

Road to Far Far Away

Par Nathan Coustance & Donovan Feuillastre

Fiche signalétique

Nom du jeu : Road to FarFarAway

Type de jeu : Reversed Tower Defense

Mode : Solo

Support : PC

Moteur : Unity 2020.3.23f1

Pitch : Aidez Shrek et ses amis à sauver chasser Lord Farquaad de son trône. Rendez-vous à Far Far Away et prenez toutes les forteresses sur votre chemin pour libérer le pays de son oppresseur.

Inspirations

Shrek



TABS



Advance Wars



Gameplay

Units

- **Allies** : ce sont les unités possédées par le joueur. Chacune possède des points de vie, une cible, ainsi qu'un ordre, donné par le joueur, indiquant quel type de cible prendre en priorité.
- **Enemies** : ce sont les unités ennemies. Leur seul objectif est de défendre les tours en attaquant l'unité *Allies* la plus proche.

Unit Properties

- **Damage** : montant de dégâts d'une attaque.
- **Fire rate** : délai entre chaque attaque.
- **Max Health** : montant maximum des points de vie.
- **Current Health** : points de vie actuels de l'unité.
- **Move Speed** : vitesse de déplacement de l'unité.

- **Cost** : coût d'envoi de l'unité.
- **SearchingMethod** : type de recherche des autres unités ennemis.

Enemies

- **Soldats** : séparant le joueur des tours, ces unités sont les seuls défenseurs du royaume.
- **Tours** : servant de garnisons aux unités ennemies, ces tours sont les cibles principales du joueur. À chaque fois qu'une est détruite, les autres s'améliorent et gagnent en puissance.

Spawning

- **Canon** : le joueur dispose d'un canon lui servant à envoyer ses unités au combat. Il dispose d'un choix entre plusieurs unités et est libre de les envoyer où il souhaite sur la map.
- **Tours** : les unités ennemies apparaissent depuis les tours. Chaque tour fait apparaître régulièrement des unités pour se défendre.

End Conditions

- **Winning condition** : le joueur a détruit toutes les tours ennemies.
- **Losing condition** : le joueur n'a plus assez d'argent pour se battre.

Level Design

La création de la map est gérée de manière procédurale.

Le placement d'obstacles est tiré de cet article de recherche :

https://www.researchgate.net/publication/224180076_Automatic_generation_of_game_elements_via_evolution

Il s'agit de placer les obstacles à la manière des déplacements d'un cavalier dans un jeu d'échecs.

On place plusieurs cavaliers sur la map, puis on génère un obstacle à chacune de ses positions possibles de déplacement afin d'obtenir une répartition naturelle des obstacles.

Après avoir placé les obstacles, on crée le chemin en plaçant 2 points, chacun sur une extrémité du terrain. On va ensuite utiliser l'algorithme de pathfinding A* afin de trouver un chemin reliant ces 2 points parmi les obstacles.

Si aucun n'est trouvé, on essaye d'enlever aléatoirement des obstacles afin de dégager la voie pour un chemin.

Après avoir exécuté les étapes précédentes, on obtient une map correcte mais pouvant parfois être décevante. Pour cela, nous avons implémenté un algorithme génétique permettant d'obtenir systématiquement de bons résultats grâce à une fonction de fitness et des paramètres adaptés.

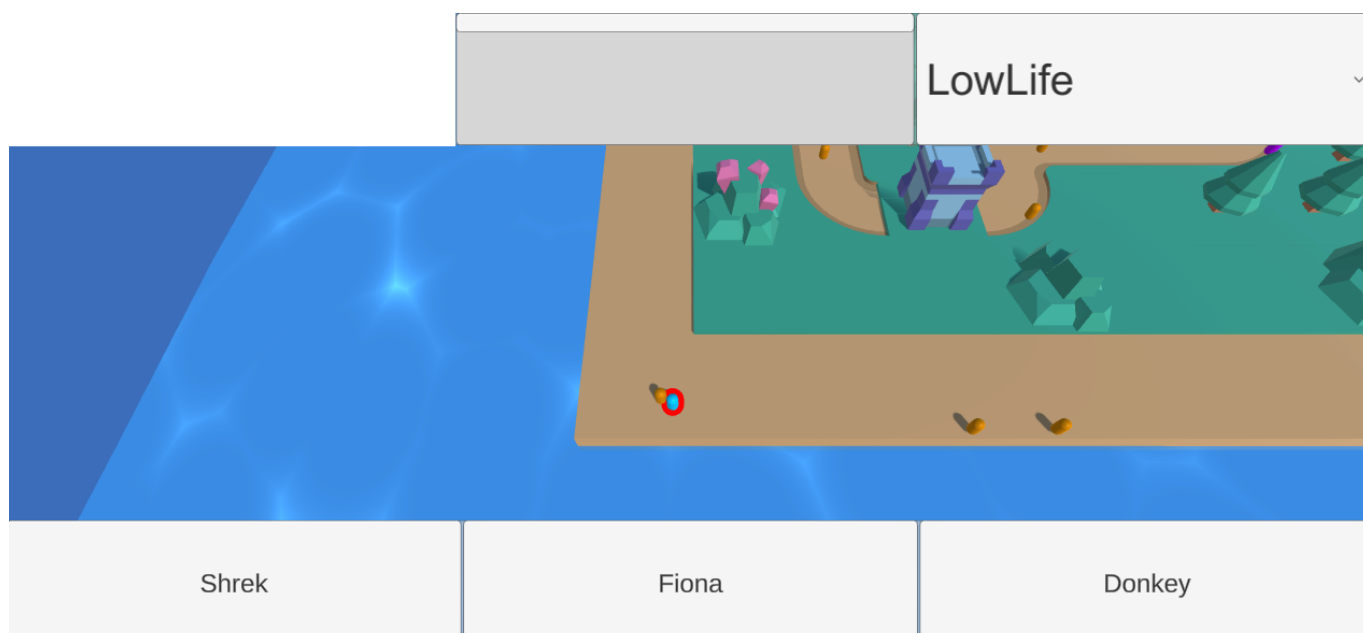


UX/UI

Les éléments d'informations sont affichés directement sur le HUD.

En bas on trouve les différentes unités que le joueur peut envoyer au combat. Il suffit de cliquer sur l'un des boutons pour sélectionner l'unité correspondante et ainsi pouvoir l'envoyer en cliquant sur le terrain.

En haut se trouvent les informations de l'unité actuellement sélectionnée. Les unités sélectionnables sont celles se trouvant sur le terrain. On peut ainsi consulter leurs points de vie actuels ainsi que modifier leur stratégie de combat.



Défi

- Level Design procédural
- State Machine, Pattern d'ennemi, Intelligence Artificielle (*et/ou réseau*)
- Beaucoup d'intelligence artificielle en même temps

Enjoy !

