**Princess Kidnapper**

**Document Technique**

27/11/2018

Guillaume Jeannin

**Sommaire**

**Description**  **1**

Scénario

Principe du jeu

Conditions de Victoire/Défaite

**Étapes du projet**   **2**

**Mécaniques**  **2**

Caméra

Capacités du joueur

Contrôles

Capacités des ennemis

**Arborescence**  **3**

**Fonctionnement global des scripts**  **4**

**Problèmes rencontrés 5**

**Conclusion**  6

Réalisé

Améliorations possibles

Impression personnelle

**Description**

**Scénario**

Le joueur incarne le Baron Jean-Pierre, un épéiste talentueux qui souhaite étendre sa famille en mariant ses fils aux princesses des royaumes voisins. Un seul problème, ses fils sont extrêmement laids, les seigneurs voisins refusent de marier leurs filles avec eux.

Jean-Pierre décide de faire usage de ses talents de combattant pour kidnapper ces princesses pour les marier de force avec ses fils. Il devra se frayer un chemin à travers les terres de ses voisins jusqu’à leur château où se trouvent les princesses.

**Principe du jeu**

Le joueur incarne Jean-Pierre et doit atteindre la princesse au bout de chaque niveau. De nombreux ennemis tenteront de lui barrer la route en l’attaquant et chacune de leurs attaques est mortelle.

On retrouve comme ennemis des Paysans, Soldats, Archers, Mages et Dragons.

Jean-Pierre peut les éliminer avec son épée, au corps à corps ou en chargeant une onde de choc pour attaquer à distance.

Chaque niveau est composé de 2 parties, la première à l’extérieur sur les terres d’un seigneur voisin, la seconde dans son château. Les niveaux sont faits de plateformes, d’ennemis et d’un checkpoint à la transition entre les parties.

Jean-Pierre peut s’y déplacer en marchant, sautant et avec un dash horizontal.

**Conditions de Victoire/Défaite**

Le seul moyen de gagner et d’atteindre la princesse avec Jean-Pierre.

Le joueur peut augmenter la difficulté du niveau pour le réussir dans chaque difficulté.

Il est possible de perdre en étant tué par un ennemi, en tombant dans un trou, en marchant sur un piège ou en tuant la princesse.

**Étapes du projet**

Ci-dessous une liste chronologique des éléments initialement planifiés :

* Déplacements de base du joueur
* Plateforme simple
* Princesse
* Attaque du joueur au corps à corps
* Ennemi : Paysan
* Attaque chargée du joueur à distance
* Ennemi : Soldat
* Ennemi : Archer
* Menu principal
* Menu pause
* Ennemi : Mage
* Ennemi : Dragon
* Dash du joueur
* Premier niveau
* Contrôles mannette

**Mécaniques**

**Caméra**

Le jeu est un platformer 2D en vue de profil orthogonale suivant le joueur. La vue est décalée vers l’avant et le haut, le joueur peut aussi regarder en bas.

**Capacités du joueur**

Déplacement vers la gauche et vers la droite

Saut

Accélérer brièvement sa vitesse horizontale

Attaque de corps à corps à l’épée, en face de lui ou vers le bas

Attaque chargée à distance, tout droit ou en diagonale en avant et vers le bas

**Contrôles :**

|  |  |
| --- | --- |
| -> | Se déplacer vers la droite |
| <- | Se déplacer vers la gauche |
| Espace, ( A ) | Sauter (uniquement si sur une plateforme ou au sol) |
| Flèche du bas | Déplacer la caméra vers le bas / Orienter les attaques vers le bas |
| Clic droit, ( B ) | Augmentation de la vitesse horizontale (le faire pendant un déplacement vers la droite ou la gauche pour dasher) |
| Clic gauche, RB | Attaque au corps à corps |
| Clic gauche, RB (maintenir 2 à 4 secondes puis relâcher) | Attaque chargée à distance |
| P, Start | Pause |
| Échap | Ferme l’application |

**Capacités des ennemis**

**Paysan** : Se tourne et se déplace horizontalement en direction du joueur, armé d’une fourche tuant au touché, ne se déplace que si le joueur est assez proche

