

Rapport de projet

Jeu ***HILE***

Projet de programmation avancée

Groupe : Auterieux Titouan, Briand Laura

Table des matières

Table des matières.....	2
0 Introduction.....	4
0.1 Présentation du projet.....	4
0.1.1 Objectif du projet.....	4
0.1.3 Technologies utilisées (C#, frameworks, outils).....	4
1 L'univers du Jeu, spécificités et gameplay.....	5
1.1 Description de l'univers.....	5
1.1.1 Thème.....	5
1.1.2 Histoire et contexte narratif.....	5
1.2 Spécificités de l'univers.....	5
1.2.1 Environnements et décors.....	5
1.2.2 Personnages principaux et leurs rôles.....	6
1.2.3 Règles et mécanismes fondamentaux.....	7
2. Possibilités des Joueurs et Déroulement d'une Partie.....	7
2.1 Possibilités des joueurs.....	7
2.1.1 Actions et interactions disponibles.....	7
2.1.2. Compétences et capacités des personnages.....	8
2.1.3. Système de progression.....	8
2.2. Déroulement d'une partie.....	9
2.2.1. Objectifs et missions.....	9
2.2.2. Conditions de victoire et de défaite.....	10
3. La Modélisation Objet Réalisée.....	10
3.1. Diagramme de classes UML.....	10
3.2.1. Présentation du diagramme.....	10
3.2.3. Justification des choix de modélisation.....	13
4. Les Tests Réalisés pour Vérifier le Bon Fonctionnement du code.....	13
4.1. Types de tests effectués.....	13
5. La Gestion de Projet.....	13
5.1.1. Rôles et responsabilités des membres de l'équipe.....	14

5.1.2. Mode de communication et collaboration.....	14
5.2. Planning de réalisation.....	14
6. Bilan Critique sur le Projet.....	14
6.1. Points forts du projet.....	15
6.1.1. Aspects réussis et innovations.....	15
6.1.2. Retour des utilisateurs/testeurs.....	15
6.2. Difficultés rencontrées.....	15
6.2.1. Problèmes techniques et organisationnels.....	15
6.2.2. Défis imprévus.....	15
6.3. Leçons apprises.....	15
6.3.1. Améliorations possibles.....	15
6.3.2. Recommandations pour de futurs projets similaires.....	15

0 Introduction

0.1 Présentation du projet

0.1.1 Objectif du projet

Dans le cadre du module de programmation avancée dispensé au semestre 6, le projet suivant nous permet de mettre en pratique les concepts clés de la programmation orientée objet comme l'héritage, le polymorphisme, l'encapsulation, ...

Le cadre du projet est fixé dans le document consigne et nous impose de réaliser un programme de type console en respectant les contraintes de la convention c# donnée en cours.

0.1.3 Technologies utilisées (C#, frameworks, outils)

Nous utilisons le framework **Dotnet 6.0** du langage C# pour programmer ce mini-jeu en lançant une configuration projet de type console avec la commande

```
>> dotnet new console -o "projet-poo"
```

Nous disposons pour ce projet d'un accès libre à la **documentation dotnet** ainsi qu'à l'ensemble des fonctionnalités qu'offre le langage et son framework en termes de classe existante.

1 L'univers du Jeu, spécificités et gameplay

1.1 Description de l'univers

1.1.1 Thème

D'après le thème imposé qui est celui d'un monde de pirate nous choisissons d'implémenter un monde marin. Le thème sera représenté par la présence d'îles, de trésors et d'un bateau pirate par exemple. Nous incorporons également les éléments constitutifs d'un monde fantastique comme un magicien, une licorne et des monstres.

1.1.2 Histoire et contexte narratif

Dans le monde **HILE** les aventuriers se font rares. Les îles sont regorgées d'étrangetés et souvent les objets que l'on y trouve sont étranges et inspirés de toutes sortes d'univers parallèles. Les monstres sont nombreux et peu en ont réchappés. Mais on dit qu'il existe un trésor, un objet si fabuleux de beauté et d'étrangeté qu'il peut vous garantir la vie éternelle. Osez-vous engager votre vie pour confronter la mort ?

