Logan Callery
Email: lcallery@pdx.edu
Project Title: *Lord of the Lawn – Clicker Game*

Project Vision

For my final project, I'm planning to create a clicker-style game called *Lord of the Lawn*. The main idea is that the player runs a small lawn mowing business, and every click represents mowing one square foot of grass. Each "job" or lawn will have a randomly generated total size, meaning the player will need to keep clicking until the entire lawn is mowed before moving on to the next one. The overall structure is inspired by the *Universal Paperclips* game that was provided as an example, I want to recreate that same sense the example provided of but within the setting of a lawn mowing business. As the player mows more lawns and earns money, they'll unlock upgrades that make their work easier or more efficient. The first big upgrade will be the ability to hire an employee who automatically mows one square foot per second. From there, the player will be able to buy more employees, better equipment, or even manage multiple lawns at the same time.

In addition to upgrades, the game will include some simple metrics to track the player's progress. These might include:

- Total funds – how much money the player currently has.
- Price per square foot – how much the player will charge per square foot of lawn
- Number of active lawns – how many jobs the player is currently working on at once

My initial plan is to build it as a terminal-based game first. This will let me focus on getting the logic, timing, and upgrade systems working correctly before worrying about a graphical interface. Eventually a simple graphical interface resembling the *Universal Paperclips* game, will be added

Implementation Plan

The first step will be to set up the core game loop which includes handling user input (clicks), updating the number of square feet mowed, and generating new lawns when the current one is finished. Once that's working, I'll add the economy system, which includes money, upgrades, and automated employees. Each upgrade will modify the game state in some way, like increasing the rate of mowing or changing how much a lawn pays.

I'll need to store key values like:

- Current lawn size and progress
- Total funds earned
- Upgrade levels
- Number of employees and their productivity rate

For the automation system (the employees who mow on their own), I will need to experiment with Rust's async features or threading to simulate them working in the background while the player clicks. This could be a challenge, but it would make the gameplay more dynamic

Issues of Concern

There are a few areas that I think might be challenging or time-consuming. The first is scope management even though clicker games seem simple, adding upgrades, balancing the economy, and making everything feel rewarding takes a lot of iteration. I want to make sure I don't spend too much time trying to perfect the balance at the cost of finishing the core mechanics. I'm also a little concerned about timing and concurrency if I use asynchronous code for the automated workers. Rust's async and threading systems are powerful, but they can be tricky to implement correctly. Finally, balancing the random generation of lawn sizes and upgrade prices will take some testing. If lawns are too large or upgrades too expensive, the game could feel slow and boring; but if everything is too easy, the sense of progression is lost. I'll likely playtest and adjust numbers as I go to find a good balance.