

Rapport LIFProjet

Sujet AM2 Animation procédurale des mouvements d'un personnage
et jeu vidéo sous Unity

EuhMDR

Julien Croynar

Victoria Bekker

Gaspard Essevaz-Roulet

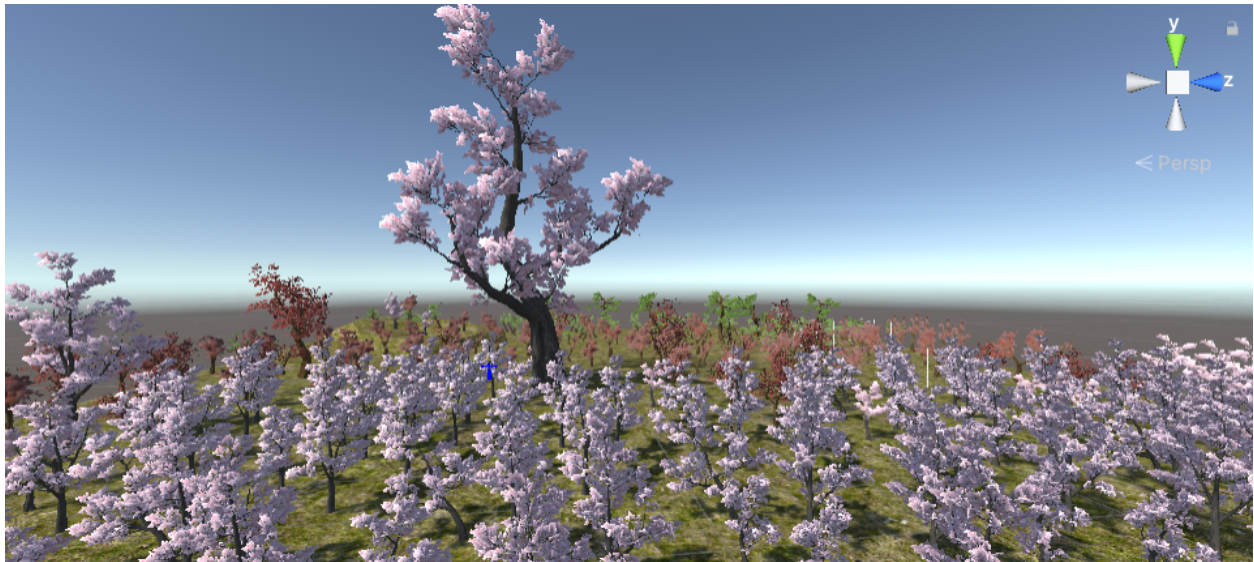
Encadrant: Alexandre Meyer

3 mai 2022

Travail Réalisé

Monde

Le monde est une forêt japonaise remplie de cerisiers, d'érables et de pins. Les arbres ont des hitboxes afin d'empêcher le joueur de passer à travers.



Joueur

Le joueur est un ninja qui se déplace grâce à un contrôleur. Une caméra suit le ninja et permet de faire des rotations pour changer la direction de déplacement et de ses tirs. Ses animations sont gérées par de l'animation procédurale qui sera expliquée plus bas.

Le joueur dispose de 3 possibilités au niveau du tir:

- Un tir basique qui retire 1 point de vie à l'ennemi touché.
- Un Super tir qui permet de tuer en 1 coup un ennemi.
- Un tir permettant de détruire des parties du décors.

Ces 3 tirs ont une durée de chargement plus ou moins longue.

Le joueur possède des points de vie, si le joueur se fait trop tirer dessus, il “meurt” en perdant tout son contrôle et se fait pousser par les ennemis.

Ennemis

Les ennemis sont des hommes en costumes qui suivent le joueur et essayent de lui tirer dessus et de le tuer. Ceci est géré par un script qui calcule la distance entre le joueur et l'ennemi, et si la distance est assez faible, l'ennemi commence à suivre le joueur et à lui tirer dessus.

Si un ennemi se fait tirer dessus par un Super tir ou un certain nombre de tirs normaux, il meurt en disparaissant.



