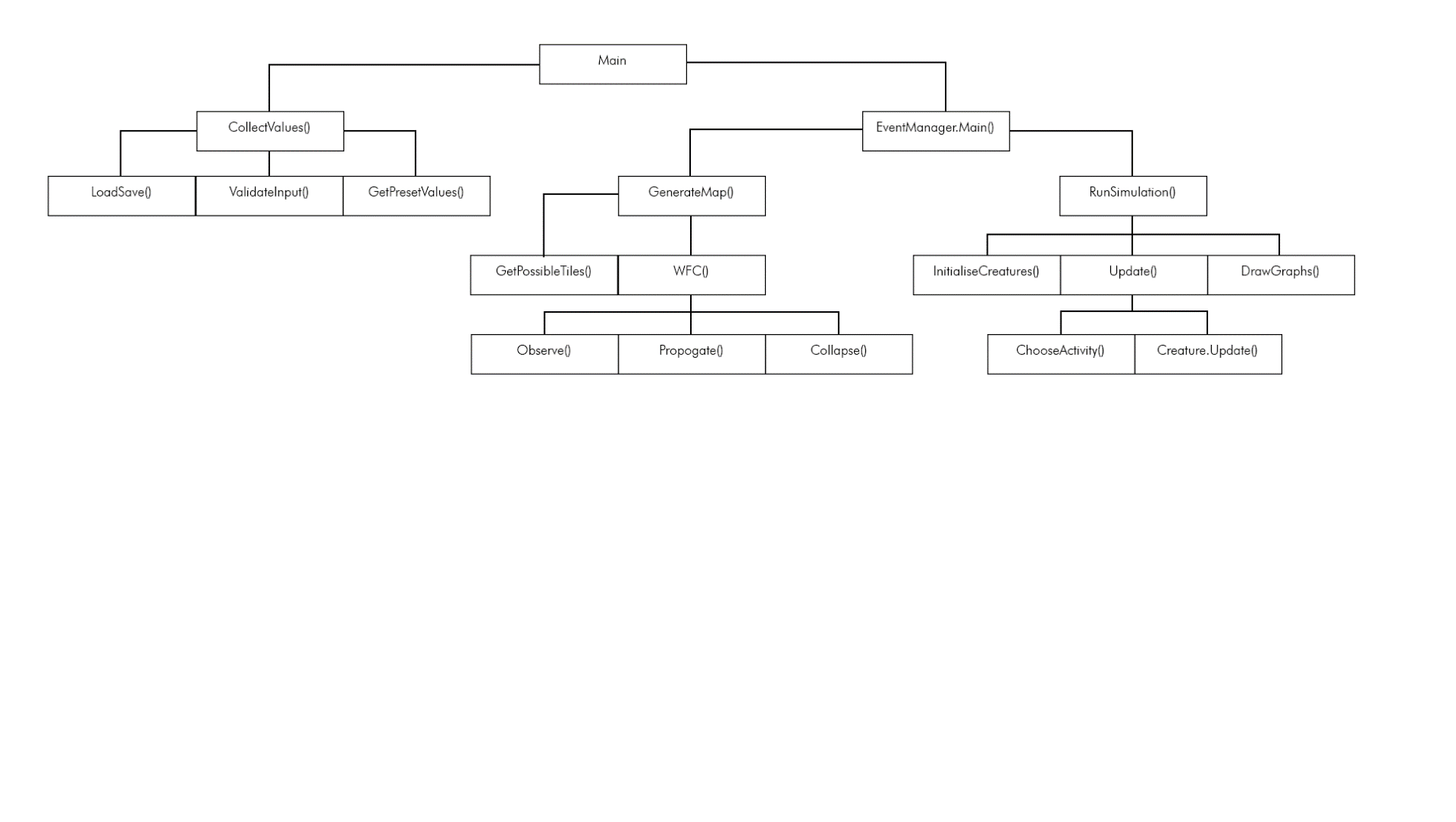
