For all players of our game, the flow of playing a game is as follows:

1. Players joined the game can choose to create a team when the number of team is less than 4, and can join any team if the number of members in a team is less than 10.
2. After all the players are ready, the game will start after a 10-second counting finishes.
3. All the members in one team will be placed randomly in a city in the map. Different teams are located in different cities in the beginning.
4. There are several prefixed paths from city to city, what the player needs to do is just to choose one path and the vehicle used to travel.
5. Players can know the information about the city they are in, including the next destination city available, vehicle available and its cost, weather. Different time and money cost will leads to different scores players gain.
6. The team score is calculated by the accumulation of all team members.
7. Players can access to the score they own, their team score, and other teams’ score.
8. When all the members of a team succeed travelling around the world, which means, all members go back to the original city, that team wins. If all the teams do not manage to do so after 15 minutes, the team with highest score wins the game.

API Usage:

The API includes IPlayer, IServer, and IGame three stubs. We seperate client and server program in one machine. IServer is a bit like IPerson in HW08, it is used for avoiding routing and fetch a specific IGame according to client's request. First of all in the client program, an initial frame will show up  that can connect to a specific server via IP address, it is like the way we get IPerson stub in HW08. Then the client can choose to create a game or just join a game. After that, a lobby shows up for a client to create or join a team. The IPlayer stub can be used to generate ITeam objects, which is a field of IGame. Then all game data update can be done by interaction between IPlayer and IGame, like chooseNextCity, chooseTransportation, which will leads to different scores gained by an IPlayer and the whole ITeam. When the last client request that satisfies the winning criteria, the server will broadcast a winning message and losing message to corresponding teams.

The program MVC is used to connect, choose game and choose team. When the game starts, a new miniMVC will be created to handle game issues. Also, the controller of the game will be sent to the client who request to join or create a game. They interact with each other with messages when game starts. Before that, the interaction is implemented by method calls.