VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Project Proposal: Eden Project

Bryony Gatehouse

Supervisor: Zohar Levi

March 18, 2021

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honours in Software.

Abstract

This document details the proposal for an ENGR489 project. This project will implement a realistic tree model based of research papers for users to place in a simulation.

Contents

1. Introduction

The engineering problem to be explored in this project is the implementation of algorithms to produce realistic 3D tree models. The final product will be evaluated on the recreation of the original results of the algorithm, the effectiveness of the algorithms, and the visual performance of the 3D model. This project will use Godot, a free game engine, as its main coding and simulating platform, and GitLab for version control. Research and theoretical papers are accessed through my supervisor or from Victoria University Library.

2. The Problem

The result of this project should be an application or game where the user is able to create realistic-looking trees with some indication to how the new tree impacts the CO2 levels in the air. Thus, the problem is how to implement an algorithm to produce a 3D tree model which reflects natural tree growth patterns.

Tree models can be created by using multiple methods. One such method is by employing a graphics designer to build a 3D model from scratch, probably based off an image or real tree. This model can then be edited slightly for each new tree required. However, this takes a lot of time.

The axial tree is the most simplest tree modeling algorithm. It it made up of nodes and segments, where each node has a straight segment continuing the branch, and can have zero to two side segments branching off from the node at a predefined angle [1]. This algorithm isn't entirely accurate but, with the use of controlled randomised angles, can create tree models which closely match a real tree.

Finally, the other method that is widely used, parses in images or 3D point sets of trees. The parsed data is separated into branches and leaf clusters which are used to reconstruct the tree as a 3D model [3] [6]. This method can accurately portray a tree from a certain angle, but might be missing branches from other angles which was hidden in the photo or point set. Using this method, an entire street of trees can be replicated quickly and efficiently.

An extension to the final product can be the inclusion of environmental effects to the tree's growth, specifically the response to a light source and possible collision between buds. The extended problem is how to implement algorithms to edit the tree model so it reacts to its environment.

Trees collect energy from the sunlight using photosynthesis. The more buds the tree has exposed to the light, the more energy it is able to collect. Thus, trees tend to bend or grow their branches towards the sunlight. This is called Phototropism. Brightly lit buds are able to grow quicker, and tend to split more often, than the buds left in the shade. Low-illuminated buds will reduce their branching tendencies in order to put all their energy into growing towards a more illuminated spot. However, the lack of resources could mean that the low-lit bud could die before reaching the sun [5].

Another method to increase a buds own energy store is to avoid growing through another buds' area of interest. This would result in the two buds having to share the energy from the sun. Thus, the buds tend to avoid each other as much as possible by detecting possible collisions within their "sphere of interest", which is the spherical area around the bud [4]. An extension to this idea is the avoidance of other physical objects, like a wall, which could invade the bud's "sphere of interest".

3. Proposed Solution

The project will be coded using the **Godot engine**. Godot is a game engine which has a quick learning curve, allowing new programmers to quickly start creating. It offers full C++ support for the coding, includes a built-in debugger, and has a file system that works great with Git [2]. Two weeks have been budgeted for learning and exploring this engine. Though I will be continuing to learn new features or better methods throughout the project timeline.