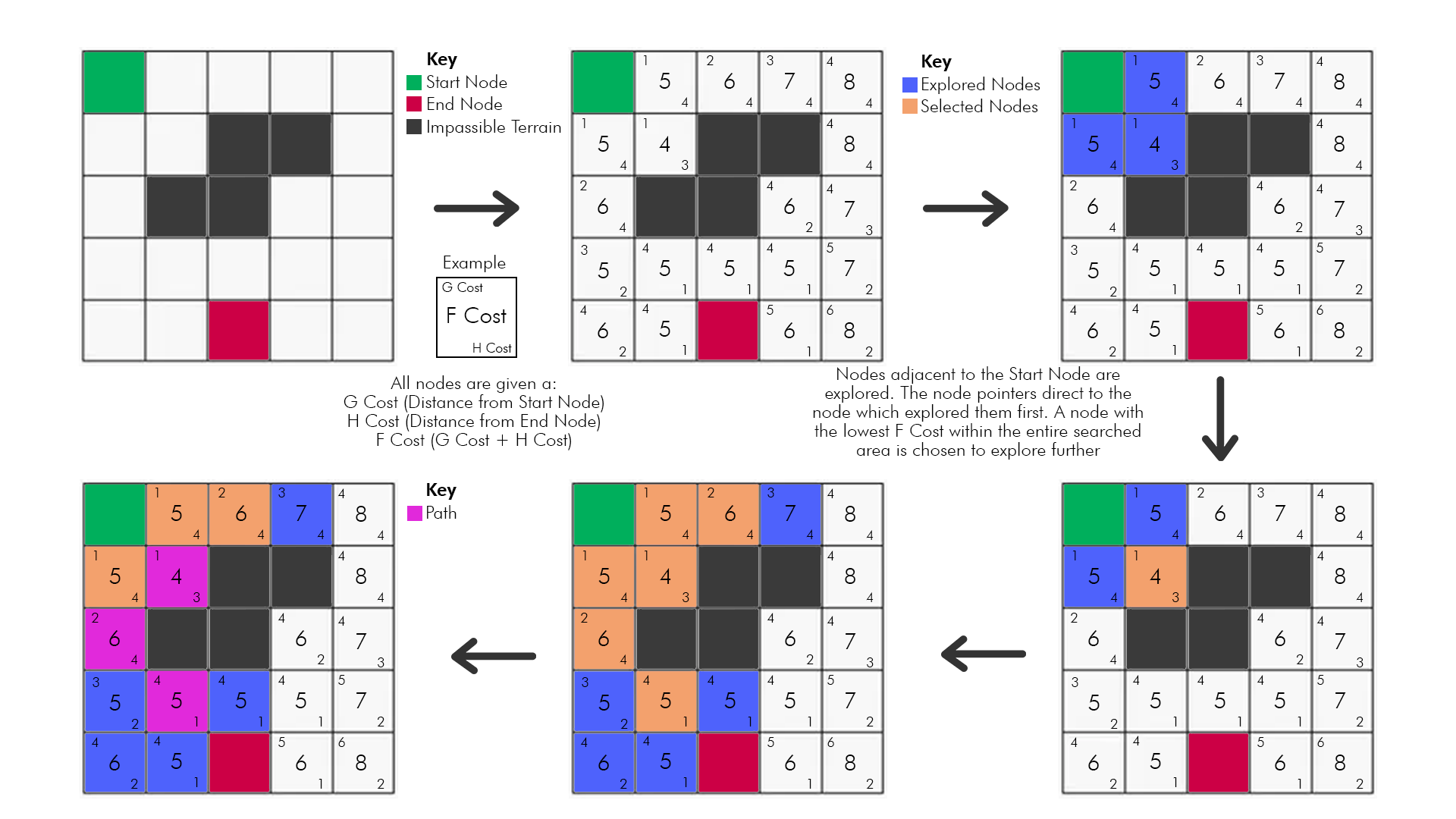
Design

