Projet PCII

Nous avons pour but de réaliser un jeu en temps réel qui se base sur le jeu Fourmizz. L’objectif de ce projet est de réaliser un jeu de la forme 1 vs IA, où chaque joueur possède une liste d’Unités et de Bâtiments. Chaque joueur possède différents types de ressources et le gagnant sera désigné quand l’un des 2 joueurs atteindra une limite de ressource (un système de course à la ressource).

### Règles de ce jeu :

Chaque joueur commence la partie avec une fourmilière qui lui sert de bâtiment principal (càd que c’est l’endroit où se stockera les ressources récoltées par les fourmis ouvrières.

Chaque joueur commencera avec un nombre de ressources basique afin que chacun puisse débuter le jeu sans soucis.

Afin de pouvoir créer des fourmis combattantes, il faudra pour cela implanter un nouveau bâtiment (nommé Caserne) sur la grille de jeu. Mais avant de pouvoir ces futures fourmis, il faudra avant tout avoir les ressources nécessaires.

Pour la récolte des ressources, chaque joueur peut cliquer sur une ouvrière et lui ordonner d’aller récolter des ressources sur la case indiquée par le joueur.

Au cours d’une expédition pour récolter des ressources, les joueurs peuvent décider d’envoyer, en plus des ouvrières, un certain nombre de fourmis combattantes afin d’assurer une certaine sécurité. Néanmoins, il se peut que lors de cette même expédition, des fourmis combattantes ennemies viennent tout interrompre et fassent échouer cette expédition en repartant avec ces mêmes ressources.

Lors d’un duel entre fourmis ennemies, en fonction des unités de combats, la fourmi gagnante se fera en fonction de plusieurs paramètres (vitesse d’attaque, coup d’attaque et de défense)

Comme annoncé précédemment, pour pouvoir gagner la partie, chaque joueur doit récolter un maximum de ressources, le plus vite possible avant son adversaire. Cependant, la partie peut se finir d’une autre manière. En effet, la fourmilière adverse peut se faire attaquer et se faire dépouiller l’intégralité de ses ressources, laissant ainsi l’adversaire sans ressource, sans moyen de pouvoir créer de nouvelles ouvrières.

Dans un premier temps, nous avons réalisé un espace de travail respectant le MVC afin de pouvoir programmer sans avoir une quelconque erreur lors de l’affichage ou le changement d’état. De plus, nous y avons inclus des classes abstraites comme Unités qui sera utilisé afin de potentiellement créer différentes unités.

Dans un deuxième temps, nous nous sommes séparé le travail afin de pouvoir exécuter une première version de l’espace de jeu ainsi qu’un prototype d’IA naïve afin d’avoir une idée de comment représenter tous nos éléments un à un et d’ainsi définir de nouvelles classes comme Joueur, Bâtiments…