1.2 Spécificités de l'univers

1.2.1 Environnements et décors

L'univers principal de jeu est constitué d'une carte unique dont l'environnement change de manière aléatoire entre chaque niveau. Le terrain qui modélise l'île est

généralisé de manière aléatoire au milieu d'une carte initiale d'océan. On visualise en jaune le sable d'une île et en bleu la mer. Pour des facilité de contraste et afin de garder le jeu accessible on distingue le bateau en rouge afin de ne pas le confondre avec la mer (bleu).

1.2.2 Personnages principaux et leurs rôles

Un personnage principal est joué par un joueur unique en troisième personne vue de haut. On peut jouer à deux joueurs en 1 VS 1 afin de s'affronter sur une carte, dans ce cas, les deux joueurs contrôlent deux personnages principaux choisis en début de partie.

Les personnages jouables sont

Pirate	Magicien	Licorne	Jean-Norbert
--------	----------	---------	--------------

Spécificité des personnages :

Jean-Norbert : Le personnage Jean-Norbert est un cogniticien aventureux qui est en quête de la compréhension de l'univers et notamment de l'axe rostro-caudal. Si il découvre le livre "Cognitique: Science et pratique des relations à la machine à penser" il termine le jeu en gagnant. Il est relativement facile de trouver le livre mais la survie de Jean Norbert en tant que non guerrier n'est pas facilement assurée. Il sera beaucoup plus en danger que les autres personnages face aux monstres.

Pirate : Le pirate est féroce et sanguinaire, ses attaques sont puissantes et il ne fait pas bon de le croiser.

Magicien : Le magicien lance des sorts de loin et peut ainsi attaquer à une distance de 5 cases. De même, il peut se téléporter pour se déplacer (une fois tous les 5 tours en moyenne).

La licorne : La licorne peut aléatoirement charger et donc avancer plus vite, elle possède également la possibilité de se régénérer seule de manière aléatoire. C'est un personnage agile et utile.

1.2.3 Règles et mécanismes fondamentaux

Chaque personnage peut se déplacer à volonté sur l'île sur laquelle il apparaît. Nager, creuser, attaquer un monstre coûte de l'énergie au personnage, et en l'absence d'énergie, le personnage perd un point de vie. Chaque trésor est enterré entre 1 et 9 mètres de profondeur et chaque action de creuser permet de creuser 1m supplémentaire. Une fois le trésor trouvé, il rentre dans l'inventaire du joueur qui dispose de 5 places. Il est possible à tout moment de déposer du matériel dans le coffre du bateau.

2. Possibilités des Joueurs et Déroulement d'une Partie

2.1 Possibilités des joueurs

2.1.1 Actions et interactions disponibles

A chaque tour, le joueur peut effectuer différentes actions :

- **Se déplacer** : le déplacement du joueur peut être effectué à chaque tour dans la direction choisie par le joueur (haut, bas, gauche ou droite) à l'aide des touches **Z, Q, S et D ou sur les flèches du clavier** . Si le déplacement est effectué sur une case eau, le joueur perd de l'énergie.
- **Creuser** : lorsque le joueur se trouve sur une case trésor, il peut creuser en appuyant sur la touche **ENTRER** de son clavier. Il doit creuser plusieurs fois (9 fois au maximum) pour trouver un ou plusieurs objets qui correspondent aux trésors.
- **Manger** : à tout moment de la partie (pendant son tour), le joueur peut appuyer sur la touche **M** pour manger. Ainsi, si il possède un objet comestible dans son inventaire, cet objet disparaît (il est mangé) et le joueur gagne de l'énergie. En revanche, si aucun des objets présents dans son inventaire n'est comestible, un message "vous n'avez rien à manger dans votre inventaire" s'affiche.
- **Boire** : comme pour l'action manger, le joueur peut boire à n'importe quel moment de la partie en appuyant sur la touche **B**. Ainsi, s'il possède un objet buvable dans son inventaire, cet objet est bu et le joueur gagne ou perd de l'énergie en fonction de l'objet avalé.
- **Attaquer** : lorsque le joueur se trouve sur une case à côté de celle d'un monstre ou d'un joueur adverse, il peut attaquer ce dernier en appuyant sur la touche **A**. La cible perd alors des points de vie à chaque coup du joueur reçu
- **Déposer** : lorsque le joueur est sur le bateau, il peut déposer les objets se trouvant dans son inventaire en échange de points de score (différents selon les objets). Pour effectuer cette action, il doit appuyer sur la touche **ESPACE**.

