The Drawing Game

Group 8
Mikael Rydmark
Joel Lasses
Ting-Hey Chau
David Alison

6. Functional Test Cases

6.1 General Game

6.1.1

Function being tested:

A player shall be able to create a game.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user has logged in to the system and is currently located in the "start frame".

Input:

The user clicks on "Create Room".

Expected Output:

A new window appears where you can set the game preferences.

Instructions:

1. Click on "Create Room".
2. Set game preferences.
3. Click on "Start Game".

6.1.2

Function being tested:

A player shall be able to join a game.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user has logged in to the system and is currently located in the "start frame".

Input:

The user clicks on "Join Room".

Expected Output:

A new window appears where you can join one of two teams and wait for the room creator to start the game.

Instructions:

    1.Click on "Join Room".
    2. Join a team.
    3. Wait for game creator to start the game.

6.1.3

Function being tested:

A Player shall be able to leave a game.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user has logged in to the system and is currently located in a game room.

Input:

The user clicks on "Leave Room".

Expected Output:

The current window is closed and the user is returned to the "Start Frame".

Instructions:

    1.Click on "Leave Room".

6.1.4

Function being tested:

The system shall store profile information.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and located at the profile page.

Input:

The user clicks on "Save settings".

Expected Output:
The user profile shall now be saved.

Instructions:

    1.Edit user profile.
    2. Click on "Save Settings".
    3. Verify by entering your profile page that the profile is showing.

6.1.5

Function being tested:

You can click on a persons name to see his/her profile provided that the person is a member.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and located in the "Start Frame".

Input:

The user clicks on a persons name in the user list.

Expected Output:

A new window appears showing the profile of the clicked person.

Instructions:

    1. Click on a persons name in the user list.

6.1.6

Function being tested:

Registered members shall be able to edit a profile page containing personal information about themselves.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

A registered member is logged in and located in the "Edit Profile Page".

Input:

The user edits user information.

Expected Output:

The new changes to the profile are now shown in the profile page.

Instructions:

1. Edit some user information.
2. Click on "Save Settings".

6.1.7

Function being tested:

You have to be at least four players to play; there are only two teams in one game session.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and located in a game room.

Input:

-

Expected Output:

As long as there are fewer than four players present in the game room you are unable to start the game. When in the game room you have only two teams to choose from.

Instructions:

1. Verify that the "Start Game"-button is shaded while there are fewer than four players present in the game room.
2. Verify that there are only two teams to choose from.

6.1.8

Function being tested:

There has to be at least two players in each team.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and located in a game room.

Input:

-

Expected Output:

As long as there are fewer than two players in each team you are unable to start the game.

Instructions:

    1. Verify that the "Start Game"-button is shaded while there are fewer than two players in each team.

6.1.9

Function being tested:

The system evens the teams out if a player exits a game making the teams uneven.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and playing a game.

Input:

A player leaves the game making the teams uneven.

Expected Output:

When a player leaves a team making one team have more than one player more on their team than the other team, one player from that team joins the team with fewer players.

Instructions:

    1. Play a game with one team with n players and the other team with n+1 players.
    2. See to it that one player from the team with n players leaves the game.
    3. Verify that the system then moves one player from the n+1 team over to the other team.

6.1.10

Function being tested:

If a player is inactive for a specific period of time he is kicked out of the game.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and playing a game.

Input:

-

Expected Output:

The player who is being inactive is kicked from the game.

Instructions:

1. Be inactive in a game
2. Verify that you are kicked from the game.

6.1.11

Function being tested:

A dice will tell the teams how many steps on the board they should move.

Requirement document reference:

Functional requirements, page 9

Initial system state:

The user is logged in and playing a game.

Input:

The user guesses the correct word.

Expected output:

The guess word is correct; you will receive a "Your guess is correct!" output on the display.

The team's piece on the game board will move as many steps shown by the dice.

Instructions:

1. Choose the "Game chat window" by clicking in it
2. Type in a word and press "Enter"
3. Verify that "Your guess is correct!" output is showing.
4. Verify that the piece on the game board moves as many steps as shown by the dice.

6.1.12

Function being tested:

There shall be a game board that contains the path from finish to goal, where the first team that gets to the goal wins.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and is playing a game.

Input:

-

Expected Output:

In the game window there is a game board displayed.

