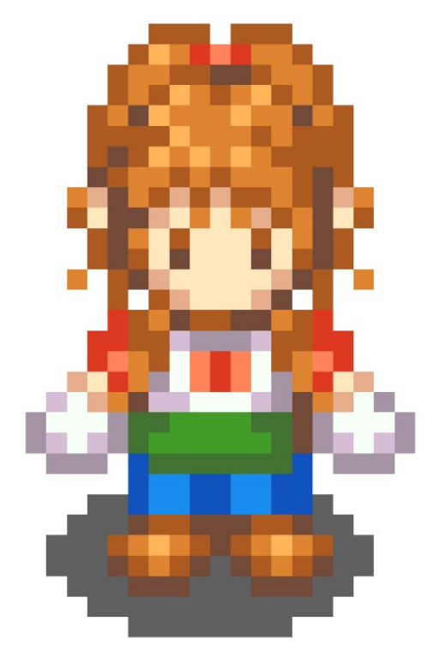


Harvest Moon 2.0

Kendall Haworth, Computer Science

Mentor: Dr. Adam Summerville

Kellogg Honors College Capstone Project

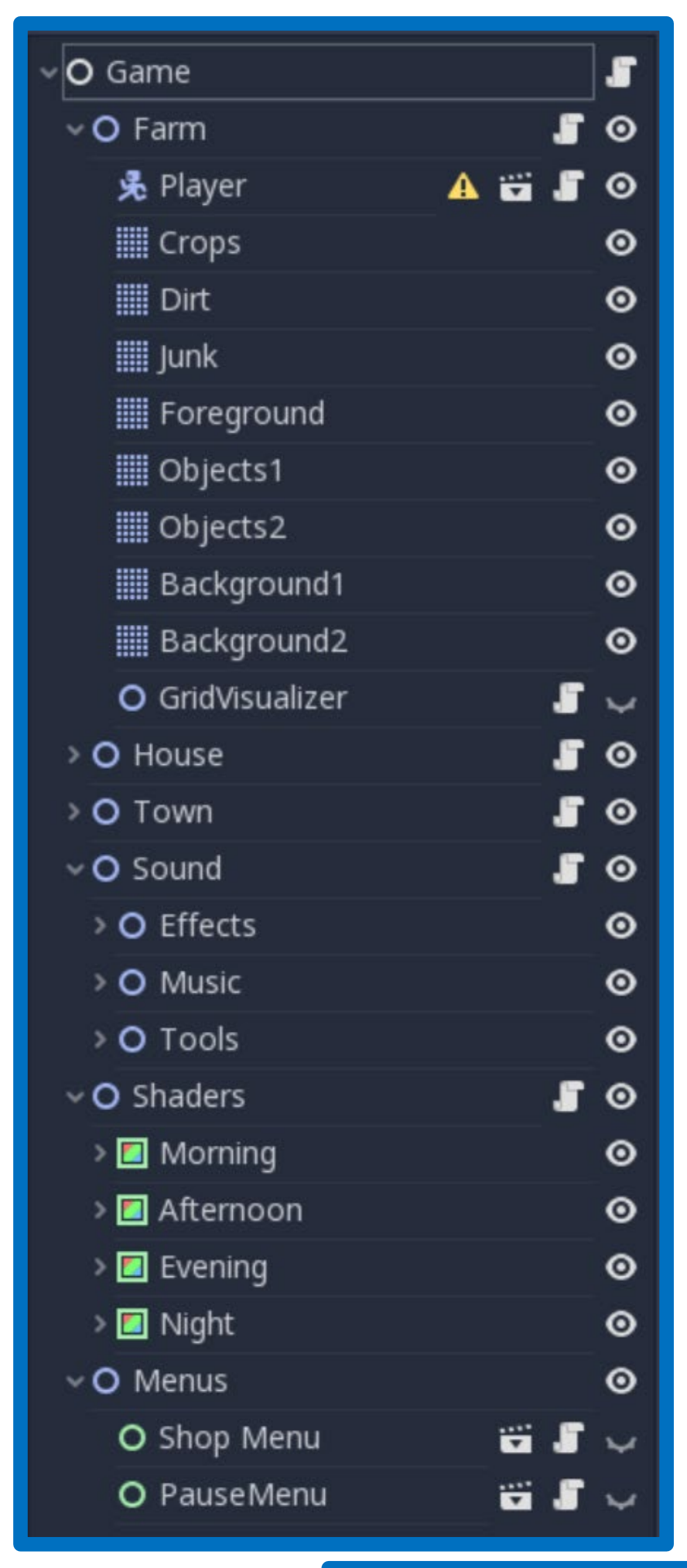
Introduction

Harvest Moon 2.0 is an open-source farming and exploration video game developed in Godot. It is directly inspired by *Innocent Life: A Futuristic Harvest Moon*, a PSP video game. The purpose of this capstone project was to further my own skills in software engineering and game development.

