Sullivan Honnet

Jules Vittone

26/01/2019

**Rapport succinct du projet C++**

Pour ce projet, nous avons fait plusieurs choix dès le début du projet. Notre objectif était qu’il soit très facile de rajouter de nombreux modules car nous voulions rajouter de nombreuses choses. Le premier objectif était que des unités puissent être ajoutées sans problème. Pour ce faire, nous avons créé des super-classes : Unité, Bâtiment et Soldat.

Ces super-classes sont utilisées par le reste du programme qui n’a pas à savoir quelles sont les différentes unités qui existent. Actuellement, je voudrais rapatrier un certain nombre de fonction dans différents morceaux du programme dans ces super-classes mais par manque de temps ça ne sera pas possible.

Nous avons également choisi de rajouter des âges. Actuellement un seul âge existe mais il ne serait pas très compliqué de rajouter plusieurs autres âges. Il est à noter que chaque changement d’âge entraînera un changement de musique.

Nous n’avons pas prévu que les mécaniques du joueur puissent évoluer grandement, la plupart des mécaniques du joueur sont ainsi fixées et compliquées à changer.

Il nous semblait en revanche important de pouvoir modifier la taille du terrain, c’est pourquoi la taille du terrain est une variable générale qui peut être modifiée à l’envie sans que cela pose de problème. De même, nous avons choisi de faire un terrain composé de case qui sont une classe spécifique car nous voulions nous offrir la possibilité de créer des cases avec des effets spéciaux. Ainsi, il est possible en créant une classe qui hérite de Case de très rapidement créer un type de Case spéciale.

Enfin nous voulions pouvoir rajouter des bâtiments supplémentaires et des unités pour construire et réparer ces bâtiments mais nous avons manqué de temps pour le faire. Mais on retrouve dans le code tous les éléments qui permettrait de le faire, notamment une super-classe Bâtiment qui hérite de la même que Soldat, Unité et la possibilité de stocker des Bâtiments en plus des Soldats sur les Cases. De même, les joueurs possèdent une base, une liste de soldat et une liste de bâtiment. La base sert à savoir quel est la base du joueur et la liste de soldat quels sont les unités de ce joueur. La liste de bâtiment aurait eu le même rôle que la liste de soldat si nous avions eu le temps de les rajouter.

La classe Terrain est un singleton, en effet, le constructeur ne peut pas être appelé en dehors de la classe mais il est possible d’obtenir une instance de Terrain. L’intérêt de ce Terrain est double, premièrement, il centralise toutes les informations qui peuvent être nécessaire n’importe où dans le programme, deuxièmement, il sert à lancer les différentes fonctions de toutes les classes qui existent dans le projet. Par exemple, c’est lui qui lance le tour du joueur 1 puis du joueur 2.