

Table des matières

Introduction

Dans le cadre de notre formation, nous sommes amenés à réaliser un jeu réaliser dans le langage JAVA appelé TetraWord. Ce jeu est un mélange entre Tetris et Boggle. En effet, chaque cube comprend une lettre, ce qui permet ainsi de former des mots.

Dans une première partie, nous allons rappeler les règles du jeu. Dans une seconde partie, nous allons parler de la conception et développement du jeu. Enfin, dans une dernière partie, nous allons présenter l’extension et le résultat obtenu.

Règle du jeu

# Tétris

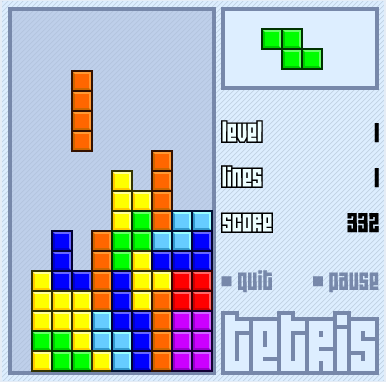


Figure 1 : Tetris

Tetris est un puzzle game. En effet, le but du jeu est de former des lignes à l’aide des pièces de couleurs et des formes qui tombent. Il est possible de tourner ces formes de 90 degrés la forme. Ces dernières sont formées de quatre pièces. Lorsqu’une ligne est complète, elle disparait. Le jeu ce termine lorsque l’écran est remplit.

# Boggle



Figure 2 : Boggle

Le but du Boggle est de trouver un maximum de mot dans une grille de lettre, pouvant être formés à partir de lettre adjacentes (horizontalement, verticalement ou en diagonale) dans un temps à partie.

# TetraWord

## Déroulement du jeu

La partie commence comme un jeu de Tetris classique. Cependant, chaque pièce de la forme contient une lettre. Une fois qu’une ligne est remplie, le mode anagramme s’active. Le joueur doit trouver un mot dans la ligne qui comporte au moins … % du meilleur anagramme. Si cette condition est remplie, la ligne est détruite. Tout comme Tetris, lorsque la grille du joueur est remplie, ce dernier a perdu.

## Les bonus

Le jeu contient également des bonus qu’il est possible d’activer :

* Le bonus **worddle** : Ce dernier active le mode worddle. Ainsi dans toute la grille, le joueur doit trouver dans un temps à partie, le maximum de mot formés à partir de lettres adjacentes. Tout comme pour le jeu Boggle.
* Le bonus **échange** : Ce bonus permet d’échanger les plateaux des deux joueurs.
* Le bonus **retournement** : Le plateau du joueur d’adverse se retourne. Les formes vont donc de bas en haut.
* Le bonus **malus** qui basse les points du joueur adverse
* Le **bonus** qui ajoute des points au score

Conception du jeu

Nous avons départagé le projet en plusieurs étapes :

* L’étape de **conception** : elle consiste à la réflexion sur l’architecture, les classes.
* L’étape de **design** : elle consiste à la réflexion sur l’ergonomie et la réalisation du design pour le jeu.
* L’étape de **développement** du jeu : elle consiste en la réalisation du jeu. Donc coder tous le jeu en lui-même.

# Conception

Avant de commencer le développement du jeu en lui-même. Nous avons choisi d’avoir une étape de conception. De cette façon, nous partons tous de la même base. Ainsi, nous savons tous, les fonctionnalités et variables utilisés et implémentés de chaque classe. De cette façon, il est plus facile de coder en simultané puisque nous connaissons les liens entre chaque classe. De plus, cette étape permet de gagner du temps au moment du développement. Pour cela, nous avons d’abord réalisé un user-case avant de faire une UML.

