**Name of Project**

**Date: 07/12/12**

**Student Names:** *Ivan Mashalov.*

***Statement of requirements***

1. *We assume that there are two players, who can speak English, have 15 checkers of a different colour each. Each player has two die. The checkers are placed in the universal setup at the start of the game. The board is of regulation size and has the standard 24 triangles, twelve on each side and 6 in each quarter (which are numbered to aid the rules of the game).*
2. *Input: the player will input data with the mouse. Output: the game will output the die number and possible movements of the checkers.*
3. *The game will be coded in the language of Java.*

**Requirements**

**Functional Requirements**

*R1. It shall be possible to select the participants who will be taking place in the game.*

*R2.The game shall include 2 die that randomly generate 2 numbers between 1 and 6*

*R3. If the player rolls a double then the player gets twice the value of the two die which can be used on 4 different checkers.*

*R4. The game shall only allow 5 checkers, of one colour, on one triangle at a time.*

*R5. When a player lands a counter on a triangle and one of another colour lands on the triangle then the original counter shall be sent back to the start.*

*R6. The player who rolls the highest dice before the game shall have the first play in the game.*

*R7. The games shall have a menu system which allows the player to start new games and save or load games*

*R8. The game shall inform the players when and who the winner is by displaying a message on the screen when a player wins.*

*.*

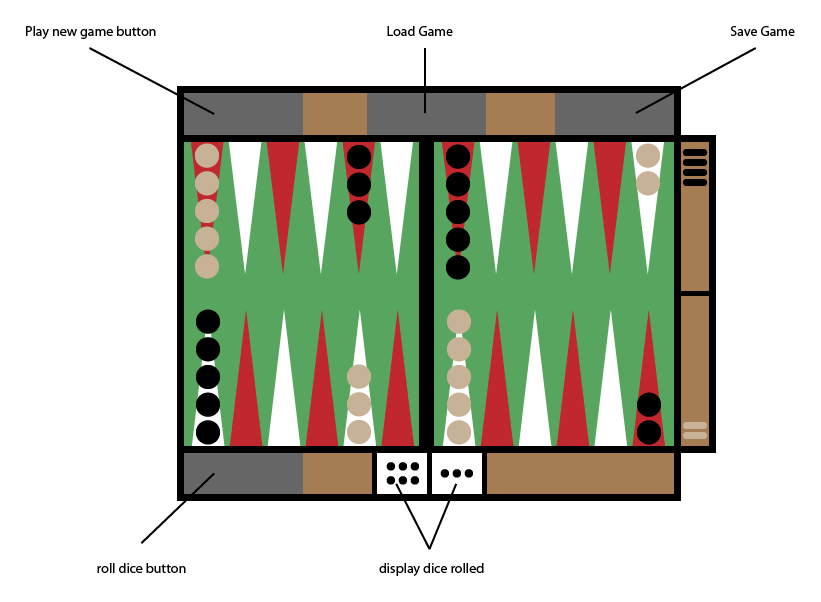
**Non-functional Requirements**

R1. It shall be easily used by the players of the game.

R2. The game shall be available for play on multiple platforms.

R3. The systems running on must have the java system running on it.

**User Interface**

**

**User Cases**

**USER**: Opens program and starts game.

**SYSTEM**: Sets up board and rolls dice.

**SYSTEM**: Decides which player starts based on dice rolled.

**USER**: Clicks to roll dice.

**SYSTEM**: Updates board.

**USER**: Clicks save.

**SYSTEM**: Saves game.

**USER**: Clicks load.

**SYSTEM**: Loads game.

**USER**: Completes game

**SYSTEM**: Displays winner on screen.

**Classes**

***Candidate Classes***

|  |  |  |
| --- | --- | --- |
| **Candidate Classes** | **Accept/Reject** | **Reason for Rejection** |
| Players | Rejected | Function of dice |
| Move | Rejected | The same as pointers |
| Menu | Rejected | Repetition of board |
| Game | Rejected | Too broad |
| Dice | Accepted |  |
| Board | Accepted |  |
| Pointers | Accepted |  |
| Score | Rejected | The game is win or lose |
| User | Rejected | Repetition of players |

***Class Descriptions including Responsibilities, Fields and Methods***

**Backgamon1**

* **Method**

player2Move

player1Move

die2

die1

drawBoard

newGame

menu

* **Fields**

Display new board with everything cleared

Takes a save file and reprints it to the board

Stores all data on board

Takes integers returned from DiceRolled and shows possible moves for the user to play.

***Class Diagrams***

******

***Activity Diagrams***

******