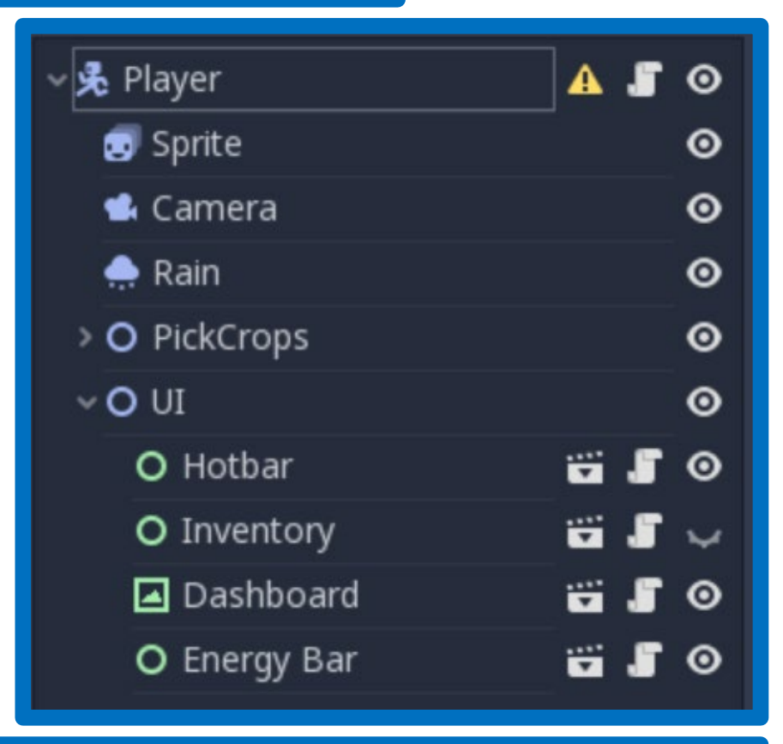
Features

Harvest Moon 2.0 features a full game loop and the following gameplay mechanics:

