User requirement specifications

MDW – Ludo game

FONTYS UNIVERSITY OF APPLIED SCIENCES

June 26, 2016

Authored by: Monica Stoica , Dimitar Vikentiev, Rosen Danev

Table of Contents

Introduction

Our group consists of three members: Rosen Danev, Monica Stoica and Dimitar Vikentiev, students of class EI8S2.

The following document describes the implementation of an object-oriented software product using UML techniques.

The goal of this software system is to allow users to play the Ludo game. In addition, the User Requirements Specification (URS) will be described such as functional and non-functional requirements and user interface.

The functional requirements are represented by use cases and MoScoW. We have chosen the most suitable use cases so that the most functionality of our system will be covered. In this way, we were able to determine the most appropriate user-friendly interface.

About the game

Ludo, is a board game for two to four players in which players race their four pawns from start to finish according to dice rolls.

Before the beginning of the game, each player’s tokens are out of play staged in one of the large corner areas of the board (called the player’s yard).

When able to, the players will enter their tokens one per time on their respective starting squares, and proceed to race them contraclockwise around the board along the game track (the path of squares not part of any player's home column). When reaching the square below his home column, a player continues by racing tokens up the column to the finishing square. The rolls of a cube 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. The others often continue play to determine second-, third-, and fourth-place finishers.

To enter a token into play from its staging area to its starting square, a player must roll a 6. If the player has no tokens yet in play and does not roll a 6, the turn passes to the next player. Once a player has one or more tokens in play, he selects a token and moves it forward along the track the number of squares indicated by the die roll. Players must always move a token according to the die value rolled, and if no move is possible, pass their turn to the next player.

When a player rolls a 6 he may choose to advance a token already in play, or alternatively, he may enter another staged token to its starting square. The rolling of a 6 earns the player an additional ("bonus") roll in that turn. If the additional roll results in a 6 again, the player earns an additional bonus roll. If the third roll is also a 6, the player may not move a token and the turn immediately passes to the next player.

A player may not end his move on a square he already occupies. If the advance of a token ends on a square occupied by an opponent's token, the opponent token is returned to its owner's yard. The returned token may only be re-entered into play when the owner again rolls a 6. There are no "safe" squares on the game track which protect a player's tokens from being returned. A player's home column squares are always safe, however, since no opponent may enter them.

Functional requirements

Moscow

Nb.	Requirement	MoSCoW
1.	The player can throw the dice	M
2.	The game will start automatically after all 4 players have joined. If there are less than 4 players but more than 2, then any of them can start the game.	M
3.	The colours will be assigned based on priority	M
4.	The players can exchange messages	S
5.	A player can log in with a username and password	S
6.	A player can log out	S
7.	A player can quit the game at any time without affecting the game	M
8.	Only one player per square	M
9.	Ranking	S
10.	A player can sign in	S
11.	Other players will be informed when a player choses to exit the game or is kicked out of the game	M
12.	Players can invite other players to join	C
13.	A player can pause and resume the game	S
14.	At the end of the game, an overview is provided	S
15.	Allow other users to watch the game	C
16.	Replay of the game	C
17.	The player has to move the pawn according to the dice	M
18.	If rolled a 6, the player has the opportunity to start with a new pawn and the user can roll the dice again	M
19.	When a player wins, the game will end for him/her. However, he will have the opportunity to watch the other players compete for the second and third place	M
20.	Ranking of the players	S
21.	Visitors can watch the game	C

Use Cases

The following use-cases represent the functional requirements that Ludo game will be providing.

I.

Goal: Log in

Actor: Player

MSS:

1. The actor starts the game
2. The system displays the log in window
3. The actor enters the username and password.
4. The actor presses the 'Login' button.
5. The system checks if the account exists and if the username and the password match.
6. The system displays the lobby of the game.

Ext:

3a. The account details, username and password, are not recognized. The system notifies the actor and the actor is sent to step 3.

II.

Goal: Sign in

Pre-Condition: The log in form is displayed

Actor: Player

MSS:

1. The actor presses the 'Sign in' button.
2. The system displays the Sign in form.
3. The actor enters the needed information and presses 'Submit' button.
4. The system check if the username is available
5. The system creates a new database entry with the specified details.
6. The system closes the form and opens the lobby window.