Animation procédurale

L'animation procédurale pour la marche est gérée par un seul script qui peut être divisé en plusieurs étapes:

- L'initialisation consiste à gérer les valeurs initiales des positions du corps et des pieds, de la distance et hauteurs des pas, ainsi que de calculer la distance entre les pieds et le corps et de déterminer le premier pied qui se déplace.
- Déterminer s'il faut prendre un pas grâce à `step()`. Si la distance entre le pied qui doit bouger et le corps est plus grande que la distance initiale, la fonction retournera un `True` pour indiquer qu'il faut faire un pas.
- Déterminer la direction du déplacement grâce à `isMoving()`. La procédure permet de changer la valeur de deux entiers, `forward` et `moving_right`, le premier indique si le personnage recule, avance ou ne se déplace pas dans l'axe X et le deuxième est pareil mais pour l'axe Z.
- Calculer le futur placement du pied grâce à `nextStep()`. Les positions x et z sont calculées en ajoutant ou soustrayant la distance souhaitée aux positions courantes s'il y a un déplacement. La position y est calculée grâce à un Raycast pour éviter que le pied passe sous le sol. Finalement, la rotation du pied est calculée pour que le pied soit au bon angle sur des surfaces inclinées.

- Le déplacement des pieds est géré par `FootMovement()`. Grâce à de l'interpolation, les positions x et z du pied qui doivent bouger sont calculées. Pour la position y du pied, la fonction `Sin` permet de calculer la hauteur. L'autre pied ne change pas de position. Si aucun pied ne doit faire un pas, les pieds ne changent pas de position.

Jeu final

Le jeu final est un petit shooter où le joueur se fait poursuivre par des ennemis. Le joueur peut tirer sur ses ennemis ainsi que faire disparaître les arbres. Il peut aussi s'amuser à se cacher dans le baril ou à essayer de grimper des arbres grâce à des exploitations liées aux hitboxes précises des arbres.

Organisation

L'objectif de départ était de créer un jeu 3D dans lequel le joueur pourrait se déplacer dans un petit monde et se battre contre des ennemis.

Le travail a initialement été réparti de la façon suivante:

- Création du monde par Gaspard
- Création du personnage principal par Julien
- Création des ennemis par Victoria

Par la suite Victoria est passée entièrement sur l'animation procédurale du Joueur. Julien est passé de son côté sur les mécanique de tir du Joueur et des ennemis, ainsi que les différents scripts liés à ces derniers. Gaspard a également aidé sur la conception du gameplay, notamment sur le déplacement du joueur.

Difficultés

Le projet est rapidement devenu trop grand pour GitLab, nous avons donc dû partager le projet en utilisant WeTransfer ce qui nous a donné moins d'opportunités pour mettre en commun nos travaux et a ralenti la vitesse pour faire des modifications vers la fin.

Le concept d'animation procédurale était aussi difficile à implémenter pour un personnage bipède à cause d'un manque de tutoriaux précis.

Bilan

Finalement, nous avons créé un petit jeu drôle et jouable mais bugué avec beaucoup de place pour des améliorations.

Avec une meilleure organisation et plus de mise en commun du code, nous aurions pu mettre plus de fonctionnalités et avoir un jeu plus fluide en termes de gameplay et d'animation. Nous aurions aussi pu prendre plus de temps pour apprendre Unity.

Le problème principal fut la motivation qui n'était pas au rendez-vous. Plusieurs membres du groupe étaient dans une situation particulière (redoublement, réorientation) donc il n'y eut pas assez de sérieux dans le projet. Nous avons l'impression d'avoir toujours du temps jusqu'à en manquer.

Nous avons cependant appris pendant ce projet de nombreuses choses car on apprend beaucoup des erreurs et nous en avons fait un certain nombre. Nous avons appris :

- Il faut avoir des objectifs réalisables pour ne pas se perdre.
- La communication est primordiale.
- Il faut bien préparer le terrain en amont et mieux se former aux outils utilisés.
- Il faut savoir prendre ses responsabilités et motiver son équipe pour avancer ensemble, et ne pas rester dans son coin à ne rien faire.
- Un projet est bien plus complexe que juste produire du code, l'organisation derrière est au moins aussi importante