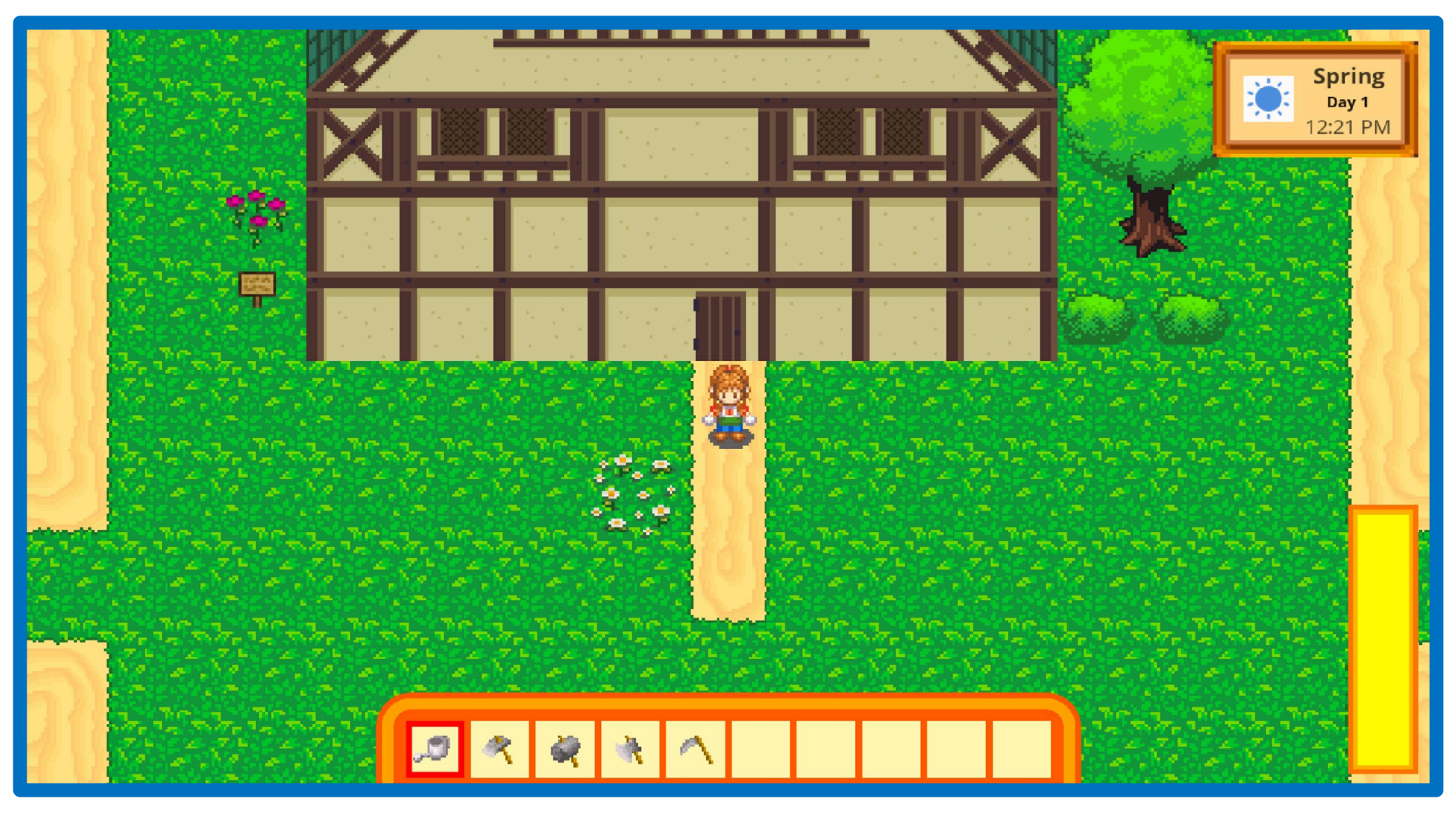
- Plant, water, and harvest crops
- Visit the town to sell your crops and buy seeds and tools
- Sleep in bed to progress to the next day
- The player will become tired and forced to sleep if it grows too late in the day
- Logs, weeds, and stones clog your fields over time - clear them!
- The time of day gradually changes from morning to night
- Weather changes between sunny and rainy
- Custom inventory system for equipping tools and organizing items
- Save Game system - you will never lose your progress between sessions, pick up right where you left off
- Energy Bar - working uses energy! Sleep to regain your energy

