Rapport du jeu – Adventure of Lolo

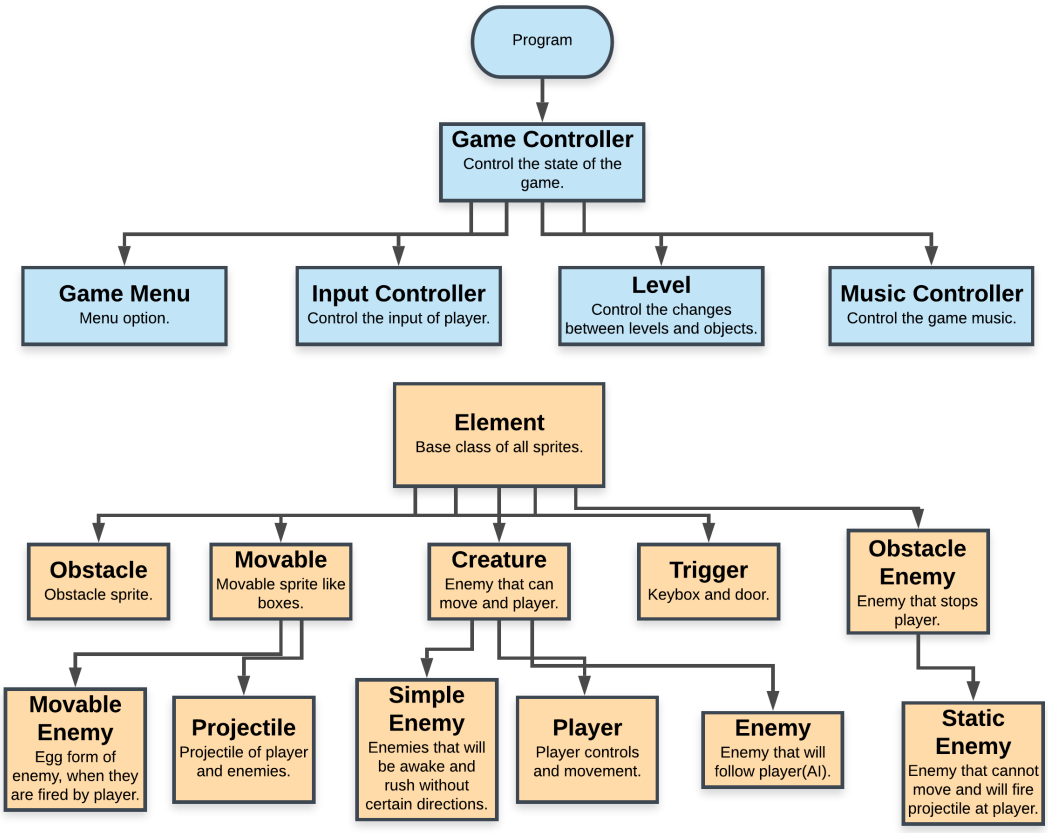
MTI3D M1 Zemin XU

1. Introduction

Dans ce projet j'utilise C++ et la librairie SFML pour reproduire le jeu « Adventure of Lolo ». C'est un jeu de 2D d'aventure et jeu de réflexion. Je suis bien fasciné par le fait que nous devons bien décider où nous allons avant de bouger notre personnage, qui a besoin de la réflexion et la réaction de joueur. Dans ce jeu, j'utilise les textures de mon jeu de type « BomberMan » préféré, « QQ Tang ». L'univers est sombre et les monstres sont mignons. Le joueur incarne un ange qui essaie de passer l'enfer.

2. Système de jeu

La structure de jeu ressemble celui de Unity. Il y a un « Game Controller » pour lancer, surveiller la programme. Je sépare le jeu en deux états différents, le menu et les niveaux. Le « Music Controller » et le « Input Controller » est ici pour aider manipuler les saisies de joueur et le son. À part de cela, un système de « Element » avec ses sous-classes sont construits. Vous pouvez regarder l'illustration dessous pour les détails.

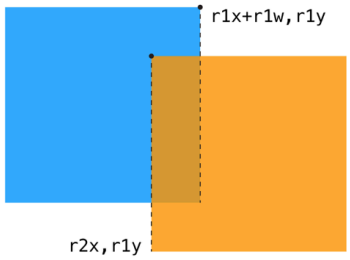


Dans le script de « Level » , il y a plusieurs listes comme la liste de « enemy », la liste de « obstacle », etc. Dans le jeu il existe 5 niveau, chaque fois avant l'initiation de niveau, les listes sont vidés. Pour créer la carte, j'utilise un chiffre pour représenter un élément sur un tableaux de deux dimensions.

3. Implémentations compliquées

- Collision

La difficulté de ce jeu est la collision, surtout la collision entre les boites qui peux bouger avec les autres boites ou autres éléments. Dans la classe « Element », une fonction « DetectCollision » va examiner si la forme de deux rectangles touche. Si c'est vrai, une autre détection va examiner les fantômes, de ces deux formes et aura les effets correspondants. Pour préciser, le fantôme est le rectangle invisible un peu plus petit que la forme. Un tutoriel l'explique bien.



Chaque élément va avoir un index, qui est nommé « kind » pour lui identifier. Tous ces détections auraient lieu dans la classe « Level » lorsqu'un élément bouge, comme joueur, projectile ou enemy.

- Map

L'initialisation de la carte va créer tous les éléments en précisant leur texture et leur position et dimension. La valeur « playerLife » resta inchangé lors du changement de niveau.

- Render & Update

Les fonctions « Render » sont différentes que celui de « Update », parce que « Render » va exécuter une fois chaque frame, mais « Update » aura beaucoup plus que « Render ». Dans la classe « GameController » il y aura une fonction « UpdateTime » qui décide le longueur de chaque frame.

- Enemy

Plusieurs genres d'ennemie sont présentes dans ce jeu. Le « Obstacle Enemy » représente un bouchon dans la route. « Static Enemy » est celui qui lance le projectile vers Lolo. « Simple Enemy » bouge aléatoirement et « Enemy » suivra le personnage. « Movable Enemy » est l'œuf, qui est un état lorsque le joueur lance le projectile sur ennemie.

Les comportements des ennemies au lancement de niveau sont différents que le moment où il reste un cœur(champignon) à ramasser. Ce gameplay est identique que l'origine. Prenant exemple de « Simple Enemy », ce dernier va être activé à la fin de niveau. Donc il y aura deux états différents pour lui.

L'idée de la variable « kind » est utile lorsqu'un œuf va revenir en état d'ennemie. Il suffit de stocker cette variable et calculer le temps passé, une fois le temps écoulé, l'œuf va recréer cette ennemie et détruire lui-même.

4. Améliorations possibles

- Ennemie

Il existe des bugs dans ce jeu. Par exemple, l'ennemie dans le niveau 4 n'est pas assez intelligent pour suivre le personnage. Dans la version future, je vais l'améliorer avec algorithme A*.

- Design pattern

L'organisation du code n'est pas encore propre. Je voulais pratiquer la théorie de design pattern pour l'organisation, pour que la modification soit facile à faire et la fonction peut-être facilement à ajouter.

- Gameplay

Pour l'instant, le « level design » est presque identique que le jeu d'origine. Je voulais que la version future pourrait avoir un gameplay de plus, original. Ce qui est pas mal est de l'avoir sur plate-forme de site web et la mode de multi-joueur.