Project Hellknöw

Group 3

Henrik Sandström

Jonas Lindmark

Carl-Fredrik Sundlöf

Tim Hao Li

Usecase 1: Gamer practice

Primary Actor: Gamer

Stakeholders and Interests:

-Gamer: Want to get better at Hellknöw, while getting a distraction from ordinary life.

-Game developers: Wants the play session to work properly since otherwise they will have to address the potential bug and/or error in the game.

Preconditions: Gamer has the game installed on his computer that meets the requirements of the game.

Success Guarantee (post-conditions): The game has closed, and does not affect the computers memory or processor anymore. The gamer is satisfied and not as tense or bored as before.

Main Success Scenario (or Basic Flow):

1. The gamers starts the game and it it loads the main menu.
2. The gamer selects practice mode and the game starts.
3. The gamer moves his character around the map and tries the different weapons he has been assigned.
4. Repeat step 3 until the gamer thinks he has played enough.
5. At some point the player decides that he is done playing and therefore presses the Esc key.
6. The user is returned to the main Options menu where he presses the Exit button.
7. The program terminates.

Extensions:

3. The game crashes

If a crash of any kind occurs the game should close down and the player should be able to restart it.
It will not be possible to resume any state that the game was in.

Special Requirements:

-A user should have a functioning keyboard and mouse.

-The gamer need to have a computer that supports Hellknöw.

Frequency of Occurrence:

Once a week for around 10 minutes.

Usecase 2: Gamer vs Gamer

Primary Actor: Gamer Hosting Game

Stakeholders and Interests:

-Gamer: Wants to play the game to reduce boredom and relax for a while. He also want to do something he enjoys together with his friend.

-Friend: Wants to play the game to reduce boredom and relax for a while. He also want to do something he enjoys together with his friend.

Preconditions: The gamer has a copy of Hellknöw installed and is currently communicating with the other player. Both the players need to have their computers connected to each other either through a local area network or over the Internet.

Success Guarantee (postconditions): The game successfully connects both players. The stage loads successfully with both players assigned weapons. The game ends when one of the players quits the game and successfully shuts it down. The gamers is satisfied and not as tense or bored as before.

Main Success Scenario (or Basic Flow):

1. The gamer starts the game.
2. The gamer chooses Create Game
3. A box appear with the message "Waiting for other player".
4. The gamer sends his IP to his friend.
5. When his friend has successfully connected the stage begins to load.
6. The game start and both players are assigned two random weapons.
7. The players starts moving around the map using the default assigned keys and the mouse.
8. The players now try to kill each other with the weapons they have been assigned.
9. When one player dies the game updates the score for this session and a new round starts. The game returns to point 6.
10. At some point the players agree that they are done playing and therefore presses the Esc key.
11. The user is returned to the main Options menu where he presses the Exit button.
12. The program terminates.

Extensions:

3-5: Initial connection of the two players fail, resulting in a message informing the players that the connection between them failed.

6-9: The program crashes or otherwise is interrupted: The gamer forcefully closes the program using his own knowledge of Windows XP to do so. Afterwards the gamer restarts his client to initiate a new session.

6-9: The connection is lost during gameplay: The user receives a message informing him that he has lost connection to the server and is given the choice of either "Await reconnect" or "Forfeit".

1. The user chooses "Await connect": The user then has to wait for his opponent who has the server to try to reestablish the connection.
2. The user chooses "Forfeit": The user is returned to the main menu screen.

Special Requirements:

-A user should have a functioning keyboard and mouse.

-The gamer need to have a computer that supports Hellknöw.

-A user should have working network connection.

Frequency of Occurrence: Expecting two to three sessions per week.

Open Issues:

-In case of game crash/freeze the users must know how to operate the Task Manager or know of an equivalent method to end the game.

-Provide different stages for more varied play.

Usecase 3: Gamer vs Gamer

Primary Actor: Gamer connecting.

Stakeholders and Interests:

-Gamer: Wants to play the game to reduce boredom and relax for a while. He also want to do something he enjoys together with his friend.

-Friend: Wants to play the game to reduce boredom and relax for a while. He also want to do something he enjoys together with his friend.

Preconditions: The gamer has a copy of Hellknöw installed and is currently communicating with the other player. Both the players need to have their computers connected to each other either through a local area network or over the Internet.

