

Rapport de création du jeu MagicDoom

Manon BONNOT, Baptiste CLOCHARD

10 décembre 2023



Travail réalisé dans le cadre du cours programmation pour le jeu vidéo, master 2 Gamagora.

Encadrant : M. Carlos CRISPIM.

Année scolaire 2023/2024

Table des matières

1	Introduction	2
1.1	Répartition des tâches	2
1.2	Commentaires dans le code	2
2	Mécaniques	3
3	Interface	3
3.1	Signs et feedbacks	3
3.1.1	Signs	3
3.1.2	Feedbacks	3
3.2	Les différents écrans	4
4	Animations	4
5	Problèmes rencontrés	5
6	Différences entre le préprojet et le jeu	6
7	Conclusion	6

1 Introduction

Le jeu *MagicDoom* a été réalisé dans le cadre du cours programmation pour le jeu vidéo, encadré par M. Carlos CRISPIM. Vous jouez un sorcier qui souhaite défendre ses chaudrons contre des hordes d'ennemis. Les ennemis laissent des gemmes en mourrant, qui permettent une fois récoltées et placées dans les chaudrons de lancer des sorts. Le joueur peut alors utiliser les sorts mis à sa disposition pour attaquer les vagues d'ennemis.

1.1 Répartition des tâches

Nous allons détailler ci-après les tâches accomplies par chacun des membres du groupe.

Pour Manon :

- animations du personnage et des monstres ;
- ajout des sons au jeu ;
- création et modification des assets du jeu ;
- ajout de la logique de déplacement, de collision et des chaudrons ;
- ajout de l'interface et des paramètres.

Pour Baptiste :

- code review ;
- débogage ;
- ajout de la logique de vie, de dégât et des gemmes ;
- écriture du rapport

1.2 Commentaires dans le code

Nous avons essayé d'utiliser des fonctions avec des noms explicites et d'apporter un maximum de clarté au code. Cette volonté implique que nous n'avons commenté notre code seulement aux endroits qui nous paraissaient nécessaires.

2 Mécaniques

Les différentes mécaniques implémentées ont été représentées sous forme de graphe (figure 1)

