

Projet tuteuré Semestre 2



Jeu Pac-Man en Java

Sommaire

1	Présentation du jeu.....	4
1.1	Introduction.....	4
1.2	Règles.....	4
1.3	Améliorations.....	4
2	Objectifs de réalisation.....	5
2.1	Fonctionnalités.....	5
2.1.1	Fonctionnalités du menu.....	5
2.1.2	Fonctionnalités du jeu.....	5
2.2	Aspect graphique souhaité.....	5
2.2.1	Aspect graphique du menu.....	5
2.2.2	Aspect graphique du jeu.....	5
2.3	Contraintes et problèmes éventuels.....	5
2.3.1	Ressources graphiques.....	5
2.3.2	Taille de fenêtres.....	6
2.3.3	Génération procédurale de map.....	6
3	Démarche et organisation.....	6
3.1	Planning.....	6
3.1.1	Jalonnement du projet en étapes.....	6
3.1.2	Diagramme de Gantt.....	6
3.2	Répartition des tâches.....	7
3.3	Outils.....	7
3.3.1	Gantt.....	7
3.3.2	Github.....	7
3.3.3	Google Drive.....	7
3.3.4	Discord.....	7
3.4	Modalités de réunion.....	8

1 Présentation du jeu

1.1 Introduction

Pac-Craft est un jeu dérivé du célèbre Pac-Man, consistant à contrôler un petit personnage dans un labyrinthe pour y récolter un maximum de pièces tout en échappant à des monstres. Nous voulons transformer l'aspect graphique du jeu en s'inspirant du célèbre Minecraft mais en gardant une vue du dessus (2D).



1.2 Règles

But : Ramasser toutes les pièces en parcourant le labyrinthe sans toucher les monstres.

Les monstres se déplacent en direction du personnage. Si les monstres touchent le personnage, celui-ci perd une vie.

La partie est terminée quand le personnage n'a plus de vie. Le joueur passe au niveau suivant lorsqu'il ramasse toutes les pièces (nouvelle map).

1.3 Améliorations

Changement des textures de la map et de l'apparence des monstres et du personnage (différents types de monstres).

Génération procédurale d'une map pour chaque nouveau niveau.

2 Objectifs de réalisation

2.1 Fonctionnalités

2.1.1 Fonctionnalités du menu

Présence d'un bouton jouer, quitter et d'une image d'arrière plan.

2.1.2 Fonctionnalités du jeu

- Déplacement du personnage (haut, bas, droite, gauche). Détection de collisions avec les murs et entres le personnage et les monstres.
- Pièces que l'on peut ramasser et compteur du nombre de pièces ramassées.
- Monstres cherchant à se rapprocher du personnage (calcul de chemin).
- Système de tilemap afin d'afficher la map, génération procédurale de niveau, (nombre de niveaux infini).
- Boutons pause et quitter.

2.2 Aspect graphique souhaité

2.2.1 Aspect graphique du menu

Similaire au menu principal de Minecraft afin de conserver l'ambiance du jeu.

2.2.2 Aspect graphique du jeu

Le terrain sera composé de blocs comme dans le jeu Minecraft (carré pixelisés). Les personnages et les monstres seront inspirés de ceux de Minecraft mais vues du dessus (2D).

2.3 Contraintes et problèmes éventuels

2.3.1 Ressources graphiques

Il sera impossible de récupérer les ressources graphiques de Pac-Man et/ou Minecraft. Il faudra donc trouver ou fabriquer les ressources graphiques nécessaires.

2.3.2 Taille de fenêtres

La gestion du redimensionnement et de la taille des fenêtres semble un défi important, nous opterons sans doute pour une fenêtre de jeu de taille modeste qui ne pourra pas être redimensionnée afin d'éviter tout problème.

2.3.3 Génération procédurale de map

La génération procédurale de map pourrait être un autre défi important, il s'agit non seulement d'avoir une map que l'on puisse parcourir de bout en bout mais il faut également qu'elle ne soit pas trop difficile pour le joueur (cul-de-sac à proscrire). Des recherches sur les théories de graphes et de création de labyrinthe pourraient s'imposer.

3 Démarche et organisation

3.1 Planning

3.1.1 Jalonnement du projet en étapes

- Création du groupe et établissement du cahier des charges du jeu du 4 février au 24 février.
- Phase de recherche et d'apprentissage vis-à-vis des technologies inconnues telles que JavaFX du 24 février au 18 mars.
- Rencontre avec notre tuteur de projet puis rédaction du rapport du 18 mars au 4 avril.
- La réalisation technique du 4 avril au 9 mai.
- Tests et corrections du 2 mai au 16 mai.
- Réalisation de la présentation orale du 9 mai au 23 mai.

3.1.2 Diagramme de Gantt

Mise en place d'un diagramme de Gantt (voir extrait ci-dessous).

Classe personnages

Sous-classes Hero+Monstres

Collisions personnages / monstres

Test unitaires

Présentation orale

3.2 Répartition des tâches

Division des effectifs en groupes de deux afin d'avancer sur plusieurs aspects du projet en parallèle :

- **Henry/Aurélien** : communication avec le tuteur, rapport, tilemap, boucle principale
- **Laurent/Thomas** : objets mouvants (personnages, monstres) et leurs collisions
- **Léo/Tristan** : UI, effets sonores, ressources graphiques, détection de collision sur la tilemap

3.3 Outils

3.3.1 Gantt

Afin de gérer et répartir les tâches dans le temps.

3.3.2 Github

Utilisé pour mettre le travail en commun et aussi versionner le projet.

3.3.3 Google Drive

Serveurs afin de stocker en ligne les divers ressources documentaires (pdf, rapports, présentations).

3.3.4 Discord

Outil de communication.

3.4 Modalités de réunion

- échanges réguliers sur discord
- appel discord toutes les fins de semaine
- réunion en présentiel tous les 1-2 semaines