Success Guarantee (postconditions): The game successfully connects both players. The stage loads successfully with both players assigned weapons. The game ends when one of the players quits the game and successfully shuts it down. The gamers is satisfied and not as tense or bored as before.

Main Success Scenario:

1. The gamer loads up the game.
2. The gamer chooses Join Existing Game, the client displays a box with two buttons, Ready and Cancel.
3. The gamer presses Ready and receives his opponents IP which he inputs into his client.
4. The gamer waits for his adversary to initiate the connection and the game, after a couple of seconds the box disappears and the stage starts loading indicating that the connection was successful.
5. The gamer starts up and the gamer is appointed an avatar and two random weapons.
6. The gamer presses the "Return" key to initiate the ingame chat function, types a message and presses "Return" again to send.
7. The message is displayed in semi-transparent text on the screen and after a while his adversary's answer is displayed on top of the gamers message.
8. While occasionally typing messages the user roams the stage using the predefined keys and tries to utilize the weapons to vanquish his adversary.
9. At some point either the gamer or his adversary's avatar dies. The game then updates the score for this session and starts a new round. Returning to point 5.
10. At some point the players agree that they are done playing and therefore presses the Esc key.
11. The user is returned to the main Options menu where he presses the Exit button.
12. The program terminates.

Extensions:

2-5: Initial connection of the two players fail: The gamer receives a message saying that the connection could not be established.

6-9: The program crashes or otherwise is interrupted: The gamer forcefully closes the program using his own knowledge of Windows XP to do so. Afterwards the gamer restarts his client to initiate a new session.

6-9: The connection is lost during gameplay: The user receives a message saying that he has lost connection to the server and is given the choice of either "Await reconnect" or "Forfeit".

1. The user chooses "Await connect": The user then has to wait for his opponent who has the server to try to reestablish the connection.
2. The user chooses "Forfeit": The user is returned to the main menu screen.

Special Requirements:

-A user should have a functioning keyboard and mouse.

-The gamer need to have a computer that supports Hellknöw.

-A user should have working network connection.

Frequency of Occurrence: Expecting two to three sessions per week.

Open Issues:

-In case of game crash/freeze the users must know how to operate the task manager or know of an equivalent method to end the game.

-Provide different stages for more varied play.

-Provide better means of communication within the game.

Non-functional requirements

It should take a small amount of time for the game to start up.

Rationale: Because the game is mainly going to be played for short periods of time, it is important for the user that the actual gameplay can be quickly accessed.

It should be simple to set up a game and invite a friend over the Internet using our game.

Rationale: The process of start up must be fast because the users does not want to spend time waiting instead of playing.

The menu system throughout the game should be easy to understand and navigate through.

Rationale: If complicated very few gamers will be able to play the game.

The update rate of frames in the game should be so high that the user feels that the game flows. (Assuming the user has a computer that fulfills our specification)

Rationale: The goal is for this game to be enjoyable and fun to play, with action being a major part of it. Action that lagging in terms of response to the user is not fun in any way, therefore we will need high responsiveness and high update rates. Dropped frame rates will decrease the users enjoyment of the game and it will also take the users focus off of the actual gameplay. Dropped frames would also effect a users ability to react in a multiplayer situation, with drop frames he might miss his opponent or the opponent might be appearing in the wrong place.

An experienced Java developer should be able to understand how to add weapons to our game by looking at our documentation and source code.

Rationale: The documentation and implementation of the game should make it possible to implement new weapons without changing all the code. This requirement will also set a good standard for the developers so that different parts and features of the game can be written by different developers. This feature is present because the users will want to expand the game in new directions.

Functional requirements

The user should be able to install the game on a computer with Windows XP and version six or higher of Java Runtime Environment.

Rationale: The users interest is to play the game, and therefore he should be able to install he game.

The user shall be able to see the main menu and move the cursor up and down to choose the desired option.
Rationale: The player need to be able to navigate within the game.

The user shall be able to see a submenu with details and options after selecting an option on the main menu.
Rationale: The user needs to be able to know what the different menu options do.

The stage in the game shall be a 2D picture containing a combination of pits, flat plains, stairs, ladders, ramps, cliffs, and obstacles. The contour of the map shall be the surface on which the avatar in the game shall move.
Rationale: The player wants to enjoy the game and have terrains to explore.

