



ÉTABLISSEMENT : UNIVERSITÉ DE FRANCHE-COMTÉ,
FACULTÉ DE SCIENCES ET TECHNIQUES

Rapport - L1ST – APOO – Projet TP 2023

ENCADREMENT : DORINE TABARY

Jeu de blocage avec polyominos

Mai 2023

Dédicaces

Je dédie ce travail à :

À ma grand mère, qui nous à quitter ce début de mois et qui a su me montrer son enthousiasme tout le long de mes études.

À mon encadrant, pour son précieux accompagnement, ses conseils avisés et sa disponibilité tout au long de ce projet.

À mes amis, pour leur soutien constant.

À toutes les personnes qui ont contribué à la réalisation de ce projet, pour leur collaboration et leur aide précieuse.

Remerciements

Durant le projet, j'ai eu la chance d'avoir l'aide de Kannelle Goudjo-Guillemot, mon amie, et je tiens à la remercier pour son temps dédié et sa contribution dans l'écriture de ce rapport.

Table des matières

Remerciements	3
Introduction	6
1 Analyse technique du sujet proposé	7
1.1 Présentation rapide	7
1.2 Définitions	8
Organisation	9
1.3 Ressources humaines	9
1.3.1 Fréquence de travail	9
1.3.2 Communication pour le travail	9
1.4 Ressources logicielles	10
1.4.1 Pour le projet	10
1.4.2 Écriture du code	10
1.4.3 Écriture du rapport	10
2 Description des classes	11
2.1 Main	11
2.2 Jeu	11
2.3 Scene	11
2.4 Menu	11
2.5 Niveau	11
2.6 Plateau	12
2.7 Cellule	12
2.8 Joueur	12
2.9 Humain	12
2.10 Machine	12
2.11 Polyomino	13
2.12 Domino, Triomino, Tetromino	13
2.13 Saisies	13

3	Fonctionnalités implémentées	14
3.1	Initialisation	14
3.2	Visuels	14
3.3	Placement d'un polyomino	14
3.4	Fin de partie	15
3.5	Stratégie de la machine	15
	Conclusion	16

Introduction

Ce rapport présente le travail réalisé dans notre projet du module d'algorithmique et programmation orientée objet (APOO).

Dans le cadre du deuxième semestre de licence en sciences fondamentales de l'université de Bourgogne Franche Comté, nous sommes amenés à acquérir plusieurs compétences de développement orienté objet, telles que les classes, les objets, les constructeurs, les références, l'encapsulation, la statique ou l'instance, ainsi que la notion d'héritage.

Ce projet nous donne l'occasion de manipuler ces compétences de développement.

Le thème du sujet consiste à l'implémentation d'un jeu de blocage sur grille, avec des pièces de type polyominos. Sa réalisation doit être effectuée d'ici le 19 mai 2023, sous forme d'un ensemble de fichiers en Java (un langage de programmation orienté objet), accompagné d'un rapport, que vous êtes en train de lire.

Il nous est donc donné environ deux semaines pour réaliser, en binôme, une version fonctionnelle (même incomplète) du jeu, ainsi qu'un rapport décrivant le jeu, ses contraintes, les solutions et les choix techniques envisagés dans sa réalisation.

Afin de rendre compte de notre travail, nous allons vous décrire les axes les plus importants.

Chapitre 1

Analyse technique du sujet proposé

1.1 Présentation rapide

Le jeu oppose deux personnes, ici humain contre machine, sur un plateau de forme rectangulaire. Les joueurs ont des pièces, des polyominos qu'ils placent chacun leur tour. La fin du jeu est décidée par un blocage, c'est-à-dire lorsque l'un des joueurs ne peut plus poser aucune de ses pièces, il est alors perdant.

Les affichages se font sur la console. Le mode de jeu demandé est humain contre machine. La stratégie de jeu de la machine est libre. Au niveau du code, il nous est imposé de travailler en programmation orientée objet et d'utiliser l'encapsulation.

1.2 Définitions

Programmation orientée objet : Un paradigme de programmation qui organise les programmes autour d'objets, qui représentent des entités ayant des caractéristiques (attributs) et des comportements (méthodes). L'approche orientée objet favorise la modularité, la réutilisabilité et la flexibilité du code.

Par exemple on va définir une chaise comme ayant un nombre de pieds des matériaux un coût une résistance, et d'autre part une personne avec son nom sa taille son poids, puis on définit leurs comportements : la personne pourra s'asseoir sur la chaise, la déplacer mais pas la chaise sur la personne évidemment.

Encapsulation : Un principe de la programmation orientée objet qui consiste à regrouper les données (attributs) et les méthodes qui les manipulent dans une même entité appelée objet. L'encapsulation permet de protéger les données internes d'un objet en les rendant accessibles uniquement par des méthodes spécifiques. En reprenant l'exemple de la classe Personne décrite ci-dessus, on peut imaginer que celle-ci possède un ensemble de meubles, dont des chaises, on dit alors qu'il y a encapsulation.