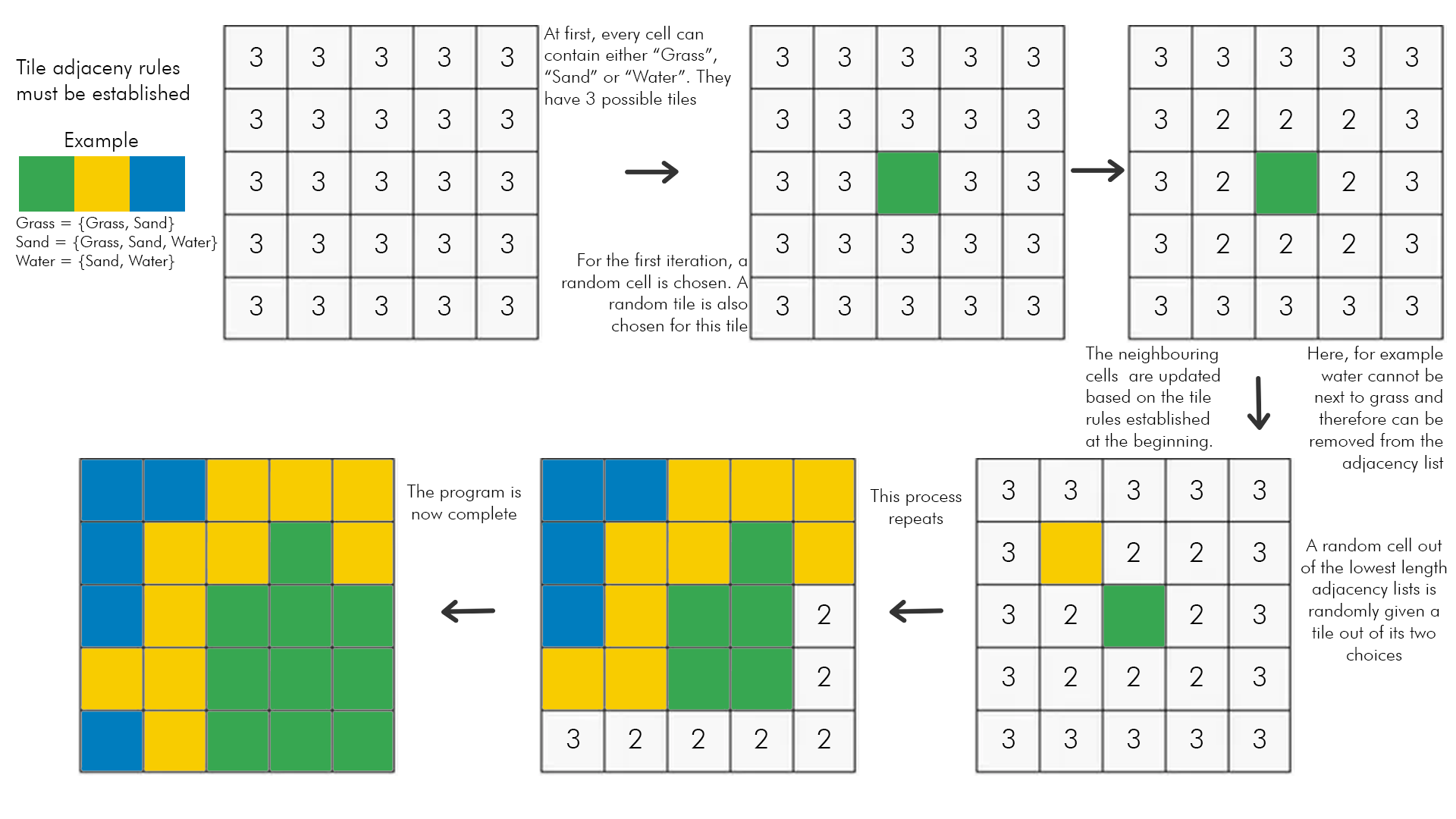
Overview

Algorithms



A Diagram to show A\* Pathfinding

While the simulation is running, each creature will make a decision on what they can do, either wandering, eating, or reproduction. This algorithm is used during the “ChooseActivity()” function which choses one of these 3 options and pathfinds towards it’s objective. A location is given as a parameter, then a queue of steps to take from your current location is returned,

Diagram to show my “Wave Function Collapse” algorithm.

This algorithm is called in the “GenerateMap()” function. It takes in a JSON file as a parameter, where it randomly generates a map from the data within the JSON file. The data includes, weights, tile adjacency rules, biases and each tiles name and image. The algorithm runs as shown above. A 2D list of Tile objects, the exact size of the world is returned, where index [0][0] is the top left tile.

