

Rapport Projet

I- Description du programme fourni	3
II- Conception et spécification	7
III- Evaluation de la réalisation	10
IV- Organisation du travail	12
V- Perspectives	12

I- Description du programme fourni

Le jeu de gestion de ressources et d'ouvriers est un jeu qui fait appel à la réflexion du joueur afin d'avoir le meilleur score possible en trente manches. En effet le score peut varier en fonction de la logique du joueur mais aussi de la chance du joueur.

En résumé c'est un jeu avec une série de bâtiments, dont chaque bâtiment a une action spécifique lorsqu'un ouvrier est placé dessus. Les bâtiments ont la caractéristique d'avoir l'apparence de tetromino. C'est-à-dire quatre cases présentent au maximum par bâtiment.

Cela implique qu'un bâtiment peut contenir au plus quatre ouvriers pour exercer l'action. Comme dit précédemment ce jeu consiste à faire le plus grand score, et pour cela il faut bien calculer ce qui se passera dans les prochains pour ne pas être bloqué en matière de pose de bâtiments ou d'ouvriers.

Dans ce jeu le joueur peut avoir plusieurs actions possibles, en commençant par la pose d'un bâtiment quelconque sur le plateau de jeu. Pour cela il se trouve un bouton à côté de chaque bâtiment.

Lorsqu'un appui est détecté sur l'un de ces boutons, il y a une vérification pour vérifier que le bâtiment se trouve bien dans la main du joueur, et s'il est présent, le bâtiment est directement placé dans l'extrémité haute du côté gauche du plateau.

Une fois le bâtiment posé il est déplaçable dans tout le plateau (avec les flèches du clavier), il est aussi pivotable avec la touche "T" du clavier (mis à part l'O.-Block). Une fois que le joueur a fini de placer son bâtiment, il doit le valider avec la touche "ENTRÉE" du clavier.

Ensuite le joueur peut mettre fin au tour (même s'il n'y a pas d'intérêt à passer au prochain tour sans placer d'ouvriers) ou sinon il peut placer un ouvrier avec la même optique que le placement d'un bâtiment.

Lors de la validation de l'ouvrier, des vérifications sont effectuées pour ne pas placer un ouvrier autre part qu'un bâtiment. Une fois que ses actions ont été effectuées le joueur peut passer au prochain tour à l'aide du bouton "Fin de tour".

Si l'utilisateur a besoin d'aide, il peut cliquer sur le bouton "aide" qui affichera une nouvelle fenêtre avec les commandes du jeu mais aussi les détails sur chaque bâtiment.

C'est-à-dire de voir leurs prix en matériaux, leurs consommations, leurs productions (énergie ou matériel), mais aussi pour certains bâtiments qui ont des actions spécifiques lorsqu'un ouvrier est positionné dessus.

Les actions des bâtiments s'enclenchent uniquement si un ouvrier est présent sur le bâtiment et à la fin de chaque tour. Si une ressource manque l'ouvrier du bâtiment sera éjecté du bâtiment.

De plus si le joueur veut visualiser ses statistiques il peut appuyer sur le bouton "Statistiques", qui affichera une nouvelle fenêtre avec les informations sur la partie en cours.

Et enfin la partie se termine une fois que le joueur ait atteint trente tours, une fenêtre s'affichera pour montrer le score final de la partie au joueur.

Vous avez ci-dessous quelques captures du jeu :

Vue d'aide :

Aide ×

<div style="background-color: #008000; color: white; padding: 5px; margin-bottom: 5px;"> I-Batiment : <ul style="list-style-type: none"> - Action : Usine - Prix : 4 matériaux - Consomme : 1 énergie - Produit : 2 matériaux </div> <div style="background-color: #90EE90; color: black; padding: 5px; margin-bottom: 5px;"> O-Batiment : <ul style="list-style-type: none"> - Action : Parc - Prix : 8 matériaux - Action détaillée : Permet de déplacer un ouvrier dans un autre bâtiment vide. </div> <div style="background-color: #FFA500; color: black; padding: 5px; margin-bottom: 5px;"> T-Batiment : <ul style="list-style-type: none"> - Action : Panneau Solaire - Prix : 2 matériaux - Produit : 2 énergies (dans les tours pairs) </div> <div style="background-color: #8B0000; color: white; padding: 5px;"> L-Batiment : <ul style="list-style-type: none"> - Action : Appartement - Prix : 8 matériaux - Consomme : 1 énergie - Action détaillée : Si 2 ouvriers sont placés sur ce bâtiment, ils retourneront automatiquement hors du plateau, et un ouvrier s'ajoutera au total des ouvriers. </div>	<div style="background-color: #800080; color: white; padding: 5px; margin-bottom: 5px;"> J-Batiment : <ul style="list-style-type: none"> - Action : Laboratoire - Prix : 8 matériaux - Consomme : 4 énergies - Produit : 2 matériaux - Action détaillée : Pioche un bâtiment de la pile. </div> <div style="background-color: #00FFFF; color: black; padding: 5px; margin-bottom: 5px;"> S-Batiment : <ul style="list-style-type: none"> - Action : Mine - Prix : 2 matériaux - Produit : 1 matériel </div> <div style="background-color: #40E0D0; color: black; padding: 5px; margin-bottom: 5px;"> Z-Batiment : <ul style="list-style-type: none"> - Action : Décharge - Prix : 2 matériaux - Consomme : 1 énergie - Action détaillée : Enlève un bâtiment non souhaité de la main. </div> <div style="background-color: #3CB371; color: white; padding: 5px;"> Comment jouer ? -Pour placer un bâtiment ou un ouvrier appuyez sur les boutons. -Déplacements effectué avec les flèches directionnelle -Pivots des bâtiment réalisable avec la touche T -Validation des ouvriers et bâtiments avec la touche ENTER. -Supprimer un bâtiment ou déplacer un ouvrier avec les boutons "Choisir ouvrier" et "Supprimer". </div>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Vue de fin de partie :

