



PROJET REALISE PAR :

OUAAQA YOUSSEF | BOUFARHI AYMAN

ENCADRE PAR :

LOTFI ELAACHAK | IKRAM BEN ABDELOUAHAB

OUTILS :



LIEN GITHUB :

Sommaire

1 – le processus de développement

2 – les options développés

3 – les difficultés rencontrées

1 – le processus de développement

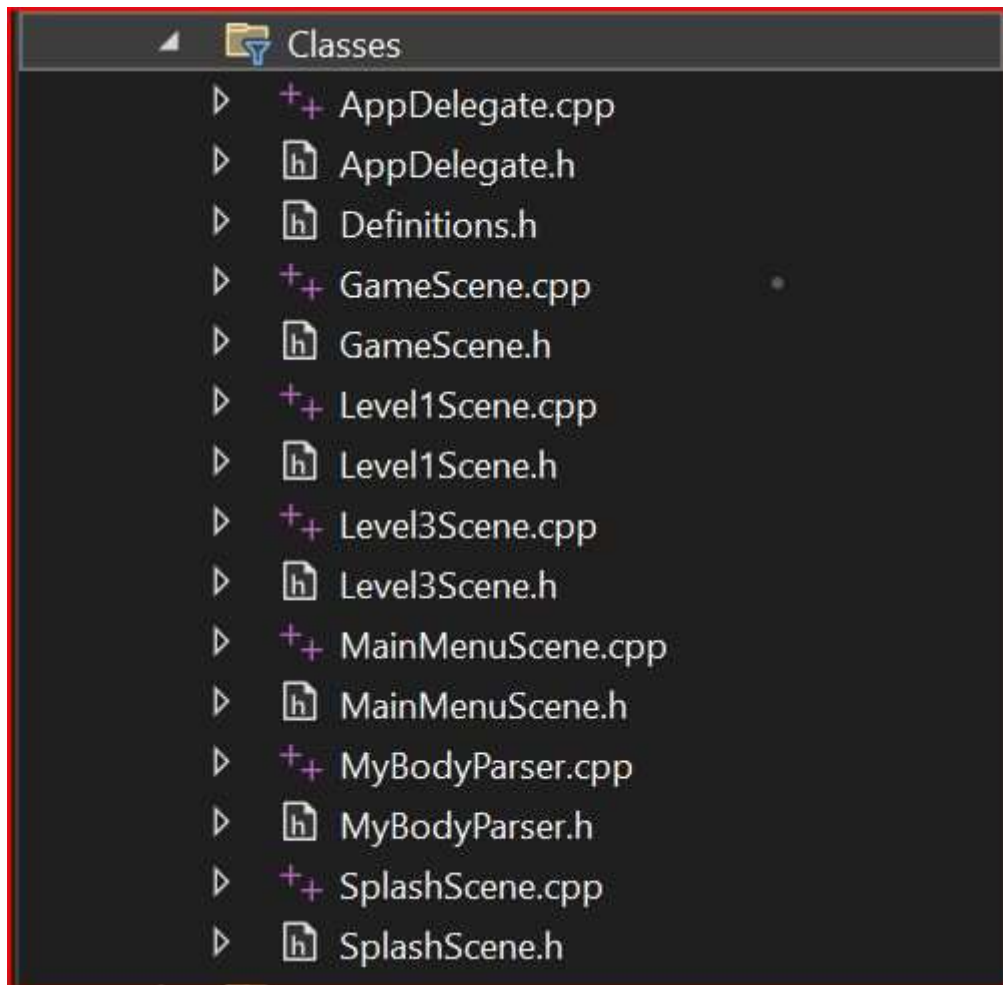
Après avoir pris une base en langage de programmation C++ au sein de la Faculté, nous avons d'abord commencé à explorer Cocos2d-x pour avoir une idée générale, puis nous nous sommes réunis pour passer aux choses sérieuses.

Nous avons commencé par le menu, nous avons fait deux menus pour à la fin nous choisirons le meilleur, puis nous avons ajouté le personnage principales et nous avons créé des cartes démo, afin que nous puissions tous les deux essayer chez nous de manipuler les cartes, ajouter d'autres choses, déplacer le personnage.

Enfin, nous nous sommes réunis pour partager et combiner nos travaux pour finalement proposer ces trois niveaux.