Ext:

4a. The username is taken by another player. The system displays an error and the actor is sent to step 3.

III.

Goal: Join a game

Pre-Condition: The lobby window is displayed

Actor: Player

MSS:

1. The actor presses the 'Join game' button.
2. The system closes the current window.
3. The system displays the game window.

Ext:

1a. There are already 4 players in the game. The system informs the actor that the game cannot be joined.

IV.

Goal: Throw a die

Pre-condition: At least two players in the game

Actor: Player

MSS:

1. The actor presses on the 'Roll the dice' button
2. The system generates a random number between 1 and 6 and shows it on the screen.
3. The system shows a die picture corresponding to the number rolled
4. The system plays the 'Roll a die' sound

Ext:

1.a. It is not the actor's turn. The system displays a message informing the actor that he has to wait for his turn to come.

V.

Goal: Move the pawn

Pre-condition: The dice has been rolled

Actor: Player

MSS:

1. The actor clicks on the pawn he/she wants to move.
2. The system calculates and moves the pawn on the proper square
3. The system disables the current actor
4. The system enables the next actor to roll the dice.

Ext:

1.a. The actor has no pawns out yet. The system checks if the thrown number is a 6. If yes, the actor clicks on one of the paws. The system draws the pawn on the screen. In both cases, the system disables the current actor and enables the next actor to roll the dice.

2.c There is a pawn on that square. The already existent pawn will be put back on the starting spot and the new pawn will take its place.

VI.

Goal: Players exchange messages

Pre-Condition: The game has been started

Actor: Player

MSS:

1. The actor types a message in the message text box.

2. The actor presses the 'Send' button.

3. The system updates the chat window with the new message for all users of the system.

VII.

Goal: Pause game

Pre-Condition: The game is already running in a game phase.

Actor: Player

MSS:

1. The actor presses the 'Pause' button.

2. The system will pause the game for all the players

VIII.

Goal: Quit the game

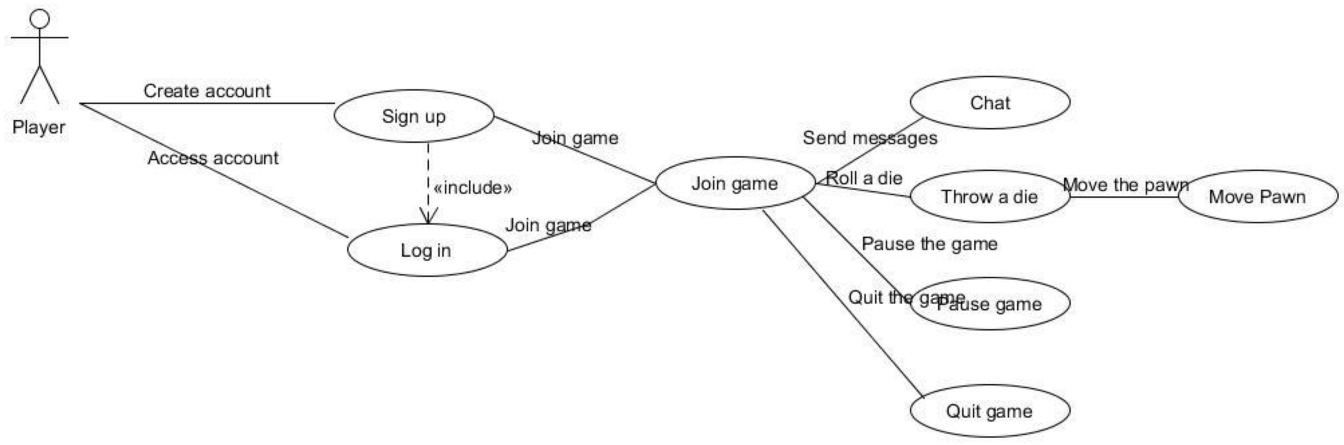
Pre-condition: The game is running

Actor: Player

MSS:

