|  |  |  |
| --- | --- | --- |
| USER REQUIREMENTS SPECIFICATIONS | May 15  2015 | |
| LUDO WCF APPLICATION | | 1. GROUP: Todor Tsekov Ivana Raykova Zair Thiel |



Table of Contents

[1. INTRODUCTION 2](#_Toc419460867)

[2. FUNCTIONAL REQUIREMENTS AND USE CASES 3](#_Toc419460868)

[2.1 REQUIREMENTS 3](#_Toc419460869)

[2.2 USE-CASE DIAGRAM 0](#_Toc419460870)

[2.3 USE CASES 0](#_Toc419460871)

[ Registration in the game 0](#_Toc419460872)

[ Log into the game 0](#_Toc419460873)

[ Chat during the game 0](#_Toc419460874)

[ Global Chat 1](#_Toc419460875)

[ Roll die 1](#_Toc419460876)

[ Place token 1](#_Toc419460877)

[ Move token 1](#_Toc419460878)

[ Remove token 2](#_Toc419460879)

[ Start Game 2](#_Toc419460880)

[ Create Game 2](#_Toc419460881)

[ Invite a Player to a game 2](#_Toc419460882)

[ Join Game 3](#_Toc419460883)

[ Exit Game 3](#_Toc419460884)

[ Choose Color of Token 3](#_Toc419460885)

[ Spectate Game 3](#_Toc419460886)

[ Replay Game 4](#_Toc419460887)

[ Pause Game 4](#_Toc419460888)

[ Resume Game 4](#_Toc419460889)

[3. GUI 5](#_Toc419460890)

[3.1 Login form 5](#_Toc419460891)

[3.2 Registration form 6](#_Toc419460892)

[3.3 Lobby form 7](#_Toc419460893)

[3.4 Game Creator form 8](#_Toc419460894)

[3.5 Join the game form 9](#_Toc419460895)

[3.1 Game form 10](#_Toc419460896)

[4. Non-functional requirements 10](#_Toc419460897)

[5. Work division 10](#_Toc419460898)

# 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 after each move | 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 at the end 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 | M |
| 19 | When starting a game, players are allowed to make private rooms(games protected with password created by the game creator) | W |
| 20 | Players can log in to play the game(if have enough time we will implement it, so it stays a Should) | S |
| 21 | Players can register to play the game(if have enough time we will implement it, so it stays a Should) | 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-CASE DIAGRAM



## 2.3 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 3. The clicks the “login” button 4. The system logs the user into the game and re directs him/her to the Lobby form |
| 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 | The game has already started |
| Actor | User1 User2 |
| MSS | 1.The User1 types a message in the chat box  2.The system sends the message in the chat box  The User2 receives a message in the chat box(example) |
| Extension | 2.1 The message of User1 can be seen from all players/users in the game  All of this can be repeated with User2 sending a message and User1 receiving it. |
| Post Condition |  |

# Global Chat

|  |  |
| --- | --- |
| Goal- level | To chat with all registered players that are online |
| Pre – condition | The User is logged in the game and he/she is in the lobby |
| Actor | User1 User2 |
| MSS | 1.User1 types a message in the Global chat box  2. The system shows the message in the Global chat box to all portal users  User2 receives a message in the Global chat box(example) |
| Extension | All of this can be repeated with User2 sending a message and User1 receiving it. |
| Post Condition | User1 send a message to all online players. |

# Roll die

|  |  |
| --- | --- |
| Goal-level | Roll the die |
| Pre-condition | Current player (user) turn |
| Actor | User, Players |
| MSS | 1.The user clicks on the die  2. The system shows the die result to all of the players in the game |
| Extension |  |
| Post Condition | The die is rolled |

# 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, Players |
| MSS | 1.The user clicks on the desired token that he wants to place on the board  2. The system places the token on the starting area and shows the result to all players. |
| 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, Players |
| MSS | 1.The user clicks on the token  2. The system moves the token and shows all the players in the game and shows them the new position of the token. |
| 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  (consequence of the move token use-case) |
| Post-Condition | Players token takes the position of enemy token |

# Start Game

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

# Create Game

|  |  |
| --- | --- |
| Goal | User creates a game |
| Pre-condition | The User is logged in the game and he/she is in the lobby |
| Actor | User |
| MSS | 1.User clicks the create game button  2. System shows the user the Game creator screen. |
| Extension |  |
| Post-Condition | The user created a game |