Polyominos : Un assemblage de carrés, collés les uns aux autres par leurs côtés, mais ne pouvant pas être assemblés par un coin. Les polyominos existent dans différentes tailles, les plus courants étant les tétrominos (pièces utilisées dans Tetris) et les dominos. Dans ce jeu, nous utiliserons des polyominos composés de 2, 3 ou 4 carrés.

Organisation

Ce chapitre présente l'organisation mise en place pour mener à bien le projet du module d'algorithmique et programmation orientée objet (APOO).

1.3 Ressources humaines

1.3.1 Fréquence de travail

J'ai travaillé seul sur l'ensemble du projet. La fréquence de travail était variable en raison de contraintes personnelles, administratives et logistiques. Cependant, ces interruptions dans le travail étaient prévues et anticipées, ce qui m'a permis de trouver des solutions adaptées. J'ai choisi de travailler sur mon temps libre, dès que possible, pour avancer sur le projet.

Lorsque j'avais des séances de travail de courte durée, j'ai privilégié des tâches qui demandaient peu de ressources, comme tester les fonctions pour m'assurer qu'elles remplissaient correctement leur tâche. En revanche, lors de séances plus longues, j'ai consacré du temps à des tâches qui demandaient plus de ressources, telles que réfléchir aux différentes possibilités pour certaines fonctionnalités et prendre des décisions concernant leur implémentation.

1.3.2 Communication pour le travail

Étant seul sur le projet, la communication était très simple, car tout se déroulait dans ma tête. Je n'avais pas besoin d'échanger avec d'autres personnes pour prendre des décisions ou coordonner des actions. Cela m'a permis d'avancer de manière autonome et de suivre ma propre méthodologie de travail.

1.4 Ressources logicielles

1.4.1 Pour le projet

J'ai utilisé GitHub pour créer un dépôt distant afin de mieux suivre et organiser le projet. Cela m'a permis de sauvegarder mon travail de manière sécurisée, de suivre les modifications apportées aux fichiers et de collaborer avec moi-même en utilisant différentes branches pour développer des fonctionnalités spécifiques.

1.4.2 Écriture du code

J'ai utilisé l'éditeur Scite pour écrire le code du projet. J'ai travaillé avec Java 1.8, comme demandé dans les spécifications du projet.

1.4.3 Écriture du rapport

Pour l'écriture du rapport, j'ai utilisé TexMaker, un éditeur LaTeX convivial. LaTeX m'a permis de produire un rapport de qualité professionnelle avec une mise en page soignée, une gestion des références, des tableaux et des images, ainsi qu'une génération automatique de la table des matières et des numéros de page. Cela m'a également facilité la rédaction des sections structurées du rapport.

Chapitre 2

Description des classes

2.1 Main

La classe Main contient la méthode principale qui lance l'exécution du jeu.

2.2 Jeu

La classe Jeu représente le jeu lui-même et gère des scènes, et l'arrêt du programme.

2.3 Scene

La classe Scene est simplement représentative des différentes scènes du jeu, le Menu et le Niveau en lui-même.

2.4 Menu

La classe Menu hérite de Scene, gère l'affichage et la gestion des menus du jeu, tels que le menu principal et le menu des options. Avant de lancer une nouvelle partie (Niveau).

2.5 Niveau

La classe Niveau hérite de Scene, elle représente la scène du jeu en lui même, c'est la qu'on va demander la saisie tour à tour des polyominos.

2.6 Plateau

La classe Plateau représente le plateau de jeu, qui est une grille de cellules où les pièces sont placées.

2.7 Cellule

La classe Cellule représente une cellule individuelle sur le plateau, pouvant contenir une pièce, avec un motif donné.

2.8 Joueur

La classe Joueur est une classe abstraite qui représente un joueur dans le jeu, avec des méthodes pour placer des pièces et prendre des décisions.

2.9 Humain

La classe Humain hérite de Joueur qui implémente les actions d'un joueur humain. La différence avec la machine est principalement dans le placement d'un polyomino qui se fait par saisie pour un humain.

2.10 Machine

La classe Machine hérite de Joueur qui implémente les actions d'un joueur contrôlé par l'ordinateur, avec une stratégie de jeu spécifique. (Une partie y sera dédiée plus loin dans le rapport.)

2.11 Polyomino

La classe Polyomino est une classe abstraite qui représente un polyomino, avec des méthodes pour le positionner sur le plateau ou le tourner par exemples.

2.12 Domino, Triomino, Tetromino

Les classes Domino, Triomino et Tetromino héritent de Polyomino qui représentent différents types de polyominos, respectivement composés de 2, 3 et 4 carrés.

2.13 Saisies

La classe Saisies gère les saisies utilisateur, telles que la saisie des coordonnées pour placer une pièce.

