

Réalisation du jeu «*Miniville*»
sous forme de programme

Projet tutoré semestre 2

Étudiants :

Jonathan Selle
Florent Dewilde
Adam Bernouy
Brice Mautalent

Professeurs :

Philippe Le-pivert
Bruno Legrix

Projet réalisé dans le cadre d'une évaluation en DUT informatique du
19/06 au 24/06 2017

TABLE DES MATIÈRES :

1. Introduction	5
2. Démarches et choix du projet	6
1. Organisation de la semaine	6
1. Liste des objectifs	6
2. Répartition des tâches	7
2. Les ressources	8
1. Informations sur le jeu	8
2. Recherches personnelles	8
3. Les outils utilisés	9
1. Différents éditeurs de texte	9
2. Les bibliothèques java	9
3. Git/GitHub	10
3. Les différents apprentissages liés à ce projet	10
1. Apprentissages liés à l'écriture du code	10
2. Autres apprentissages	11
4. Avis du groupe sur la semaine de projet	12
1. Les apports de ce projet	12
1. Apports méthodologiques	12
2. Autres apports	13
2. Si il y avait quelque chose à améliorer	13
5. Conclusion	14
6. Annexes	15

I- Introduction

Pendant notre dernière semaine avant les vacances d'été, il nous a été imposé, à nous, étudiants du département informatique de l'IUT du Havre, de réaliser un projet parmi deux choix proposés. Notre choix s'est orienté vers la reproduction du jeu de société « *Miniville* » sous forme d'un programme. Le programme final devait nous permettre de jouer de façon intuitive au jeu de société « *Miniville* », avec en Bonus une interface IHM GUI, une possibilité de jouer en réseau, la probabilité que le joueur courant obtienne un certain nombre de pièces, la gestion des cartes mauves et la sauvegarde d'une partie en cours. Le projet devait être réalisé en une semaine afin de faire la démonstration et la soutenance le Lundi 26/06/2017.

II- Démarches et choix du projet

II.1 Organisation de la semaine

II.1.1 Liste des objectifs

Le projet comportait différents objectifs à faire du plus simple au plus compliqué. Les professeurs nous ont proposé 5 étapes avant de passer à la phase finale :

1. Ecriture d'un programme qui initialise la pioche ouverte, et la main de deux joueurs. Les joueurs disposent de 20 pièces. A leur tour, le jeu lance **un seul** dé, et ils décident d'acheter ou non un établissement. Les monuments ne sont pas encore gérés. On devrait pouvoir visualiser la situation complète de la pioche ouverte, ainsi que pour chaque joueur, le nombre de chaque carte et l'argent qu'il possède.
 2. Gérer uniquement les Règles des cartes vertes.
 3. Gérer en plus les règles des cartes bleues.
 4. Gérer en plus les règles des cartes rouges.
 5. Gérez maintenant les monuments, achat et effet.
 6. Gérez 2 à 4 joueurs.
-

II.1.2 – Répartition des tâches

- Pour l'organisation du projet, il était nécessaire de définir les objectifs appropriés à chacun. Les premières étapes du projet n'étaient pas forcément des tâches complexes, c'est pour cette raison que chacun a fait une partie des classes sans répartition forcée. Par la suite, les différentes étapes attribuées ont été distribuées à chacun. Florent s'est occupé de l'interface CUI, Jonathan s'est occupé principalement des règles de cartes, Adam et Brice se sont occupés des étapes de code secondaires, ainsi que du débogage et correction du code lorsque cela était nécessaire. Adam a également intégré la possibilité de lancer l'application depuis n'importe quel endroit, et Brice s'est occupé du diagramme UML.

Pour la phase finale du projet, l'IHM GUI est gérée par Florent, la partie réseau par Jonathan, les parties de test sont gérées par Adam, et les Cartes mauves Brice et Adam.

Un planning a été réalisé sous forme d'un tableau pour chaque jour afin de mieux voir les tâches effectuées par chacun de nous.

II.2 Les ressources

II.2.1 Informations sur le jeu

Pour débiter le projet, un certain nombre d'informations nous ont été données à propos du jeu, telles que les règles du jeu, le principe de fonctionnement etc. Il fallait évidemment reproduire le jeu avec exactement les mêmes règles de cartes ainsi que le même fonctionnement.

Certaines informations nous étaient manquantes, les professeurs avaient donc mis à disposition une F.A.Q (Foire aux questions) afin de nous aider si jamais nous rencontrions des difficultés sur une étape du projet.

II.2.2 Recherches personnelles

D'autres informations non pas à propos du jeu mais plutôt à propos de certaines parties du code ne nous étaient pas encore tout à fait connues. Il a fallu utiliser certaines ressources des bibliothèques java qui nous étaient à l'heure actuelle encore inconnue. Nous avons donc effectué nos recherches sur les Javadoc (Documentation complète java sur une classe en particulier)

Cf : Javadoc [par ici](#)

Pour la version imprimée, il suffit de chercher sur un moteur de recherche quelconque « Javadoc », suivi de la méthode ou classe que vous souhaitez consulter

II.3 Les outils utilisés

II.3.1 Différents éditeurs de texte

Pour ce qui est de l'édition de texte, Adam a choisi «*Notepad++*», un éditeur peu gourmand en ressources, utile pour sa machine peu performante.

