ASSIGNMENT 1

REQUIREMENTS AIR HOCKEY (MOSCOW-METHOD)

AND

USE CASE DIAGRAMS AND DESCRIPTIONS

SEM Group 45

Table of contents

Requirements

Use Case Diagrams and Descriptions

Must have

<u>Functional</u>

Authentication

- The player is able to type in a username
- The player is able to type in a password
- The game is able to connect to a database to check if the inputted username and password are correct
- If the username and password are correct, the player is logged in and able to start a game
- If the username and password are incorrect, an error page is shown and the player has to type in their credentials again
- The player is able to go to a signup page to create an account
- Once the player has put in a username and password on the signup page, the account should be saved in the database
- If the player uses a username that is already in use, an error should be shown

Gameplay

- The player should be able to press a start button to start a game
- Once the start button has been pressed, the game should place a pod at the starting position
- The game shall place a puck in the middle of the field
- The player should be able to move around the pod with the mouse only
- The player should be able to hit the puck with their pod
- The puck is able to go into slot at the other side of the field
- After a goal is scored, the puck is placed back in the middle
- Once a player scores seven goals, the game ends
- If the player scored seven goals first, a victory screen is shown
- After the game ends, the player is able to enter a name with an associated score

Scoring

- The score of both players has to be shown
- After the game is finished, the player should be able to enter name with associated score
- The game is able to save the score of the game in the database
- After the game is finished, the player should have to option to see the top 5 scores
- The game should be able to show top 5 scores

<u>Non-functional:</u>

- For every goal, the player gets points
- For every goal conceited, the player loses point
- For every game won, the player gets points
- For every game lost, the player loses points
- The top 5 scores are the total points accumulated by a player
- The game must be programmed in Java

Databases

- The game is connected with an SQL Database
- The API JDBC is used to connect the game with the database
- The game has prepared statements to counter code injection attacks

Should have

Functional:

Authentication

- If the player uses a username that is already in use, an error should be shown

Gameplay

- The game shall not allow the player to move the pod outside their half of the field
- After a goal, the player has to be reset to their goal position
- The player is able to reset the game through a reset button
- The player is able to pause the game through a pause button
- The player is able to quit the game through a quit button
- The player is able to play against an AI opponent
- The puck is able to go into slot at the players side of the field (own goal)
- If the player concedes seven points first, a defeat screen is shown
- The time played is shown
- When the puck hits the pod, it bounces off the pod with the direction depending on the point of contact
- The game keeps track of the winning streak of the player

Non-functional:

- The player is able to create an account using their email

Could have

<u>Functional:</u>

Authentication
- The game can advise the player on the strength of their password
- When the user creates an account with their email account, a confirmation email is sent to that account
- When the player is logged in, there is an option to sign-out

Gameplay
- The player can choose to play online multiplayer mode (against another player online)
- The player can choose to play split-screen multiplayer mode (against another player on the same screen)
- The player is able to change the colour of the pod through the selection menu
- The player is able to save his top winning streak
- Winning games grands the player MMR points on which their opponents (with similar MMR) is based
- After 5:00 minutes played, the game goes to golden goal mode (first player to score wins)

Scoring
- If the player wants to save their score but does enter a name, the score will be saved with their username