# Invite a Player to a game

|  |  |
| --- | --- |
| Goal | User 1 invites User2 to a game. |
| Pre-condition | User1 created a game, there is at least one other player online. |
| Actor | User1, User2 |
| MSS | 1. User 1 browse through the Player list.  2. User 1 right-clicks on User2’s name and chooses the invite option.  3. System informs User1 that he/she invited User 2 to a game.  4. System informs User 2 that he/she was invited to a game.  5. User 2 clicks the Accept button. |
| Extension | 2.1 The players chosen by User1 have to be online.  5.1 User 2 clicks the Decline button. |
| Post-Condition | User1 invited user 2 to a game. |

# 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 |

# Choose Color of Token

|  |  |
| --- | --- |
| Goal | User want to choose a token color |
| Pre-Condition | User is creating a game or he/she is a guest in a game |
| Actor | User1 User2 |
| MSS | 1. User1 clicks on one checkboxes showing the different token colors.  2. System indicates that this User 1’s token color and informs User 2. |
| Exceptions | * 1. If the color is already occupied User1 will receive an appropriate error message that he needs to choose another token color.   2.1 User 2 cannot choose the same token color as User1. |
| 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.The system redirects the User to the Game form, all the buttons are disabled |
| 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 replays the game in the Game form |
| Post-condition | User is now watching his replay of his last game |

# Pause Game

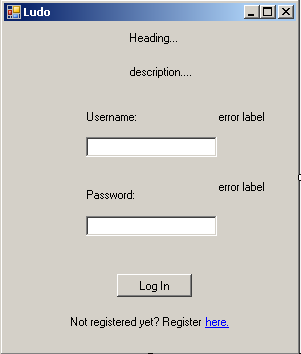
|  |  |
| --- | --- |
| Goal | User wants to pause the game |
| Pre-Condition | User is playing the game and it is his turn |
| Actor | User 1 User 2 |
| MSS | 1. User1 clicks on the pause button.  2. System pauses the game and informs User 2 that the game is paused. |
| Post-condition | User1 paused the game, user 2 is waiting for user 1 to resume the game |

# Resume Game

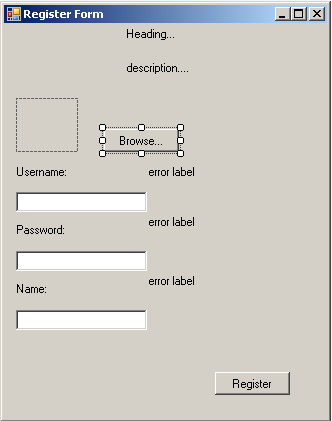
|  |  |
| --- | --- |
| Goal | User wants to resume the game |
| Pre-Condition | User paused the game |
| Actor | User1 User2 |
| MSS | 1. User1 clicks on the resume button.  2. System resumes the game, allowing User1 to continue with his/her move.  3. System informs User2 that the game is resumed. |
| Post-condition | User1 resumed the game, user 2 is waiting for user 1 to make a move. |

