Design Rationale

Jianlan Zhu

Andrew ID: jianlanz

1. As a whole, I think it is important to have a game controller to interact with other parts of the elements. In MVC pattern, there is a controller to interact with different model and view, I want to do something similar.
   1) GameController: interaction with all other elements during the game, acts as a "CPU"
   2) Player: represents players in the game
   3) Board: the game board, which has a lot of tiles on it. Mainly responsible of logics like checking completeness, checking tile placing validity etc.
   4) Tile: the tile users play with
   5) Segment: the smallest units in the game, has different features: city, city2, road, cloister, field and crossing. User can put follower on the segment.
   6) Area: composed of many segments, represent the area on the board that has the same feature. Used to determine completeness.

2. Several necessary components of the game include: the game board, players, tiles, followers, segments.

3. As to the informal questions:
   1) The user can do start game operation, choose number of players, really start the game, choose where to place the tile given, rotate the tile when needed, place followers onto the tile, ask for changing tile when there is nowhere to place the given tile, get the game result when finished.

   2) Different players share the same screen. The current player will see his role highlighted on the screen.

   3) User will be given the opportunity to decide whether to put follower onto a tile after the tile is validated. If they do not want to do that, they can click the "next Player" button on the screen so that the turn will switch to the next turn. Of course, they can also click on the tile to place followers.

   4) The tile rotation is represented with an integer variable direction, where 0 means no rotation, 1 means rotate to the right 90 degrees etc. The range is from 0 to 3.

   5) When a new tile is placed, the gameController will use board to determine if there is feature that is complete by using the open field in Area class. As to determine the follower conflicts, we just need to check if the area has already contained any followers. If it does, there will be a conflict.

   6) The game support different scoring methods by referring to the feature types. There are different algorithms to calculate different completed features. For roads, cities and farms, I will use the open field to judge completeness and calculate the score. I will separately determine the completeness of cloister.

7) The game will handle exceptions by throw and dealing with different exceptions. There will also to Boolean return value to help determine that.

8) The GUI will provide users with:
    a. The whole board: including some game information like the game name etc.
    b. Button: "Add Player", "Confirm Name", "Start Game", "Next Round", "Next Player", "Rotate".
    c. Displayer of different players and their current scores