## 1ère Version : Construction de la grille de jeu + IA

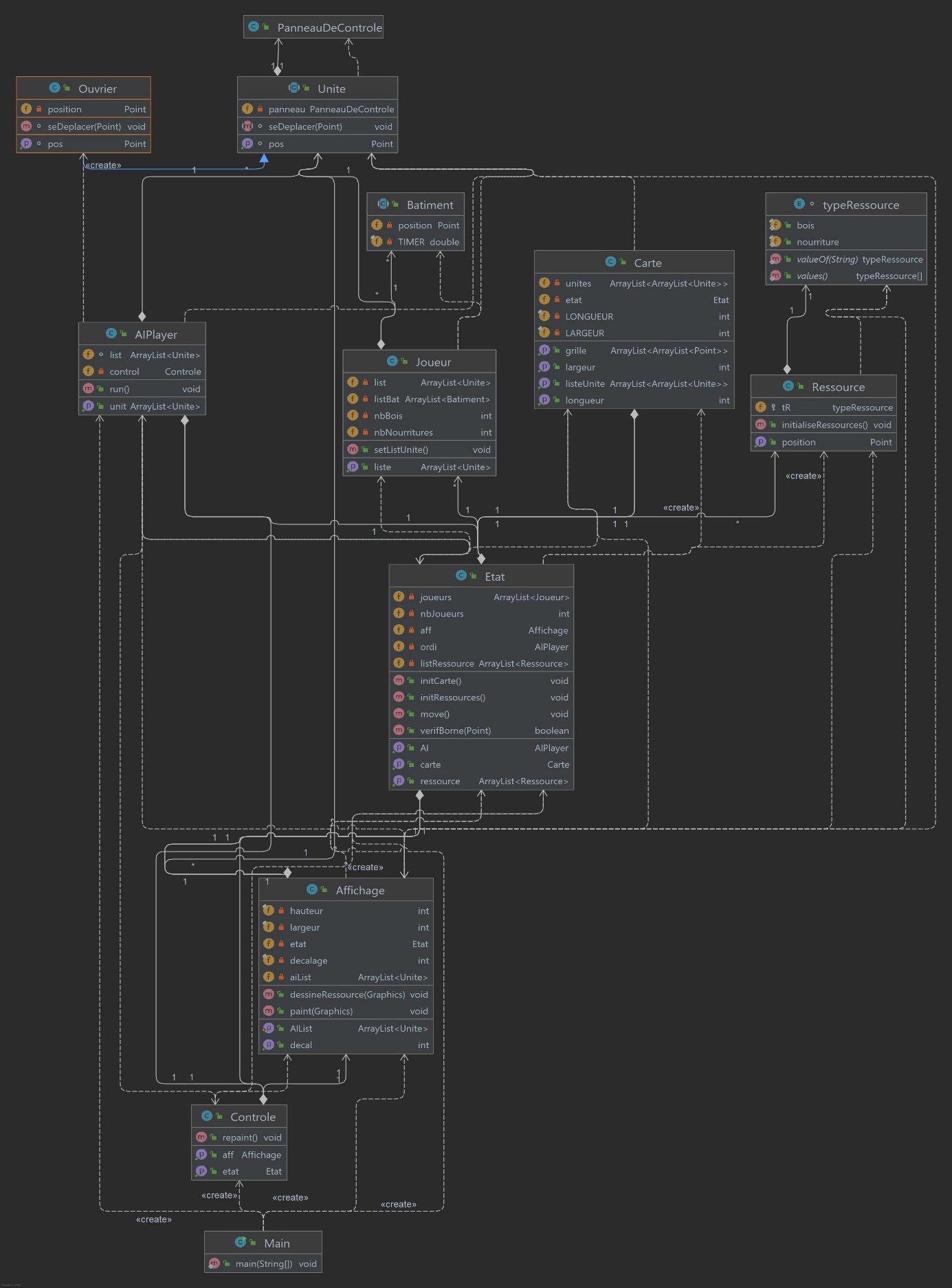
Au cours de cette 1ère semaine de travail, nous sommes partis sur la réalisation de l’affichage de la grille de jeu, ainsi que d’une première version d’une IA (Qui pour l’instant ne va pouvoir que se déplacer sur notre grille de jeu). Pour pouvoir réaliser cela, nous avons défini une première unité, qui est l’unité basique de ce jeu, l’ouvrière. En effet, cette unité aura la possibilité de récupérer toutes les ressources dispersées sur la grille de jeu et de ramener ces ressources à sa fourmilière. Concernant l’IA, afin de gérer le déplacement de cette dernière, nous avons décidés d’encapsuler cette unité dans un Thread afin de ne pas avoir à gérer cela de nous-même.

## 2ème Version : Implémentation des ressources sur la grille de jeu + Bâtiments

(Suite documentions Fin semaine 3: - Notre conception du jeu avec les différentes unités et bâtiments précisément et leurs interactions (aussi on fait 1 ou 3 interfaces : les deux fourmilières + grille de base avec les ressources ?)

Pour cette 3ème semaine, nous avons continuer notre avancée en y implémentant cette fois-ci des bâtiments et des ressources. Afin de pouvoir réaliser cela, nous nous sommes d’abord demandé sur comment aller être géré la génération des ressources sur notre grille (car étant donné que notre jeu est en temps réel, il faut créer un léger mécanisme qui génèrera les ressources quand il y en aura besoin). Par la suite, nous avons dû nous interroger sur comment afficher un bâtiment sur notre grille.

L’utilité de ces bâtiments est de pouvoir créer des futures unités de combats et de pouvoir occuper un slot de notre grille de jeu.

A la fin de cette 3ème semaine, nous obtenons le diagramme suivant : Comme nous pouvons le constater, nos classes sont reliés entre-elles afin de pouvoir avoir un jeu fonctionnel et sans difficulté dans l’exécution de notre jeu. En revanche, ce diagramme n’est que provisoire, donc il n’est pas à prendre en considération pour l’intégralité de notre projet.