**FICHE PRODUIT**

Le produit rendu est un logiciel « maquette » de notre projet de recherches, permettant de représenter le fonctionnement de la génération procédurale de façon plus visuelle et concrète.

# Intitulé du logiciel :

Le logiciel est une sorte de prototype de jeu, nommé « Lecroduge » dont le but pour le joueur est pour l’instant de se déplacer dans un monde généré de façon procédurale.

Plateforme : Linux, Windows et MacOs

Nombre de joueurs : 1

Langages utilisés : C++

IDE utilisée : QT Creator

# Améliorations possibles :

Le logiciel est un prototype que nous souhaitons développer plus tard dans un autre projet, et sa version finale devra avoir ces caractéristiques :

Style : RPG

Plateforme : Linux, Windows et MacOs

Nombre de joueurs : 1

Langages utilisés : C++

IDE utilisée : QT Creator

Dans cette version finale du jeu, la génération procédurale ne sera pas le seul type d’aléatoire utilisé. En effet, des ennemis seront placés de façon aléatoire sur la map, et nous prévoyons de créer des mini jeux qui seront déclenchés lors d’affrontements avec des ennemis, au moment où le joueur entrera en contact avec eux. Le résultat du mini jeu décidera de la victoire ou de la défaite du joueur.

\*map = carte

# Principe du jeu dans sa version actuelle :

Fonctionnement de la génération procédurale :

Pour générer la carte procéduralement et selon certaines contraintes (les différentes zones devant avoir une taille moyenne définie, les bords de la carte devant être des océans, et des compatibilités devant être repectées, nous avons utilisé l'algorithme de Voronoï.

L’objectif de l’algorithme de Voronoï est de placer aléatoirement sur une carte des points nommées centres, qui s’étendront ensuite pour former des zones.

Pour cela, nous avons implémenté plusieurs fonctions : biomeGen, centerGen et biomeAttribution.

Tout d’abord, la fonction biomeGen crée une matrice de biomes (types de zones), ceux adjacents étant compatibles, qui sera utilisée ensuite pour donner un type aux centres. Pour cela, les biomes sur les bords sont mis à Océan, puis un parcours en spirale de la matrice est effectué, pour donner à toutes les cases de la matrice un biome compatible à ceux adjacents.

Ensuite, centerGen place aléatoirement des centres dans des zones prédéfinies (la carte est découpée en carrés dont la taille correspond à la taille moyenne d’un biome), et à chaque centre est attribué le biome correspondant dans la matrice créée par biomeGen.

Enfin, biomeAttribution parcourt à chaque frame la matrice de centre pour ne garder que les suffisamment proche, puis détermine quelles cases sont visibles à l'écran et les parcours toutes en leurs attribuant le biome du centre le plus proche.

Ainsi, la carte est générée aléatoirement (l’aléatoire est utilisé dans le choix des biomes dans biomeGen et dans le placement des centres dans centerGen), les contours de la carte sont bien des océans (car biomeGen met à chaque zone sur les bords de la carte le type océan), les zones ont bien une taille moyenne prédéfinie (on peut choisir de mettre plus ou moins de centres sur la carte), et les biomes adjacents sont bien compatibles (biomeGen ne met un biome dans une case que s’il est compatible à tous ceux adjacents).

# Principe du jeu dans sa version finale :

### Déplacement :

* Le joueur se déplace dans un monde généré procéduralement
* Il peut rencontrer des ennemis à combattre, variant en fonction de la zone

### Combat :

* Le combat se déroule en deux phases :
* L’attaque, ou le joueur choisit un type d’attaque (mêlée, distance, magie …), et joue à un mini-jeu généré aléatoirement selon le type de l’attaque, dont la réussite influe sur la puissance.
* La défense, où un mini jeu est choisi aléatoirement parmi les 3 types d’attaques (mêlée, distance, magie) et dont la réussite du joueur influe sur les dégâts subis.
* Le joueur reçoit des récompenses à la fin du combat

Il existe donc 3 catégories d’attaques différents, les attaques de mêlée, à distance et de magie, et chaque catégorie possède deux variantes, une pour le joueur et une pour les ennemis :

-attaques ennemies :

-mêlée : le joueur doit suivre un rythme au clavier

-magie : le joueur doit se souvenir de mots qui s’affichent à l’écran, puis les saisir après sans se tromper

-distance : le joueur doit cliquer avec précision et rapidité sur des zones qui apparaissent au fur et à mesure

-attaques du joueur :

-mêlée : il faut réussir un QTE (quick time event), c’est à dire saisir au clavier une suite de touches différentes dans le temps imparti, au fur et à mesure qu’elles apparaissent

-magie : le joueur doit se souvenir d’un certain nombre de symboles ou d’images sur des cartes, et les retrouver après en faisant des paires, une fois les cartes retournées. Le nombre de paires trouvées dans un temps imparti définit les dégâts infligés.

-distance : il faut garder la souris dans une zone de l’écran qui bouge et qui peut changer de taille

Inspirations : Deltarune

Dwarf Fortress

Paper Mario

#### Points clés :Rejouabilité grâce à la génération procédurale

Plusieurs mini jeux diversifiant le gameplay

Progression grâce aux récompenses de fin de combat