Chapitre

3

Fonctionnalités implémentées

3.1 Initialisation

La majeure partie de l'initialisation se passe dans le menu. C'est à dire au début du lancement du jeu. Le menu principal propose de lancer la partie de quitter le programme ou donner quelques informations comme les règles de jeu. Lorsque l'utilisateur lance une nouvelle partie un nouveau menu apparait, en fait dans cette seconde partie on va définir le plateau en premier, puis les joueurs, qu'ils soient humain ou machine. Donc différents plateau sont proposés, puis l'utilisateur saisi le nombre de joueurs humains.

3.2 Visuels

Il y a plusieurs affichages différents : celui des menus, du plateau du jeu de polyominos d'un joueur. Dans un premier temps il a été décidé de développer un affichage très simple, des numéros pour les lignes, des lettres de A à Z pour les colonnes, un point représente une cellule libre, sinon il y a un caractère en fonction du joueur, par défaut 'X'. Le même style a été conservé pour l'affichage des polyominos d'un joueur.

3.3 Placement d'un polyomino

Pour un plateau donné il prend en compte : une pièce, prise dans un certain sens, et une position (ligne, colonne). Dès lors un algorithme naïf pour vérifier le positionnement et de comparer le plateau et la pièce à sa position donnée. Il serait alors intéressant de chercher des mécanismes pour simplifier la complexité d'un tel algorithme.

3.4 Fin de partie

Pour un plateau donné il prend en compte : l'ensemble des pièces d'un joueur.

Dès lors une manière naïve de procéder à la vérification d'un blocage est de vérifier que pour chaque pièce il n'y a plus de place libre. Pour le placement et le blocage, nous chercherons des méthodes plus efficaces de vérification.

Concrètement des idées telles que le fait que si un domino ne peut être placé nul part, alors toute pièce plus grande ne peut pas l'être (pour des pièces de même taille il est donc nécessaire de prendre en compte la forme de celle-ci).

Et nous pourrions également parcourir seule les zones libres du plateau, et ainsi plus la partie avancera plus le temps de calcul diminuera.

3.5 Stratégie de la machine

Pour une facilité d'implémentation il sera choisit de faire une stratégie très naïve pour la machine, en fait celle-ci utilise très exactement le même algorithme que celui qui vérifie la fin de partie. Une telle stratégie n'est clairement pas intéressante pour le joueur comme le développeur. En revanche il est possible de développer une véritable stratégie se basant sur

l'apprentissage. En enregistrant les parties précédemment jouées puis en attribuant aux coups un score il est possible pour la machine d'apprendre à jouer. Dès lors en théorie elle devrait s'améliorer au fur et à mesure des parties d'un joueur en s'inspirant des meilleurs coups déjà produit. De plus un tel processus permettrait de générer un niveau de difficulté personnalisé à chaque joueur humain, car en effet celui-ci donne un nom en début de partie, la difficulté serai alors adapté au niveau de stratégie du joueur.

Conclusion

En conclusion, ce projet du module d'algorithmique et programmation orientée objet (APOO) a été une expérience enrichissante et stimulante. J'ai pu mettre en pratique mes connaissances en développement orienté objet et acquérir de nouvelles compétences tout au long de sa réalisation.

L'objectif de ce projet était de concevoir et d'implémenter un jeu de blocage sur grille avec des pièces de type polyominos. J'ai réussi à créer les différentes classes nécessaires pour représenter les éléments du jeu, tels que le plateau, les joueurs, les polyominos, et à mettre en place les fonctionnalités essentielles, comme le placement des pièces, la vérification de la fin de partie et la stratégie de jeu de la machine.

L'organisation du projet a été adaptée à mes contraintes personnelles, ce qui m'a permis de travailler sur le projet en fonction de mes disponibilités. J'ai privilégié des tâches qui demandaient peu de ressources lors de séances de travail courtes et j'ai consacré des séances plus longues à des tâches plus complexes. La communication était simplifiée puisque j'ai travaillé seul sur le projet.

J'ai utilisé des ressources logicielles telles que GitHub (<https://github.com/UnExtraterrestres>) pour la gestion du code source, Scite pour l'écriture du code en Java, et TexMaker pour la rédaction du rapport en utilisant LaTeX. Ces outils ont facilité la gestion du projet, la collaboration avec moi-même et la production d'un rapport de qualité.

En conclusion, ce projet m'a permis de consolider mes compétences en programmation orientée objet, de mettre en pratique des concepts clés tels que l'encapsulation, l'héritage et la modularité, et de développer ma logique algorithmique. J'ai également apprécié la flexibilité offerte par le projet pour m'adapter à mes contraintes personnelles.

Ce projet constitue une étape importante dans mon apprentissage de l'algorithmique et de la programmation orientée objet, et il me motive à continuer à explorer de nouveaux défis et à améliorer mes compétences en développement logiciel, tel que le LaTeX auquel je n'avais jamais touché.