**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 :**

23 décembre 2018



Animation des agents

Les agents sont animés en utilisant des sprites(dessiné par nous-même) qui sont succédé une à la suite de l’autre de manière suffisamment rapide pour que le joueur ait une impression de fluidité. Lorsqu’un événement particulier survient (mort), on charge un groupe de sprites différent. Voici les groupes utilisés pour l’agent ennemie:

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



L’avatar est composé de plusieurs animations différentes selon les parties du corps qui ont été sélectionné par l’utilisateur (voir la section de la fonctionnalité avancée ‘Personnalisation des avatars’).

Animation de l’interface

* Barre de lancée: L’image oscille selon une fonction mathf ping pong avec un easing utilisant Hermite. Cette valeur correspond au taux de force pour le lancer et le cadre est changé à chaque tour pour indiquer 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 d’un bouton du menu, un indicateur est déplacé à côté de celui-ci(personnage) et l’arrière-plan du texte paraît plus pâle. Des particules indiquent la position du curseur et laissent un sillage derrière(plus d’infos dans la section suivante).

Effets de particules

* Lors d’une victoire, les tuyaux du vainqueur s’activent et laissent s’échapper des particules solides avec de la gravité.
* Les cubes mauves du terrain produisent des particules avec une texture en halo lumineux.
* Les balles de feu sont composées de particules qui changent de couleur au cours de leur vie et restent près du centre de la balle, elles laissent aussi une légère trainée lors de leur déplacement.
* Le pointeur du menu principal est composé d’un ensemble de systèmes de particules. Il y a celui qui indique la position du curseur, il a une courte durée de vie et diminue de taille après un court temps. Il y a 2 systèmes semblables qui diminuent de taille après une certaine distance et qui possède du bruit en fonction de cette même distance. L’un des 2 a aussi une génération de couleur qui change avec le temps. Finalement, le dernier produit peu de particules et ceux-ci ont des grandeurs générée aléatoirement et éparpillée avec un fort bruit, pour ensuite disparaître après une distance aléatoire.

Système de pooling

Un système de pooling a été conçu pour générer des particules à la balle de feu tiré par les agents. Lorsque la simulation débute, 10 particules sont générées dans un pool. Lorsqu’un agent lance la balle, les particules sont transférées à la position de la balle et lorsqu’il n’y a plus de particule dans le pool, c’est la particule qui a été générée depuis le plus longtemps est utilisée.



Ambiance 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 utilisés lors de la génération. Ces paramètres sont enregistrés 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 sélectionnée.

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 parmi la liste qui lui a été fournie.

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

Les décors sont ensuite instanciés en choisissant une position aléatoire sur la zone jouable et en s’assurant que cette position possède au moins la distance spécifiée entre elle et les positions précédentes. Si elle ne respecte pas la distance, un autre point est trouvé et ainsi de suite jusqu’à trouver un point acceptable ou d’atteindre la limite de loop.

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

Une texture avec bruit de Perlin est générée, 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 formes 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 possèdent des tuyaux (généré et texturé par nous dans un autre programme) qui s’animent lors d’une victoire et des arbres qui sont générés et instanciés de manière différente en utilisant l’outil fourni par unity.

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

**Personnalisation des avatars (Alain Junior Marcoux)**

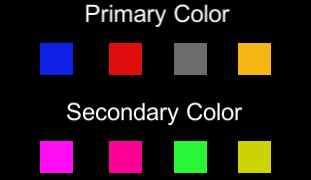
L’utilisateur peut choisir entre une sélection d’avatars préconstruits :

Dans le menu de personnalisation de l’avatar, l’utilisateur peut sélectionner l’un des trois avatars préconstruits.



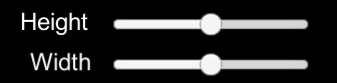
L’utilisateur peut choisir plusieurs couleurs de son avatar séparément :

Dans le menu de personnalisation de l’avatar, l’utilisateur a le choix entre plusieurs couleurs pour changer la couleur primaire et secondaire de son avatar.



Le joueur peut modifier la transformation de base de son modèle :

L’utilisateur peut modifier la largeur et la hauteur de son avatar.



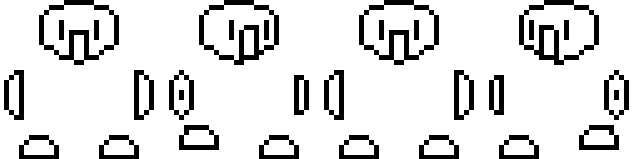
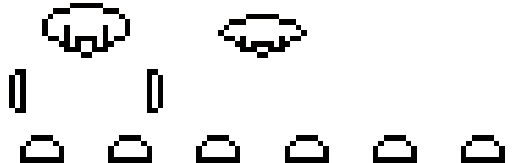
L’avatar est construit par composition d’images :

L’avatar est construit avec une composition d’images que l’utilisateur a sélectionnée parmi plusieurs choix. Toutes les parties de la composition sont animées avec des sprites sheets.

Partie du corps principal :



Parties du corps secondaire :

Yeux :

Yeux(mort) :



Bouches :

Lunette :

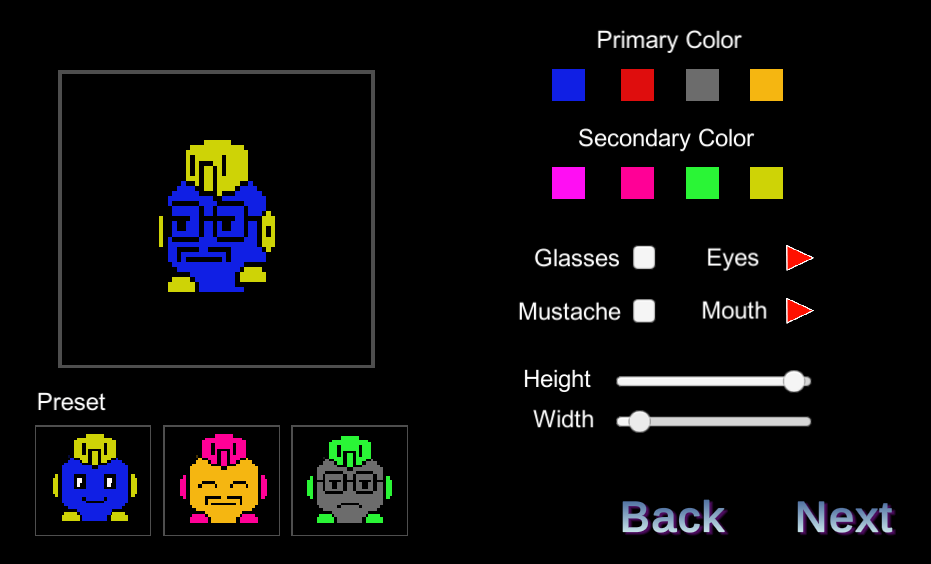


Moustache :



L’avatar est construit par combinaisons de maillages et utilise un système de masque :

Pour équivaloir ces critères avec un avatar construit avec des images, l’architecture pour la création de l’avatar a été conçue pour que lorsqu’on modifie les dimensions du corps de l’avatar, les dimensions des accessoires (lunette, moustache) et des parties du corps (yeux, bouche et parties secondaires) soient également modifiées en fonctions ce celui-ci. Une interface de personnalisation a été implémentée pour modifier l’avatar de manière conviviale et pour permettre de voir directement les changements apporter sur celui-ci.



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