2.1.2. Compétences et capacités des personnages

L'importance des dégâts fait par un joueur lors d'une attaque évolue au cours de la partie. En effet, plus le niveau du joueur est élevé, plus les dégâts sont importants. De plus, le niveau de magie des personnages permet également d'augmenter les dégâts infligés lors d'une attaque.

2.1.3. Système de progression

Les joueurs possèdent tous un niveau, des points de santé, des points d'énergie, un niveau de magie et un score. Chacune de ces 5 caractéristiques peut augmenter ou diminuer au cours de la partie.

Score : celui-ci augmente lorsque le joueur dépose ses objets sur le bateau, chaque objet rapporte un score qui lui est propre.

Niveau : celui-ci augmente au fur et à mesure de la partie lorsque le score du joueur atteint 10.

Énergie : l'énergie du joueur peut soit diminuer, soit augmenter en fonction de l'action du joueur. Si ce dernier se déplace dans l'eau, l'énergie diminue. De même si le joueur creuse ou attaque un monstre. En revanche, le joueur peut gagner de l'énergie en mangeant ou en buvant des objets qui sont dans son inventaire. Si l'énergie se retrouve égale à 0, le joueur perd un point de santé.

Santé : la santé du joueur peut diminuer d'un point si ce dernier perd toute son énergie. En revanche, elle augmente de 10 points lorsque le joueur augmente de niveau.

Magie : le niveau de magie augmente à chaque niveau obtenu par le joueur. Il permet d'augmenter les dégâts infligés lors d'une attaque.

Les valeurs de chacune de ces caractéristiques dépendent du type de personnage choisi par le joueur au début de la partie.

2.2. Déroulement d'une partie

2.2.1. Objectifs et missions

L'objectif du jeu est de récolter le **meilleur score possible** en ramenant des objets trouvés en creusant sous terre au bateau. Pour cela, le joueur doit combattre des monstres ou des joueurs adversaires, gagner des niveaux, trouver des objets spéciaux pour augmenter leur dégâts ou leur énergie tout cela en ramenant un maximum d'objets sur le bateau.

2.2.2. Conditions de victoire et de défaite

Pour une partie à deux joueurs (ou 1 joueur contre un bot), la condition de victoire est de finir la partie avec un score plus élevé que son adversaire. Pour une partie à un joueur, la condition de victoire est la découverte et le dépôt dans le bateau de tous les objets présents initialement sur la map.

En revanche, si le joueur n'a plus de vie, il perd la partie. De plus, si lors d'une partie à deux joueurs, si aucun des deux n'a perdu avant d'avoir ramené tous les objets sur le bateau, le joueur perdant est celui qui finit la partie avec le moins de points.

