Use case: UC02 - CreateMultiplayerGame

Short Description: The player wants to create a new multiplayer game

Primary actors: Player

Secondary actors: None

Stakeholders: The player gets a new multiplayer game

Overview: This feature will be used when the player wants to play online against another online opponent.

Precondition: The player is in matchmaking

Main course of events:

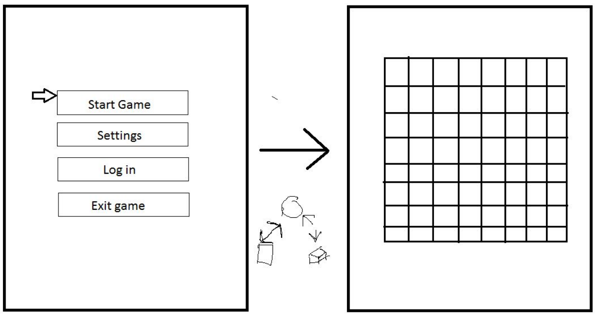
Actor: Player	Actor: Client
1. A player wants to create a multiplayer game.	
<<include>> US04 Matchmaking	
	2. The client receives the game data from the server.

Postcondition: The player has started the game

Alternative course of events: None

Cross references: (US04 - Matchmaking)

Sketch of user interface:



Sprint: 01
Responsible: Niels Nielsen, Patrick Florczak
Status: Ongoing

Use case: UC06 - TakeTurn

Short Description: Allows a player to take a turn in the game

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to take a turn, which allows the opponent to make their turn

Precondition: The player is connected to a game

Main course of events:

Actor: Player	Actor: Client	System
1. A player wants to take a turn	2. The client sends a request to take a turn for the player	3. The server takes the turn in the game
		4. The server sends a response back to both players
	5. The client gets the game data.	

Postcondition: The players has taken a turn

Alternative course of events: None

Cross references: US05 - ConnectGameServer

Sketch of user interface: None

Sprint: 05
Responsible: Martin Hubel
Status: Done

Use case: UC08 - CreateAccount

Short Description: Allows a player to create an account

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to create an account on the server with a username, password and email. The email is used to verify the account

Precondition: The player is not logged in, but is connected to the general services

Main course of events:

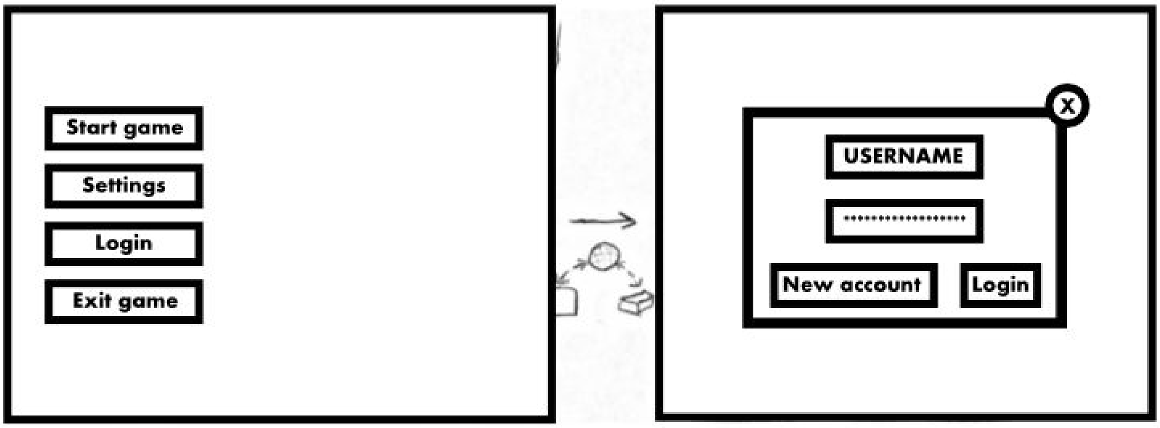
Actor: Player	Actor: Client	System
1. A player wants to create an account.		
2. A player inputs their username, password and email.	3. The client encrypts the credentials and sends them to the server.	4. The server creates an unverified account and sends an answer back to the client.