Creature Update Algorithms

Creature update is called once per frame of the simulation. It is used to perform all actions a creature can do. It first checks if the creature already has a set destination, meaning an activity has already been chosen. If the creature does not, it makes 1 of 3 choices. Either to eat, wander or reproduce. This choice is randomly made, but the chances of each option happening are based on weights which are calculated on the creatures current needs. A higher need will give a higher weight and therefore a higher chance of performing an action which will satiate their needs. “Wandering” will choose a square within the users chosen distance. “Reproduce” will make the creature look for others who are wanting to reproduce and locate an opposite sex. The female then waits, while the male pathfinds to their location. If the creature is a prey and “Eat” is chosen, the creature will pathfind to the nearest berry which has not been chosen yet. If no berries exist or they all have other prey on the way to it, the prey will wander instead. If the creature is a predator however, it will pathfind to the nearest untargeted prey. If the prey moves away, the tiles the prey step on are added to a stack. After the predator’s movement queue has been emptied, if the prey is still alive, the predator will use the stack to hunt down the prey.

GUI algorithms

The GUI is loaded within Tkinter frames. The preset data is loaded from a JSON file which is held within the “Saves” folder. The data is loaded into a dictionary for use within entry boxes. Saving works a similar way, where the data is taken from the entry boxes and then loaded into a dictionary, where it is dumped to a new JSON file with a name specified by the user. Input to boxes is validated through either a integer or float callback. This ensures that the data inputted is either an integer or a float depending on the entry box. If no value is inputted, it is caught by a try-except where it returns an error to the user.

Graph algorithms

Graphs are made using MatPlotLib, a library for python. The graphs are plotted using data from event manager, it tracks the count of each creature each frame, as well as the average gene strength.

Equations

Some of the main equations I use within my program are as follows:

Literal position = (Cell.x \* CELLSIZE, Cell.y \* CELLSIZE)

Distance between two points = ((item1.x-item2.x)2+(item1.y-item2.y)2)1/2

Weight of “Eat” choice= (BASE\_ENERGY -currentEnergy)

Weight of “wander” choice = (currentEnergy/BASE\_ENERGY)+(currentUrge /URGEREPRODUCE)\*10

