

Project: Jarl  
Design document  
Group Number: 18

Magnus Andermo  
Mattias Frånberg  
Carl Johan Gustavsson  
Pontus Stenetorp

## 6 Functional Test Cases

### FAILING NETWORK CONNECTIONS (LOBBY, CLIENT SIDE)

#### Requirements 1

**Input** None

**Observable effects** The user shall be presented with a message saying that he does not have a connection with the game server.

#### Procedure

1. Start the game.
2. Wait for the lobby to appear.
3. Pull the network cord.

### FAILING NETWORK CONNECTIONS (INGAME, CLIENT SIDE)

#### Requirements 1

**Input** None

**Observable effects** The user shall be presented with a message saying that he does not have a connection with the game server.

#### Procedure

1. Start the game.
2. Wait for the lobby to appear.
3. Join a game.
4. Wait for a game session to start.
5. Pull the network cord.

### FAILING NETWORK CONNECTIONS (INGAME, SERVER SIDE)

#### Requirements 1

**Input** A started game session.

**Observable effects** The user with a failing network connection shall be removed from the game server.

#### Procedure

1. Pull the network cord on one of the clients.
2. Look in the network log for taken actions.

### SCREEN RESOLUTION

#### Requirements 2

**Input** The game application is started.

**Observable effects** The user shall see a change of resolution (if there are more than one to choose).

### Procedure

1. Press the options button in the menu.
2. Press the resolution button in the menu.
3. Click on the drop down list in the dialog.
4. Choose a resolution.

### REGISTRATION

#### Requirements 3

**Input** The game application is started.

**Observable effects** There shall be a field in the database with the user's information.

### Procedure

1. Press the register button.
2. Fill in a valid username and password (see Table 4 in the requirements document).
3. Press ok.
4. Open the database utility.
5. Search for the user with "SELECT \* FROM users WHERE username='myuser'".

### USER STATISTICS

#### Requirements 4

**Input** A game result

**Observable effects** The database row of the account shall be changed according to the result of the game.

### Procedure

1. Check the database row representing your account and take note of the columns representing wins and losses.
2. Play a game and win.
3. Check the database row again. The value representing wins shall be the old value incremented by one and the value representing the losses shall be unchanged.

### Procedure

1. Check the database row representing your account and take note of the columns representing wins and losses.
2. Play a game and lose.
3. Check the database row again. The value representing losses shall be the old value incremented by one and the value representing the wins shall be unchanged.

## GAME SESSIONS

### Requirements 5

**Input** Data about existing games on the server

**Observable effects** Each existing game session on the server shall conform to the limits that Requirement 6 implies.

#### Procedure

1. Join a game with one player missing.
2. Then the game shall start automatically.

#### Procedure

1. Join a game with more than one player missing
2. Then the game shall start when the other spots are filled with players.

## CREATE GAME

### Requirements 6

**Input** Data to create a game.

**Observable effects** A new game shall be created.

#### Procedure

1. Click on create new game.
2. Check that its not possible to create a new game without a Name (3-30 alphanumeric characters or spaces) or password (3-30 alphanumeric characters) or player limit (integer greater or equal to 2).
3. When the game is created you shall automatically be a participant of the new game.

## JOIN GAME SESSION

### Requirements 7

**Input** None.

**Observable effects** The lobby GUI goes into the GUI state showed in fig. 9.

#### Procedure

1. Go to lobby.
2. Choose a server join.
3. Click the join button in the game GUI, type in the password in the Password field in the dialog.
4. The join should normally be successful but in the rare occurrences when an other client joined the game session and we tried to join it but did so after the other player, but before we got a update about the join event from the server.

