**Troisième livrable : Interactivité**

**Par :** Équipe 16

|  |  |
| --- | --- |
| Giguère, Gabriel | 111104259 |
| Marcoux, Alain Jr | 111 151 878 |
| Proulx, Danick | 111 187 002 |

**Réalisé dans le cadre du cours :**

IFT-3113 – Projet de jeu vidéo

**Rapport présenté à :**

François Chéné

**Remis le :**

20 décembre 2018



Animation des agents

Les agents sont animé en utilisant des sprites(dessiné par nous même) qui sont succédé une à la suite de l’autre de manière suffisament rapide pour que le joueur ai une impression de fluidité. Lorsqu’un événement particulier survient (mort), on charge un groupe de sprites différent. Voici ces groupes utilisé pour les 2 sortes d’agents:

* en vie/mouvement: loop entre 4 images
* mort: 3 images sans répétition















Animation de l’interface

* Barre de lancée: L’image oscille selon une fonction mathf ping pong qui correspond au taux de force et le cadre est changé à chaque tour pour prendre la couleur de l’équipe.
* Vies: lorsqu’un agent est défait, un X s’ajoute dans les vies au dessus de l’écran
* Pointeur: lorsque le pointeur est au dessus dun bouton du menu, un indicateur est déplacé à côté de celui-ci(personnage) et larrière plan paraît plus pâle.

Effets de particules

* Lors d’une victoire, les tuyaux du vainqueur s’active et laisse s’échapper des particules solides avec de la gravité.
* Les cubes mauves du terrain produisent des particules avec une texture en hâlo lumineux.
* Les balles de feu sont composé de particules qui change de couleur au cours de leur vie et restent près du centre de la balle, elles laisse aussi une légère trainée lors de leur déplacement.

Ambience sonore

Liste des effets sonores et contexte d’utilisation

Fonctionnalités supplémentaires

**Génération procédurale de l’environnement (Gabriel Giguère)**

Lorsqu’on sélectionne une partie locale, il est possible de choisir les paramètres qui seront utilisé lors de la génération. Ces paramètres sont enregistré dans les préférences du joueur. Ces paramètres sont:

* nombre d’agents contrôlé.
* nombre d’agents à affronter.
* la largeur du niveau.
* la profondeur du niveau.
* la quantité de décors, dont le maximum varie selon la largeur sélectionné.

Une fois le niveau lancé, le script de génération instancie un à un les cubes composant le terrain jouable selon le nombre de rangées et de colonnes spécifié plus tôt. Lors de l’instanciation, le générateur choisit aléatoirement un cube de terrain préfabriqué parmis la liste qui lui a été fournit.

Ensuite, il calcul le périmètre de cette zone et instancie un mur infranchissable autour.

Les décors sont ensuite instancié en choisissant une position aléatoire sur la zone jouable et en sassurant que cette position possède au moins la distance spécifié entre elle et les positions précédantes. Si elle ne respecte pas la distance, un autre point est trouvé et ainsi de suite jusqu’à trouver un point acceptable ou datteindre la limite de loop.

Ensuite les agents sont instancié sur la zone jouable de manière aléatoire.

Une texture avec bruit de Perlin est généré, ce bruit est toujours le même résultat dune génération à l’autre ( et d’une simulation à l’autre), donc on le génère une seule fois et ensuite, lors de l’instanciation d’un cube, on applique ce bruit sur le normal map pour donner plus de forme au cube. Le magma suit le même principe, cependant on fait osciller le offset en y et déplacer le offset en x selon le temps, jusqu’à ce que le offset atteint l’autre côté du cube, ensuite on recommence. Les blocs mauves possèdent en plus l’oscillation de l’intensité émettrice de leur couleur et des particules. Les décors possède des tuyaux (généré et texturé par nous dans un autre programme) qui s’animent lors d’une victoire et des arbres qui sont généré et instancié de manière différente en utilisant l’outil fournit par unity.

Une fois la génération terminé, elle passe le relais au script de gestion de la partie.

**Personalisation des avatars(Alain Jr Marcoux)**

**Musique de fond réactive(Danick Proulx)**