

[HAI809I - projet] Compte Rendu N°1



Yahnis Saint-Val (yahnis.saint-val@etu.umontpellier.fr)

Léa Serrano (lea.serrano@etu.umontpellier.fr)

25 février 2023

Génération procédurale de cartes d'environnement



1 Depot GitHub

Nous avons choisi d'utiliser GitHub pour gérer notre projet :
<https://github.com/LeaSerrano/M1-IMAGINE-ProjetImage>

Vous y trouverez notre programme, ainsi que les différents comptes rendus.

Vous trouverez dans le dossier "CR1" le compte rendu de cette semaine, accompagné de notre diaporama.

2 Présentation de notre projet

Nous avons été sélectionnés pour réaliser le projet Images RGB-D (sujet n°4).

Pour ce projet, nous avons pour but de réaliser plusieurs des objectifs proposés :

- Génération synthétique d'images RGB-D

Pour cela nous avons prévu de générer des images afin de créer des cartes d'environnement. Nous allons utiliser différentes méthodes de traitement d'image afin d'obtenir ces cartes d'environnement à partir d'images créées aléatoirement par du bruit. On aura donc des cartes d'altitudes (avec du Perlin Noise), des cartes de rivières (avec du Worley Noise) nous combinerons ensuite ces cartes pour obtenir notre carte d'environnement finale.

Nous pourrions aussi éventuellement combiner nos images avec des images de textures afin d'avoir une texture d'herbe au niveau des plaines, une texture d'eau au niveau de la mer,...

- Visualisation 2.5D à partir d'images RGB-D

Nous aimerions aussi réaliser une partie visualisation 3D des cartes d'environnement que l'on va générer. Pour cela, nous voulons générer un maillage en 3D, ou un nuage de points, représentant notre environnement, et ayant donc les propriétés (hauteur, couleur, texture...) correspondantes à ce qui est décrit dans nos cartes d'environnement.



3 Objectifs pour la semaine qui arrive

Pour la semaine qui arrive, nous avons prévu de commencer à réaliser l'état de l'art afin d'avoir une meilleure connaissance du sujet dès le départ.

Ensuite nous allons chercher comment réaliser les algorithmes de Perlin Noise et de Worley Noise et éventuellement, de commencer à les programmer.

Nous avons trouvé ces bibliothèques qui nous seront sûrement utiles :

[Perlin Noise C++](#)

[Worley Noise C++](#)