The player shall be able to move certain objects in the game.
Rationale: To give the game a more realistic feeling and allow players to change their strategies during a round due that players can not see through objects.

Every map need to have at least two predefined spawn points.
Rationale: Its important that the players spawn in predefined spots so that they don't spawn inside or partly inside a wall or object.

The health points of a player should decreese when hit by a weapon, object or explosion. The amount of health points lost should be determined by the weapon, object, or the cause of the explosion.
Rationale: The goal of the game is to kill each other, so the players would want to be able kill each other.

The user shall be able to see his/her own health and hitpoints status in the game.
Rationale: The user will want to see how close they are to dying so they know what risks they can take and what strategy to use.

A player should die when his health points reaches zero.
Rationale: This determines if a player is dead or alive.

It will only be possible to decrease the health points of a player.
Rationale: There should be no possible way to get more health points. This makes the implementation easier and makes it harder for players to cheat.

Once one player is dead the game should restart and spawn both players with two random weapons.
Rationale: This makes the game more fair since if a player is lucky and gets to spawn with two really good weapons, he will hopefully not spawn with the same weapons the next round and therefor only have the advantage during one round.

The score should update before every restart. If player #1 kills player #2, player #1's score should increase by one and if player #2 kills player #1, player #2's score should increase by one.
Rationale: This enables the users to keep track of the score and it makes the scoring fair.

The user shall be assigned two weapons at the start of each game. The assignment shall be randomized.
Rationale: The random factor of the game is supposed to further coerce spontaneous gaming. The winner of the

previous round pertains no advantages in the next round. We chose two weapons because we wanted the user to be able to switch weapons in game to further excite the gameplay.

The user shall be able to use the pre-configured control keys to move the avatar and the mouse to change avatar's view.
Rationale: To ease gameplay, the fastest way to be able to look around is to move the mouse which will be in the interest of the gamer. The character will then be moved around with a predefined set of keys. This allows the user to for example move in one direction and look in the other, like a real life scenario.

The user shall be able to shoot and attack with their weapon using the predefined keys.
Rationale: The users want to attack and deal damage to the other player.

The user shall be able to select the option Single Player/Practice mode in the main menu. After selecting Single Player the user shall automatically enter a map.
Rationale: We want to provide a single player version of the game mainly because we think users will want to get familiar with the course in a more controlled environment rather than multiplayer. We also think that the user will use the single player mode to play around and learn how to use all the weapons in the game for the same reasons.

The user shall be able to select the multiplayer option in the main menu. After selecting the multiplayer option the player is able to choose between creating a game or joining an existing game.
Rationale: This allows the player and his friend to choose who is supposed to create the game. The one who creates the game is preferably the player with the fastest computer.

In the multiplayer menu the player should be able to his IP adress.
Rationale: The user wants to connect to his friends and therefore he/she needs to know his/her IP.

The user shall be able to chat with the opponent using the chat function provided by the system.
Rationale: The game is supposed to be fun and alleviate stress. The user's will be less stressed if he can communicate with his opponent in the game.

The user shall only be able to see the opponent when they are in sight. Being in sight means directly in contact with the eyes of the avatar without any obstacles between the avatar and the item or the opponent. However, the user shall always be able to see the whole map.
Rationale: A large part of the game will be strategy. The user will want to exploit and utilize the stage as much as possible to sneak around and surprise his/her adversary. By adding Fog of War that prevents the user from seeing parts of the course that the avatar can't possibly see we add both realism and added excitement.

When the user changes the avatar's sight with the mouse the Fog of War will disappear where the avatar is looking and appear where he is not looking.
Rationale: The user want visual representation of that which is avatar is looking at while maintaining the Fog of War where his attention is not present.

If the network connection is lost in a multiplayer mode, the system shall notify both users about the disconnection and prompt the user with options to either reconnect to the other user or quit the game and return

to the menu.

Rationale: Connection issues are unfortunately unavoidable and it is fair that the users wants to be given as straightforward information as possible about this issue. Often the problem is external and the users will in most cases know what the cause is or troubleshoot their own Internet connection themselves. It is also regular that smaller interruptions can disrupt connection between the users, in this case we want to provide a means to resume

The user shall be able to get instructions on how to play the game through the options on the main menu.

Rationale: The user would be able to learn the game as easily as possible.