Les Composants du Projet

Le projet est constitué de plusieurs scripts principaux, chacun jouant un rôle dans la génération, le dessin et l'affichage de l'arbre. Voici un aperçu des principaux composants :

a. LSystem (LSystem.cs)

Le cœur du projet est la classe **LSystem**, qui contient la logique de création et de génération des arbres à partir d'un **axiome** initial et de **règles** définies. Elle applique ces règles de manière itérative pour générer une chaîne qui représente la structure de l'arbre. Cette chaîne est ensuite utilisée par la "tortue" pour dessiner l'arbre en 3D.

Fonctionnement :

- **Axiome** : L'axiome est le point de départ du L-System. C'est la chaîne de caractères initiale qui est progressivement transformée par l'application des règles.
- **Règles** : Les règles sont définies comme des transformations des symboles dans l'axiome. Par exemple, un symbole "X" peut être remplacé par "F+F-X".
- **Itérations** : Le système applique les règles sur l'axiome un nombre défini de fois, chaque itération produisant une nouvelle chaîne de caractères.
- **Tortue (Turtle)** : La tortue est responsable du dessin de l'arbre. Elle suit les instructions (comme avancer, tourner, etc.) fournies par la chaîne générée par le L-System.

b. Turtle (Turtle.cs)

La **Tortue** est un élément clé de ce projet, car elle interprète la chaîne générée par le L-System pour dessiner les branches et les feuilles de l'arbre. La tortue suit des règles de mouvement basées sur les symboles de la chaîne :

- **'F'** : La tortue avance d'une certaine distance et crée une branche.
- **'+' et '-'** : Ces symboles indiquent une rotation de la tortue autour de son axe Y (pour tourner la branche).
- **'[' et ']'** : Ces symboles permettent de sauvegarder et restaurer la position et l'orientation de la tortue, ce qui est essentiel pour créer des branches secondaires (comme des ramifications).

c. TreeSpawner (TreeSpawner.cs)

Ce script permet de générer un arbre lorsque l'utilisateur clique sur le sol dans la scène. Lors du clic, un rayon est projeté depuis la caméra pour détecter la position sur le sol. Un arbre est ensuite créé à cet endroit en utilisant un L-System choisi au hasard parmi plusieurs options disponibles.

- **Raycasting** : Le script utilise le raycasting pour détecter la position du sol où l'utilisateur a cliqué.
- **Sélection aléatoire de L-System** : Il choisit aléatoirement un système L à partir de plusieurs pré-configurés.
- **Affichage d'un arbre** : Une fois le système L choisi, l'arbre est généré et affiché à la position du clic de la souris.

---

3. Le Fonctionnement Global dans le Cadre d'un L-System

a. Définition des Règles

Chaque arbre est défini par un ensemble de **règles** qui dictent la manière dont la structure de l'arbre doit se développer. Par exemple :

- La règle "F → FF" peut signifier que chaque fois que la tortue rencontre un "F", elle va avancer et créer deux segments de branche (plutôt qu'un seul).
- Des règles comme "X → F+F-X" peuvent être utilisées pour créer des bifurcations ou des ramifications dans l'arbre.

b. Axiome Initial

L'axiome, qui est la chaîne de départ, peut être quelque chose de simple, comme "F" ou "X", qui sera ensuite transformé par les règles à chaque itération. À chaque itération, les symboles de la chaîne sont réécrits en fonction des règles définies.

c. Itérations

Chaque **itération** du L-System prend l'axiome (ou la chaîne générée lors de l'itération précédente) et applique les règles pour créer une nouvelle chaîne. Plus il y a d'itérations, plus la structure de l'arbre devient complexe.

d. Interprétation avec la Tortue

Une fois que la chaîne a été générée, elle est passée à la **Tortue**, qui l'interprète pour dessiner l'arbre. La tortue suit les instructions de mouvement et de rotation :

- **'F'** : La tortue avance et crée une branche.
- **'+' et '-'** : La tortue ajuste son angle de rotation.
- **'[' et ']'** : La tortue pousse et tire son état pour gérer les bifurcations.

4. Affichage des Arbres

Les arbres sont affichés dans la scène Unity en utilisant des **prefabs** pour les branches et les feuilles. À chaque itération, la tortue crée de nouveaux objets qui sont ensuite assemblés pour former l'arbre complet.

- **Branches** : Chaque "F" génère un nouveau segment de branche, représenté par un prefab de branche.
- **Feuilles** : Certaines règles ou symboles (comme "]") peuvent générer des feuilles à l'extrémité des branches.
- **Combinaison de Meshes** : Pour améliorer les performances, les maillages des branches et des feuilles peuvent être combinés en un seul mesh à la fin de la génération.

5. Interaction avec l'Utilisateur

Le **TreeSpawner** permet à l'utilisateur de cliquer sur le sol pour générer un arbre. Lorsqu'un clic est détecté, un arbre est créé à l'endroit où le rayon touche le sol. Plusieurs arbres peuvent être créés de manière aléatoire, et chaque arbre peut avoir des caractéristiques légèrement différentes en fonction des paramètres des L-Systems.