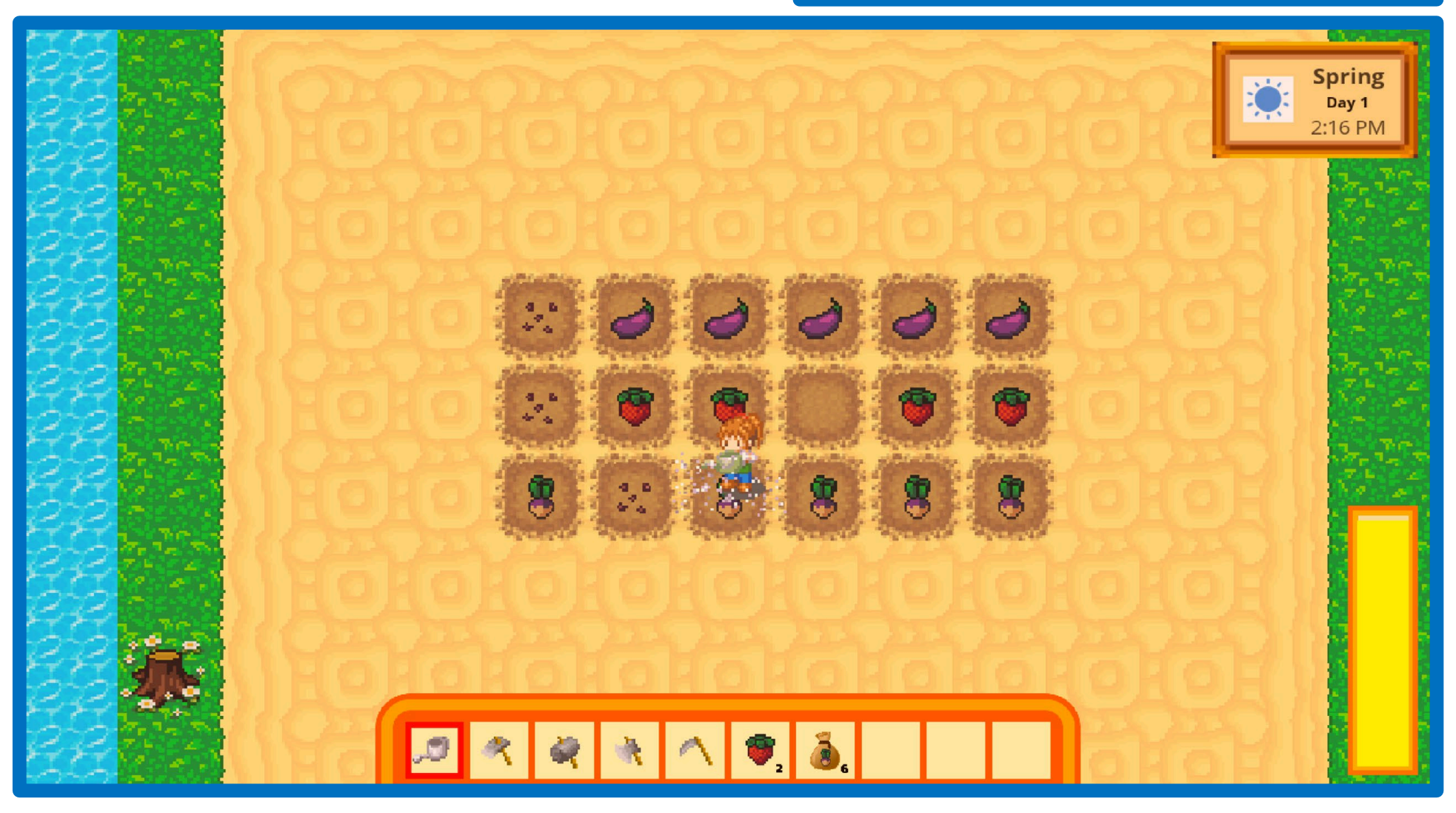


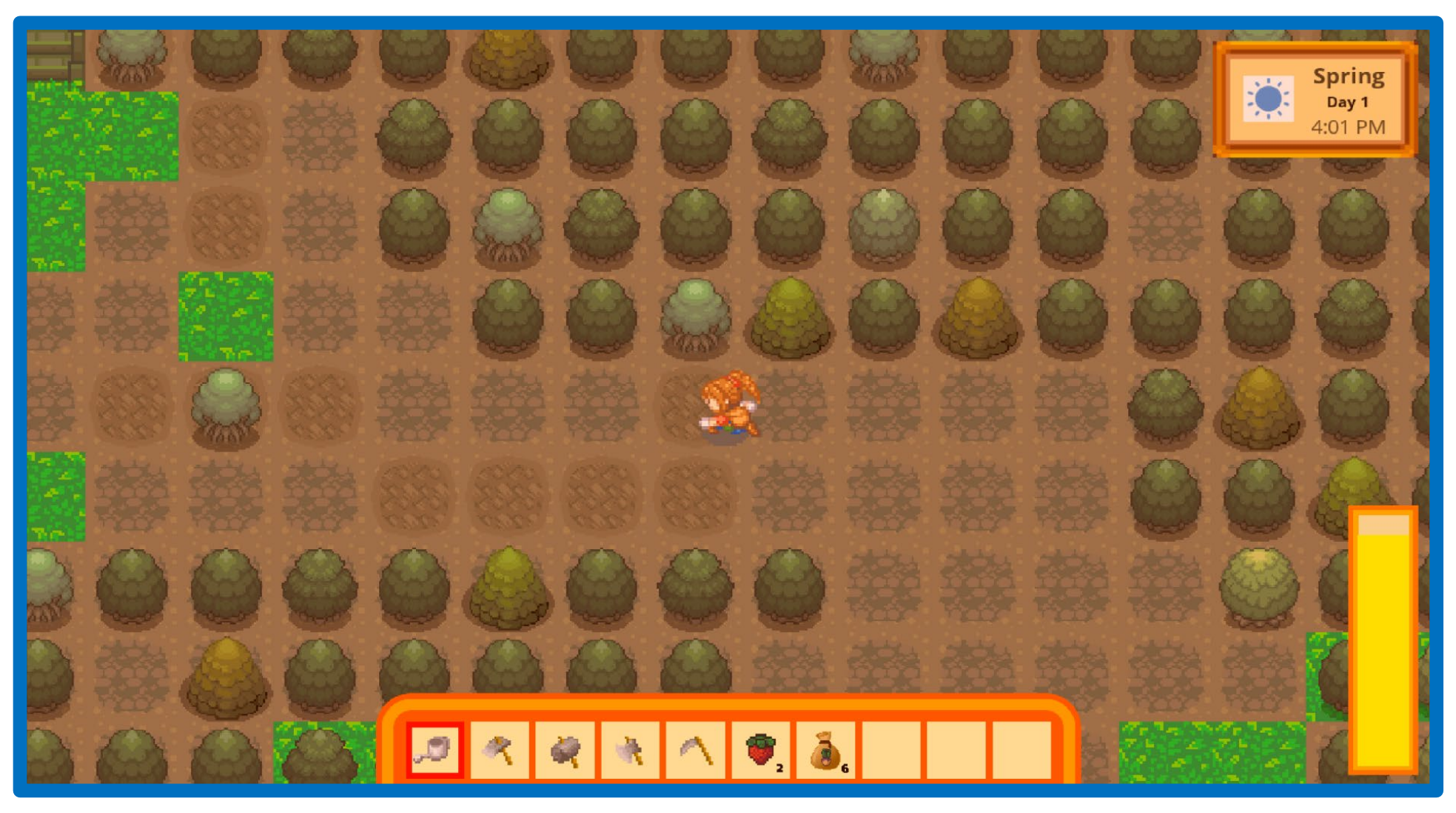
```
Game
  Farm
    Player
    Crops
    Dirt
    Junk
    Foreground
    Objects1
    Objects2
    Background1
    Background2
    GridVisualizer
  House
  Town
  Sound
    Effects
    Music
    Tools
  Shaders
    Morning
    Afternoon
    Evening
    Night
  Menus
    Shop Menu
    PauseMenu
```

