|  |  |  |
| --- | --- | --- |
| USER REQUIREMENTS SPECIFICATIONS | April 21  2015 | |
| LUDO WCF APPLICATION | | 1. GROUP: Todor Tsekov Ivana Raykova Zair Thiel |



# TABLE OF CONTENT

# 1. INTRODUCTION

The goal of this project is to build a Ludo Application. Ludo is a board game for 2-4 players. Every player is assigned 4 pieces, a dice and one of 4 possible starting areas. The goal is to get all your pieces around the game board to the finish line. Players take turns throwing the dice, which decides how far a piece can be moved. If you get a 6 you get an extra turn or you may enter another staged token to its starting square. When a player lands a piece on top on an opponent’s piece, the opponent’s piece is sent to the starting area. The general strategy is to make sure that your own pieces are not send to your starting area, and at the same time try to send the pieces of your opponents to their starting area. The rolls of a die control the swiftness of the tokens, and entry to the finishing square requires a precise roll from the player. The first to bring all their tokens to the finish wins the game.

# 2. FUNCTIONAL REQUIREMENTS AND USE CASES

# 2.1 REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| Nr | Description | MOSCOW |
| 1 | A player rolls a dice | M |
| 2 | A player moves his piece to the starting square | M |
| 3 | System notifies players of their opponents rolls | M |
| 4 | System updates to show opponents piece movements | M |
| 5 | A players chooses his color | S |
| 6 | When a player’s piece lands on an opponent’s piece, the opponents piece is then removed from its position and returned to the rest area | M |
| 7 | When a player’s piece lands on his own piece, the system will display both piece as being next to each other | S |
| 8 | A player can communicate with other players through the chat window | S |
| 9 | A player moves the number of squares he rolls | M |
| 10 | The system allows players to move his piece towards the victory area only when the roll equal to or smaller than the amount of squares between the players piece and the last unoccupied square of the victory area | M |
| 11 | A player can choose which piece he wishes to move | M |
| 12 | A player rerolls when rolling a six | M |
| 13 | Minimal 2 players required to play | M |
| 14 | A player can only move his own piece | M |
| 15 | System displays the winner of the game | M |
| 16 | Players are able to join an ongoing game | C |
| 17 | Players are able to spectate ongoing games | W |
| 18 | When a room is full, players can no longer join | W |
| 19 | When starting a game, players are allowed to make private rooms | W |
| 20 | Players can log in to play the game | S |
| 21 | Players can register to play the game | S |
| 22 | System tells players whose turn it is | M |
| 23 | System decides who starts at the beginning of the game | M |
| 24 | A player can invite other players to his game | S |
| 25 | A player can challenge a player to a game | S |

# 2.2 USE CASES

# Registration in the game

|  |  |
| --- | --- |
| Goal- level | Registering in the game |
| Pre-condition | The user is not registered in the game yet |
| Actor | User |
| MSS | 1.The User fills the registering blank with his data  2. The User chooses an username and a password  3. The User clicks on the button “register”  4. System notifies User if his register was successfully |
| Extension | 2.1 If the username already exist, the system notifies the user to choose another username |
| Post Condition | The user successfully registered. |

# Log into the game

|  |  |
| --- | --- |
| Goal- level | To log in the game |
| Pre – condition | The user has to be already registered in the game |
| Actor | User |
| MSS | 1. The User types the username 2. The User types the password |
| Extension | * 1. If the User has typed the username wrong, the system will notify the User with a message to retype it   2. If the User has typed the password wrong, the system will notify the User with a message to retype it |
| Post Condition | The user successfully logged in the game |

# Chat during the game

|  |  |
| --- | --- |
| Goal- level | To chat with opponent during the game |
| Pre – condition | 1.The game has already started |
| Actor | User |
| MSS | 1.The User type a message in the chat box  2.The User receives a message in the chat box |
| Extension |  |
| Post Condition |  |

# Roll die

|  |  |
| --- | --- |
| Goal-level | Roll the die |
| Pre-condition | Current player (user) turn |
| Actor | User |
| MSS | 1.The user clicks on the die |
| Extension |  |
| Post Condition | The system shows the die result |

# Place token

|  |  |
| --- | --- |
| Goal-level | Place a token on the board |
| Pre-condition | 1.Current player (user) turn  2.The die result must be 6 |
| Actor | User |
| MSS | 1.The user clicks on the desired token that he wants to place on the board |
| Extension |  |
| Post Condition | The token is placed on the starting area |

# Move token

|  |  |
| --- | --- |
| Goal-level | Move token |
| Pre-condition | Token should be on the board |
| Actor | User |
| MSS | 1.The user clicks on the die  2.Clicks on the token that he wants to move |
| Post Condition | Token moved to the new position |

# Remove token

|  |  |
| --- | --- |
| Goal-level | Remove a token from the board |
| Pre-condition | You land on an enemy token |
| Actor | System |
| MSS | 1.System checks if a player lands on a token that does not belong to him  2.System removes enemy token and places the user’s token on that spot |
| Extension | 1.a token belongs to the user  1.b user token gets overlapped by his own token  1.a.1Token lands on opponents while on the opponents starting square  1.a.2 Token is not removed from the board |
| Post-Condition | Players token takes the position of enemy token |

# Start Game

|  |  |
| --- | --- |
| Goal | Begin the game |
| Pre-condition | Other Players are readied and awaiting the user |
| Actor | User |
| MSS | 1.User clicks the start game button  2.System informs players that the game will begin  3.Game begins |
| Extension | 2.aNot all players are ready  2.b System informs user that not all players are ready  2.c System does not start the game |
| Post-Condition | GUI changes to the game GUI |

# Join Game

|  |  |
| --- | --- |
| Goal | Join a game |
| Pre-condition | User is logged in and gets invited |
| Actor | User |
| MSS | 1.User clicks on the join game button |
| Post-Condition | User joins game lobby |

# Exit Game

|  |  |
| --- | --- |
| Goal | Exit a game |
| Pre-Condition | User is in game |
| Actor | User |
| MSS | 1.User clicks on the exit game button  2.System asks user if he wants to leave |
| Post-Condition | User leaves the game |

# Spectate Game

|  |  |
| --- | --- |
| Goal | User wants to spectate a game |
| Pre-condition | User in the lobby |
| Actor | User |
| MSS | 1.User selects one of the players that are in game  2.System drops down a menu  3.User clicks the spectate button  4.System changes GUI of that of the game |
| Post-condition | User is now spectating an ongoing game |

# Replay Game

|  |  |
| --- | --- |
| Goal | User wants to replay the actions of the game |
| Pre-Condition | User just finished a game |
| Actor | User |
| MSS | 1.User clicks on the replay button  2.System changes current GUI |
| Post-condition | User is now watching his replay of his last game |

# 3. GUI

# 4. Non-functional requirements

|  |  |
| --- | --- |
| Nr | Description |
| 1 | GUI should be easy to use |
| 2 | System should display outputs from one user to the others users within the same session |
| 3 | System should respond fast to user inputs |
| 4 | System should have stable connectivity in order to provide stable user interaction |
| 5 | System should relay information from one user to the other |
| 6 | System should display the output of the session to all current users within the session before ending the session |