Fin ×

Fin de partie, votre score est de : 40

Vue de début de partie :

Menu ×

Nouvelle Partie

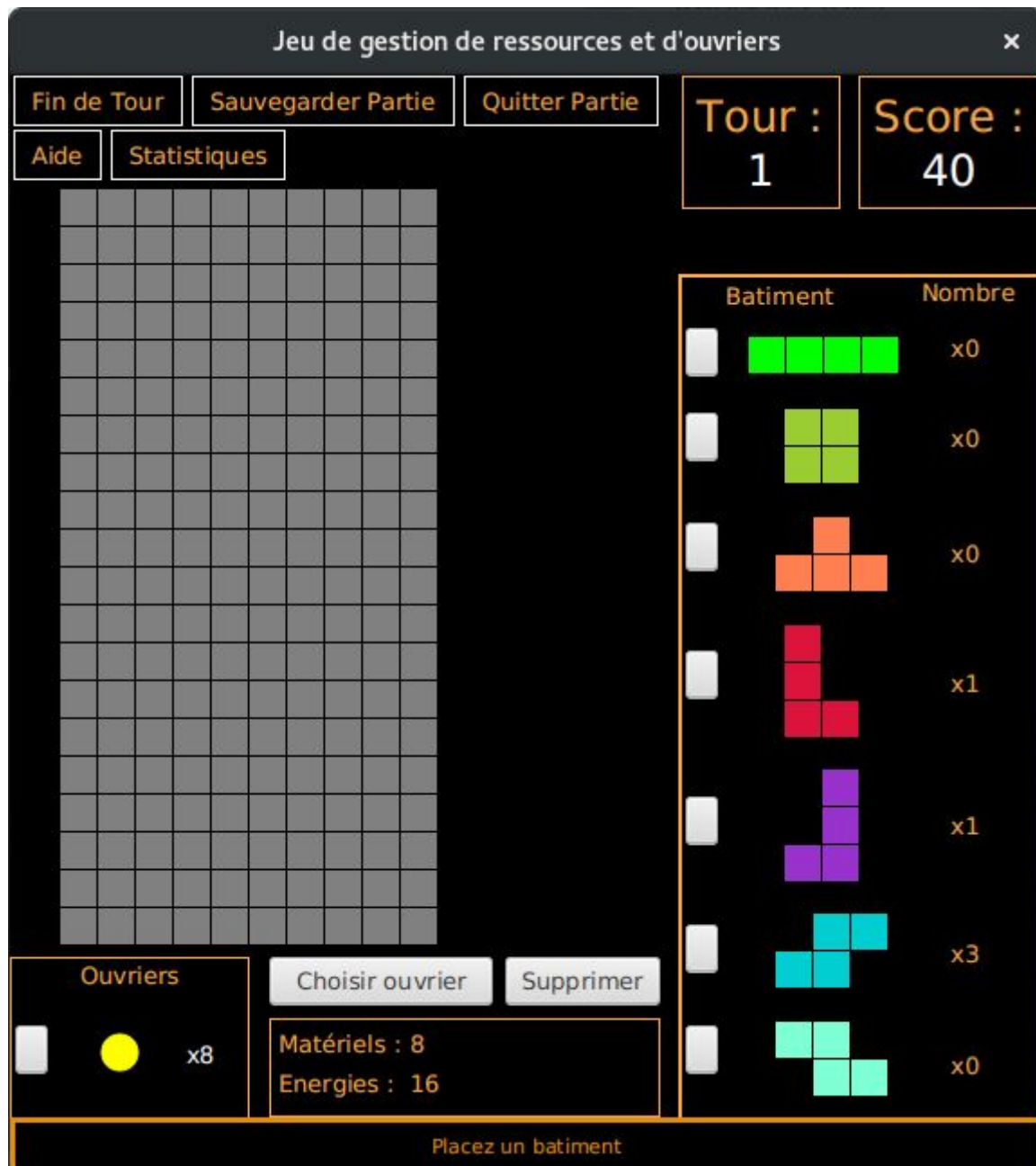
Charger Partie

Quitter

Vue de statistiques :