Weight of “Reproduce” choice = (URGE\_REPRODUCE-currentUrge)\*0.1

Data Dictionary

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Used for |
| world | 2D list | Holds all the Cell objects in the map at their position relative to others. |
| creatureworld | 2D list | Holds all the Creature cell objects in the map relative to their position in “world” |
| berryList | List | List of all berrys |
| preyList | List | List of all prey objects |
| predatorList | List | List of all predator objects |
| renderer | Renderer | Holds the current world renderer |
| engine | Engine | Holds the current world engine |
| tilelist | List | List of tile objects which are possible in this world instance |
| tiledict | Dictionary | Dictionary of all tile types and their tiles |
| fertilelist | List | Holds all the cells which are fertile ground |
| spawnableList | List | List of all cells where creatures can spawn |
| deadPredatorList | List | List of all dead predators |
| deadPreyList | List | List of all dead prey |
| preyLookingForMate | List | List of all prey objects currently looking for mates |
| predatorLookingForMate | List | List of all predator objects currently looking for mates |
| preyListLength\_perframe | List | List of the amount of prey in each frame |
| predatorListLength\_perframe | List | List of the amount of predators in each frame |
| position | Tuple | Holds the position of the object relative to the grid |
| tile | Object | Stores the current tile of a cell |
| f\_cost | Integer | Stores the sum of a cell’s distance from two points |
| traversable | Boolean | If the tile is traversable or not |
| currentpath | Stack | The path a creature would have to follow in order to reach a position |
| foodTarget | Cell | The target cell of the target food |
| mate | Creature | The target mate of the creature |
| sex | String | The sex of the creature |
| alive | Boolean | If the creature is currently alive |
| age | Integer | The age in frames of the creature |
| timebetweenmates | Integer | The time after last mate |
| lookingForMate | Boolean | If the creature is currently looking for a mate |
| extramovement | Queue | A queue of steps the creature will have to take to adjust its target |
| img | Pygame.surface | The creature image |
| preyTarget | Prey | The food target of the predator |
| mateTarget | Cell | The cell position of the mate |
| energy | Integer | The current energy level of the creature |
| urgeReproduce | Integer | The urge of the creature to reproduce |
| deathage | Integer | The age that the creature will die of old age |
| possibleTiles | List | List of possible tiles a cell can have |
| bias | Dictionary | Dictionary of tiles and the biases those tiles have |
| hasTarget | Boolean | If the cell has a creature targeting it |
| adjacencyList | List | A list of tiles which a tile can lay next to |
| name | String | Name of the tile |
| weight | Integer | The weight the tile has when a tile is randomly being chosen |
| fertile | Boolean | If a tile is fertile and can grow berries or not |
| dt | Float | The time elapsed since last frame |
| tiles | Json | A json file containing the image data, name, adjacency list, bias, weight and it’s traversability |
| FPS | Integer, Constant | The amount of frames per second the program will cap at |
| SCREENWIDTH | Integer, Constant | The width of the screen in pixels |
| SCREENHEIGHT | Integer, Constant | The height of the screen in pixels |
| CELLSIZE | Integer, Constant | The width and height of each cell in pixels |
| WFCWEIGHTS | List, Constant | Weights of a tile being chosen |
| MAXWEIGHT | Integer, Constant | The maximum weight a tile can hold |
| MINWEIGHT | Float, Constant | The minimum weight a tile can hold |
| MAIN\_FOLDER | Directory, Constant | The location of the main folder of the program |
| ASSETS\_FOLDER | Directory, Constant | The location of the assets folder of the program |
| TILES\_FOLDER | Directory, Constant | The location of the tiles folder of the program |
| CREATURE\_FOLDER | Directory, Constant | The location of the creature folder of the program |
| BERRYIMG | Directory, Constant | The location of the berry image |
| NONTRAVERSABLE\_MOVEMENT\_MODIFYER | Integer, Constant | The extra cost of walking over a non-traversable tile |
| BERRYCONST | Float, Constant | The rate of which berries spawn in the world |
| PREYCOUNT | Integer, Constant | The number of prey in the world |
| PREDATORCOUNT | Integer, Constant | The number of predators in the world |
| MAXWANDERDISTANCE | Integer, Constant | The maximum distance a creature can wander in one walk |
| BASECHOICEWEIGHTS | List, Constant | The base weights of a creatures decision making |
| MAXCHOICEWEIGHT | Integer, Constant | Maximum weight a choice can be changed by |
| URGELOSSPERSTEP | Integer, Constant | The urge to reproduce increase per step |
| SEXLIST | List, Constant | A list of possible sexes for the creatures |
| BASE\_ENERGY\_PREY | Integer, Constant | Starting energy for a prey object |
| MINDEATHAGE\_PREY | Integer, Constant | The minimum time in frames where the prey will die of old age |
| MAXDEATHAGE\_PREY | Integer, Constant | The maximum time in frames where the prey will die of old age |
| MAXOFFSPRING\_PREY | Integer, Constant | The maximum amount of offspring a prey can have |
| MINOFFSPRING\_PREY | Integer, Constant | The minimum amount of offspring a prey can have |
| URGE\_REPRODUCE\_PREY | Integer, Constant | The starting urge to reproduce of the prey |
| TIMEBETWEENMATES\_PREY | Integer, Constant | The minimum age between mates of prey |
| MINREPROAGE\_PREY | Integer, Constant | The time spent as a child before maturing |
| ENERGYLOSSPERSTEP\_PREY | Integer, Constant | The amount of energy lost per step for prey |
| BASE\_ENERGY\_PREDATOR | Integer, Constant | The starting energy of a predator |
| URGE\_REPRODUCE\_PREDATOR | Integer, Constant | The starting urge to reproduce of the predator |
| TIMEBETWEENMATES\_PREDATOR | Integer, Constant | The minimum age between mates of predator |
| ENERGYLOSSPERSTEP\_PREDATOR | Integer, Constant | The amount of energy lost per step for predator |
| MINDEATHAGE\_PREDATOR | Integer, Constant | The minimum time in frames where the predator will die of old age |
| MAXDEATHAGE\_PREDATOR | Integer, Constant | The maximum time in frames where the predator will die of old age |
| MINREPROAGE\_PREDATOR | Integer, Constant | The time spent as a child before maturing |
| Preset | Json | A json file of preset data for the simulation |
| Settings | Json | A json file of the current settings. |

File Structure

Inside the main folder is all main program files, as well as the asset folder.

