Nom du projet

AlgoWorld : Génération procédurale

Contexte et but du projet

Le but est de comprendre et d'implémenter les bases de la génération procédurale en un temps limité. Le projet se concentre sur un ou deux algorithmes essentiels (comme le bruit de Perlin ou Voronoï) et leur visualisation simple en 2D.

Porteur(s) du projet

- **Lylian Hay :** Responsable du développement et de la visualisation.

Objectif fonctionnel

1. **En tant que développeur, je veux** implémenter un algorithme simple de génération procédurale (par exemple, le bruit de Perlin/ Voronoï).
2. **En tant que développeur, je veux** visualiser les résultats sous forme de carte 2D (reliefs ou biomes).
3. **En tant que développeur, je veux** permettre l'ajustement de quelques paramètres (fréquence, amplitude) pour observer leur impact.

Environnement technique / technologique

Langage : Python.
Bibliothèques :

a. **NumPy** : Calcul matriciel.
b. **Matplotlib** : Affichage graphique (heatmaps ou cartes 2D).
c. **OpenSimplex** : Implémentation prête à l'emploi du bruit de Perlin/Simplex.

Description du livrable

1. **Programme Python** :

    a. Implémentation d'un ou deux algorithmes de génération procédurale.
    b. Affichage simple des résultats sous forme de heatmaps.
2. **Documentation concise :**
    a. Explication de l'algorithme choisi.
    b. Guide rapide pour ajuster les paramètres.
3. **Résultats :** Quelques captures d'écran des visualisations générées.

Organisation et temporalité

Jour 1 : Recherche et préparation

- Étudier les bases théoriques du bruit de Perlin (ou Simplex) et/ou des diagrammes de Voronoï.
- Installer les bibliothèques nécessaires (NumPy, Matplotlib, OpenSimplex)

Jour 2 : Implémentation de l'algorithme principal

- Implémenter le bruit de Perlin
- Tester l'algorithme pour générer une matrice de valeurs.
- Créer une visualisation de base (heatmap).

Jour 3 : Ajout de paramètres et exploration

- Ajouter la possibilité d'ajuster des paramètres (fréquence, persistance, etc.).
- Tester différentes configurations pour observer leur impact.
- Si le temps le permet, implémenter un deuxième algorithme simple (Voronoï).

Jour 4 : Finalisation et présentation

- Affiner la visualisation (ex. : ajout de couleurs pour représenter des biomes).
- Documenter le projet (explications rapides des algorithmes, captures d'écran).
- Préparer une présentation ou un fichier README explicatif.

Livrables à la fin des 4 jours

1. **Code Python fonctionnel :**
    a. Génération d'un terrain procédural simple (Perlin/Simplex).
    b. Visualisation 2D (heatmap).
2. **Documentation succincte :**

a. Une description de l’algorithme implémenté.
b. Instructions pour exécuter le programme.

3. **Exemples :**
    a. Quelques captures d’écran des terrains générés avec différents paramètres.