Statistiques ×	
Tour	1
Score	40
Energies	16
Matériaux	8
Nombre de batiments sur le Plateau	0
Nombre d'ouvriers sur le plateau	0
Nombre d'ouvriers à l'extérieur	8
Nombre de batiments dans la pile	130

Vue du jeu (fenêtre principale) :



II- Conception et spécification

Les changements dans la spécification que j'ai apporté par rapport à la spécification du départ sont :

- La scène principale avec l'affichage des boutons, en effet je n'avais prévu que trois boutons mais, j'ai envisagé l'utilisation de boutons pour placer des bâtiments et des ouvriers.
- De plus j'ai ajouté aussi des boutons afin d'afficher des scènes qui ont pour but d'informer le joueur sur la partie (statistiques), ou bien de l'informer sur le jeu, pour aider celui à comprendre la mécanique du jeu.
- J'ai dû abandonner le label qui affichait le numéro du joueur qui jouait car je devais me contenter des fonctionnalités minimales (problème de travail avec mon binôme)
- Dans les fonctionnalités minimales j'ai tout fait mis à part la sauvegarde par faute de temps que je n'ai pas pu finir
- Dans mes prédictions des fonctionnalités possibles, j'en ai fait aucune comme expliqué avant
- En vu du code j'ai beaucoup modifié la structure du projet, en prenant en compte des problématiques de visibilité etc...

Les aménagements effectués par rapport à cette spécification sont :

- Dans la spécification j'avais prévu d'utiliser une Classe Partie, pour y implémenter la logique du jeu, cette classe a été remplacé par une classe GameLogic, mais j'avais pour but d'effectuer les mêmes opérations. L'idée d'utiliser une classe pour chaque bâtiments, qui dérivait d'une classe abstraite a été conservé .
- J'ai du ajouter plusieurs classes contrôleur de fichiers fxml, notamment la GameController qui gère globalement toute les interactions de la vue principale du jeu. Je n'ai pas utilisé de classe Action, tout ce que j'avais prévu de faire dans cette classe s'effectue dans la classe GameController.

J'ai gardé l'idée de la classe Ouvrier afin de gérer les déplacements d'ouvriers.

Avant de connaître la technologie des fichiers fxm1 je pensais tout implémenter en java Fx, mais après avoir m'être initié à cette technologie, cela a dû réduire considérablement la taille des classes de vue si j'utilisais uniquement java Fx.

Pour les opérations de rotations et déplacements de bâtiments j'ai utilisé une classe Case, qui permet de localiser le bâtiment sur plateau. Pour les déplacements d'ouvriers j'ai utilisé une classe PositionOuvrier pour localiser les ouvriers quand on les déplacent et les placent.

Pour le plateau de jeu j'ai gardé l'optique de la classe Plateau, dans lequel on effectue des actions sur un tableau deux dimensions de String. J'ai ajouté une classe Cellule qui permet le dessin du plateau à chaque actions effectués. Cette classe permet aussi le dessin des bâtiments pour les visualiser dans l'aide et dans la main du joueur.

Les rôles des classes :

- package buildings.data :
 - Batiment : classe mère
 - Case : classe permettant de constituer chaque bâtiments à l'aide de tableaux de deux dimensions d'instance de cette classe.
 - Toutes les classes qui dérivent de Batiment sont les classes de bâtiments spécifiques
- package gameLogic :

La classe GameLogique se charge de calculer la globalité des opérations du programme, et se charge de stocker les listes de Batiment, et les listes Ouvrier. Cette classe se charge aussi de faire des opérations dans les listes.
- package jeu de gestion des ressources et des ouvriers.scene :

- Toutes les classes de vue sont présentent, mais aussi leur fichiers fxml assimilé aux classes de vues.
- la classe GameView, gère toutes les interactions apportés dans l'interface graphique mais aussi par défaut de structure de code, elle gère aussi certains éléments que GameLogic doit gérer.
- package Ouvrier :
 - La classe Ouvrier permet de changer les valeur des cases du tableau deux dimensions du plateau
 - La classe PositionOuvrier permet de localiser les ouvriers sur le plateau
- package PlateauDeJeu :
 - La classe plateau se charge de dessiner le plateau de jeu, d'effectuer les mis à jours du plateau après chaques actions effectués sur le plateau (que ça soit des déplacement et pivots de bâtiments, ou même déplacement et changement de position d'ouvriers grace au Batiment O).