<u>Non-functional:</u>
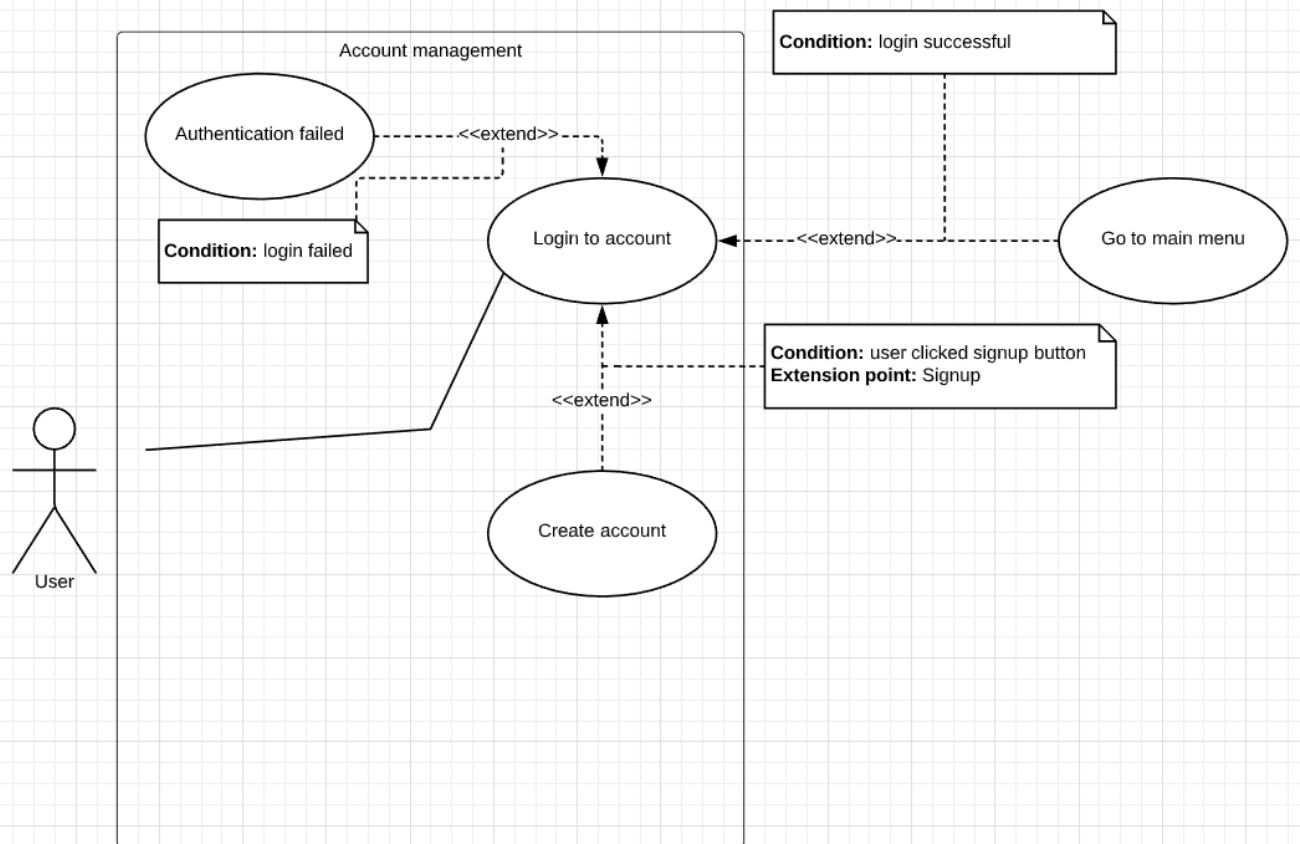- The game has an anti-cheat mechanism to check for cheating players

Won't have

Functional:

Gameplay

- The player can select different kinds of player boards
- The player can select a mode to play in the game in 3D
- The player is able to join an online competition
- The player is able to select a 'unlimited' mode (with no maximum goals or time)

Won't have

Use case diagram 1

Account management

Authentication failed

<<extend>>

Condition: login successful

Condition: login failed

Login to account

<<extend>>

Go to main menu

Condition: user clicked signup button
Extension point: Signup

<<extend>>

Create account

User

Use case description 1

Use Case: Authentication

Author: Nick Ouwerkerk

Date: 28/11/2019

Purpose: Authenticate to be able to get into the game

Overview: The user starts up the program and gets shown a login page. If the user does not have an account, he/she can click on a signup button, which leads to the signup page. When an account has been made, the user gets back to the login page. On the login page, the user can enter his/her credentials. If the credentials are correct, the user is brought to the main menu. If the credentials are incorrect, an invalid authentication error is shown.

Actors: the user (player)

Pre-conditions:

- The database is online and running

Post-conditions:

- The user is in the main menu of the game

Normal flow of events:

| User actions	System actions
1. The user inserts his/her credentials	2. The system checks if credentials are correct
	3. The system brings user to main menu

Alternative flow of events:

Step 1: The user does not have credentials. The user clicks on the signup button and creates a new account.

Step 2: Authentication failed. An error message is shown. The user is able to enter his/her credentials again.