Table

Description automatically generated

Graphical user interface, text

Description automatically generatedThe “assets” folder contains 3 more folders, leading to the tile images and ruleset, creature images, and map objects

Logo, company name

Description automatically generatedCreatures contains images of the creatures

Objects contains images of map objects

Graphical user interface

Description automatically generated

A picture containing diagram

Description automatically generatedTiles has both the images of tiles and the JSON file of the tiles

Settings contains the settings JSON file



Saves contains the preset save, as well as any user saves



Text, letter

Description automatically generatedPredator, Prey lifecycle

Libraries

I use Pygame for drawing images to the screen and input management for my program. I use Tkinter for my GUI and data input. I also use MatPlotLib to draw graphs from the data I collect in the program.

GUI overview

Main Menu

A picture containing graphical user interface

Description automatically generated

Create Simulation

Text

Description automatically generated

Save Parameters

Graphical user interface, application

Description automatically generated

World Viewer

A screenshot of a computer

Description automatically generated with medium confidence

Load Simulation

A picture containing chart

Description automatically generated

Settings

Chart

Description automatically generated

Class Hierarchy charts

|  |
| --- |
| Creature |
| +position: Cell  +worldmap: [[Cell]]  +currentpath: Stack [Cell]  +renderer: Renderer  +foodTarget: Creature  +mate: Creature  +sex: String  +alive: Boolean  +age: Integer  +timebetweenmates: Integer  +lookingForMate: Boolean  +extramovement: Queue [Cell]  +world: [[CreatureCell]] |
| +RequestMate(mate: Creature): Bool  +PotentialMateFound(mate: Creature): Bool  +GetDistanceBetween(item1: Creature, item2:Creature): Float  +CreateCreatureWorld():[[CreatureCell]]  +FindPath(target: Cell): Stack [Cell]  +Wander(spawnableList: [Cell]): Cell |

|  |
| --- |
| Predator(Creature) |
| +img: Pygame.surface  +preyTarget: Creature  +mateTarget: Creature  +energy: Integer  +urgeReproduce: Integer  +deathage: Integer |
| +LocateMate(lookingForMate: [Creature])  +AdvancePath(): Integer  +Hunt(preyList: [Creature]): Cell  +Update(preyList: [Creature], spawnableList: [Cell], predatorLookingForMate: [Creature]): Integer  +ChooseActivity(): String |

|  |
| --- |
| Prey(Creature) |
| +img: Pygame.surface  +hasPredator: Boolean  +predator: Creature  +energy: Integer  +urgeReproduce: Integer  +deathage: Integer |
| +LocateMate(lookingForMate: [Creature])  +AdvancePath(): Integer  +Forage(berryList: [Cell]): Cell  +Update(berryList: [Cell], fertileList: [Cell], spawnableList: [Cell], preyLookingForMate: [Creature]): Integer  +ChooseActivity(): String |

|  |
| --- |
| EventManager |
| +world: [[Cell]]  +berryList: [Cell]  +preyList: [Creature]  +predatorList: [Creature]  +renderer: Renderer  +engine: Engine  +tilelist: [Tile]  +tiledict: {Tile: [Cell]}  +fertileList: [Cell]  +spawnableList: [Cell]  +deadPreyList: [Cell]  +deadPredatorList: [Cell]  +preyLookingForMate: [Creature]  +predatorLookingForMate: [Creature]  +preyListLength\_perframe: [Integer]  +predatorListLength\_perframe: [Integer]  +preycount: Integer  +baseenergyprey: Integer  +mindeathageprey: Integer  +maxdeathageprey: Integer  +timebetweenprey: Integer  +energylprey: Integer  +predatorcount: Integer  +baseenergypredator: Integer  +mindeathagepredator: Integer  +maxdeathagepredator: Integer  +timebetweenpredator: Integer  +energylpredator: Integer  +berryconst: Float  +maxwander: Integer  +fps: Integer  +screenwidth: Integer  +screenheight: Integer  +cellsize: Integer |
| +SplitWorld()  +CreatureWorld(): [[Cell]]  +SpawnBerry()  +BerryUpdate()  +InitialiseCreatures()  +Update()  +Main()  +InitializeValues(preycount: Integer, predatorcount: Integer, baseenergyprey: Integer, mindeathageprey: Integer, maxdeathageprey: Integer, energylprey: Integer, baseenergypredator: Integer, mindeathagepredator: Integer, maxdeathagepredator: Integer, energylpredator: Integer, berryconst: Float, maxwander: Integer, preyTBM: Integer, predatorTBM: Integer)  +InitializeSettings(screenheight: Integer, screenwidth: Integer, cellsize: Integer, fps: Integer)  +TempMapViewer() |