2 – les options développés

Les classes :



Menu :

un bouton cliquable qui se change lorsqu'on clique pour passer au premier niveau

```
auto playbtn = MenuItemImage::create ("playbtn.png", "playbtnclicked.png", CC_CALLBACK_1(MainMenuScene::GoToLevel1Scene, this));
playbtn->setScale(0.5);
playbtn->setPosition( Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y) );

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

return true;
}

void MainMenuScene::GoToLevel1Scene(cocos2d::Ref* sender)
{
    auto scene = Level1Scene::createScene();
    Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME, scene));
}
```

Les niveaux :

Ajout du personnage :

```
auto pico = Sprite::create("pico.png");
pico->setPosition(Vec2(visibleSize.width * 0.1, visibleSize.height * 0.5));
pico->setScale(0.5);
pico->setName("pico.png");
MyBodyParser::getInstance()->parseJsonFile("pico.json");
auto spriteBody= MyBodyParser::getInstance()->bodyFormJson(pico, "pico.png", PhysicsMaterial(0.2, 0.2, 0.2));
spriteBody->setGravityEnable(true);
spriteBody->setDynamic(true);
spriteBody->setContactTestBitmask(1);
spriteBody->setCollisionBitmask(1);
spriteBody->setCategoryBitmask(1);
spriteBody->setRotationEnable(0.0f);
pico->setPhysicsBody(spriteBody);

this->addChild(pico, 1);
```

Ajout du map :

La terre :

```
auto ard = Sprite::create("120.png");
ard->setPosition(250,-30);
ard->setScale(0.9);
MyBodyParser::getInstance()->parseJsonFile("nn.json");
auto ardBody = MyBodyParser::getInstance()->bodyFormJson(ard, "120.png", PhysicsMaterial(0.2, 0.2, 0.2));
if (ardBody != nullptr) {
    ardBody->setDynamic(false);
    ard->setPhysicsBody(ardBody);
}
this->addChild(ard, 1);
```

La porte :

```
auto bab = Sprite::create("33.png");
bab->setScale(1.2);
bab->setPosition(350, 144);
MyBodyParser::getInstance()->parseJsonFile("kk.json");
auto babBody = MyBodyParser::getInstance()->bodyFormJson(bab, "33.png", PhysicsMaterial(0.2, 0.2, 0.2));
if (babBody != nullptr) {
    babBody->setDynamic(false);
    bab->setPhysicsBody(babBody);
}
this->addChild(bab, 1);
```

Background :

```
auto instructionbackground = Sprite::create("instruction.jpg");
instructionbackground->setScale(1.0);
instructionbackground->setPosition(Point(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
this->addChild(instructionbackground);
```

Déplacer le personnage :

```
auto eventListener = EventListenerKeyboard::create();
eventListnr->onKeyPressed = [](EventKeyboard::KeyCode keyCode, Event* event) {
    Vec2 loc = event->getCurrentTarget()->getPosition();
    switch (keyCode)
    {
        case EventKeyboard::KeyCode::KEY_LEFT_ARROW:
        case EventKeyboard::KeyCode::KEY_A:
            event->getCurrentTarget()->runAction(MoveBy::create(0.09f, Vec2(-20, 0)));

            break;
        case EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
        case EventKeyboard::KeyCode::KEY_D:
            event->getCurrentTarget()->runAction(MoveBy::create(0.09f, Vec2(20, 0)));
            break;

        case EventKeyboard::KeyCode::KEY_SPACE:
        case EventKeyboard::KeyCode::KEY_W:

            event->getCurrentTarget()->runAction(JumpBy::create(0.5, Vec2(30, 0), 30, 1));
            break;
        case EventKeyboard::KeyCode::KEY_DOWN_ARROW:
        case EventKeyboard::KeyCode::KEY_S:
            event->getCurrentTarget()->runAction(JumpBy::create(0.5, Vec2(-30, 0), 30, 1));
            break;
    }
}
```

Passer au niveau suivant :

```
auto doorbtn = MenuItemImage::create("33.png", "b.png", CC_CALLBACK_1(Level1Scene::GoToGameScene, this));
doorbtn->setScale(0.5);
doorbtn->setPosition(450, 144);

auto door = Menu::create(doorbtn, NULL);
door->setPosition(Point::ZERO);
this->addChild(door);

return true;
}

void Level1Scene::GoToGameScene(cocos2d::Ref* sender)
{
    auto scene = GameScene::createScene();
    Director::getInstance()->replaceScene(TransitionFade::create(TRANSITION_TIME, scene));
}
```

3 – les difficultés rencontrées

Le premier défi auquel nous avons été confrontés est d'ajouter la carte, dans de nombreux sites Web, nous avons lu que nous avions besoin d'une carte.tmx conçue en TILED, ce que nous avons fait, mais nous n'avons pas réussi à l'ajouter au jeu.

Le deuxième défi consiste à faire avancer automatiquement le personnage au niveau suivant lorsqu'il atteint un point spécifique sans appuyer sur aucun bouton.

Le troisième défi est après que le personnage saute, pendant quelques secondes il perce la terre et il revient en haut.

Le dernier défi est une scène de jeu, après que le personnage soit tombé dans un trou.