

### **TP3 Update**

Design changes:

- Added cooking mechanic / “home” location in game where player can obtain items from cabinet/fridge, cook recipes, and go to sleep to move onto next day
- Added inventory that stores fruits/other food items and recipe book
- Clouds only pass by once to make gameplay cleaner and less laggy
- Journal can now display multiple pages and colors change depending on past 3 days (in game world)
- Did not implement sitting mechanic / birds / picking flowers

### **TP2 Update**

Changes to algorithmic plan

Perlin noise - No longer using perlin to draw trees, only clouds. Trees are hand-drawn. Clouds now change depending on different “octaves” of perlin noise.

Changes to design of journal - am now reading / writing to three external files that are included in the zip file

No other changes

### **Project Proposal:**

#### **Project Description:**

Title - Perennial summer

My project is simulation 2D isometric game that takes place during the summer. The user plays as a girl navigating an infinitely generated terrain composed of hills, trees, grass, and clouds. The game is aimed to be peaceful and reflective; one of the main features for user interaction is a journal where the user can write down their thoughts; the game’s colors change in response to emotions the user displays. In addition, the user can interact with the landscape by performing actions such as picking fruits and sitting down.

#### **Competitive Analysis :**

Of course, my project is a much smaller scope than many of the similar projects I’m highlighting, but my main inspiration was drawn from the games Animal Crossing, farming simulation games like Stardew Valley and Harvest Moon, and Minecraft.

Animal Crossing/ Stardew Valley -

My game is similar to Animal Crossing in that it focuses on giving the user a relaxing experience with a focus on nature imagery. However, Animal Crossing focuses on allowing the user to change their island/town/place of inhabitation by altering the landscape and inviting fellow villagers; landscape alteration is not a focus of my game; instead, I want to focus on the user’s internal thoughts, so in a way my project is less “externally” focused. There are still some similarities as the player will be able to pick fruit from trees and pick up flowers, both of which are

heavily influenced from AC mechanics. Similar to Animal Crossing are farm simulation games. I was inspired by these games because of their day/night cycles; in addition, Stardew Valley records the passage of time through seasons and days; a similar mechanic appears in my project.

Minecraft -

My project is similar to Minecraft in that it has an infinite generation feature and is mainly built off of “blocks”. However, my project is in isometric (2.5D) instead of 3D, and the user cannot modify the terrain directly. My project also focuses less on exploration than Minecraft, instead enabling the user to see their internal state reflected in the landscape on the screen.

### **Structural Plan:**

The entire game’s code takes place within main.py:

Main.py:

- appStarted/restartApp
- Functions for diamond square algorithm - generates terrain height
  - diamondSquare, calculateDiamondValues, addSquaresToList
- Functions for perlin noise - generates cloud and tree shapes
  - calcGradVec, calcDotProduct, interpolate, normalize, perlin
- 2D to isometric or iso to 2D conversion functions
  - getCellBoundsinCartesianCoords, 2DtoIso
- Functions for journal
  - displayJournal
  - detectWords - detects certain words corresponding to different emotions (ex. “Happy” or “sad”) and changes app variables that control the game’s colors
- User input detection functions
  - keyPressed detects player movement and also input for journal
- Drawing functions
  - drawBoard, drawPlayer, drawCell, drawGrass, drawHome
- Additional functions for changing colors/ aesthetic purposes
  - drawColors, drawFlowers, drawFruit

### **Algorithmic Plan:**

The two parts of my project that are most algorithmically complex are the terrain height generation and tree/cloud shape generation. For the terrain height, I am using the diamond square algorithm, and for the tree/cloud shape generation I am implementing perlin noise.

Diamond square algorithm - This algorithm is used for randomly generating heights for a terrain so that it looks natural. It is composed of a “diamond step” and a “square step”, which together generate an array of “heights” (integers) for any array of size  $2^n + 1$ . My diamond square algorithm is composed of a main recursive function, in which the base case returns the filled board, and the recursive step performs the diamond step/square step. It uses two helper functions calculateDiamondValues (square step) and addSquareToList (diamond step). When combined with my drawing function, it generates random heights that give the shape of low-poly hills.