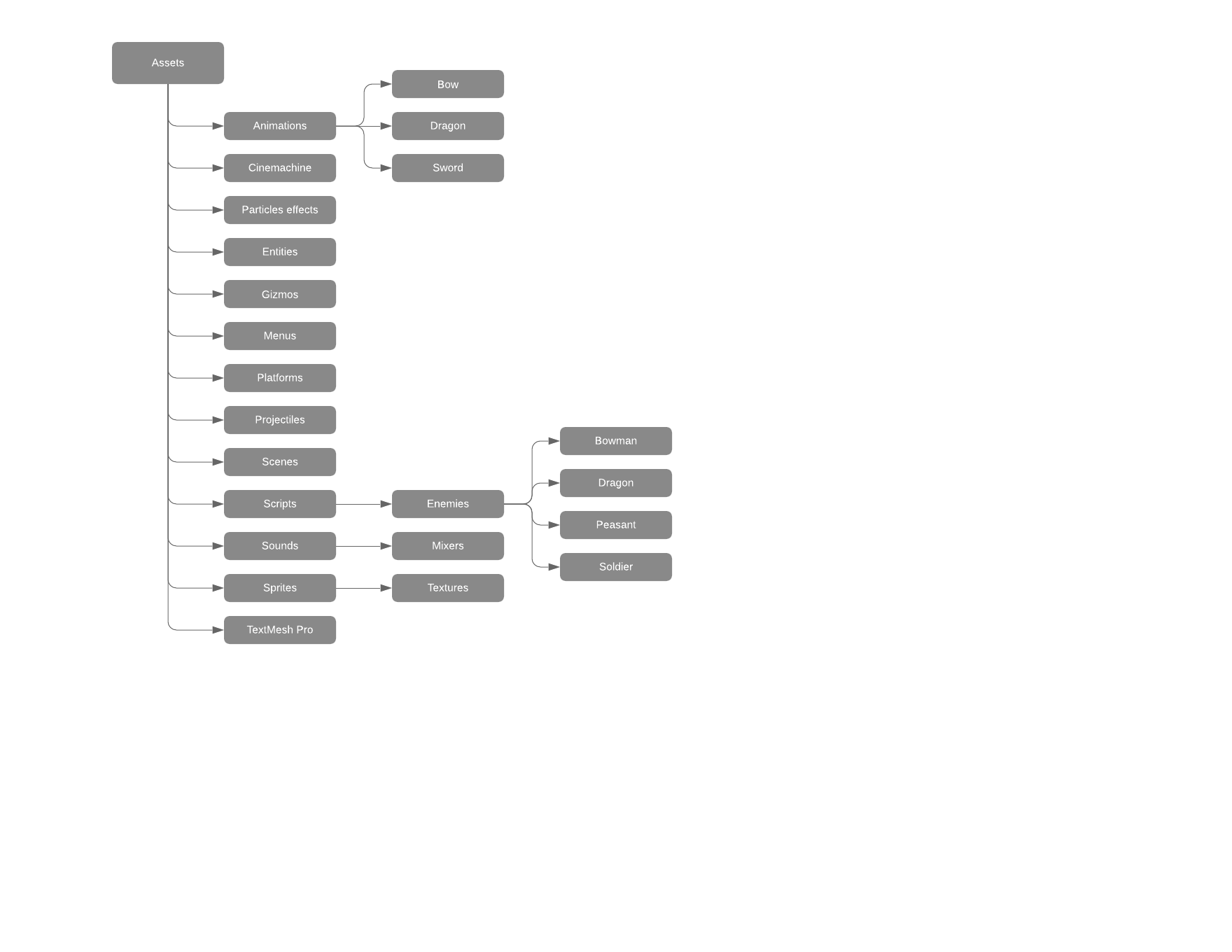
**Soldat** : Se tourne et se déplace horizontalement en direction du joueur en restant dans une zone limitée, équipé d’un bouclier bloquant les attaques du joueur et d’une lance tuant au touché, oriente sa lance devant, en dessous ou au-dessus de lui selon la position du joueur par rapport à la sienne

**Archer** : Se tourne en direction du joueur, se déplace horizontalement dans le sens opposé en restant dans une zone limitée, oriente son arc vers le joueur, tire des flèches à trajectoire balistique sur le joueur s’il est dans sa zone de tir