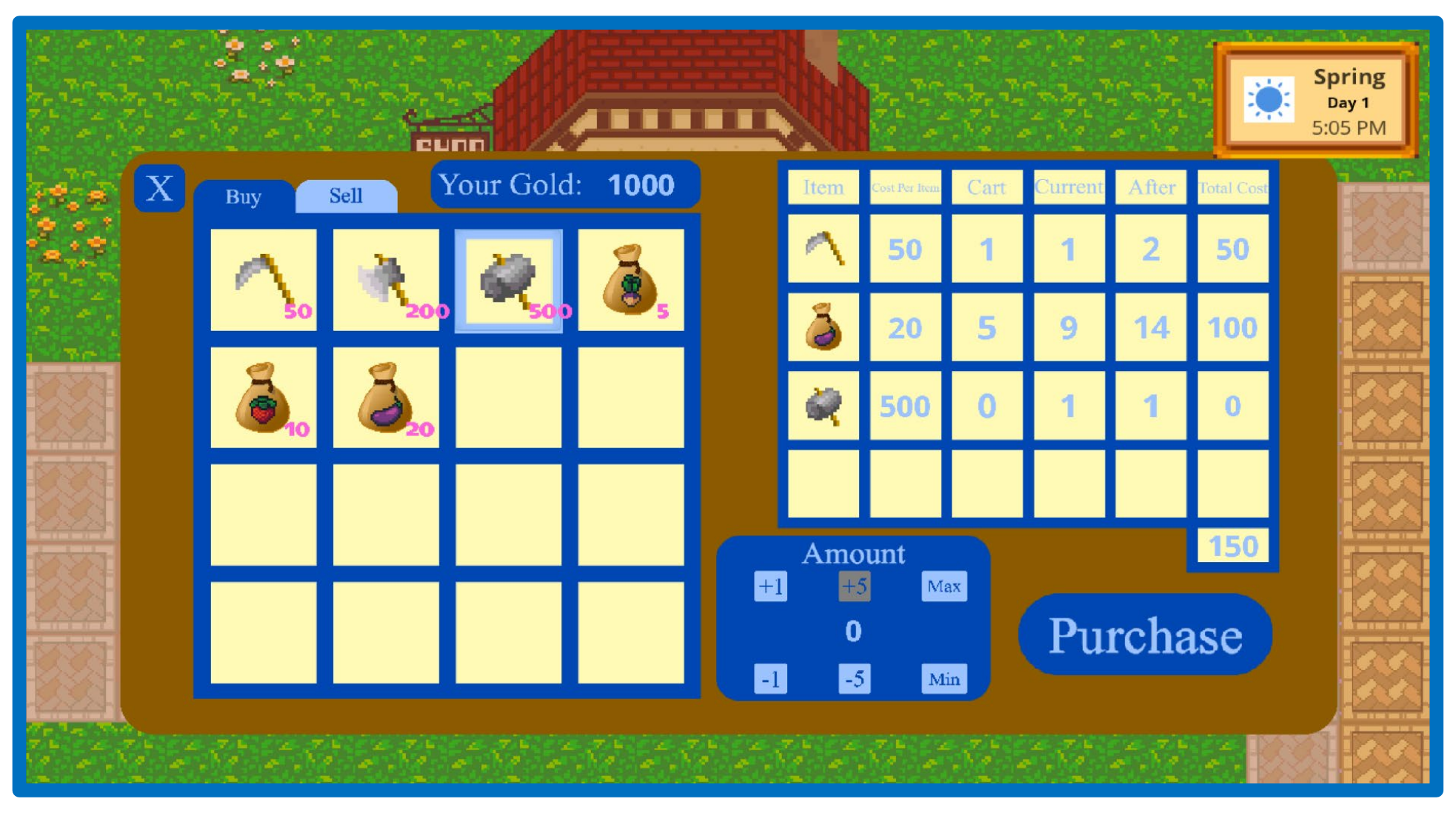


```
Player
  Sprite
  Camera
  Rain
  PickCrops
  UI
    Hotbar
    Inventory
    Dashboard
    Energy Bar
```