|  |
| --- |
| Engine |
| +clock: Clock  +t1: Float  +dt: Float |
| +update\_dt() |

|  |
| --- |
| Tile |
| +img: Pygame.surface  +adjacencylist: [Tile]  +name: String  +bias: {Tile:Integer}  +tempbias: Integer  +weight: Float  +traversable: Boolean  +fertile: Boolean |
| +UpdateTileList(tilelist: [String]) |

|  |
| --- |
| Cell |
| +position: (Integer,Integer)  +possibletiles: [Tile]  +tile: Tile  +bias: {Tile:Integer}  +hasBerry: Boolean  +hasTarget: Boolean  +target: Creature |
| -\_\_cmp\_\_(other: Cell): Integer  -\_\_str\_\_(): String |

|  |
| --- |
| CreatureCell |
| +position: (Integer,Integer)  +tile: Tile  +g\_cost: Integer  +h\_cost: Integer  +f\_cost: Integer  +pointer: CreatureCell  +traversable: Boolean |
|  |

|  |
| --- |
| Queue |
| +fPointer: Integer  +rPointer: Integer  +list: [Cell] |
| +AddToQueue(item: Cell)  +RemoveFromQueue(): Cell  +ClearQueue()  +BackOfQueue(): Cell |

|  |
| --- |
| Stack |
| +stack: [Cell]  +size: Integer |
| -\_\_iter\_\_(): [Cell]  +AddToStack(item: Cell)  +RemoveFromStack(): Cell  +ClearStack() |

|  |
| --- |
| PathFinder |
| +world: [[CreatureCell]]  +creature: Cell  +target: Cell  +exploredcells: [Cell]  +path: Stack [Cell]  +PathFound: Boolean |
| +GetDistanceBetween(item1: Cell, item2: Cell): Float  +DetermineFandHCost()  +Explore(position: Cell): Boolean  +GetLowestFCost(): Cell  +PathFind()  +GetPath()  +InitiatePathFind() |

|  |
| --- |
| Renderer |
| +screen: Pygame.surface  +berryimg: Pygame.surface |
| +DrawCell(cell: Cell)  +ClearScreen()  +RenderWorld(world: [[Cell]])  +DrawCreature(creature: Creature)  +RenderBerry(cell: Cell)  +DrawText(text: String) |

|  |
| --- |
| GUI(tk.TK) |
| +title: String  +geometry: String  +validint: root.Register  +validfloat: root.Register  +titlefont: tk.Font  +textfont: tk.Font  +guifont: tk.Font  +errorfont: tk.Font  +container: tk.Frame  +fullscreen: Boolean  +frames: {Object: tk.Frame}  +presetdata: {String:Integer}  +preycount: Integer  +baseenergyprey: Integer  +mindeathageprey: Integer  +maxdeathageprey: Integer  +timebetweenprey: Integer  +energylprey: Integer  +predatorcount: Integer  +baseenergypredator: Integer  +mindeathagepredator: Integer  +maxdeathagepredator: Integer  +timebetweenpredator: Integer  +energylpredator: Integer  +berryconst: Float  +maxwander: Integer  +settingsdata: {String:Integer}  +fps: Integer  +screenwidth: Integer  +screenheight: Integer  +cellsize: Integer |
| +show\_frame(cont: Object)  +clear\_widgets(frame: tk.Frame)  +Quit(event: Object)  +toggle\_fullscreen(event: Object) |