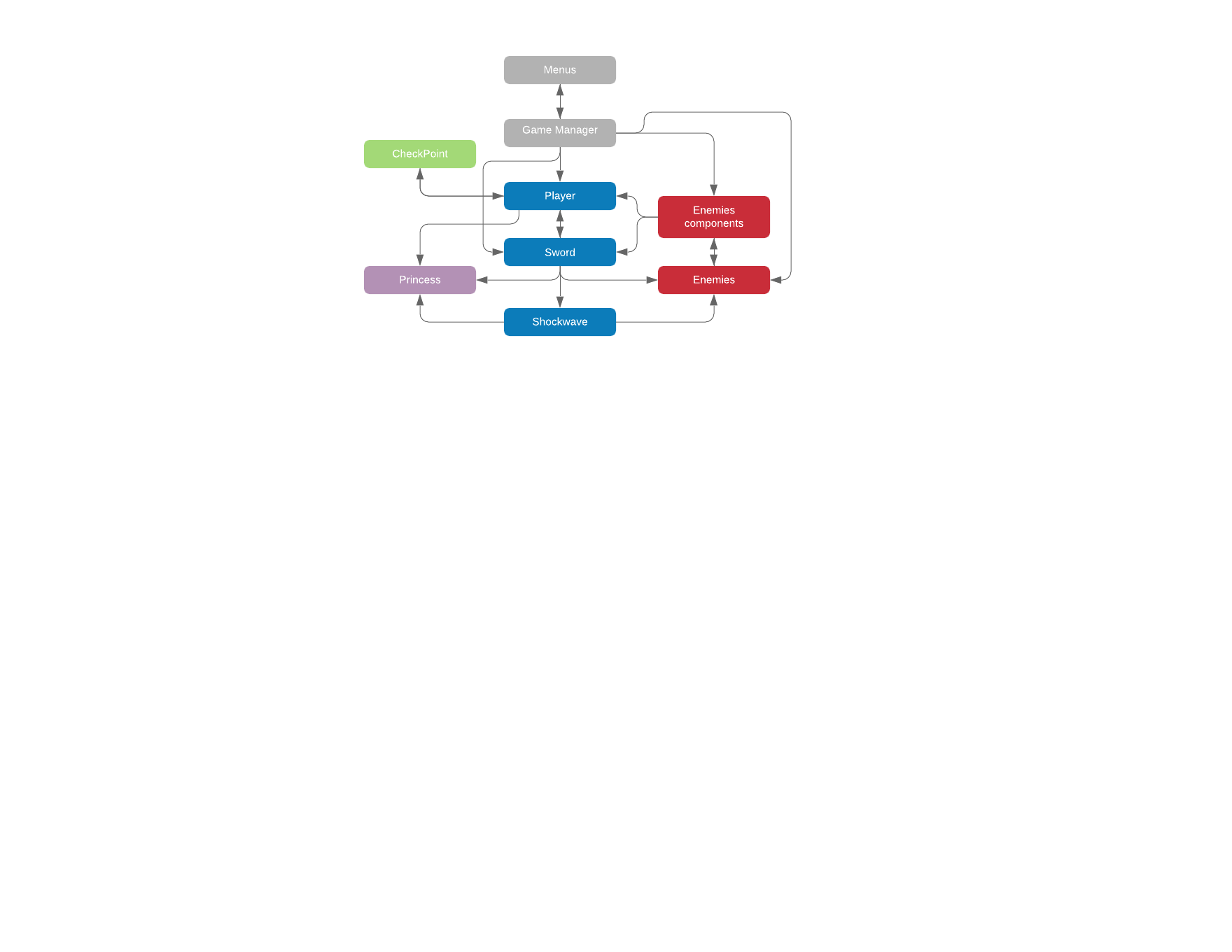
**Mage** : Se tourne en direction du joueur, oriente ses mains en direction du joueur, projette des boules de feu qui suivent le joueur et rétrécissent jusqu’à disparaitre

**Dragon** : Se tourne en direction du joueur, se déplace verticalement dans une limite de position, oriente sa gueule vers le joueur, crache du feu sur le joueur s’il est assez proche (le joueur peut monter dessus) fait apparaitre son cadavre lors de sa mort

**Arborescence**



**Fonctionnement global des scripts**



Les changements de scène sont gérés par le Manager et les Menus.

L’épée du joueur interagit avec les ennemis et avec la princesse pour les tuer au bon moment et les composants d’attaque des ennemis appellent un changement de scène du Manager quand ils tuent le joueur.

La princesse appelle un changement de scène quand elle est kidnappée.

La difficulté est gérée par le Manager.

La pause est gérée par le Manager qui fige l’état du jeu, et peut être interrompue par le menu pause.

**Problèmes rencontrés**

Gérer les cooldown

Pour plusieurs éléments du projet il me fallait un moyen d’évaluer le temps écoulé depuis une certaine update. J’ai résolu ça en incrémentant un timer de deltaTime à chaque update depuis la fin de la capacité jusqu’à ce qu’il atteigne la valeur du cooldown.

Gérer l’attaque chargée

Je devais trouver comment savoir si le joueur gardait une touche enfoncée et pendant combien de temps. J’ai réutilisé le système de timer jusqu’à ce que la touche soit relevée.

Manipulation des quaternions

Pour orienter des objets vers le joueur, j’ai mis beaucoup de temps à trouver comment changer la rotation de l’objet correctement. Je m’en suis sorti à l’aide de la documentation Unity et de plusieurs threads d’unity answers.

Rendre le dash fonctionnel

La première version du dash que j’ai faite augmentait énormément la vitesse du joueur pendant une seule frame. Cette méthode permettait parfois de traverser des ennemis et n’avait parfois aucun effet.

J’ai alors mis en place un nouveau système pour le dash, où la vitesse du joueur est augmentée pendant un bref moment, ce qui fonctionne bien mieux.

**Conclusion**

**Réalisé**

Tous les éléments que j’avais prévu ont pu être implémentés dans les temps, j’ai aussi fait un checkpoint et un modificateur de difficulté que je n’avais pas prévus au départ.

**Améliorations possibles**

Voici une liste non exhaustive des améliorations et ajours possibles au jeu :

* Un niveau tutoriel
* Des graphismes et sons propres au jeu
* Une cinématique courte introduisant le scénario
* Des bonus récoltables comme un bonus d’invincibilité temporaire
* Un comportement plus développé pour les ennemis
* Un boss à la fin du niveau 1
* D’autres niveaux
* D’avantage de types d’ennemi
* Un score s’affichant à la fin du niveau selon le temps d’accomplissement et les ennemis tués

**Impression personnelle**

Le projet s’est globalement bien déroulé, je suis content d’avoir implémenté tout ce que je voulais dans les temps. J’aime bien le résultat malgré le manque de graphismes et de sons.

J’ai beaucoup appris sur l’utilisation d’Unity, ce qui me motive à en apprendre davantage.