Requirement references:

- If the username and password are correct, the player is logged in and able to start a game
- If the username and password are incorrect, an error page is shown and the player has to type in their credentials again
- The player is able to go to a signup page to create an account

Use case diagram 2
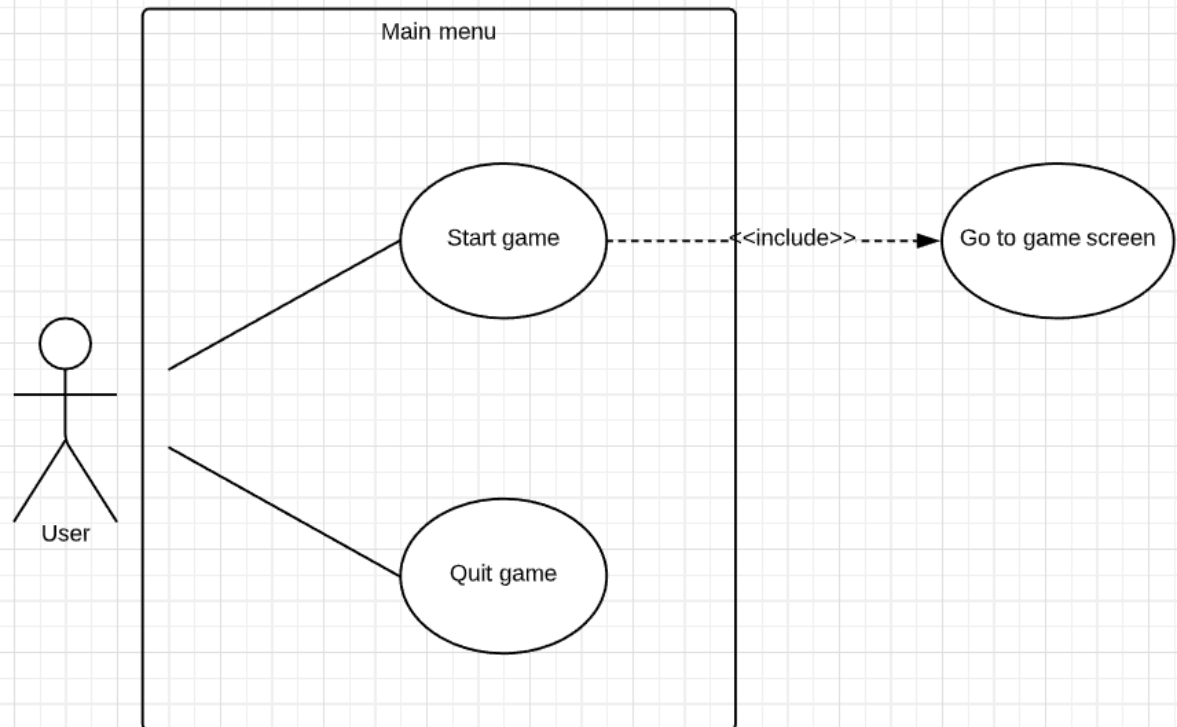
Use case description 2

Use Case: Starting and quitting the game

Author: Nick Ouwerkerk

Date: 28/11/2019

Purpose: Being able to start and quit the game in the main menu

Overview: Once the user has entered the main menu, he/she is able to start the game by pressing the start button, which leads the user to the game screen. Likewise, the user is able to quit the game by pressing the quit button.

Actors: The user (player)

Pre-conditions:

- The user has been able to login to his/her account

Post-conditions:

- The user is not in the main menu anymore

Normal flow of events:

