Software Design Patterns Final Project Proposal

Hanuman Chu, Caden Swartz, Kaelem Deng, Pratik Asarpota

Project Summary

This project simulates an alien army spreading across a land(2D grid of tiles), each with distinct properties and resources. Over a series of time units/steps, the alien forces expand over the terrain, depleting resources as they go. The simulation supports multiple spreading strategies (Least Resistance, Random, Greedy) and various resource consumption behaviors (Instant, Constant, or Decreasing over time). The environment and initial conditions are defined through a JSON configuration file which specifies tile types, map dimensions, resource values, and alien force parameters. The simulation runs until a specified duration is reached, producing different logs at different levels (debug or default) and exporting the final state to a JSON file. This will be written in Java.

This is an overview of the functional components:

- 2D array of tiles
    - Each tile has properties to simulate upon
        - Spreading difficulty - how hard it is for aliens to move here.
        - Resources - aliens can consume this to grow their forces.
    - Certain tiles have extra properties that can happen at a random time
        - Rebel forces
        - Efficient resource generator
- Alien forces that spread across the grid over time steps with decisions made based on strategies
    - Spreading strategy - how aliens move their forces to other tiles
    - Resource consumption strategy - how aliens use tile resources to grow
- Environment/initial conditions specification via a JSON file
- Simulation output
    - Logging for each time step
    - Outputting the final state information to json file

Key Challenges

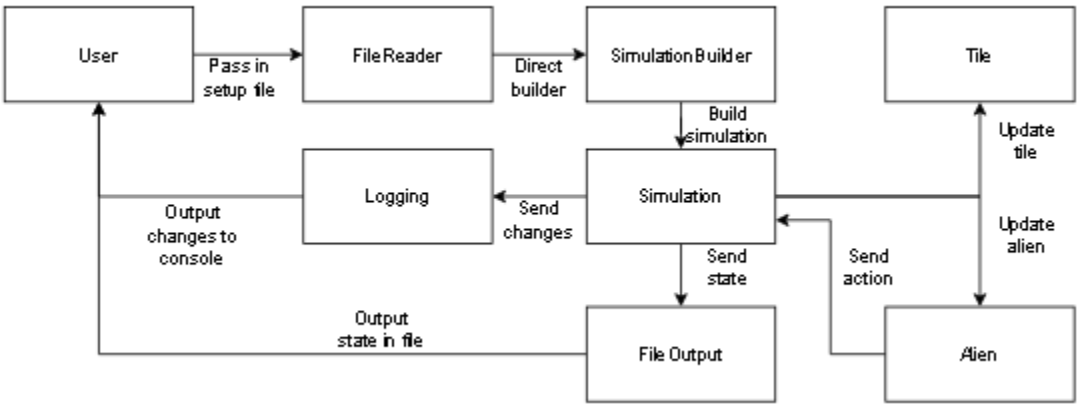
- Complex object initialization (Builder) - Because we will create our simulation environment based on user specified files, we will need a sophisticated and modular way to initialize our simulation.
- Multiple kinds of alien forces (Strategy) - Because we want there to be different kinds of alien forces, we need these to behave differently while interacting with the same interface, and having the same information available.

Planned Patterns

Structural - We will use the Decorator pattern for tiles with some examples being a rebel presence affecting spreading, friendly biosphere that has more efficient resource usage, etc. with the possibility of multiple at a time.

Creational - We will use the Builder pattern to create the simulation itself, based on the specifications from an input file. Using a builder means that we can separate out the construction of the simulation from its dynamics during the simulation time steps. Thus, we can easily construct the simulation environment from a file specification without changing underlying simulation code.
Behavioral - We will use the Strategy pattern to allow for simulations to handle different kinds of alien behavior. Some aliens may try to spread out evenly, while others may prefer to move all their alien forces to the most vulnerable tile. The Strategy pattern allows us to slot in all these different behaviors while reusing the same structural code.

Subsystems Diagram



Roles

- Pratik Asarpota - User input/output
    - File specification definition
    - Final state output
- Caden Swartz - Building the simulation environment and main loop
    - Creating tiles, alien forces
- Hanuman Chu - Updating tiles/alien behavior
    - Alien Strategies
    - Interaction with environment
- Kaelem Deng - Logging + Statistics
    - Resources/aliens left
    - Rate of change of resources/aliens