## User-case

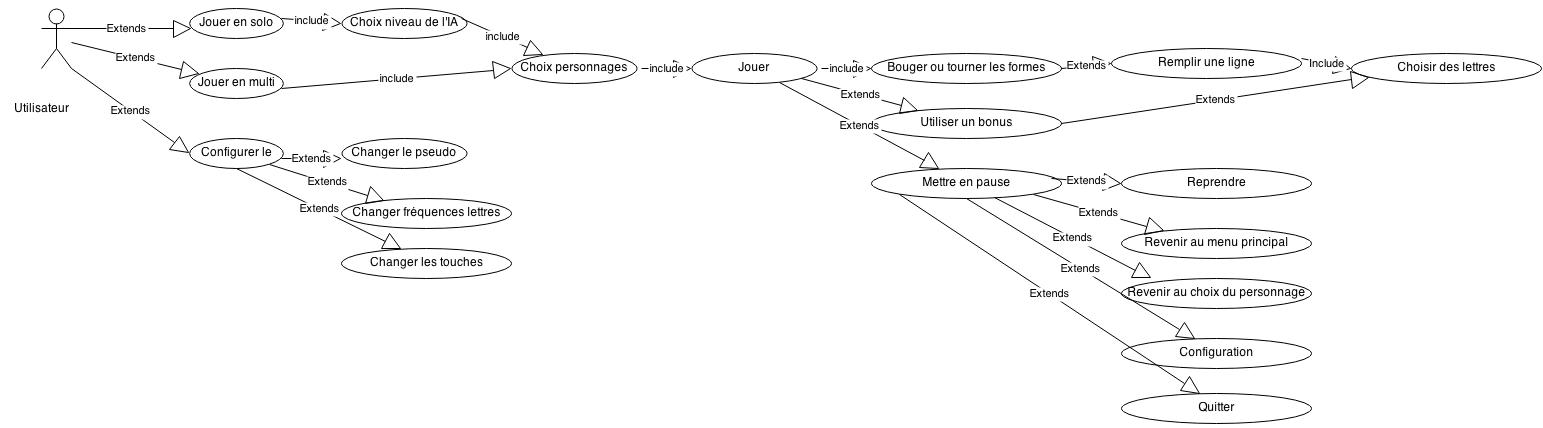


Figure 3 : User-case Utilisateur

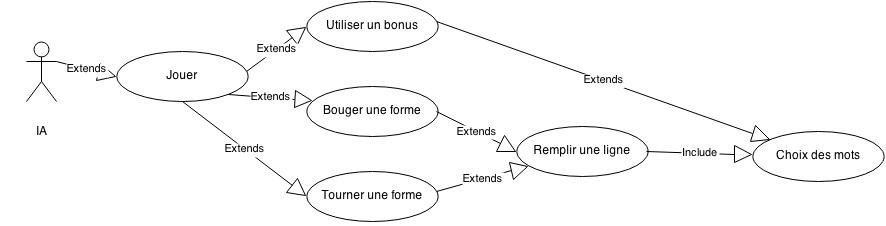


Figure 4 : User-case IA

## UML



Figure 5 : UML simple

# Ergonomie et design

En ce qui concerne l’ergonomie et le design, nous avons principalement utilisés l’user-case pour définir les différents types d’affichage à réaliser. Nous avons également fait des recherches sur l’existant afin garder les points forts au niveau ergonomie.

Nous avons choisi de refaire le design de zéro, c’est-à-dire sans reprendre des éléments existants. De cette façon, nous pouvons affirmer que nous avons entièrement réalisé le jeu. Nous avons donc fait différents test avant de se mettre d’accord sur un design particulier.

Nous sommes ainsi arrivés au résultat suivant :

# Développement

## Structure

On peut remarquer que la structure a quelque peu changé. En effet, nous n’avions pas pris en compte SWING.

## Les problèmes rencontrées

## Les solutions apportées

Extensions

Nous avons apportés quelques améliorations au programme afin d’augmenter le côté fun du jeu. Ainsi, nous avons ajouté des personnages au jeu. Au début de la partie, le joueur doit choisir son avatar (un panda (don’t say no to panda), un robot, un ninja ou un démon). Chaque avatar est caractérisé par un terrain, et par une attaque spéciale.