|  |
| --- |
| MainMenu(tk.Frame) |
| -buttonwidth: Integer  -buttonrelief: String |
| +MoveToSimulationMenu(parent: GUI, controller: tk.Frame)  +LoadtoSimulationMenu(parent: GUI, controller: tk.Frame)  +Settings(parent: GUI, controller: tk.Frame)  +Quit(controller: tk.Frame) |

|  |
| --- |
| CreateSimulationMenu(tk.Frame) |
| -preyCount: tk.IntVar  -predatorCount: tk.Intvar  -preyBaseEnergy: tk.Intvar  -predatorBaseEnergy: tk.IntVar  -preyMaxDeathage: tk.IntVar  -predatorMaxDeathage: tk.IntVar  -preyMinDeathage: tk.IntVar  -predatorMinDeathage: tk.IntVar  -preyEnergyLoss: tk.IntVar  -predatorEnergyLoss: tk.IntVar  -preyTimeBetweenMates: tk.IntVar  -predatorTimeBetweenMates: tk.IntVar  -maxWanderDistance: tk.IntVar  -berryConst: tk.IntVar  -cyclescount: tk.IntVar |
| +Back(parent: GUI, controller: tk.Frame)  +ShowGeneGraphs(parent: GUI, controller: tk.Frame)  +ShowPopulationGraphs(parent: GUI, controller: tk.Frame)  +OpenViewer(parent: GUI, controller: tk.Frame)  +StartSimulation(parent: GUI, controller: tk.Frame, preycount: tk.Entry, predatorcount: tk.Entry, baseenergyprey: tk.Entry, mindeathageprey: tk.Entry, maxdeathageprey: tk.Entry, energylprey: tk.Entry, baseenergypredator: tk.Entry, mindeathagepredator: tk.Entry, maxdeathagepredator: tk.Entry, energylpredator: tk.Entry, berryconst: tk.Entry, maxwander: tk.Entry, preyTBM: tk.Entry, predatorTBM: tk.Entry)  +SaveSimulation(parent: GUI, controller: tk.Frame, preycount: tk.Entry, predatorcount: tk.Entry, baseenergyprey: tk.Entry, mindeathageprey: tk.Entry, maxdeathageprey: tk.Entry, energylprey: tk.Entry, baseenergypredator: tk.Entry, mindeathagepredator: tk.Entry, maxdeathagepredator: tk.Entry, energylpredator: tk.Entry, berryconst: tk.Entry, maxwander: tk.Entry, preyTBM: tk.Entry, predatorTBM: tk.Entry)  +Save(parent: GUI, controller: tk.Frame, preycount: tk.Entry, predatorcount: tk.Entry, baseenergyprey: tk.Entry, mindeathageprey: tk.Entry, maxdeathageprey: tk.Entry, energylprey: tk.Entry, baseenergypredator: tk.Entry, mindeathagepredator: tk.Entry, maxdeathagepredator: tk.Entry, energylpredator: tk.Entry, berryconst: tk.Entry, maxwander: tk.Entry, preyTBM: tk.Entry, predatorTBM: tk.Entry)  +Close(tkinter: tk.Frame) |

|  |
| --- |
| Settings(tk.Frame) |
| -fps: tk.IntVar  -screenwidth: tk.IntVar  -screenheight: tk.IntVar  -cellsize: tk.IntVar |
| +Back(parent: GUI, controller: tk.Frame, cellsizeEntry: tk.Entry, screenWidthEntry: tk.Entry, screenheightEntry: tk.Entry, fpsEntry: tk.Entry) |

|  |
| --- |
| LoadSimulation(tk.Frame) |
| -filelist: [String]  -cyclescount: tk.IntVar |
| +Back(parent: GUI, controller: tk.Frame)  +Run(parent: GUI, controller: tk.Frame, option: String)  +ShowGeneGraphs(parent: GUI, controller: tk.Frame)  +ShowPopulationGraphs(parent: GUI, controller: tk.Frame) |

|  |
| --- |
| ImageLabel(tk.Label) |
|  |
| +load(im: string)  +unload()  +next\_frame() |