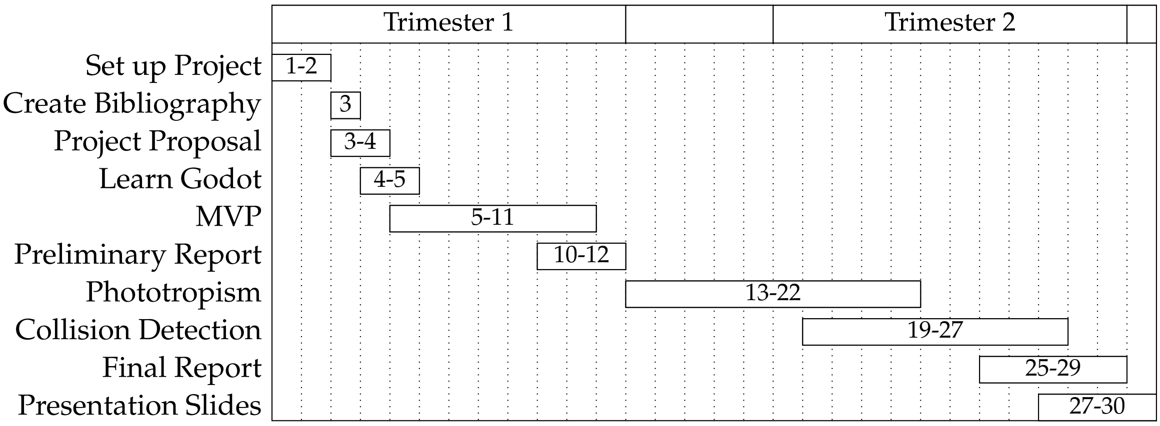

Figure 1: Proposed timeline for Project

The **Minimum Viable Product (MVP)** is the most basic product that meets the project requirements. For this project, this would be a product which has implemented at least one tree modeling algorithm, and uses the created models in a simulated world. The models do not need to respond to any external factors, but should create a realistic tree. Current plans aim to have the MVP finished before the break between the two Trimesters so Trimester 2 can be focused on the extensions, and writing the final report.

The **Preliminary Report** is a requirement for the ENGR489 course. It is due at the end of Trimester 1, on the 24th of May at midnight. The purpose of this report is to show the current progress of the project, including how I am dealing with problems and setbacks. This might include outstanding issues where feedback is required. Considering the progress from the Trimester, a revised timeline should also be included.

During Trimester 2, there is the opportunity to expand the product with extensions. **Phototropism** allows trees to look more realistic in their environment - like they'd grown there. The plan is to implement an algorithm for this idea over the mid-year break, and during the start of Trimester 2. **Collision Detection** will adapt the tree model to grow around obstacles. This means it will take into account the environment and surroundings. Depending on progress during the project, this is aimed to be implemented before the end of the second Trimester.

The **Final Report** is a requirement for the ENGR489 course, and is the most important component for the individual project. It is due at the end of Week 12 of Trimester 2. The purpose of this report is to outline what I've done, why I did it, what I could have done differently, and how it could improve things.

Finally, the **Presentation Slides** and **Presentation** are the last requirements for the ENGR 489 course. The slides are due on the 14th October at midnight and they will be presented the next day. The presentation should be 7 minutes long, not including questions. It should cover the project's motivation, the problem statement and a discussion of possible approaches, a technical discussion of my solution, and the results or findings.

4. Evaluating your Solution

The main problem to be solved during this project is the implementation of an algorithm to produce a 3D tree model which reflects natural tree growth patterns. Thus, the evaluation should prove that the final product is able to meet all the criteria laid out in this problem:

- Implemented an algorithm for a 3D tree model
  - Found an algorithm
  - Understands the algorithm
  - Implemented the algorithm in code
  - The implementation can recreate the original results of the algorithm (or similar)
- Construct a visual version of the 3D tree model in a simulation
  - Uses a simulator or similar
  - Implemented the algorithm in the simulator
  - Constructs the 3D tree model using materials (leafs, bark, etc)
- Reflects the natural tree growth patterns
  - Algorithm follows tree growing patterns
  - 3D tree model looks like a real tree (or similar)

The extended problem is the implementation of algorithms that edit the tree model so it reacts to its environment. If the extension was able to be completed, the evaluation should prove that the final product is able to meet the following criteria:

- Implemented two algorithms
  - Found an algorithm
  - Understands the algorithm
  - Implemented the algorithm in code
  - The implementation can recreate the original results of the algorithm (or similar)
- Implemented an algorithm to react to the direction of the light source
  - Algorithm matches trees tendency towards light
  - 3D tree model appears to have grown naturally towards the light
- Implemented an algorithm to avoid collisions between buds
  - Algorithm matches tree bud's tendency to avoid growing so their ”spheres of interest” collide with something.
  - 3D tree model appears to have grown naturally around obstacles

5. Resource Requirements

This project will mainly use Godot, a free game engine available from the internet. It will also use a GitLab repository for version control. Eclipse or similar may be used for committing code to GitLab if Godot hasn't the capacity. This is because it allows a separation between the code and documentation committing, and shows the differences between the to-be-committed file and the latest-committed version. All these programs are free to use.

Currently the papers which the algorithms will be sourced are accessed through my supervisor or the university library system.