3. La Modélisation Objet Réalisée

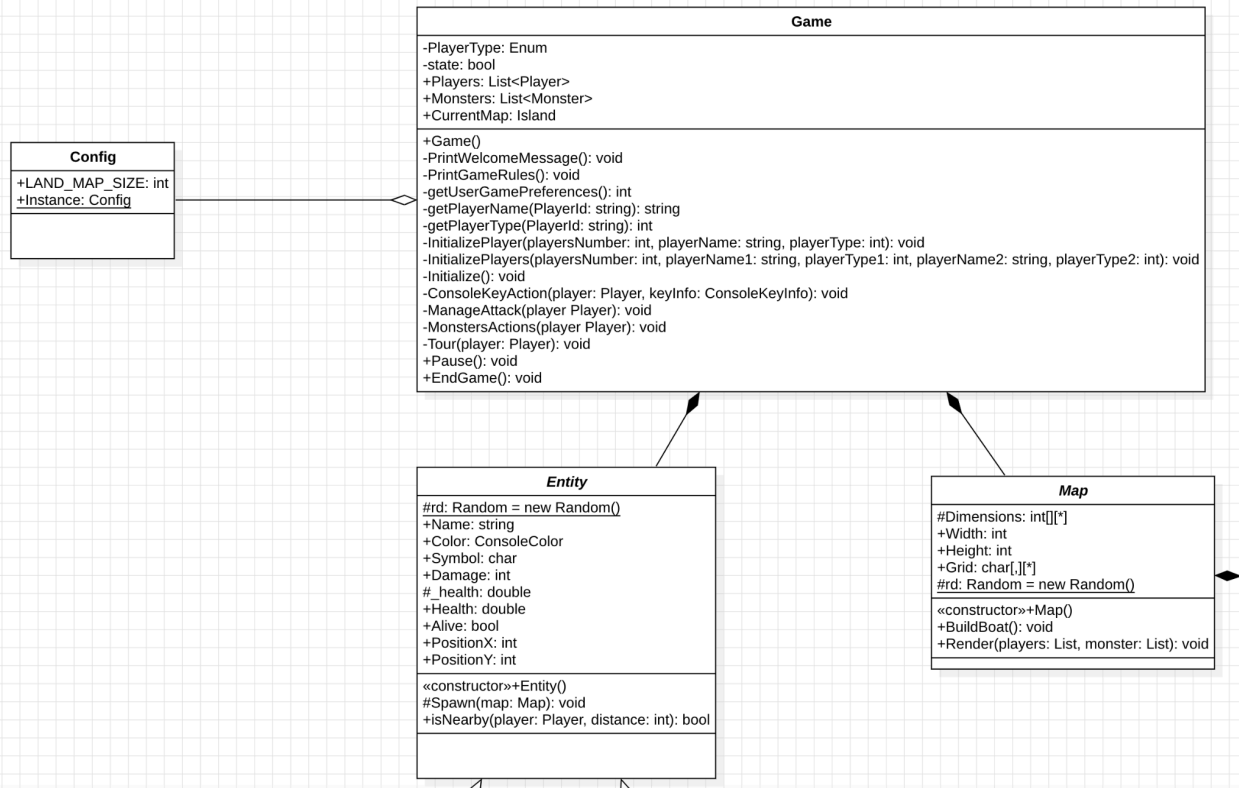
3.1. Diagramme de classes UML

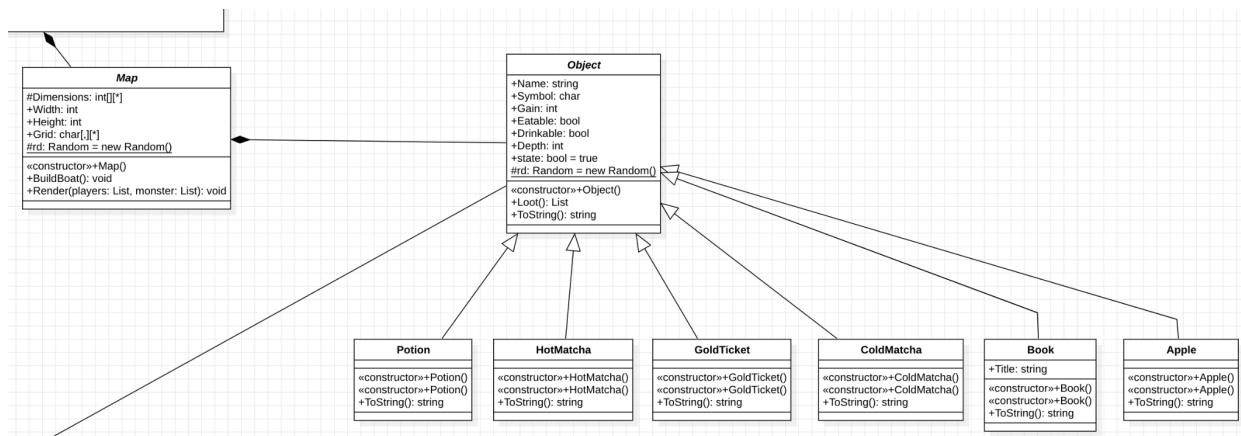
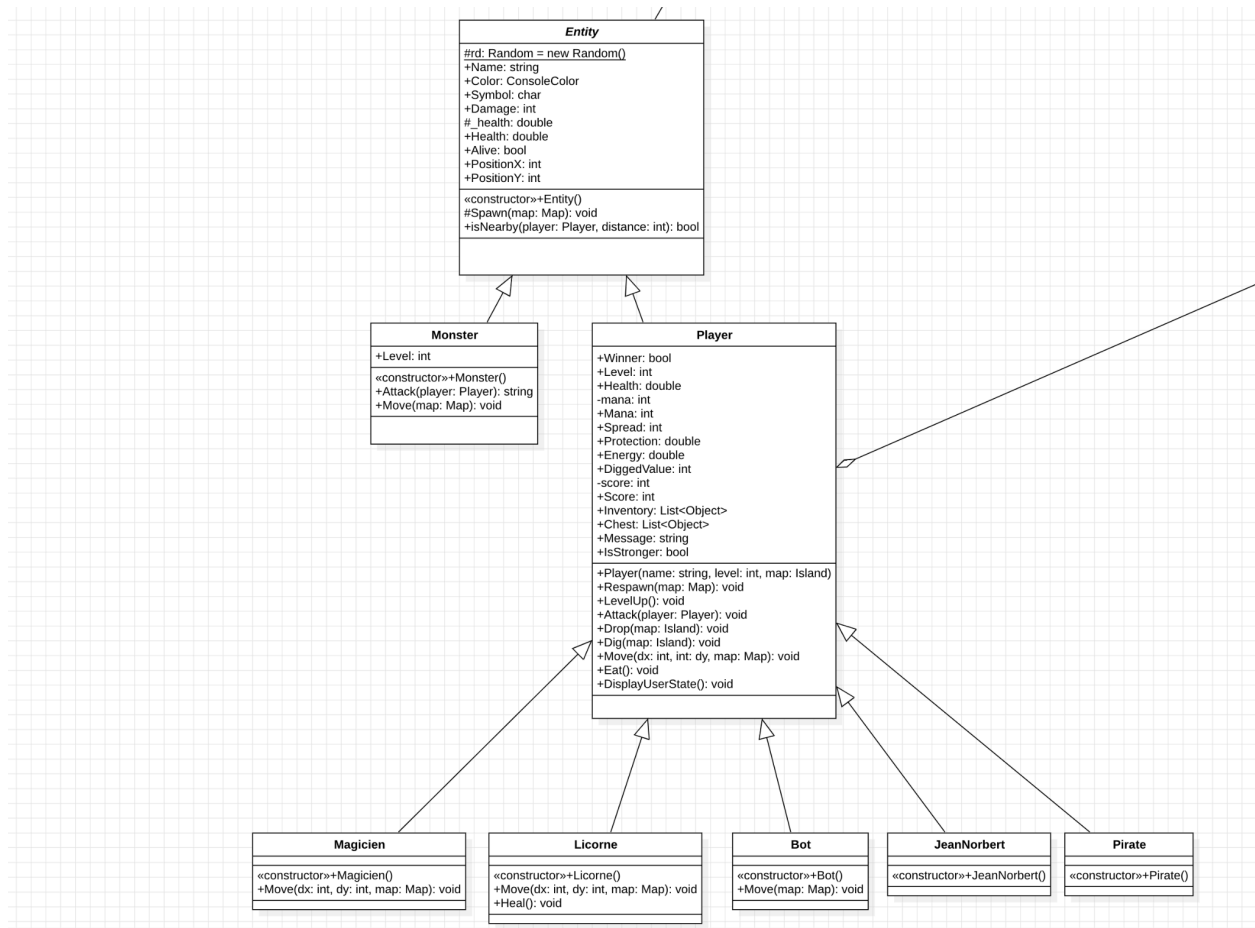
3.2.1. Présentation du diagramme

Le diagramme UML est organisé en trois parties majoritaires : le jeu (déroulement, articulation des acteurs, fonctionnalité de traitement du joueur), la carte (bateau, terre, mer, couche du terrain, couche des objets, etc ...) et les entités (joueurs et ses déclinaisons, bot, monstres).



3.2.2. Classes principales et leurs relations





3.2.3. Justification des choix de modélisation

Nous choisissons de bien séparer les éléments de terrain (fixes et positionnés précisément) des éléments mobiles. Ainsi, au long de la partie, la carte n'a pas accès aux informations de position des joueurs, excepté lors du rendu. Cela causerait une agrégation en double sens qui alourdirait grandement le fonctionnement du programme.

Les objets sont sauvegardés dans une grille de terrain que l'on peut visualiser comme une couche d'abstraction.

La première couche est celle du terrain, une simple grille `char[][]` permet de représenter à la fois les zones de mer et de terre.

Une deuxième couche virtuelle représentée par une grille `Object[][]` sert à représenter les objets et leur positions sur la MAP. Leur profondeur est établie aléatoirement à la création de l'objet.