Techniques

This section will discuss various techniques used to implement some of the features in the game.

Graphics
Harvest Moon 2.0 uses 32x32 pixel tiles to create the entirety of the environment. The player's sprite and animation frames were taken directly from a sprite sheet. The inventory, shop menu, pause menu, and other menus were all custom made with the help of graphics software such as GIMP and paint.net.

Sound
Sound in Godot is played through an "AudioStreamPlayer" node. However, these nodes have no pause and resume feature, which is necessary when accessing the pause menu or loading a save. If an audio node is stopped and played again it will restart from the beginning of the audio file. Fortunately, audio nodes can report on their playing status, their playback position, and be started at a particular time stamp. Thus, when pausing or saving the playback position, all sound nodes are looped through to check their playing status. If they are playing, they are saved to a dictionary in the format (Sound Node Name, Playback Position). The proper nodes can then be resumed when the player resumes the game or the game is loaded back from a save file.

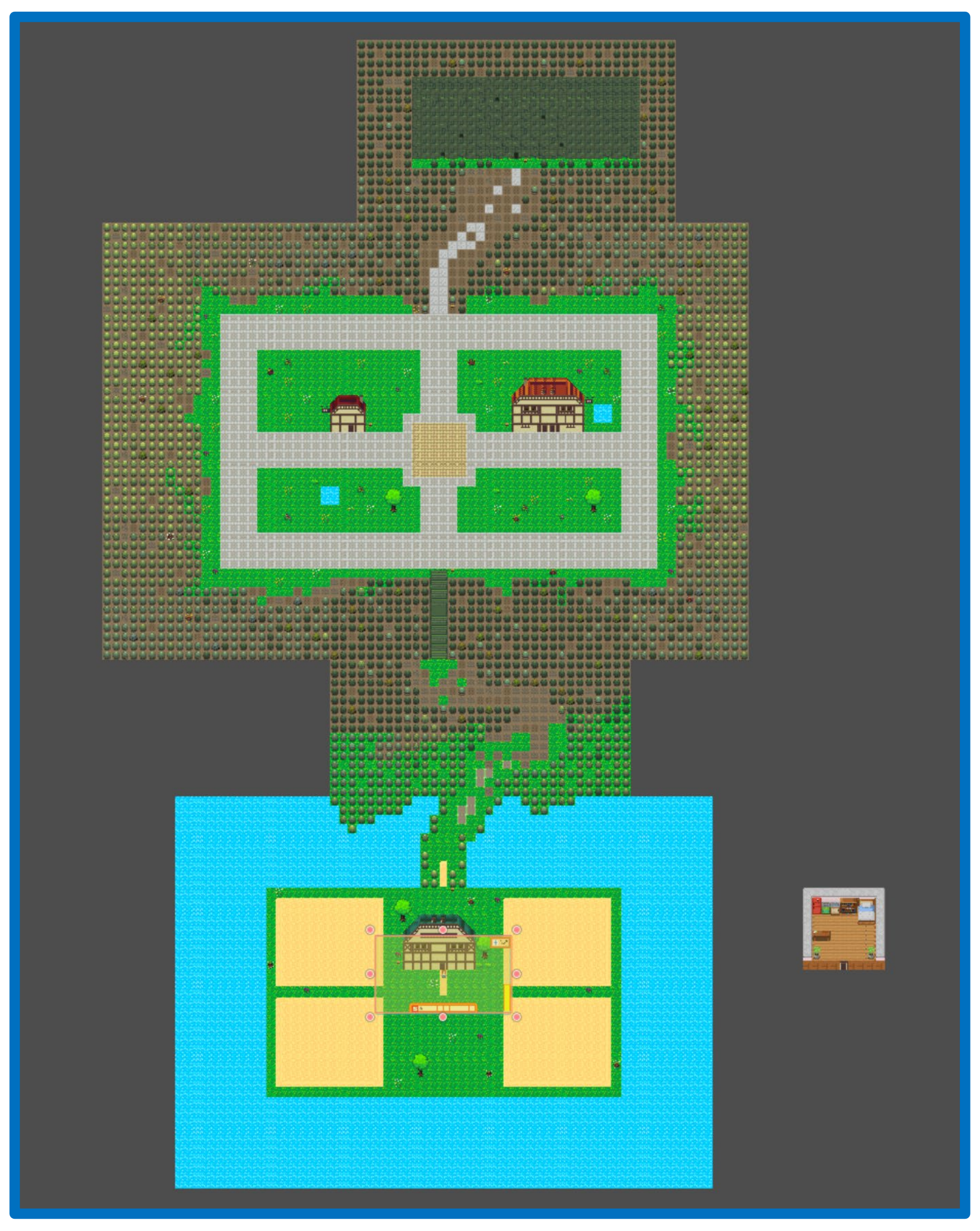
Save Game
In order to save the player's progress, all non-constant variables and objects must be saved and reloaded upon launching the game again. This data is saved in a json format in a .txt file and read back in upon choosing to reload the save. However, this presents a problem. Primitive variables, such as strings, floats, integers, and booleans, can be directly saved to the text file, but objects, such as images, sounds, and shaders, cannot be directly saved. The workaround is to save data about these objects that indicates their current state so that they can be reinstantiated when the game is loaded. For example, rather than saving the image, the name of the image can be saved (e.g., hammer) and then the proper image can be reloaded based upon the image name.