Instructions:

1. Start a game.
2. Verify that a game board is showing.

6.1.13

Function being tested:

The squares in the path shall have a color corresponding to a category of words.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and playing a game.

Input:

-

Expected Output:

The game board's squares have different colors.

Instructions:

1. Start a game.
2. Verify that the squares of the game board have different colors.

6.1.14

Function being tested:

There shall be two kinds of guessing squares a team can land on. Double guess where both teams have a chance to guess the right word and single guess where only the team that lands on the square may guess.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and playing a game.

Input:

-

Expected Output:

The game board contains two kinds of squares, one for single guess and one for double

guess.

Instructions:

    1. Start a game.
    2. Verify that the game board contains two kinds of squares.

6.1.15

Function being tested:

During the game there shall be a team chat available, where you can chat with your team members.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in and playing a game.

Input:

Keyboard input in the team chat.

Expected Output:

The game window is displaying a team chat where messages from other teammates are displayed.

Instructions:

    1. Write something in the team chat window.
    2. Verify with another user in the game not on your team that what you wrote is not visible to him.
    3. Verify with a user on your team that what you wrote is visible to him.

6.1.16

Function being tested:

A tutorial with complete instructions for the game is available before and during game play.

Requirement Document reference:

Functional Requirements, page 9.

Initial system state:

The user is logged in to the system.

Input:

The user clicks on the tutorial button.

Expected Output:

A new window appears showing the game rules and a tutorial.

Instructions:

1. Click on the tutorial button.
2. Verify that a window with game rules and a tutorial is displayed.

6.1.17

Function being tested:

It is possible to use a microphone.
Requirement Document reference:

Functional Requirements, page 10.

Initial system state:

The user is logged in to the system.

Input:

Sound from a microphone.

Expected Output:

Sounds from the microphone of other players are played out of the speakers.

Instructions:

1. Talk in your microphone
2. Listen to other players talking in their microphones.

6.1.18

Function being tested:

The ability to store the drawn picture for later use.

Requirement Document reference:

Functional Requirements, page 10.

Initial system state:

The user is logged in and playing a game.

Input:

A drawn picture.

Expected Output:

The picture is saved on your computer.

Instructions:

1. Draw a picture in the game.
2. Save the picture.
3. Verify that the picture is stored on your computer.

6.2 Drawing

6.2.1

Function being tested:

The team members take turns drawing. You cannot choose which one who should draw.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

Play enough number of guesses, to see that the team member's takes turn in drawing.

Initial system state:

The game has started

Input:

Keyboard input from guessing of the word in the Game chat window

Mouse input when drawing a word.

Expected output:

The client display will show the user typed letters on the screen.

If the guess word is correct, you will receive a "Your guess is correct!" output on the display.

Another team member will be up next to draw, the order cannot be changed

Instructions:

1. Choose the "Game chat window" by clicking in it
2. Type in a word and press "Enter"
3. Verify that "Your guess is correct!" output is showing.
4. Verify that another team member's is drawing the next word.
   (The order cannot be changed)
5. If it is the users turn to draw, draw the word
6. Verify that a different drawer will be up next
7. Repeat the above steps over again
8. Verify that the same order of players take turns in drawing

6.2.2

Function being tested:

The player will draw inside of a canvas that everybody can see.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

Draw using the mouse, inside of the drawing area (canvas).

Initial system state:

The game has started and it is your time to draw the word.

Input:

Mouse input when drawing the word.

Expected output:

Display everything that the drawer has drawn.

Every client will display the drawn word.

Instructions:

1. Choose the "Drawing window" by clicking in it
2. Draw something by choosing a specific tool in the toolbox
3. Verify that the drawn figure is shown.
   (Both on the drawers display and on the other players display)

6.2.3

Function being tested:

Before the drawer starts drawing the system shall play a specific sound to let everybody know that the drawing begins.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

Have a computer that supports sound and have the speakers on.

Initial system state:

The game has started and it is time for someone to draw a word.

Input:

-

Expected output:

A specific sound from the speakers

Instructions:

1. Listen to the specific sound

6.2.4

Function being tested:

When a player is drawing, chat and use of the microphone for that player is disabled.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

Have a computer that supports sound, have the speakers on and a additional microphone.

Initial system state:

The game has started and it is time for the user to draw a word.

Input:

Mouse to draw the word

Microphone to talk to other players

Expected output:

Display everything that the drawer has drawn.

