItemImage::create("Menu_Button.png", "Menu_Button(Click).png",
    CC_CALLBACK_1(PauseMenu::GoToMainMenuScene, this));

auto menu = Menu::create(resumeItem, retryItem, mainMenuItem, NULL);

menu->alignItemsVerticallyWithPadding(visibleSize.height / 4);

this->addChild(menu);

return true;
```

Remarque: Pour mettre le jeu en pause, il faut cliquer sur la gauche en haut.

- Dans la **Game Scene**, nous ajoutons l'arrière-plan du jeu qui est mis en place dans **canva** :

```
(pauseItem->getContentSize().width / 2) + origin.y));

auto menuTitle = MenuItemImage::create("Play_Level.png", "Play_Level.png");
float rx = visibleSize.width / menuTitle->getContentSize().width;
float ry = visibleSize.height / menuTitle->getContentSize().height;

menuTitle->setScaleX(rx);
menuTitle->setScaleY(ry);

menuTitle->setPosition(Point((visibleSize.width / 2) + origin.x, (visibleSize.height / 2) + origin.y));

auto menu = Menu::create(pauseItem, menuTitle, NULL);

//menu->alignItemsVerticallyWithPadding(visibleSize.height/4);
menu->setPosition(Point::ZERO);

this->addChild(menu);
```

- Dans la **Game Scene**, nous ajoutons le pico:

```
mySprite = Sprite::create("rose.png");

mySprite->setPosition(Point((visibleSize.width / 2) + origin.x, (visibleSize.height / 2) + origin.y));

this->addChild(mySprite);

//auto action = MenuButton::create(2, Point(100, 50));
```

- Nous ajoutons de la musique de jeu dans le **Main Menu Scene**:

```
CocosDenshion::SimpleAudioEngine::getInstance()->setBackgroundMusicVolume(0.25);

CocosDenshion::SimpleAudioEngine::getInstance()->preloadBackgroundMusic("picopark.mp3");

CocosDenshion::SimpleAudioEngine::getInstance()->playBackgroundMusic("picopark.mp3");
```

Bibliothèque:

Cocos2d-x Game Development Essentials (PDFDrive)

<https://www.youtube.com/@SonarSystemsCoUk>

Realisé par:

Bendahrass mohammed (grp2)

Bouktaib Khadija (grp2)