Procedural TNM084 Project Plan

David Styrbjörn

November 2021

1 Summary

The short summary of the project is a simple game where the user waters and takes care of a tree. The tree is procedural and generated with some algorithm that is yet to be decided. The objective of the game is to make the tree grow and grow by passively watering it. The game will have some type of incentive by rewarding the user when the tree reaches certain sizes.

When the game is closed a timestamp is saved and when the game is started up again the tree can be loaded and the amount of time since the game was last started can be used to interpolate the progress of the tree.

Some light research has lead me to L-Systems[1] and the possibility of applying those to generate the tree. I'm imagining a very modular procedural system with alot of paremters for how the branches are made etc which could all be influenced by user input.

2 Base-level Implementation

- An algorithm that can generate a tree-like structure with a number of parameters to make each tree look unique (in 2D)
- A working game mechanic where the player has the ability to water said generated tree, in the beggining through simple input like a button-click with some very basic SFX and VFX.
- A working system which keeps track of the tree's health (water-level to begin with) without any form of serializing of data between play sessions.
- Graphical effects like post-proccesing =¿ chromatic abberation and vignette to get a unique look for the game.

[1] https://en.wikipedia.org/wiki/L-system

3 Further-development Implementation

- Extend the algorithm to be able to generate bushes around the tree
- Serialize data between play sessions, making it so that the game doesn't start over each time the player closes it down
- Extend the water-mechanic by adding more tree-factors, an idea is to have a feature where the tree can drop saplings and perhaps make it recursive so that the game revolves around creating a forest with increasing size
- Even more generated content to fill the game-screen, gradient-background, clouds, more fauna types

4 Tools

Might be changed during the development but these are the starter tools that will be used.

- C (MinGW-w64/MSVC)
- Visual Studio Code
- CMake
- Raylib[2]

[2] www.raylib.com