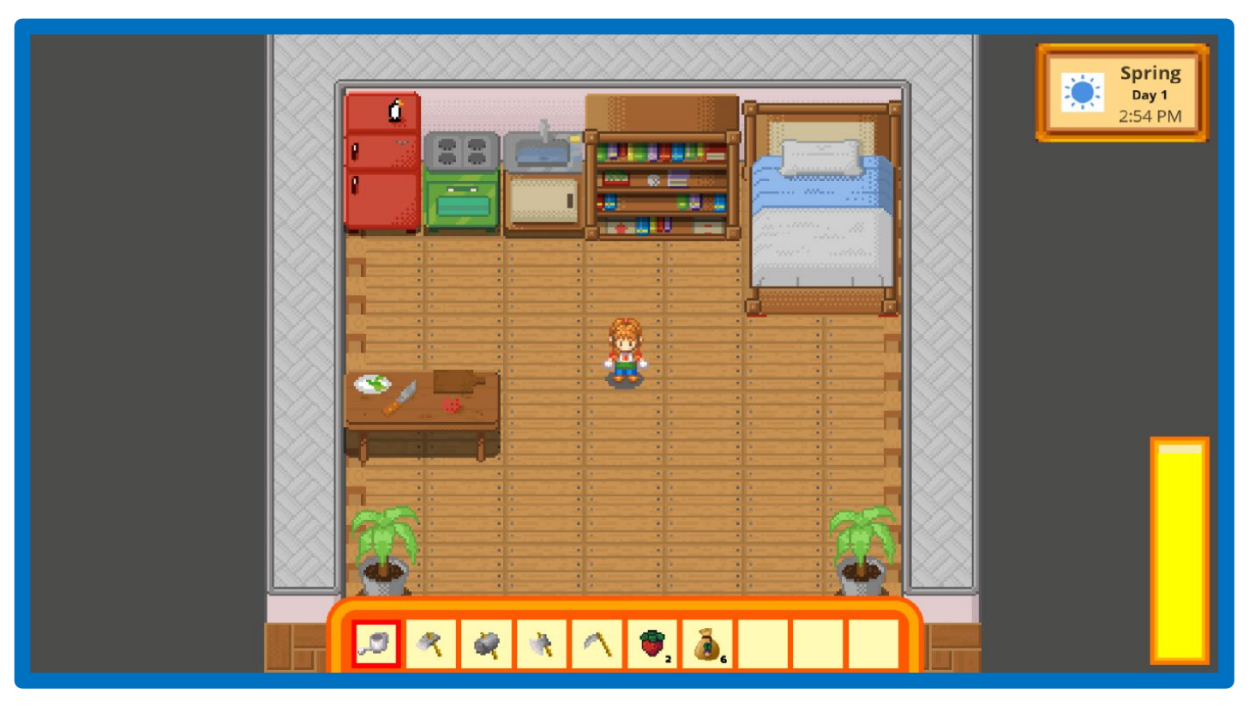
Shaders
Harvest Moon 2.0 features a full day and night cycle. The day starts out slightly yellow as in early morning sun rays and slowly burns off to full brightness. As evening approaches the world becomes orange and red, and then this fades into the deep blue and black of night, as shown in the images to the left and below. To achieve this effect, a yellow, red, and blue image are overlayed on the screen. Their opacity is increased and decreased by slowly tweening their values according to the time of day.