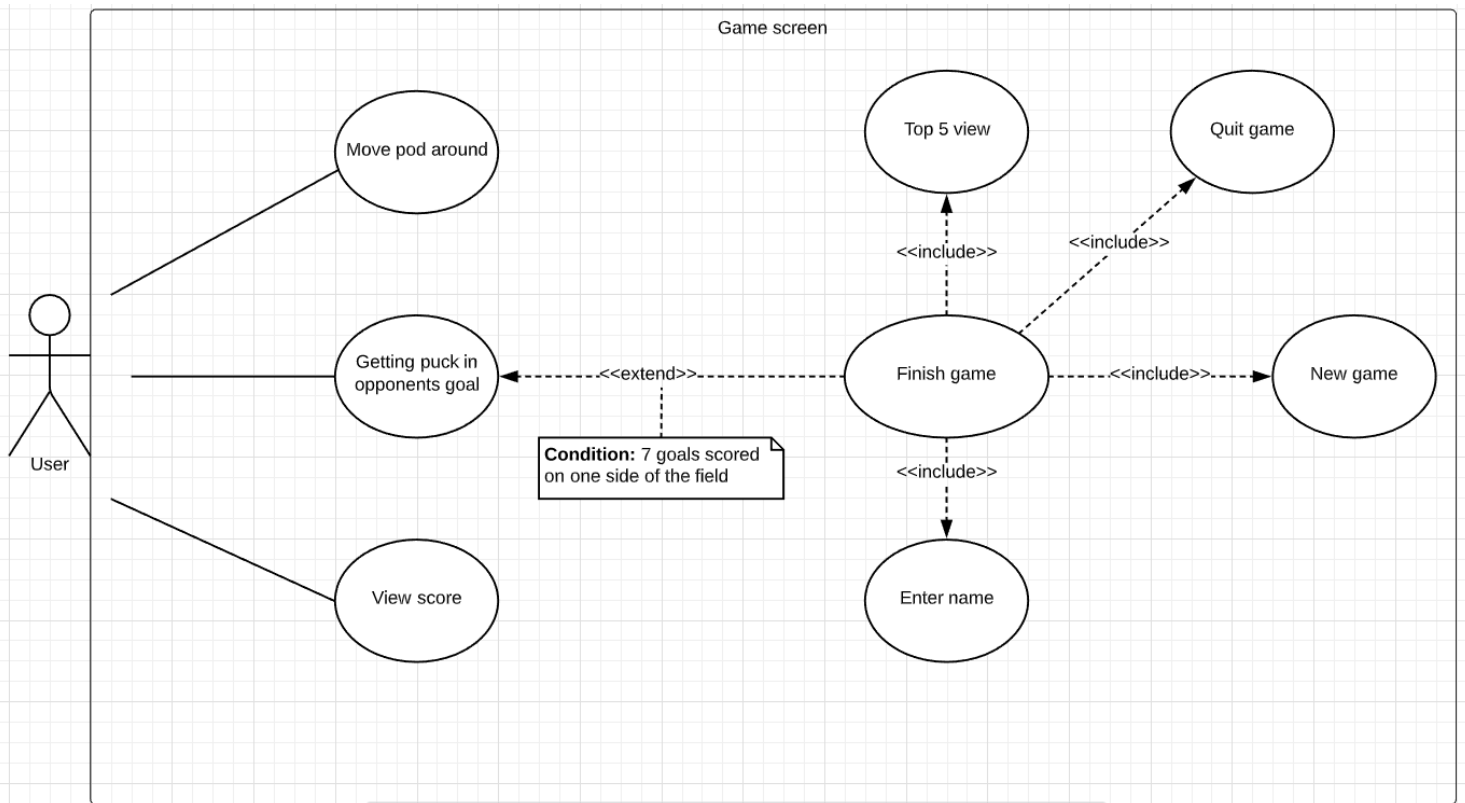| User actions	System actions
1. The user presses the start button	2. The system brings the user to the game screen

Alternative flow of events:

- **Step 1:** The user presses the quit button. This will exit the program.

Requirement references:

- The player should be able to press a start button to start a game
- The player is able to quit the game through a quit button (*should have*)

Use case diagram 3

Game screen

Move pod around

Top 5 view

Quit game

Getting puck in opponents goal <<extend>> Finish game <<include>> New game

Condition: 7 goals scored on one side of the field

<<include>>

<<include>>

<<include>>

View score

Enter name

User

Use case description 3

Use Case: Playing the game

Author: Nick Ouwerkerk

Date: 28/11/2019

Purpose: Being able to play the game

Overview: In the game screen, the user is able to move around the pod with their mouse movement, view the current score and get the puck into the opponent's goal by hitting it with the pod. When the user scores in the opponent's goal, his/her score is incremented by one and the puck is put back in the middle of the field. Once the user scores seven points, the game is finished. The user is able to save their score by entering a name. The user is also able to view the top 5 scores. Furthermore, the user is able to start a new game or quit the game.

If the user tries to move the pod out of the field, the system will keep the pod in bounds.

Actors: The user (player)

Pre-conditions:

- The user pressed the start button in the main menu

Post-conditions:

- The user is playing a game

Normal flow of events:

User actions	System actions
3. The user starts a new game	
4. The user moves the pod around	
5. The user hits the puck	
6. The user scores a goal	7. The system resets the puck to the middle of the field

Alternative flow of events:

Step 7: Seven goals are scored. The game will end, and the user has the option to save their score, view the top 5 scores, quit the game or start a new game.

Requirement references:
- The player should be able to move around the pod with the mouse only
- The player should be able to hit the puck with their pod
- The score of both players has to be shown
- When the game is finished, the player should be able to enter name with associated score
- When the game is finished, the player should have to option to see the top 5 scores