Projet Du Module CPP

Game: „PicoPark“ :

Realise Par:

Sara Isaoui

Table De Matiere:

Cocos2dx:

Cocos2d-x est un moteur de développement de jeux open source écrit en C++. Il est utilisé pour créer des jeux et d'autres contenus interactifs pouvant être exécutés sur diverses plates-formes, notamment iOS, Android, Windows, Mac et Linux. Cocos2d-x fournit un ensemble d'API pour créer des graphiques 2D, des animations et des simulations physiques, ainsi que pour gérer les entrées de l'utilisateur et d'autres fonctionnalités liées au jeu. Il est populaire parmi les développeurs de jeux en raison de sa compatibilité multiplateforme, de ses performances et de sa facilité d'utilisation.

Tiled:

Tiled est un éditeur de cartes de jeu open source utilisé pour créer des niveaux pour les jeux vidéo. Il permet de créer des cartes en utilisant des tuiles (ou des blocs) prédéfinis et de les assembler pour créer un niveau. Tiled

prend en charge plusieurs formats de fichier de carte, y compris TMX, qui est utilisé avec le moteur de jeu cocos2dx.

1. *Main Menu:*

Ce code est pour une scène dans un jeu Cocos2d-x. La scène représente un menu principal du jeu et comprend les éléments suivants :

a.Une image de fond
b.Une image de logo
c.Un bouton de démarrage et un bouton de sortie, qui sont tous deux des éléments de menu.

Le bouton de démarrage fera passer le jeu à la scène "LevelsMenu" lorsqu'il sera cliqué, tandis que le bouton de sortie fermera le jeu lorsqu'il sera cliqué. Le code comprend également des méthodes de transition entre les scènes et de fermeture du jeu.

Le code comprend plusieurs instructions d'inclusion en haut, qui apportent les déclarations des différentes classes et fonctions utilisées dans le code. La fonction createScene crée une instance de la scène MainMenu et la renvoie. La fonction init initialise la scène et ajoute l'arrière-plan, le logo et le menu à la scène. La fonction GotToLevelsMenu fait passer le jeu à la scène

"LevelsMenu" lorsqu'elle est appelée, et la fonction menuCloseCallback termine le jeu lorsqu'elle est appelée.

2. *Levels Menu:*

scène représente un menu qui permet au joueur de choisir quel niveau jouer. Il comprend les éléments suivants :

a.Une image de fond
b.Une image de logo
c.Trois boutons de niveau (niveau 1, niveau 2, niveau 3) et un bouton de retour.

Les boutons de niveau feront passer le jeu aux scènes de niveau respectives (Niveau1, Niveau2, Niveau3) lorsqu'ils sont cliqués, tandis que le bouton de retour fera revenir le jeu à la scène du menu principal lorsqu'il sera cliqué. Le code comprend également des méthodes de transition entre les scènes. Le code comprend plusieurs instructions d'inclusion en haut, qui apportent les déclarations des différentes classes et fonctions utilisées dans le code. La fonction createScene crée une instance de la scène LevelsMenu et la renvoie. La fonction init initialise la scène et ajoute l'arrière-plan, le logo et les boutons à la scène. Les fonctions GoBackToMainMenu, GoToLevel1, GoToLevel2 et GoToLevel3 font passer le jeu à la scène du menu principal ou aux scènes de niveau lorsqu'elles sont appelées.

3. *LevelObject:*

Ce code définit une classe appelée "LevelObject" dans le moteur de jeu Cocos2d-x. La classe a plusieurs méthodes et variables membres liées à un corps physique et un sprite Box2D. La classe est destinée à être une classe de base pour d'autres classes qui représentent des objets dans un niveau de jeu.

Certaines des méthodes notables de cette classe incluent :

a. „***addBodyToWorld***“ : cette méthode crée un corps Box2D et l'ajoute au monde spécifié. Le corps est positionné au même emplacement que le nœud LevelObject.
b. „***addCircularFixtureToBody***“ : cette méthode ajoute une fixation circulaire au corps, avec le rayon spécifié.
c. „***addRectangularFixtureToBody***“ : cette méthode ajoute une fixation rectangulaire au corps, avec la largeur et la hauteur spécifiées.
d. „***createFixture***“ : cette méthode ajoute un appareil au corps en utilisant la forme spécifiée. L'appareil possède certaines propriétés par défaut telles que la densité, la friction et la restitution.