III- Evaluation de la réalisation

1)Fonctionnalités réalisés

Dans ce projet, ce qui a été réalisé par sont toutes les fonctionnalités minimales à faire sauf la sauvegarde et le chargement.

C'est-à dire que les matériaux, énergies, ouvriers, bâtiments sont tous géré. Les ouvriers et bâtiments permutent bien entre les listes. Chaque actions de bâtiments sont réalisés le nombre de fois que d'ouvriers est présent sur le bâtiment.

Si les ressources ne le permettent pas l'ouvrier ou les ouvriers présents sur le bâtiment, sont expulsés dans la main.

Si il n'y a plus de bâtiments dans la main, au tour suivant un bâtiment est généré automatiquement dans la main. La vue de statistiques est présente.

Les bâtiments sont pivotables et déplaçable.

Chaque tour le jeu respecte le nombre d'actions réalisables. Le jeu est entièrement développé avec Java, il n'y a pas d'images ajoutés.

L'affichage du score se met à jour à la fin de chaque tours.

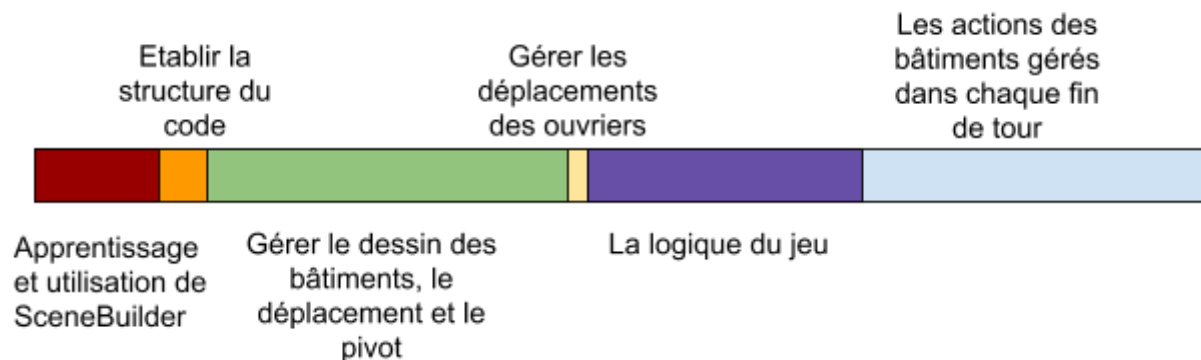
2) Techniques d'utilisations regretté

Les techniques que j'ai regretté d'utilisé sont :

- Pour commencer le mode de jeu, c'est à dire que j'utilise le clavier pour poser des bâtiments ou des ouvriers, ça aurait été plus ergonomique d'utiliser la souris pour effectuer ces actions, parce que dans ce cas, le joueur utilise la souris et le clavier pour jouer ce qui est vite agaçant de permuter entre la souris et le clavier.
- Ensuite la méthode dont je dessine le plateau, je dessine des cases en fonction des valeurs du tableau deux dimensions du plateau. Alors que je pouvais directement utiliser des Group pour les placer avec la souris.

- La structure de mon code j'aimerais la revoir du début parce qu'il y a des des fonctions qui se ressemblent et quand je veux factoriser il y a des conflits dans le code.
- Je regrette aussi d'avoir créé que deux branches sur Git alors que le plus performant est de créer une branche pour chaque fonctionnalités.

IV- Organisation du travail



Cette organisation est à titre personnel. Les tâches ont été réalisées uniquement par moi. Je me suis organisé de façon m'avancer le plus possible en restant après les cours à l'IUT, le dépôt Git a servi juste à récupérer le travail que j'effectuais à l'IUT sur mon PC ou l'inverse.

V- Perspectives

Si j'aurais pu avoir les ressources nécessaires, j'aurais d'abord voulu terminer toutes les fonctionnalités minimales en développant un code propre.

Et aussi il serait intéressant de produire une interface graphique qui a un bon design mais aussi d'avoir un jeu avec une bonne ergonomie d'utilisation.

Mais sinon dans le cas d'une équipe performante, j'aurais bien aimé à séparer les tâches à réaliser avec chaque membre de l'équipe. De plus investir dans l'organisation du projet, car c'est primordiale qu'un projet soit bien organisé sinon il ne peut pas avancer rapidement, et cela peut poser problème dans les critères de temps, pour preuve nous avons mon cas.