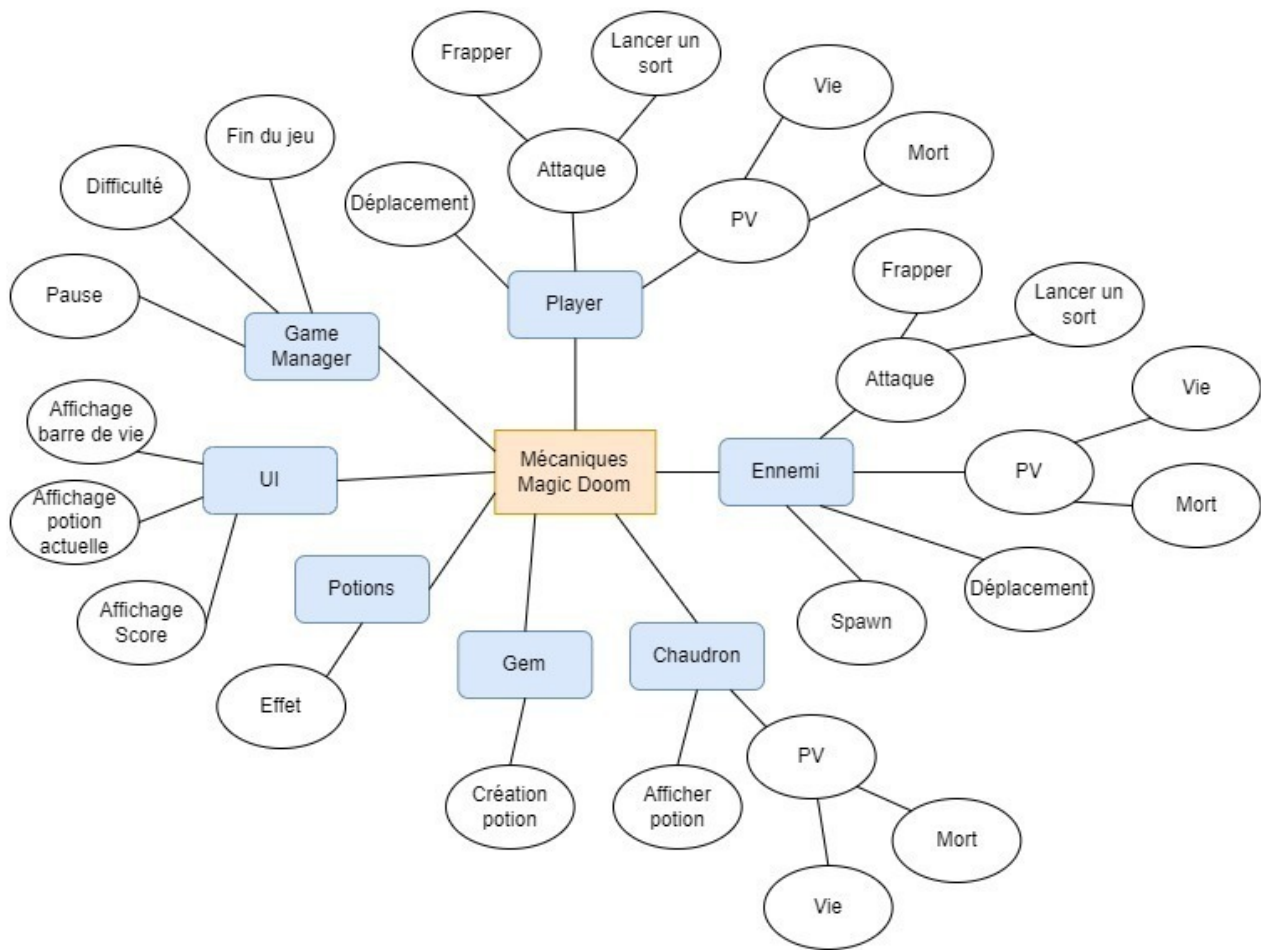


FIGURE 1 – Le graphique des mécaniques.

3 Interface

3.1 Signs et feedbacks

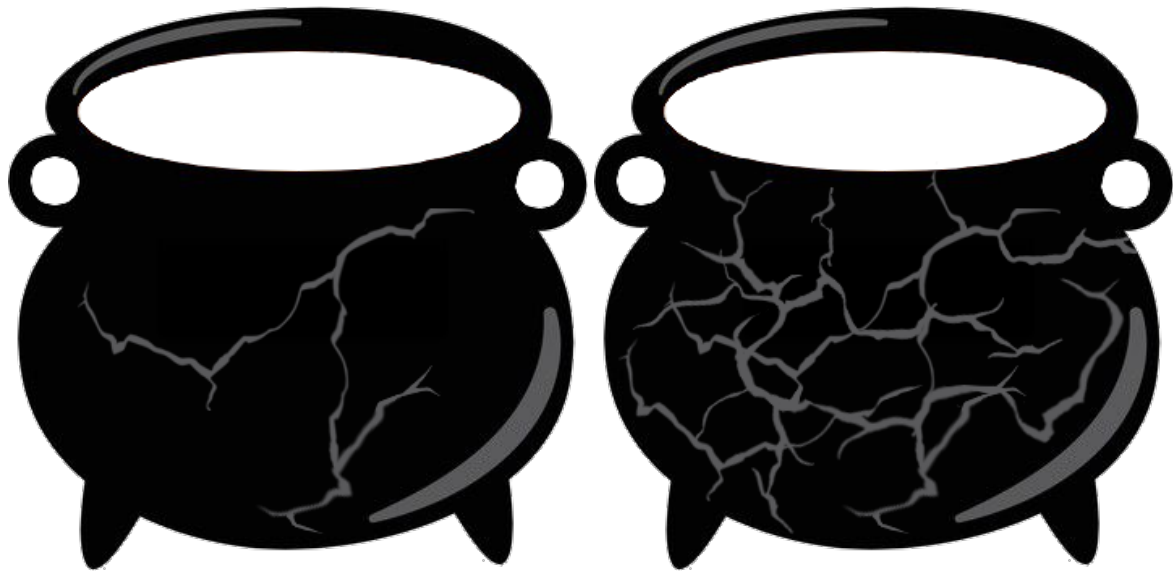
Pour aider à la compréhension du jeu, nous avons ajouté des signs et des feedbacks, sonores ou visuels.

3.1.1 Signs

Un exemple de sign utilisé est les fissures sur les chaudrons. En fonction des points de vies du chaudron il apparaîtra plus ou moins fissuré à l'écran (figure 2).

3.1.2 Feedbacks

Les feedbacks sont les informations qui permettent au joueur d'avoir un retour sur une action qu'il a effectuée. Un exemple de feedback est l'animation lorsque le joueur prend des dégâts (figure 3).



(a) Chaudron avec de légères fissures

(b) Chaudron sur le point de se briser

FIGURE 2 – Exemple de sign sur le chaudron

3.2 Les différents écrans

Différents écrans ont été utilisés dans le jeu :

- menu principal ;
- réglages ;
- choix de partie ;
- combat ;
- crédits ;
- tutoriel ;
- fin de partie (avec plusieurs alternatives possibles).