4. Les Tests Réalisés pour Vérifier le Bon Fonctionnement du code

4.1. Types de tests effectués

Un programme de test `Tests.cs` permet de visualiser les principales fonctionnalités du jeu indépendamment et d'isoler une erreur de fonctionnement grâce à un système

Try() Catch()

5. La Gestion de Projet

5.1. Organisation de l'équipe

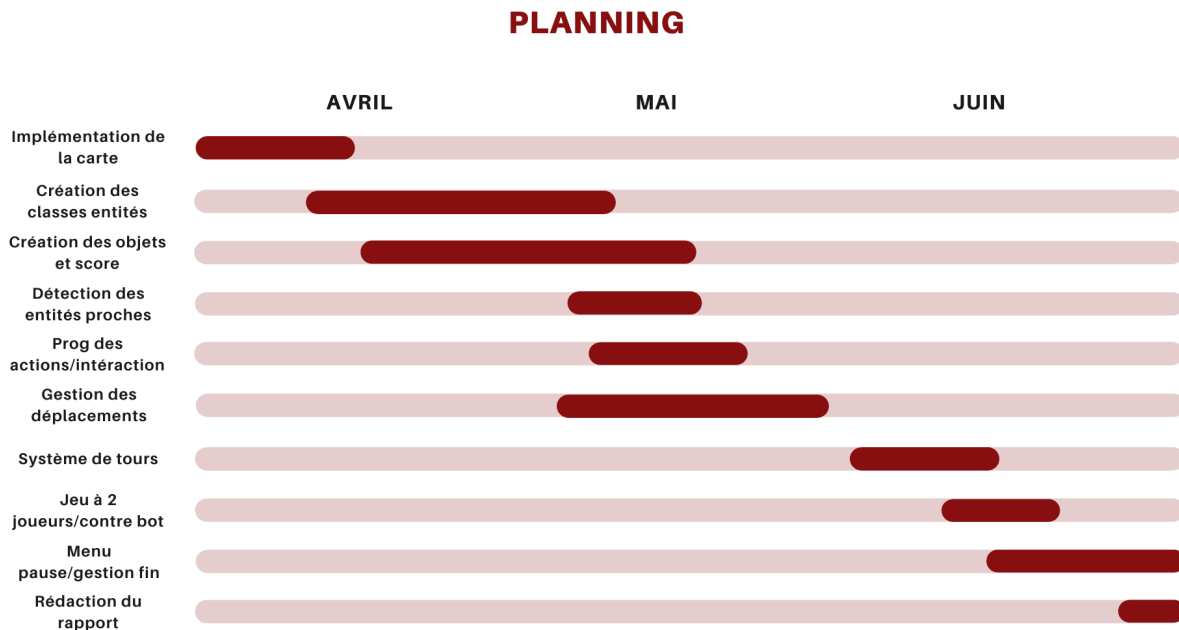
5.1.1. Rôles et responsabilités des membres de l'équipe

L'équipe est composée de deux développeurs. Nous avons donc réparti les tâches principales en moitié et l'idéation du jeu à été réalisée lors de points de réunions réguliers.

5.1.2. Mode de communication et collaboration

Comme dit précédemment, des réunions et des points réguliers ainsi qu'une communication par l'outil whatsapp, nous ont permis de faire avancer le projet correctement. Nous avons travaillé collaborativement grâce aux outils de versionnage git et github.

5.2. Planning de réalisation



6. Bilan Critique sur le Projet

6.2. Difficultés rencontrées

6.2.1. Problèmes techniques et organisationnels

La modélisation des interactions entre les entités et les objets, ainsi qu'entre les entités entre elles a majoritairement posé problème. En effet, les interactions nécessitent la détection des plus proches voisins sur une grille. Nous avons choisi une méthode "artisanale" en soustrayant les abscisses et ordonnées des positions des deux entités.

6.2.2. Améliorations possibles

Il peut être judicieux de penser à améliorer les capacités des personnages. Nous aurions pu mieux gérer l'héritage des types de personnages afin de gérer les possibilités pour le joueur. Par ailleurs, il est possible que le système permettant de gérer les types d'objets ne soit pas le plus optimisé. Il serait peut être possible de conceptualiser autrement ce système.