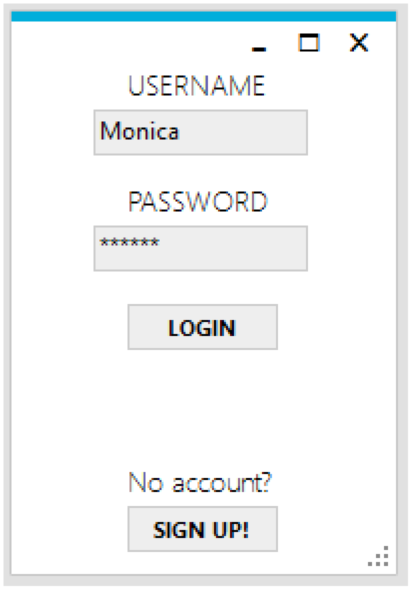
1. The actor presses the 'Quit game' button
2. The system kicks out the actor from the game
3. The system informs the other players that the actor has quitted the game.

Use case diagram



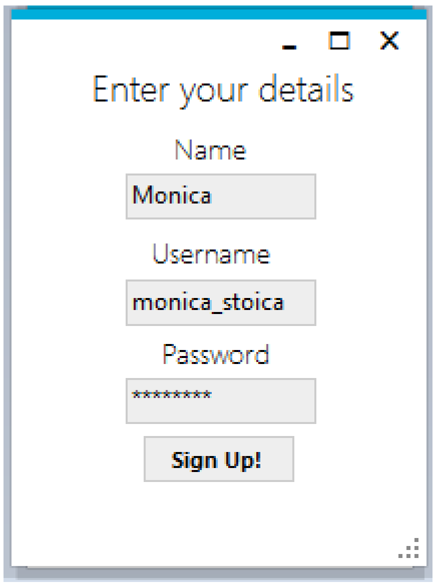
User Interface

In order to play the game, you must be logged in. When first starting up the application, a log in window will be displayed.



You can enter your username and your password and press the 'LOGIN' button. In case the combination of your password and username is not correct, a 'Login failed' will be displayed.

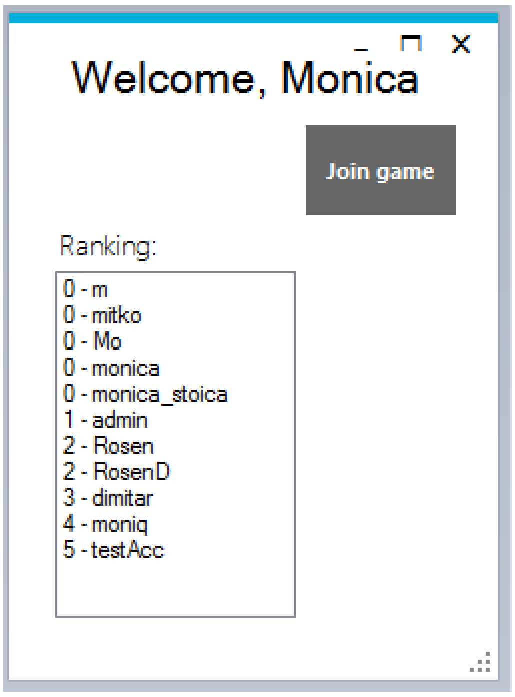
If you wish to create an account, you can press the 'Sign up' button and a new window will be displayed.



For registering, you only need to fill in the name, the username and the password. In case there is already a user registered with that username, a message will be displayed.

To finish the registration, press the 'Sign up!' button.

After the logging in or registering, you will be redirected to a new window – the lobby. However, the lobby does not contain all the basic functionality such as 'Invite players'. In the lobby you can see a list of players, ordered by their points. Every time you win, you will receive a number of points.