Postcondition: The player has created an unverified account

Alternative course of events: None

Cross references: US01 - ConnectToServer

Sketch of user interface:



Sprint: 02
Responsible: Martin Hubel, Klaus Riisom
Status: Done

Use case: UC09 - Connect lobby

Short Description: When a user log in as a player, they will then be put into a lobby with other players.

Primary actors: Player

Secondary actors: System

Stakeholders: The player will be able to easily see a list of other online players.

Overview: When a user wants to play, they login as a player. Then the player is put into a available lobby, in which the player can see other online players.

Precondition: The user has logged in.

Main course of events:

Actor: Player	System
1. A player wants to join a lobby.	
2. The player sends their token ID to verify their login status.	3. The system verifies the token ID.
	4. The system finds an appropriate lobby and adds the player.
	5. The system call back the player to notify a successful connection.

Postcondition: The player is put into a lobby.

Alternative course of events:

Cross references: US02 - Login

Sketch of user interface:

Iteration: #02 & #03
Responsible: Martin Fabricius, Patrick Blomberg Florczak
Status: Done

Use case: US01 - ConnectGeneralServer

Short Description: When the player login, the client creates a connection to the server.

Primary actors: Player

Secondary actors: Client

Stakeholders: The player gets access to the game's online features.

Precondition: None

Main course of events:

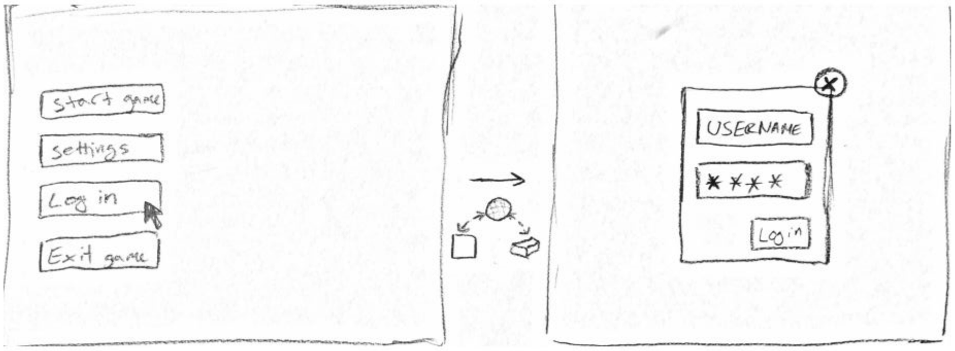
Actor: Player	Actor: Client	System
1. A player wants to connect to the general server.	2. Client sends message to Server.	3. Server receives the message and sends confirmation back to Client
	4. Client receives confirmation.	

Postcondition: Connection to the general server has been established

Alternative course of events:

Cross references: US02 - Login

Sketch of user interface:



Sprint: 01
Responsible: Martin Fabricius, Klaus Riisom
Status: Done

Use case: US02 - Login

Short Description: Allows a player to login on the server, verifying account information

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to login where they provide a matching username and password as credentials to the server.

Precondition: The player is not logged in yet

Main course of events:

Actor: Player	Actor: Client	System
1. A player wants to login.		
<<include>> US01 ConnectGeneralServer		
2. A player inputs their username and password, and submits it.	3. The client encrypts the credentials and sends them to the server.	4. The server authenticates the credentials and sends an answer back to the client.

Postcondition: The player is logged in

Alternative course of events: NewAccount

Cross references: US01 - ConnectGeneralServer

Sketch of user interface: See US01 - ConnectGeneralServer

Sprint: 01
Responsible: Martin Hubel, Rasmus Lassen
Status: Done

Use case: US02 - Login:NewAccount

Short Description: The player account does not exist, presenting the player with the option of creating a new account.

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: When the player account does not exist when trying to login, the player is required to create a new account in order to be able to continue.

