15-112 Term Project Proposal

Description:

Name: The Settlers of Python
This project will be a python implementation of the popular board game Catan.

Competitive Analysis:

The Settlers of Python seeks to create a fun Python implementation of Catan. The official online version of Catan, Catan Universe, implements 3D graphics in the GUI. This project will be limited to only 2D. However, I aim to improve on the GUI by making features clearer and easier to use.

In addition to GUI, I am also implementing an AI. There are several online versions of a Catan AI, involving NumPy and search trees. I will be implementing a Monte Carlo search function to create my AI.

Structural Plan:

Elements (GUI Features):

- Element class (/resources/gui/element.py)
    - Button class (/resources/gui/button.py)
        - Arguments
            - pos: takes a tuple (x, y)
            - size: takes a tuple (width, height)
            - text: takes a string
            - color: takes a Color instance
            - radius: takes a float/int

Catan Features:

- Board class (/board/catan.py)
    - Arguments (__init__)
        - r: number of rows in axial grid
        - q: number of columns in axial grid
    - Board Generation (__init__)
        - self.hexBoard: stores the board as a List. Calls generateAxialList().
        - self.hexCount: number of hexes in the entire board
        - self.edges: List to store all Edge objects
        - self.generateEdges(): method to create all N instances of Edge objects in board. (n = self.hexCount)
        - self.assignEdges(): method to assign the Edge instances to indices in self.hexBoard
        - self.assignTypes(): method to randomly assign the Tile types

        - self.assignNumbers(): method to assign number tokens fairly, i.e. according to standard Catan rules.
    - Tile class
        - Arguments (__init__)
            - r: row index of the Tile
            - q: col index of the Tile
        - Additional Variables (__init__)
            - self.edges: contains all Edge objects adjacent to the Tile
            - self.type: contains the resource type of the Tile
        - __repr__(self): prints Tile in a friendly format
    - Edge class
        - Arguments (__init__)
            - id: ID value of the Edge. Each Edge has a unique ID
        - Additional Variables (__init__)
            - self.hasRoad: Boolean, True if Road exists on this Edge
        - __repr__(self): prints Edge in a friendly format
        - __eq__(self): enables equality checking of Edges based on ID
        - __hash__(self): hash function for Edge
    - Node class
        -

Timeline:

- Complete Preliminary Board Representation (By 11/20/2019, Wednesday)
- Complete Board Controllers, i.e. Nodes and Roads (By 11/21/2019, Thursday)
- Complete In-Game GUI (By 11/23/2019, Friday)
    - Board
    - Cards
    - Points
- Complete VP Counting (By 11/25/2019, Sunday)
    - Settlements/Cities
    - Largest Army
    - Longest Road
- Complete Preliminary AI Algorithm (By 11/30/2019, Saturday)

Version Control Plan:

This project is on Git and is fully uploaded to a private repository on GitHub.

Module List:

1. pygame