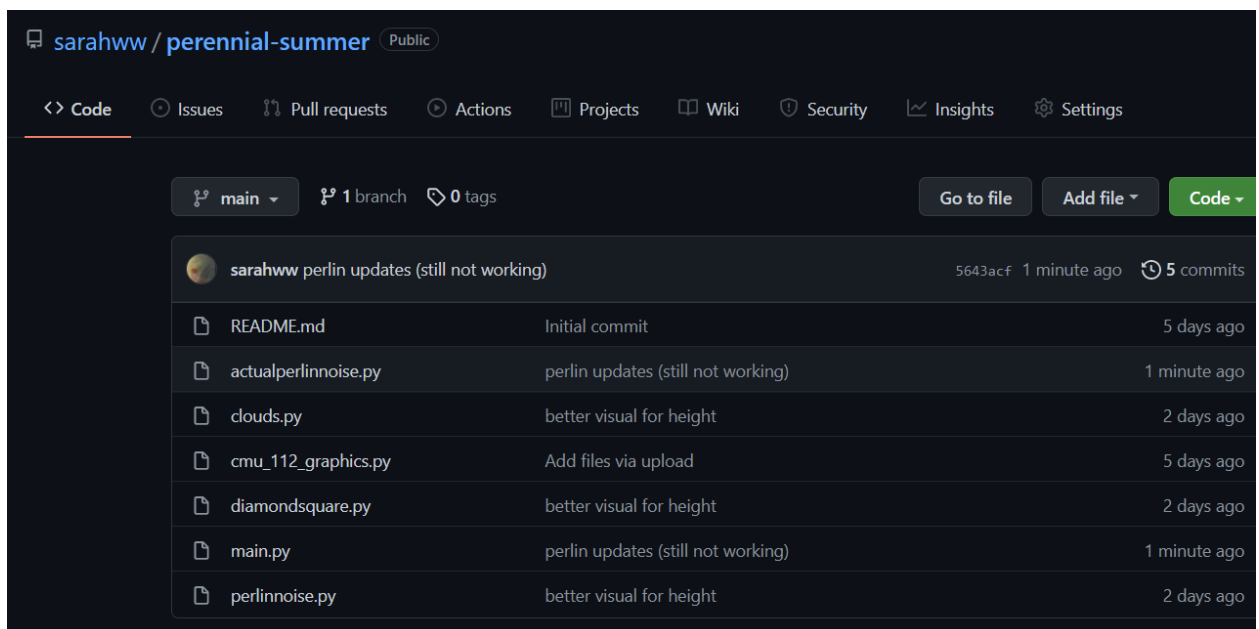
Perlin noise - This is a type of gradient noise that is used for generating a variety of natural effects such as fire, clouds, and island shapes. It uses a “grid” and generates a value from 0 to 1 for every value within that grid (including decimals). My perlin noise function is composed of the main perlin function which takes in a pixel (pX, pY), which generates 8 different vectors (gradient vectors / offset vectors based off the four corners), and interpolates them using a linear interpolation function to return a final value. This value is then multiplied by 255 and converted into rgb format (val, val, val) representing a value with 0 saturation (some shade of gray). When this is done for every pixel, it generates a black/white image that appears somewhat “cloudy” with alternating black and white areas that blend together. To use this for cloud generation, I plan to turn the black areas into blue and keep the white areas, giving the appearance of clouds. For tree shape generation, I plan to generate perlin noise with less detail so there are more “blocky” shapes in the noise, and take the black areas for tree shapes.

#### Timeline Plan:

- By Tp1 - Diamond square, (somewhat working) perlin noise, basic journal, draw grass
- By Tp2 - Changing colors in response to words in journal, aesthetic refinements for perlin noise, gameplay optimization (issues w/ drawing many objects at once)
- By Tp3 - Refinements for different emotional states in world (ex. Adding flowers, fruit, birds (this would be mostly hand-drawn art)), add player sprite, add title screen and end screen, add music

#### Version Control Plan:

I am using Github Desktop to update my files to a repository I created on Github. I am able to revert changes to previous versions if anything messes up.



Module List : None