#### **LEAVE GAME SESSION**

**Requirements** 8

**Input** None.

**Observable effects** None.

#### **Procedure**

1. Start a game session with two clients.
2. The user clicks the leave button.
3. Continues to LOOSING requirement.

#### **SEARCH GAME SESSION**

**Requirements** 9

**Input** None.

**Observable effects** The servers in the game list is sorted according to the filter settings.

#### **Procedure**

1. Type in filter settings like that he wants the game servers sorted according to.
2. Click on the filter button.

#### **SEARCH GAME SESSION**

**Requirements** 9

**Input** None.

**Observable effects** The servers in the game list is sorted according to the filter settings.

#### **Procedure**

1. Type in filter settings like that he wants the game servers sorted according to.
2. Click on the filter button.

#### **QUICK GAME**

**Requirements** 10

**Input** None.

**Observable effects** The player that choose quick game joins the server with the player with similar user statistics.

#### **Procedure**

1. Start a two game servers and ensure that they are the only game servers in the lobby.

2. Join a player with significantly different statistics on one of the servers and a player with similar statistics on the other server.
3. Start a quick game.

#### **PLAY ORDER**

##### **Requirements 11**

**Input** None.

**Observable effects** Depends on user input.

##### **Procedure**

1. Start at least three clients and join a game session.
2. At the beginning of the game, each player should not be able to send units all the way to the players castle which is located clock-wise.
3. All units sent counter-clockwise should stop half way and not approach the castle.
4. When a player is ousted, the units sent by the player ousting the player shall now continue attacking the next player which is located clock-wise if there are still more than one player with a castle.

#### **DISTRIBUTING ATTACKING AND DEFENDING UNITS**

##### **Requirements 12**

**Input** None.

**Observable effects** Depends on user input.

##### **Procedure**

1. Start at least two clients and join a game session.
2. Change the distribution of attacking and defending units.
3. Observe that the amount of attacking and defending units corresponds to the ratio given at the time when a new wave of units spawns.

#### **SPAWNING UNITS**

##### **Requirements 13**

**Input** None.

**Observable effects** Depends on user input.

##### **Procedure**

1. Start at least two clients and join a game session.
2. Observe that on given intervals waves of units shall spawn from the players castles.

#### **HERO**

##### **Requirements 14**

**Input** None.

**Observable effects** All players has a own hero just as it should be.

**Procedure**

1. Start a game with any number of players you want.

**HERO RESURRECTION**

**Requirements** 15

**Input** None.

**Observable effects** All players has a own hero just as it should be.

**Procedure**

1. Start a game.
2. Go with your hero to the enemy base, wait till you gets killed.
3. Wait till your appears again in your base.
4. Immediately go with your hero to the enemy base.
5. Wait till you gets killed, observe that you gets killed more easily this time (see req 14. Hero).

**UNITS AND BUILDINGS ATTACKING UNITS, BUILDING OR HEROES**

**Requirements** 16

**Input** None.

**Observable effects** The hero dies.

**Procedure**

1. Start a game session with two players, that you control.
2. Build a defense for the first player.
3. Select the hero for the second player, go near the first players defense tower.
4. Verify that the defense tower of the first player starts to shoot at the hero.
5. Stay till your hero dies.

**DEAL DAMAGE**

**Requirements** 17

**Input** None.

**Observable effects** A unit dies.

**Procedure**

1. See *Units and buildings attacking units, building or heroes.*

## **HOSTILE UNIT, BUILDING AND HERO**

**Requirements** 18

**Input** None.

**Observable effects** The enemy two units loses health.

**Procedure**

1. Start a game with two players.
2. Move the hero of the first player to two enemy units.
3. Utilises a stock wave.

## **LOSING**

**Requirements** 19

**Input** None.

**Observable effects** Depends on user input.

**Procedure**

1. Start two clients and join a game session.
2. Let one of the client's castle be destroyed by the other client.
3. The client who got its castle destroyed shall lose the game.
4. Then the statistics shall be updated, according to testcase 4.
5. The territory the client had which got its castle destroyed shall be added to the other client that destroyed the castle.
6. Then the client who lost shall have the option to return to the lobby or be able to observe the rest of the game and not be able to interact with the game.

