|  |  |  |
| --- | --- | --- |
|  | |  |
| PROJET TUTEURE- PYLOS | | |
| L3 DANT01/12/2020Sébastien TIXEUIL | G.O.W.M  Game On With Me | |

# PARTIE 1

## Équipe :

|  |  |  |
| --- | --- | --- |
| **Membres** | **Rôles** | **Expérience (jeux programmés en JAVA)** |
| Sinem ORUC **(**Cheffe de projet) | Coder les classes essentielles au fonctionnement du jeu | Avatar/Créatures |
| Chifaa WEHBE | Développer la partie concernant l’intelligence artificielle | TETRIS, Avatar/Créatures |
| Nahidath GASSA | Intégrer les menus dans l’espace de jeu pour que l'utilisateur puisse déclencher des événements et réagir à ce qui se passe dans la fenêtre | Avatar/Créatures, Constructeur de ligne de métro |
| Ugo MOUTYMBO | Gérer les autres composants graphiques et la bibliothèque 3D | Avatar/Créatures, Constructeur de ligne de métro |

**Avatar/Créatures :** Nous avons codé un jeu en utilisant la bibliothèque graphique Swing. Deux joueurs en local représentés par des avatars peuvent se déplacer sur la grille de jeu pour chercher de la nourriture. Les deux avatars s’affrontent, le gagnant est celui qui a le plus d’énergie.

**Tetris :** Développementdu jeu TETRIS en utilisant la bibliothèque graphique Swing. Le joueur peut changer la disposition des tétrominos grâce à son clavier.

**Constructeur de lignes de métro :** Application permettant de créer des stations de métro, puis de les relier entre elles soit manuellement, soit automatique en sélectionnant deux stations correspondant au terminus de la ligne. Utilisation des bibliothèques graphiques Swing et imageio.

## Choix de travail :

|  |  |
| --- | --- |
| **Options** | **Raisons** |
| Jeu contre un ordinateur intelligent | Approfondir la connaissance des règles du jeu |
| Améliorer et développer la rapidité, la précision et les efforts du joueur |
| Ne pas nécessiter 2 joueurs |
| Interface graphique 3D | Rendre le jeu plus agréable à jouer (presque nécessaire) |
| S'affranchir des contraintes réelles et physiques pour une illustration optimale de notre jeu |

## Architecture technique :

|  |  |
| --- | --- |
| **Langage** | **Raisons** |
| JAVA | C’est un langage maîtrisé par tous les membres de l'équipe. |
| Ce langage est apprécié pour son côté fonctionnel et son efficacité, JAVA est souvent utilisé pour coder des jeux de ce type. |
| Bibliothèques graphiques disponibles gratuitement. |
| Rendre le code plus structuré et maintenable par la suite. |
| JAVA dispose d’une bonne documentation et d’exemples trouvables sur internet. |

## Communication :

En raison de la situation sanitaire actuelle, les membres de l’équipe de développement de G.O.W.M. ne peuvent pas se réunir en présentiel. De ce fait, nous avons mis en place un répertoire **GitHub** afin de mettre en commun le code du projet. De plus, des réunions **Discord** sont organisées au besoin pour pouvoir discuter de l’avancement du projet.

# PARTIE 2

## Interfaces graphiques :

Pour gérer les interfaces graphiques (menu, boutons, fenêtre, …), nous avons décidé d’utiliser le framework jMonkey Engine. jMonkey Engine est un moteur de jeu libre spécialement conçu pour le développement 3D moderne.

### Interface 2D :

A l’exécution du programme, une fenêtre s’ouvre avec le menu ayant trois options dont « jouer à 2 », « jouer contre l’ordinateur » et « règles du jeu ».

« **jouer à 2** » permet à deux joueurs utilisant le même ordinateur de jouer une partie de pylos ensemble.

« **jouer contre l’ordinateur** » permet à un joueur de lancer une partie de pylos contre l’ordinateur.

« **règles du jeu** » affiche les règles et le fonctionnement de pylos.

### Interface 3D :

Au lancement d’une partie de jeu, une nouvelle fenêtre s’ouvre en affichant le plateau de jeu entouré des billes blanches et noires des deux joueurs.

Grâce aux fonctionnalités de jMonkey Engine, le joueur peut voir tout autour du plateau à l’aide de la souris et du clavier.

A chaque tour de jeu, un joueur clique sur une bille (du plateau ou de la bordure) afin de la sélectionner puis clique une deuxième fois sur le plateau pour la positionner.

## Intelligence Artificielle :

Pour optimiser son algorithme, l’IA déduit à chaque le déplacement le plus convenable pour empêcher le progrès de son adversaire. Voici l’ordre de priorités des actions de l’ordinateur :

1. Pour garantir sa victoire l’IA doit économiser le plus de billes possibles. Pour cela l’IA cherche à compléter un carré de sa couleur (par défaut blanc) et récupérer deux billes.

2. Comme les règles du jeu autorisent un joueur, ayant formé un carré avec sa couleur, à retirer deux billes de ce même carré, l’IA empêche la formation des carrés ennemis.

3. Il est préférable de déplacer une bille déjà présente sur le plateau vers un étage supérieur que de prendre une nouvelle bille. Ce qui permet de passer un tour sans utiliser de bille.

4. Pour augmenter ses chances de former un carré, l’IA place ses billes en priorité au milieu du plateau (valable pour le premier et le deuxième étage).

## Contraintes techniques :

Pour faire fonctionner le jeu, il faut préalablement installer Java 8 sur sa machine. De plus, il est primordial d’installer le framework jMonkey Engine 3 (disponible sur les différents systèmes d’exploitation).

Le projet Pylos de l’entreprise G.O.W.M se trouvera sur GitHub, dont nous communiquerons un lien ultérieurement, avec un guide d’installation.

A l’ouverture de jMonkey Engine, il faut importer le fichier .zip du projet, et l’exécuter.