Ticket to Ride Phase 3

Objectives

1. Finish the implementation of all operations needed to play the game. This includes operations triggered by changes in the user interface and operations sent from the server back to the client. You must provide a means for the server to send the final results page to each player.
2. Understand how to create and document a design.
3. Learn how to apply the state pattern.

New Patterns

1. State Pattern – apply it to some non-trivial part of the program.

Requirements

Design

Provide a diagram or picture of how the final results page will look.

Create and document the architecture of this phase. You should provide UML diagrams for all classes created or modified for Phase 3. These should include classes for both the client model and classes for the server model. Also, include any new classes for command management on the server.

Capabilities (These may be adjusted slightly depending on design)

In general you should provide for the following game playing activities

- Claim a route
- Ask for destination cards
- Send back 1 or 2 destination cards
- Select 1 or 2 train cards
    - You must handle all the associated rules

The following are high-level view events generated by the GUI that you must handle (depending on your design, these operations may cause the server to send commands back to the originating client). The GUI must call methods on the "Presenters". Depending on your design, the presenters will then delegate functionality to the state, which will know which actions are allowed and how to respond at any given time.

- Claim a route

- Request destination cards
- Return destination card
- Choose face-up train card
- Choose train card from deck
- Chat

High-level operations the server will send to all players

- Last round
- Route claimed by player (you may include the number of train pieces needed to claim the route)
- Current player requested destination cards
- Current player returned destination card
- Player chose train card of specific color from face-up train cards
- Player chose wild card
- Player chooses a face-down train card
- Player's turn ended
- Game history update
- Update players points
- Chat
- Display result page.

The result page need not be elaborate but must designate the winning player, and, for each player show:

- Number of points from claimed routes. The number of points for a claimed route must be proportional to the length of the route as defined in the rules.
- Number of points for reached destinations.
- Number of negative points for unreached destinations.
- Number of points for having the most claimed routes (this is a simplification from a requirement to calculate and give points for the longest route).
- The total number of points for the player.
- (Extra Credit) If the player has the longest path, show the number of points for that accomplishment (this will substitute giving points for having the most claimed routes and will be awarded extra credit).

Constraints

The design documentation must be done using both UML class diagrams and sequence diagrams.

The design documentation must include your first weekly report for Phase 3 (See template. You can find it near the bottom in the teams section)

Deliverables

Design documentation

A totally running Ticket to Ride game whose design follows the submitted design. It must provide all of the capabilities described above.

Each member of the group is to turn in a copy of the group's final weekly report for Phase 3 with their individual Phase 3 report. (For the final weekly report see template. You can find it near the bottom in the teams section)