

**Rapport de Développement / Tests**

Dans ce document, nous vous faisons part de notre rapport de développement de notre projet Monopoly. Il s’agit de faire brièvement le point sur ce que nous avons codé tout au long du projet.

Globalement, l’application n’est pas fonctionnelle excepté quelques détails. Nous pouvons lancer l’application et faire apparaître une première fenêtre qui est en réalité le menu. Sur cette fenêtre, deux boutons s’offrent à nous, l’un permet de rentrer dans le jeu et l’autre de quitter l’interface. Jusque-là, l’interface fonctionne. Si nous rentrons dans le jeu en cliquant sur le premier bouton « Jouer », nous arrivons sur une nouvelle fenêtre avec plusieurs éléments : le plateau du jeu Monopoly et les boutons « Lancer le dé », « Acheter », « Vendre », « Passer son tour », « Action Case », « Banque » et « Quitter le jeu » situés à droite. A partir de cette fenêtre, seuls les boutons « Quitter le jeu » et « Lancer le dé » sont fonctionnels. Nous ne pouvons pas changer de joueur ni interagir avec les autres boutons. Au final, nous sommes parvenus à afficher l’interface et lancer plusieurs fois le dé.

Intéressons-nous maintenant aux différentes fonctions du code. Aucunes fonctions du code n’est entièrement fonctionnelles et certaines ont mêmes étaient abandonnées. La classe Argent par exemple a été supprimée car en rajoutant quelques attributs et méthodes dans la classe Joueur, cela rendait la classe Argent inutile. De plus, certaines méthodes ont été supprimés parce qu’elles étaient inutiles. Finalement, nous avons essayé de compléter un maximum de fonctions.

Penchons-nous sur les difficultés rencontrées durant ce projet. La première difficulté rencontrée a été celle d’une image : Quand nous lancions l’application pour arriver sur la première fenêtre du menu, il y avait une image en arrière-plan. Pour ce faire, l’image avait été ajoutée dans l’IHM afin de l’afficher sur la fenêtre. Le problème, c’est qu’une fois que nous avions tous « push », la personne qui lancer ensuite l’application n’avait pas l’image en arrière-plan. Au début, nous ne savions pas pourquoi une seule personne avait l’image et pas les autres. Ceci-dit, le problème a été rapidement réglé car il s’agissait simplement du chemin emprunté par l’image ! En effet, tout le monde n’a pas le même emplacement du « Workspace » dans ses dossiers du coup l’adresse de l’image changeait pour chacun de nous. Il fallait donc à chaque fois récrire le chemin emprunté par l’image dans la classe de l’IHM si nous voulions l’afficher lors du lancement de l’application. En réalité, il s’agissait de deux images, celle du menu de la première fenêtre et celle du plateau lorsque nous arrivions sur le jeu. Ce problème est toujours présent aujourd’hui mais il ne s’agit plus réellement d’un problème mais plutôt d’une modification à apporter lors de chaque lancement du jeu.

Un autre problème nous a contrariés tout au long du développement du jeu. En effet, à chaque fin de séance de travail, nous devions tous « commit and push » notre code afin qu’il se complète avec le code des autres. Le problème ? Aucun de nous parvenait à « commit and push » correctement sans qu’il y ait des conflits ou des erreurs. Du coup, il nous arrivait de supprimer le projet et de le télécharger via le dépôt GitHub. En faisant cela, nous récupérions la dernière version qui avait été « push » et réduisons les problèmes de conflits car nous partions tous du même code. Ce problème est toujours d’actualité mais comme je l’ai dit, de temps en temps nous récupérons le dernier code « push » valide sur Git Hub.

Un dernier problème nous a distraits. Le fait de lier l’IHM à l’algorithme : En fait, il fallait relier les boutons à l’algorithme correspondant. Ce genre de problème est partiellement résolu mais nous n’arrivons pas à actualiser le plateau en fonction des actions faites précédemment.

Concernant l’application, nous la lançons depuis le « Main » en cliquant sur le bouton « Run » d’Eclipse. Apparait ensuite une première fenêtre (explication faite dans le deuxième paragraphe) puis une seconde fenêtre surgit après avoir cliqué sur « Jouer ». La deuxième fenêtre est concrètement le plateau de jeu, là où va se dérouler la partie.

Il est inutile de fournir des paramètres avant de lancer l’application.

En conclusion, notre phase « développement » c’est bien déroulé, les problèmes que nous avons rencontrés n’ont pas tous été résolu mais, au moins, nous les avons compris. Notre jeu Monopoly n’est pas terminé, nous ne pouvons pas faire une partie classique. Ce projet nous aura quand même permis d’approfondir nos connaissances en matière de « code Java ».