Precondition: The player does not have an account in the system.

Alternative course of events:
The alternative course of events begin after step 4 in the main course of events.

Actor	System
<<include>> US04 CreateNewAccount	

Postcondition: A new player account has been created.

Use case: US03 - Logout

Short Description: Allows a player to logout of the server

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to logout of the server after being logged in.

Precondition: The player is logged in

Main course of events:

Actor: Player	Actor: Client	System
1. A player wants to logout.	2. The client sends the token id to the server.	3. The server authenticates the id, logs out the client and sends an answer back to the client.

Postcondition: The player is logged out

Alternative course of events: None

Cross references: US01 - ConnectToServer

Sketch of user interface: See US01 - ConnectToServer

Sprint: 02
Responsible: Martin Hubel, Klaus Riisom
Status: Done

Use case: US04 - Matchmaking

Short Description: Allows 2 players to enter into a match against each other

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to matchmake, entering a queue waiting for another player

Precondition: The player is logged in and connected to lobby

Main course of events:

Actor: Player	Actor: Client	System
1. A player wants to start matchmaking.	2. The client sends a request to enter the player into a waiting queue.	3. The server enters the player into the waiting queue for the lobby.
		4. The server waits for more players to enter the queue.
		5. The server starts a game when another player can be matchmade.
		6. The server sends a message to the clients that they have been matchmade.
	7. The client gets the game data.	

Postcondition: The players has been matchmade

Alternative course of events: None

Cross references: US05 - ConnectGameServer

Sketch of user interface: None

Sprint: 05
Responsible: Martin Hubel, Klaus Riisom
Status: Done

Use case: US05 - ConnectGameServer

Short Description: Allows a player to connect to the game server

Primary actors: Player

Secondary actors: Client

Stakeholders: Player

Overview: A player decides to connect to the game server by supplying the server an access token.

Precondition: The player is logged in

Main course of events:

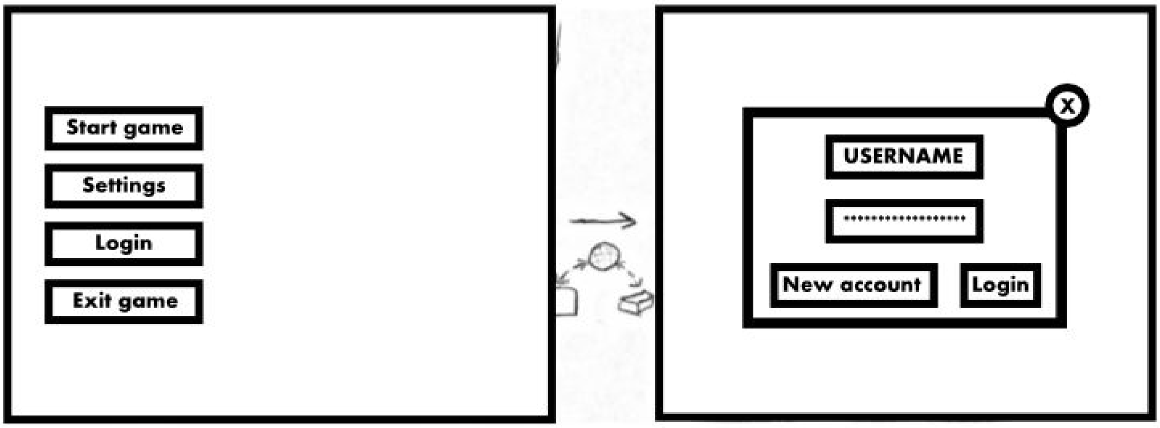
Actor: Player	Actor: Client	System
<<include>> US02 Login		
1. A player wants to connect to game server.	2. The client sends the access token to the server.	3. The server establishes an authenticated connection and sends an answer back to the client.

Postcondition: The player is connected to the game server

Alternative course of events: None

Cross references: US02 - Login

Sketch of user interface:



Sprint: 04
Responsible: Martin Hubel, Klaus Riisom
Status: Done