6. Risks and Hazards

- **Failure to meet deadlines or not being able to meet the proposed timeline.**

  This could be caused by something being harder than expected, an obstacle stopping me from completing something, or me losing motivation to complete something.

  - This should be prevented by careful planning and a resolve to stick with the plan. However, if this does happen, the extensions planned (Phototropism and Collision Detection) can be removed from the project schedule to allow extra time to finish properly.

- **Other unexpected scheduling problems.**

  This could be my laptop dying, or WiFi stopping working, or supervisor getting sick.

  - The extensions planned in the second half of the project (Phototropism and Collision Detection) could be removed from the project schedule.

- **Poor quality code.**

  This could be a bug or problem that goes unnoticed until its too late, or poorly documented code.

  - This could be avoided with careful testing and documentation. Documentation should be decided before starting to code, and should be followed throughout.

- **Loss of work due to technological problem**

  This could be the computer that all the work is saved on dying.

  - This will be avoided by using a GitLab repository to store code and regularly committing.

- **Injures that temporarily stop me from working**

  This could be me breaking an arm or getting COVID.

  - Best course of action if this were to happen would be to take the time to rest or heal so that the injury is healed in the shortest time possible. If this does happen, it needs to be communicated as soon as possible to my supervisor and the course coordinator so they know what is happening.

- **The program bringing harm to someone.**

  - The current scope of this project shouldn't include any code that could bring harm to another being. However, all code should be carefully tested and documented before being released (if it is) to be the safest it can be.

- **Supervisor being unavailable.**

  This could be them becoming sick or too busy to answer emails.

  - If this happens, I should contact the course coordinator.

- **COVID Level changing.**

  - Section 6.1 details the COVID response plan.

6.1 COVID Alert Level Responses

COVID Alert Levels are liable to change, depending on the border situation. Thus it is good to be prepared with a plan for the different levels of lockdown. Thankfully this should stay a software project, and shouldn't require me to come into the labs at uni, or in person meetings with my supervisor.

Level 1: Level 1 is about being alert in case of an outbreak. During this time, face to face meetings with my supervisor are allowed.
In order to stay safe during this time I will:

- Scan into buildings and rooms using the NZ COVID Tracer App with Bluetooth enabled in order to keep a record of my movements.
- Stay home if sick.
- Practice healthy hygiene habits.

Level 2: Level 2 is about being careful as there might be an outbreak or cases in the community. During this time, face to face meetings with my supervisor may continue.
In order to stay safe during this time I will **also**:

- Social distance.
- Wear a face covering when social distancing might not be possible.
- Scan into the University using my Student ID.

Level 3: Level 3 is about being careful knowing there is a outbreak or cases in the community. During this time, face to face meetings with my supervisor might not be a good idea.
In order to stay safe during this time I will **also**:

- Stay home if possible.
- Use Zoom rather than meeting in person.

Level 4: Level 4 is complete lockdown. During this time I will not be in Wellington. I will not be able to meet my supervisor face to face.
In order to stay safe during this time I will **also**:

- Do everything over the internet.

Bibliography

[1] Lindenmayer, A., and Prusinkiewicz, P. *The Algorithmic Beauty of Plants*. Springer New York, mar 1990.

[2] Linietsky, J., and Manzur, A. Godot: Features. https://godotengine.org/features, 2021. Accessed: 2021-03-15.

[3] Livny, Y., Pirk, S., Cheng, Z., Yan, F., Deussen, O., Cohen-Or, D., and Chen, B. Texture-lobes for tree modelling. In *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11* (2011), ACM Press.

[4] Pirk, S., Benes, B., Ijiri, T., Li, Y., Deussen, O., Chen, B., and Měch, R. Modeling plant life in computer graphics. In *ACM SIGGRAPH 2016 Courses* (2016), ACM.

[5] Pirk, S., Stava, O., Kratt, J., Said, M. A. M., Neubert, B., Měch, R., Benes, B., and Deussen, O. Plastic trees: Interactive self-adapting botanical tree models. *ACM Transactions on Graphics 31*, 4 (aug 2012), 1–10.

[6] Tan, P., Zeng, G., Wang, J., Kang, S. B., and Quan, L. Image-based tree modeling. In *ACM SIGGRAPH 2007 papers on - SIGGRAPH '07* (2007), ACM Press.