No sound from the drawer will be sounded on another client's computer.

Typing in the Game chat window and Team chat window is disabled.

Instructions:

1. Try to talk into the drawers computer microphone
2. Verify on another client's computer that the drawer's microphone is muted.
3. Try to type in the Game chat window or in the Team chat window on the drawer's client.
4. Verify that both chat windows are disabled.

6.2.5

Function being tested:

While drawing it is possible to choose colors and choose between different set of drawing functions, such as: circle, straight line, rectangle/box, eraser and a pen with different thicknesses.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

-

Initial system state:

The game has started and it is time for the user to draw a word.

Input:

Using the mouse to draw the word and choose between the different drawing tools.

Expected output:

Display everything that the drawer has drawn.

Instructions:

1. Click in the Drawing area.
2. Choose between the different drawing tools.
3. Draw using the chosen drawing tool.
4. Chang drawing tool.
5. Verify that it works to change between the drawing tools.
6. Repeat instruction 2-6 to verify that every drawing tools works.

6.3 Guessing

6.3.1

Function being tested:

One can only guess the correct word by typing it in the Game chat window.

Requirement document reference:

Functional requirements p.10

Requirements to test the function:

Guessing of the word is supposed to be in the Game chat window.

The user must know the correct word.

Initial system state:

The game has started

Input:

Keyboard input from guessing of the word in the Game chat window

Expected output:

The client display will show the user typed letters on the screen.

If the guess word is correct, you will receive a "Your guess is correct!" output on the display.

Instructions:

1. Choose the "Game chat window" by clicking in it
2. Type in a word and press "Enter"
3. Verify that "Your guess is correct!" output is showing.

6.3.2

Function being tested:

You have one minute to successfully guess the correct word that your team member is drawing. If you are unable to guess correctly it is the opposing teams turn to draw.

Requirement document reference:

Functional requirements p.10

Initial system state:

The game has started

It is your teams turn to guess the word that one of your team members is drawing

Requirements to test the function:

Guessing of the word is supposed to be in the Game chat window.

Not guessing the correct word within one minute.

Input:

Keyboard input from guessing of the word in the Game chat window

Expected output:

The client display will show the user typed letters on the screen.

The guessed word is incorrect; nothing will happen.

After a minute has passed, you will receive a message that the time is out and that it is the opposing teams turn to draw and guess a new word.

The time is counting down from 60 seconds and is displayed close to the drawing.

Instructions:

1. Choose the "Game chat window" by clicking in it
2. Type in a word and press "Enter"
3. Verify that nothing happened expect that the earlier typed word and other players chat messages are shown in the Game chat window.
4. Verify that the countdown timer is counting down correctly.
5. Verify that it is the opposing teams turn to draw and guess the word.

6.3.3

Function being tested:

Every guess is matched against the correct word. For the answer to be correct it has to be completely correct. Use of upper or lower case is ignored.

Requirement document reference:

Functional requirements p.10

Initial system state:

The game has started

It is your teams turn to guess the word that one of your team members is drawing

Requirements to test the function:

Guessing of the word is supposed to be in the Game chat window.

The user must know the correct word and type in both upper and lower case letter of the word.

Input:

Keyboard input from guessing of the word in the Game chat window

Expected output:

The client display will show the user typed letters on the screen.

The guess word is correct; you will receive a "Your guess is correct!" output on the display.

Instructions:

1. Choose the "Game chat window" by clicking in it
2. Type in a word and press "Enter"
3. Verify that "Your guess is correct!" output is showing.

6.3.4

Function being tested:

If your answer is correct, you may proceed in the game by rolling a dice and move as many steps as shown by the dice (on the game board).

Requirement document reference:

Functional requirements p.10

Initial system state:

The game has started,

It is your teams turn to guess the word that one of your team members is drawing

Requirements to test the function:

Guessing of the word is supposed to be in the Game chat window.

The user must know the correct word.

Input:

Keyboard input from guessing of the word in the Game chat window

Expected output:

The client display will show the user typed letters on the screen.

The guess word is correct; you will receive a "Your guess is correct!" output on the display.

Instructions:

The team's piece on the game board will move as many steps shown by the dice.

5. Choose the "Game chat window" by clicking in it
6. Type in a word and press "Enter"
7. Verify that "Your guess is correct!" output is showing.
8. Verify that the piece on the game board moves as many steps as shown by the dice.