## **INGAME CHAT**

**Requirements** 20

**Input** Text message to send.

**Observable effects** The message shall appear on all players involved in the game session.

**Procedure**

1. Send a text message using the ingame chat.
2. The message shall appear on all players involved in the game session.

## **MOVE CAMERA**

**Requirements** 21

**Input** Move mouse towards the left or right edge of the screen.

**Observable effects** The camera shall move in that direction



### **Procedure**

1. Move the mouse towards the left or right edge of the screen.
2. Then the camera shall move in that direction.

### **MINIMAP**

#### **Requirements 22**

**Input** None.

**Observable effects** The minimap shows a miniature of the game world and indicates which area the main view is currently viewing.

### **DYING AND DESTRUCTION**

#### **Requirements 23**

**Input** A started game session with two players (both controlled by the tester).

**Observable effects** The attacked game element shall be inoperable.

### **Procedure**

1. Attack the game element with an attacking unit or hero.
2. Wait for the hit points to reach zero.
3. When the hit points reaches zero the game element shall be removed/inoperable.

### **DYING AND DESTRUCTION**

#### **Requirements 23**

**Input** None.

**Observable effects** A unit is dead and that unit can interact with the world.

### **Procedure**

1. In any game you can wait until an archer dies.
2. That unit will not interact with the world and will not shoot.

### **CASTLE**

#### **Requirements 24**

**Input** None.

**Observable effects** None.

### **Procedure**

1. Wait in any game till a player loses.
2. See req. 19.

### **CASTLE**

#### **Requirements 24**

**Input** A started game session with two players (both controlled by the tester).

**Observable effects** One of the players loses.

#### **Procedure**

1. Don't build anything with player one.
2. Create attacking units with player two.
3. Attack player two's castle and wait for the hit points to reach zero.

#### **MOVE HERO**

#### **Requirements 25**

**Input** None.

**Observable effects** None.

#### **Procedure**

1. Build a defensive tower.
2. Move the hero to the same track as the tower.
3. If the hero is at the left side of the tower, move the hero to right, if the hero is at the right side of the tower, move the hero to the left.
4. Continue at 3.

#### **MOVE HERO**

#### **Requirements 25**

**Input** A started game session with a hero alive.

**Observable effects** The hero shall be moved in the direction initiated by the player.

#### **Procedure**

1. Press one of the keys binded to movement, the hero should move.
2. Try to move to a spot occupied by an other game element, by moving in the respective direction. The hero shall not move in this direction.

#### **CONSTRUCTING A BUILDING**

#### **Requirements 26**

**Input** A started game session and enough resources to build one building.

**Observable effects** The building shall be built.

#### **Procedure**

1. Select an affordable building by pressing its build button.
2. Build it on a buildable location (see requirement 26).

**Observable effects** The building shall not be built.

**Procedure**

1. Select an affordable building by pressing its build button.
2. Try to build it on a location not buildable (see requirement 26).
3. The system shall present information about why it could not be built.

**Observable effects** The building shall not be built.

**Procedure**

1. Select a building not affordable.
2. The system shall present information about why it could not be built.

**CONSTRUCTING A BUILDING**

**Requirements 26**

**Input** None.

**Observable effects** None.

**Procedure**

1. Build buildings a free spots until you get the message not enough resources.

**BUILDABLE LOCATIONS**

**Requirements 27**

**Input** None.

**Observable effects** A defensive tower should be built.

**Procedure**

1. Start a game with two players.
2. Try to build a defensive tower on enemy grounds, that attempt will fail.
3. Try to build a defensive tower on a non occupied spot near your own castle.

**ATTRIBUTES**

**Requirements 28**

**Input** None.

**Observable effects** None.

**Procedure**

1. See req. 16 for an example on health.

**UPGRADABLE ATTRIBUTES**

**Requirements 29**

**Input** None.

**Observable effects** New archers produced is stronger than they were before.

**Procedure**

1. Build the production building that enables you to produce archers.
2. Observe how powerful they are by watching how they perform in fight.
3. Upgrade archer.
4. Now all the old archers have the same powerfulness.
5. Observe how powerful the new archers are in battle verify that they are more powerful.