La classe LevelObject a également une variable membre appelée body qui est un pointeur vers un objet corps Box2D. Il existe également une variable membre appelée sprite qui est un pointeur vers un objet Cocos2d-x Sprite.

4. *Player:*

Ce code définit une classe "Player" qui hérite de la classe "LevelObject" dans cocos2d-x. La classe "Player" surcharge la méthode "addBodyToWorld" de la classe "LevelObject" pour définir le corps de l'objet comme étant un "bullet". Elle définit également une méthode "addFixturesToBody" qui ajoute une forme circulaire à l'objet "Player".

5. *TiledBodyCreator:*

Ce code fait partie d'un Tiled Map Loader pour le moteur de jeu Cocos2d-x. Il fournit des fonctions pour analyser un fichier de carte Tiled Map Editor au format .tmx et créer des corps physiques dans un monde Box2D basé sur les données de la carte. L'éditeur de cartes tuilées est un outil permettant de créer des cartes de tuiles 2D et il stocke les données cartographiques au format de fichier .tmx.

Le code contient plusieurs fonctions pour créer différents types de corps physiques (appareils) dans un monde Box2D, sur la base des données d'un fichier de carte en mosaïque. Ces types de luminaires incluent :

a.Polygones
b.Polylignes
c.rectangles
d.Cercles

Chaque fonction prend un objet Tiled map en entrée et renvoie un objet FixtureDef, qui contient un appareil Box2D et ses

données associées. L'objet FixtureDef peut ensuite être utilisé pour créer un corps physique dans un monde Box2D.

La classe TiledBodyCreator a également une fonction initCollisionMap, qui prend une carte en mosaïque et un monde Box2D comme entrées. Il analyse le groupe d'objets "Collision" de la carte en mosaïque et crée un corps physique dans le monde Box2D pour chaque objet du groupe. Le corps physique est créé en appelant l'une des fonctions createFixture, selon le type d'objet dans la carte tuilée.

6. *Level1:*

Il inclut les en-têtes nécessaires, notamment Level1.h, Definition.h, MainMenu.h, CCEventKeyboard.h, map, GameOverScene.h et iostream.

Il utilise le namespace NS_CC et déclare le namespace std.

Il définit une macro __FLT_EPSILON__ et déclare des variables globales _filteredUpdateDelta, sale, isJumping, mapsize et _player.

Il définit la fonction Level1::createScene(), qui retourne un pointeur vers une instance de Level1.

Il définit la fonction problemLoading(const char* filename), qui affiche un message d'erreur lors du chargement d'un fichier.

Il définit la fonction membre Level1::init(), qui initialise la scène et crée un monde de physique.

Il crée une carte à l'aide de la fonction TMXTiledMap::create() et enregistre les couches de la carte dans des variables.

Il crée un personnage à l'aide de la fonction Sprite::create() et enregistre le personnage dans la variable _player.

Il active la mise à jour de la scène en appelant la fonction scheduleUpdate().

Il crée des objets de physique en utilisant la fonction PhysicsBody::createBox() et les ajoute à la scène.

Il enregistre un écouteur d'événements de clavier en utilisant la fonction addEventListenerKeyboard().

Il enregistre un écouteur d'événements de contact en utilisant la fonction getEventDispatcher()->addEventListenerWithSceneGraphPriority().

Il ajoute un menu de pause à la scène en utilisant la fonction addChild().

7. *GameOverScene:*

Ce code est une scène cocos2d-x qui s'affiche lorsque le jeu est terminé. Il a une image de fond et un bouton "redémarrer". Lorsque le bouton est cliqué, il remplacera la scène actuelle (la scène du jeu) par la scène du menu principal, qui est spécifiée dans la méthode MainMenu::createScene(). La transition entre les deux scènes est une transition en fondu qui prend 0,5 seconde.