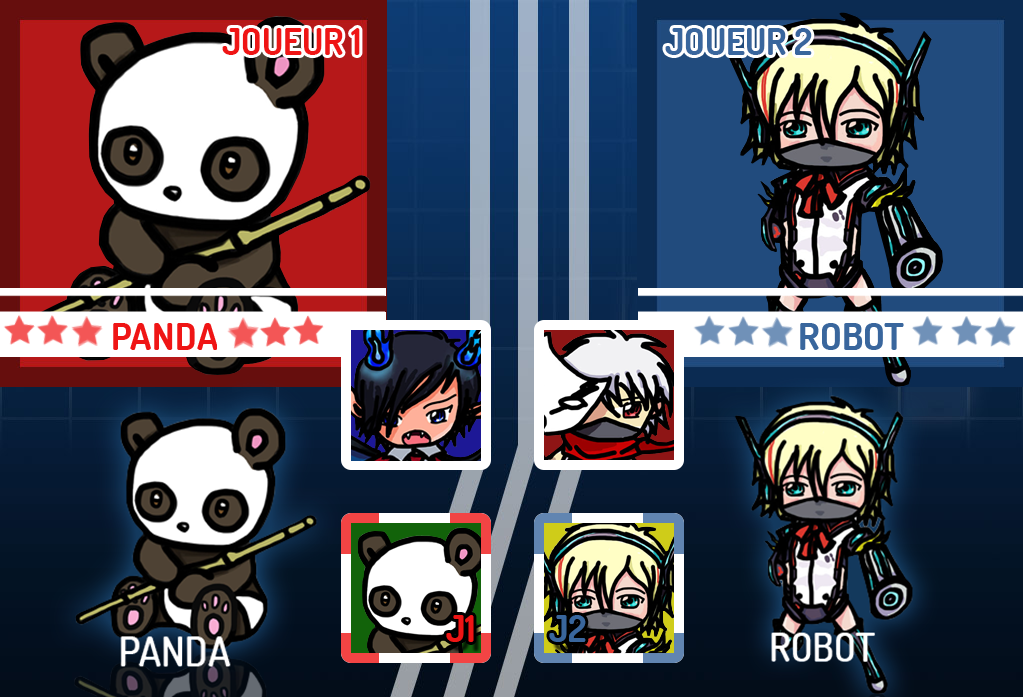


Figure 6 : Partie choix du personnage

Lorsque la partie est lancée, le choix du terrain se fait donc en fonction des personnages choisi. Ainsi, une fonction fait le choix aléatoire entre un terrain par défaut et les terrains liés aux personnages choisis.

Une fois dans le jeu, à chaque fois que le joueur gagne des points, un jauge se rempli. Il s’agit de la jauge spéciale du personnage. Lorsqu’elle est remplie, le joueur peut activer l’attaque spéciale du personnage. Cette attaque peut correspondre à un bonus. Lors de l’activation de l’attaque, une bande au milieu de l’écran apparait avec la tête du personnage en grand avant l’activation de son attaque spéciale. Nous n’avons malheureusement pas eu le temps d’implémenter cette partie.

Le résultat

Conclusion

Ce jeu fut très intéressant à concevoir et à développer. En effet, cela nous a permis de développer nos compétences en JAVA mais aussi dans la conception, réflexion avant le développement du jeu pour que la structure du jeu soit la plus optimale possible. Ainsi nous avons pu avoir une réflexion structurés tout au long du processus. Ce jeu constitue un plus dans notre portfolio.

Il est également intéressant d’observer les différents points sur lesquelles nous avons eu des problèmes, notamment sur l’UML, car en remarquant nos erreurs nos avons moins de chance de les reproduire.

De plus, ce projet nous a également permis de mettre en application les acquis du cours et expérimenter nos compétences.

Le langage JAVA est un langage structuré donc l’ensemble des classes fournis permet de créer simplement et rapidement des applications.