# 3. GUI

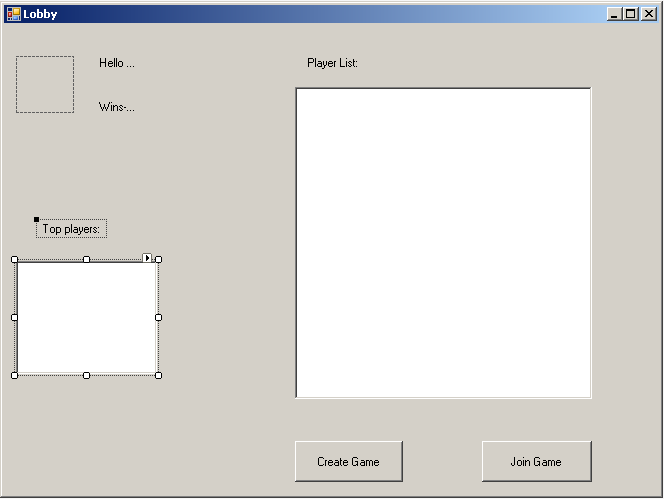
## 3.1 Login form



## 3.2 Registration form



## 3.3 Lobby form

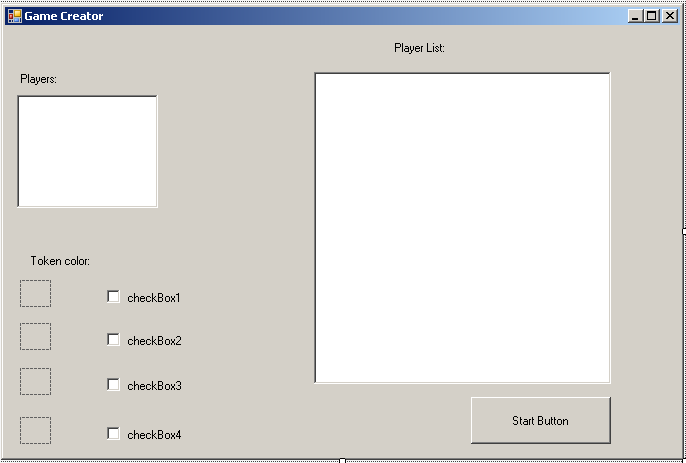


Here you can choose to create game or you can wait somebody to invite you to a game(the Join game button will be change color).

This will be the list of players.

When the user is logged in he will receive a greeting and he can see how many matches he won.

## 3.4 Game Creator form

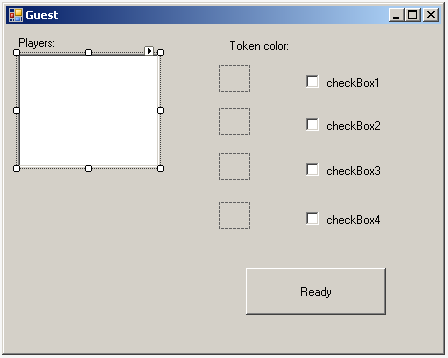


Here you can choose the color of your token.

This will be the list of players that the creator can invite in the game.

This will be the list of players joined the game

## 3.5 Join the game form

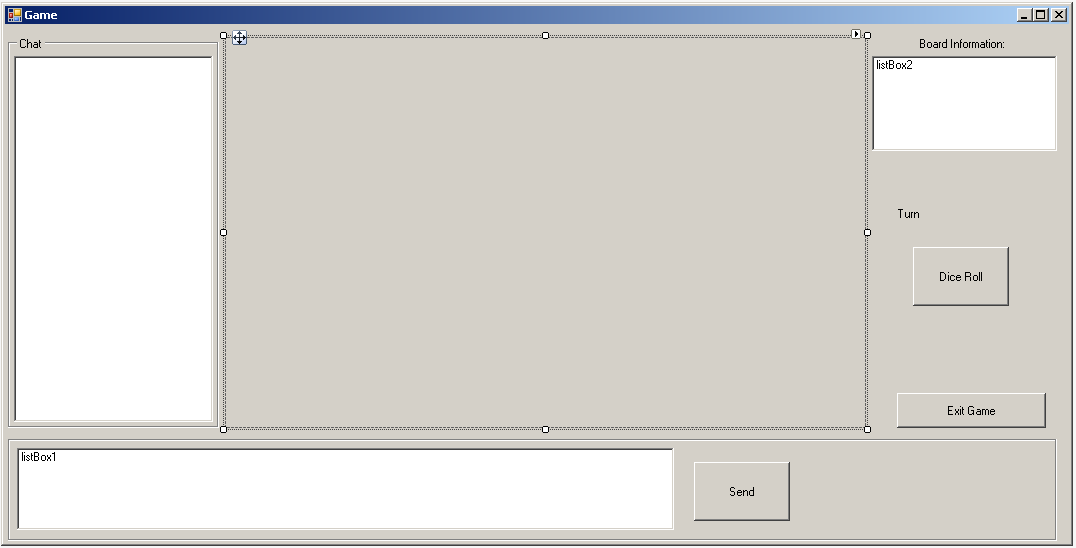


Here you can choose the color of your token.

This will be the list of players joined the game

The name of the form will be change in the application.

## 3.1 Game form



# 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 |
| 7 | The application will be programmed in C#, which means Windows operation system version windows 7 or windows 8 are the required environments for it. |
| 8 | The application will be of type WCF. |
| 9 | The application can run on a basic computer. |

# 5. Work division

|  |  |
| --- | --- |
| Name: | Tasks: |
| Todor | **GUI, 6 use-cases, use-case diagram, MoSCoW** |
| Ivana | **Introduction, functional requirements, 6 use-cases, use-case diagram** |
| Zair | **Non-functional requierments,6 use-cases, use-case diagram, MoSCoW** |