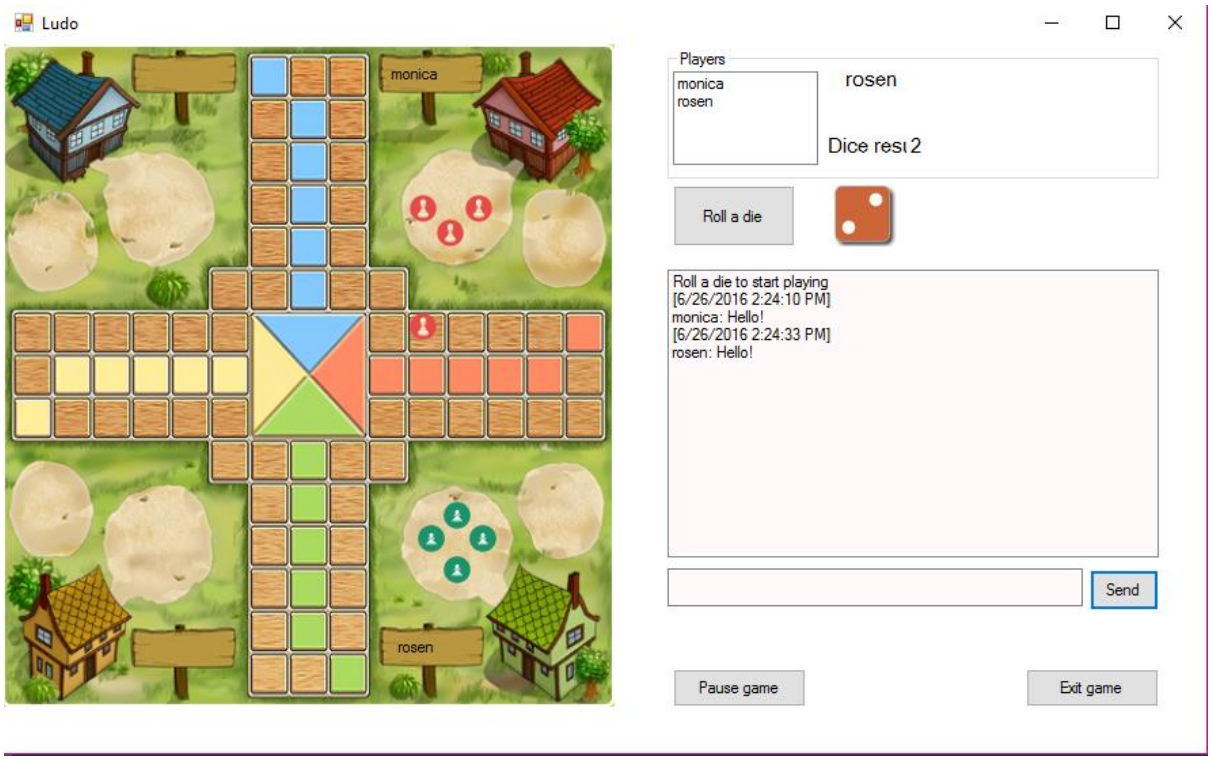


Welcome, Monica

Ranking:

0 - m
0 - mitko
0 - Mo
0 - monica
0 - monica_stoica
1 - admin
2 - Rosen
2 - RosenD
3 - dimitar
4 - moniq
5 - testAcc

After pressing the 'Join game' button, the game window will be displayed. Here you can start playing by pressing the 'Roll a die' button only if you are the first one who has entered the game. Otherwise, you have to wait for the other player(s) to start the game.



monica
rosen

rosen

Dice resı 2

Roll a die to start playing
[6/26/2016 2:24:10 PM]
monica: Hello!
[6/26/2016 2:24:33 PM]
rosen: Hello!

In the upper right corner, the user will see all players who have joined the game. As is mentioned in the description of the game a minimum of two and a maximum of four players (users) can play this game. By clicking the quit button the user will be able to leave the game.

The players can communicate with each other on the chat displayed on the right of the screen. In order to send a message, a user has to type some text in the indicated text box and press the send button.

Non-functional requirements

Due to its target audience, the game is aiming for a user friendly interface. Therefore, it does not matter how experienced a user is with the game. The only aspect that the user has to be concerned with is represented by the rules of the game. We will deliver test plans in order to achieve reliability for preventing unexpected error which might lead to unwished crashes. The final product will be an installation file.

4. The game will be written in C#
5. The game can only be run on a computer with Windows.
6. The game is running on a server.
7. Minimum two players can start a game.
8. Maximum four players can play.
9. The game will store a player's information in the database (such as Username, Password, ranking)
10. Once logged in, the player will be added to chat.