Bao Tran, Luke Lattig

CS 172

Dr. Rev. Matthew Bell

Project Requirements Final

* Board
  + Design Ascii board with \ / and || and then create roads, houses, towns, using the same Ascii symbols along with characters to indicate the state of a house/town.
  + Implement variable within structure of Ascii Design to be manipulated according to player actions and color assign them with integer values for players, available position, and invalid positions. This code should also reference pointers to arrays of Tile classes to easily check variables and display them on the board.
  + Configure a coordinate system to check nodes and sides of each tile within the Board. This must be done by manually assigning Tile object node and side arrays to a two dimensional array in tile for both node and side. This is then used to easily reference the state of a board instead of using boardcoordinates[x][y] but using referencenode/referenceside[tile][index]. This also makes it easier for the player to choose and assign where they’d want to assign a node or side.
  + Create a function to check adjacency of sides, nodes, and tile values such that no two elements, if occupied, overwrite each other. This also assigns the nodes next to the target node as -2 to make sure a player is unable to place a node if an opponent’s node is too close.
  + Create two functions that check if a node/side is valid to be assigned to a user. This must also be manual assigned and use pointers to indicate which side of which tile is adjacent to another.
  + Implement a text file to save and load player data and the position and layout of the board. As well, there will be code to save the game after each turn and the ability to either load or start a new game on start up.
  + Robber will randomly be assigned to a new tile when a 7 is rolled and the code will check the player data to see if they have more than 7 cards in their hand. If so, the code will randomly discard half of the cards. Then the person whose turn rolled the 7 will be able to assign the robber to another tile position.
  + Create a constructor to indicate how many players will be playing in this game
  + Ability to reprint the board.
  + Function to color text based on the values returned by the board for players.
  + Function to color text base on the aspect and entity values of the tiles
  + A function to randomly assign specified dice roll values to different tiles without having the same dice values adjacent to each other.
  + A function to randomly assign tile types either randomly or based on the tile values if the value is 0
  + Functions to get and set player positions, tile values, player house status, and robber status.
  + A function that uses pair<> class to return two values to convert tile input to array coordinates.
* Tiles
  + Methods to store and evaluate the nodes and sides of each created tile class.
  + Methods to check other tile classes for adjacency and overlapping resources
  + Methods to Identify which player is occupying which node / side of the specified tile
  + Check for Robber that prohibits resource gain of specified tile Robber occupies.
  + Method that uses Player method to give resources if that tile is rolled
  + Create private arrays to store tile side status, tile node status, robber status, tile type, and tile values.
  + As well create functions to get and set these values in order for Board class to use that data to perform checks.
  + Create functions to return the address of certain nodes and sides to be used in Board.
  + setNode functions must be duplicated and modified in situations of saving the game and setting up the first nodes when starting a new game.
  + A constructor that defaults all to Desert with value -1 and false robber. This allows the Board functions to easily go through and assign those with appropriate values since Board has a two dimensional array of Tile.
* Player
  + Private attributes that store player resource cards and point value
  + Player function to buy houses, road, and town upgrades // implemented in Main
  + Methods to both add and remove resources
  + Method to display players hand

Assumption

1. There will be more than one player but at max four players
2. Players will take turns without looking at other play’s resources in hand
3. The only way to get points is to buy houses and towns thus far.