Participants:	Andrew Johnson, Kyle Pierson, Norman Gifford, Rory Savage
Team:	Farkle
Project:	Multi-client Farkle Game Web Application
Company:	University of Utah School of Computing
Project Lead:	H. James De St. Germain
Date:	7 April 2015

Farkle Game Proposal

Introduction

For our project we will build a multiplayer farkle game. This game is a dice game similar to Yahtzee but is a little faster paced. The rules can be found at http://www.coht.org/resources/FARKLERULES.pdf. There are several variations on how it is played but this is the ruleset we will use. In addition to the game itself we will include a client server model to allow several players to play on their own computer. This will mean that we will need to decide what will be done by the server and what will be done by the client. Another aspect will be the communication between clients and how to update. This will have three main parts: the user interface, the client, and the server.

User Interface

The user interface will first present a screen to login as a user. If there is not an active game, the player will be able start a new game with all of the players currently logged in. Once a game is started no new users will be able to join that game. When a new user logs in they will be able to view the current game and join a new game on completion of the current game. The player will be able to exit a game in progress at which point their game score will be zero.

Once the game has started you will have the game screen. The game screen will have six dice, a cup to roll the dice, a scoring area, and a game board. The scoring area will show the users score as well as the scores of the users opponents. The game board will show the cup for rolling the dice as well as the dice that you have currently rolled. The user will be able to select the dice they want to set aside for their current score.

Client

The client provides a communication medium between the user interface and the server. The game logic itself is handled on the server, but the client will handle user action events, such as sending user actions to the server, receiving updates (regardless of whose turn it is), and updating the user interface as the game progresses. Each computer running the game will have its own copy of the client.

Server

The server will handle all of the game logic as well as managing the clients and whose turn it is via websocket connections. When it is a clients turn no other client will be able to send inputs until it is their turn. The logic will consist of starting a game, scoring, turns, and ending the game. Beginning a game will initialize a game with all of the clients currently logged into the server. The server will then assign player order randomly, listen for the players to to “roll the dice”, then handle turn taking and and scoring. At any time the server needs to handle dropped clients, from either a client leaving the game, or if a client unexpectedly loses a connection. At the end of the game the server will declare a winner and then wait for another game will be started.

Distribution of Work

Rory will work on the graphical user interface. Andrew will work on the client model. Kyle and Norman will work on the server. There are two main aspects of the server; socket communication with the clients, which Kyle will be responsible for, and game logic, which Norman will do. Note that this is a preliminary idea of how the work will distributed and there will will be adjustments depending on how the work for each task flows.

Conclusion

The main goal of this project is to provide an easy-to-use, visually appealing, Farkle game GUI provided on the website, as well as robust and efficient server-side handling of these games. This will be accomplished by using HTML5, CSS, Javascript, PHP (with Ratchet for web sockets), JQuery, and AJAX, as well as other frameworks and languages as needed. The game implementation we have chosen will be challenging but realistic, and we feel it is a good choice for this final project.