Florent a de son côté décidé d'utiliser «*SublimeText*», tandis que Brice et Jonathan ont préféré l'utilisation de «*Atom*». Il s'agit de deux éditeurs de texte permettant des actions plus avancées, comme l'édition sur plusieurs lignes, ou des raccourcis comme «**pl + Entrée** », qui évite d'écrire «**System.out.println();**».

Encore une fois, le but était d'optimiser notre temps. De plus, leur thème étant plus sombre, il était plus agréable de travailler sans que les couleurs claires nous fatiguent les yeux. Quoi qu'il en soit, tant que nous utilisions le même encodage pour l'édition de nos fichiers, les différents éditeurs cohabitaient parfaitement, et pouvaient être adaptés à chacun.

II.3.2 Les bibliothèques Java

Nous avons développé le jeu à l'aide du langage de programmation objet «*Java* » il nous a donc été nécessaire d'utiliser les bibliothèques Java comme outil afin de pouvoir développer correctement l'application.

II.3.3 Git/GitHub

Pour gagner du temps, nous avons dès le début du projet décidé de tous installer « *GitHub* ». Il s'agit d'un logiciel de partage très pratique permettant de récupérer directement les modifications de chacun, la seule contrainte étant de ne pas modifier le même fichier en même temps. Par pression d'un bouton, tout est envoyé aux autres membres du groupe, avec une description et un titre permettant d'identifier la tâche accomplie, ainsi que le pseudo de l'expéditeur.

III- Les différents apprentissages liés à ce projet

III.1 Apprentissage liés au développement de l'application

Comme dit précédemment, il nous a parfois été obligé d'aller chercher des informations sur la javadoc pour progresser dans notre code. Certaines notions nous étaient encore inconnues et nous avons donc appris des notions de programmation au cours de ce projet. Nous nous sommes également améliorés dans certaines notions déjà vues précédemment, telles que l'héritage ou encore l'utilisation de méthodes dites abstraites.

III.2 Autres apprentissages

Nous avons complètement changé notre organisation par rapport aux projets de groupe précédents. L'organisation des précédents projets était plutôt médiocre. L'utilisation de Git/GitHub pour ce projet a nettement amélioré l'organisation, puisque git nous permet une très bonne gestion des fichiers de groupe, ainsi qu'un transfert rapide à partir du moment où nous connaissons les bases de celui ci. De plus, il est multi plateforme, nous avons donc pu gérer les projets en utilisant des systèmes d'exploitation différents.

Nous avons également appris à gérer des variables d'environnement de façon à pouvoir installer le jeu depuis n'importe quel point.

IV- Avis du groupe sur la semaine de projet

IV.1 Les apports de ce projet

IV.1.1 Apports méthodologiques

En termes d'apports méthodologiques, nous nous sommes considérablement améliorés pour ce qui est de l'organisation. L'utilisation d'un outil adapté aux travaux de groupe (Ici en l'occurrence *Git*) est extrêmement pratique si nous savons l'utiliser. L'utilisation d'un planning a été également une bonne chose, la gestion du temps est plus simple et nous n'avons pas forcément eu de problème de retard ou quoi que ce soit.

IV.1.2 Autres apports

Globalement, un travail de groupe de ce genre nécessite une bonne cohésion de groupe. Chose que nous avons déjà avant de commencer ce projet, puisque nous avons déjà réalisé des projets ensemble. Ce projet a simplement permis de renforcer cette cohésion de groupe, et si nous devons refaire de nouveaux projets de groupe, nous ferions en sorte de garder ce groupe. Les travaux de ce genre nous donnent également une petite idée de ce à quoi pourrait ressembler des travaux projets en entreprise. Nous sommes encadrés par des supérieurs, avec une limite de temps et un objectif précis. C'est à nous de gérer notre projet de la meilleure des façons possibles.

IV.2 Si il y avait quelque chose à refaire

Le jeu qu'il fallait réaliser est fonctionnel, cependant certaines parties de l'étape finale n'ont pas pu être réalisées. En effet, nous n'avions pas prévu par exemple de faire le jeu en réseau. De ce fait, nous nous sommes retrouvés bloqués lorsque nous avons commencé cette partie réseau, puisque notre programme n'était pas du tout adapté à la partie réseau. Si nous devions refaire ce projet, nous ferions en sorte que le programme soit adaptatif à toute nouvelle fonctionnalité.

Malgré le planning nous avons peut être traîné sur certains objectifs pour cause d'erreurs et de dysfonctionnement du programme certaines fois. La prochaine fois il serait plus judicieux de réfléchir plus longtemps sur la façon dont nous pourrions réaliser le programme demandé avant de se lancer dans l'écriture.

V- Conclusion

Ce projet nous a permis de revoir l'ensemble de tout ce que nous avons pu voir depuis le début de l'année, et ce dans plusieurs modules (Base de la programmation et de la conception orientée objet, gestion de projet, introduction aux IHM). Ce genre de projet est vraiment très intéressant au niveau de la conception, il permet de revoir toutes les difficultés que nous avons, et de progresser dans celles ci, de façon à équilibrer notre niveau. Nous avons beaucoup apprécié ce projet et nous ferions sans hésitations d'autres projets de ce genre.

VI- Annexes

Diagramme UML :