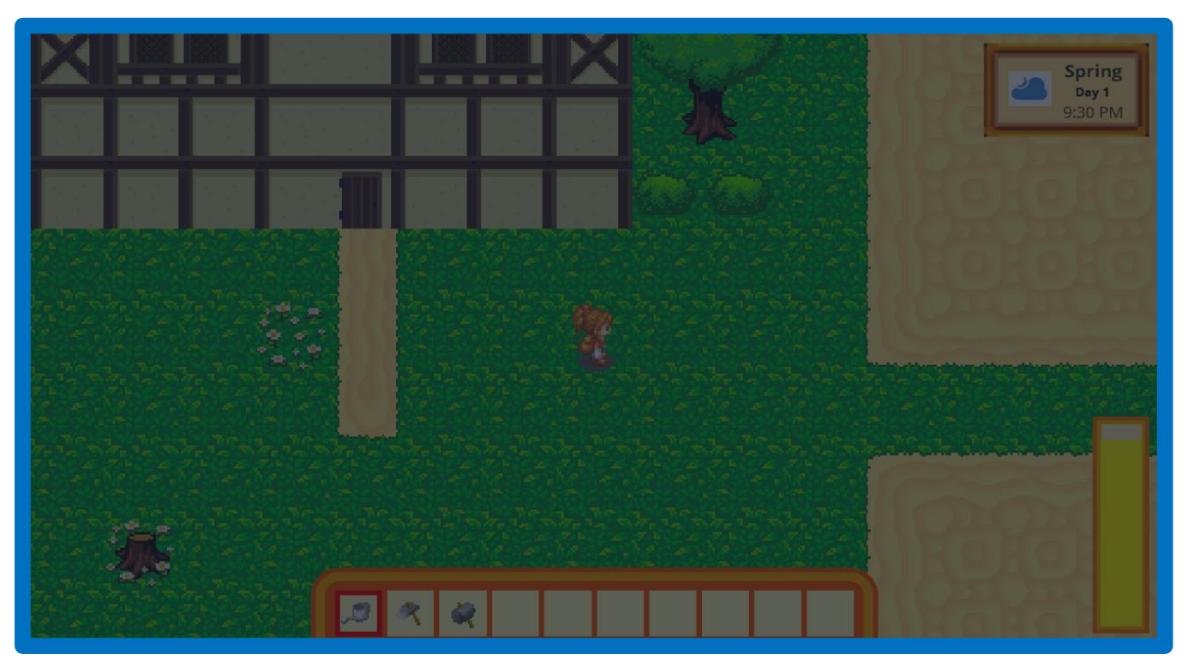
Development

Harvest Moon 2.0 was developed over the course of 10 months and has a total of 1,613 lines of code. Godot uses GDScript, a Python-lite backend scripting language, and focuses development around nodes arranged in a tree-like fashion. Different types of nodes have different properties and represent and make up everything in the program's domain – camera angles, sprites, images, animations, text, grids, buttons, special effects, and so on. At the top is an example of the node trees from the root node "Game" and the "Player" node.

A key aspect of Godot is that anything that can be done through the node editor can also be done through the script. The entire game could have been created with a single starting node with a very long script attached. However, nodes that exist for the entirety of the program and whose properties do not change during the execution of the program can be manually placed in the tree and have their properties set directly, without the need for any programming. It is best for organization, clarity, and efficiency to only use script when necessary and to instantiate all nodes with their starting properties in the editor when possible. Thus, Harvest Moon 2.0 has 511 manually created nodes, though only 28 have custom code.







Release

Harvest Moon 2.0 is available for Windows, Mac, and Linux. The source code and game can be downloaded and played for free at https://github.com/Kenny-Haworth/Harvest-Moon-2.0.