4 Animations

Nous avons animé notre joueur et nos monstres pour rendre notre jeu plus agréable à l'oeil et parfois le rendre plus compréhensible, en ajoutant une animation lorsque le joueur prend un dégât par exemple (figure 4).



(a) Le joueur lors de son état normal



(b) Le joueur lorsqu'il prend un dégât

FIGURE 3 – Exemple de feedback lorsque le joueur prend un dégât.

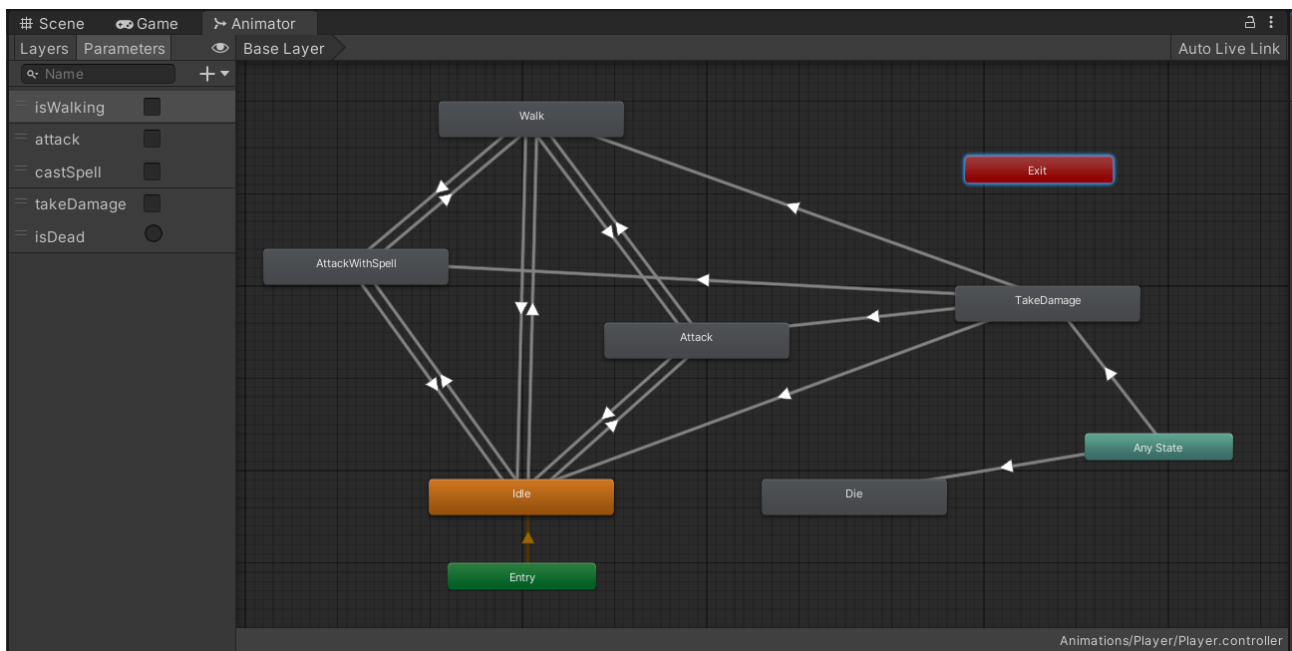


FIGURE 4 – Le graphique d'animation.

5 Problèmes rencontrés

Nous n'avons rencontré aucun problèmes majeur durant le projet.

6 Différences entre le préprojet et le jeu

Nous avons décidé d'apporter plusieurs modifications entre l'écriture du préprojet et le rendu final.

Tout d'abord, nous avons supprimé l'apparition d'une créature lors de la fin de la partie pour passer plus de temps sur l'ajout des sons et des écrans. Nous avons jugé que cette fonctionnalité demandait beaucoup de temps par rapport à ce qu'elle apportait. Nous l'avons remplacé par un écran de victoire.

Nous n'avons également pas ajouté de boss, préférant simplifier le jeu en augmentant la quantité d'ennemis qui apparaissent. Pour compenser nous avons augmenté la vie des ennemis pour qu'ils ne puissent pas être tués d'un seul coup avec une attaque de mêlée du sorcier.

La répartition des tâches a également été ajustée et est détaillée dans la section correspondante.

7 Conclusion

Ce projet nous a permis de mettre en pratique les connaissances apprises lors du cours programmation pour le jeu vidéo. Nous avons utilisé certaines notions vues en cours, comme la gestion des collisions. Nous avons également utilisé des notions apprises en autonomie, comme l'ajout d'écrans ou de musique.