

AETD - Arch-Enemy Tower Defense

Group 6

Johan Gustafson
Jonas Hellgren
Erik Nordenhök
Olof Ol-Mårs
Felix Wallén

1 Functional requirements

1.1 High priority

- **Download the program.**

The program should be available to download from our homepage.

Rationale: With a easily accessible program more users will use it.

- **Install the program.**

The user in our target audience should be able to install the program without troubles.

Rationale: Installing the program to a specified folder ensures that all files required to run the game are in the correct sub-folders and the program will be able to run as expected.

- **Play singleplayer game.**

A game-mode for only one player that does not require an Internet connection.

Rationale: To provide the users a way to play alone.

- **Build towers.**

To purchase and place a tower on the playing field.

Rationale: The user needs towers to prevent the monsters from reaching their destination.

- **Select a specific tower.**

Select a tower already built on the playing field.

Rationale: The user needs to be able to select their towers to use the functions available such as upgrading, selling or seeing additional information.

- **Kill monsters.**

Most towers attacks the monsters and when the monsters health reaches 0, they die.

Rationale: The user must kill the monsters before they reach their goal or he will lose.

- **Monster movement.**

The monsters will automatically move from one side of the playing field to their goal, always choosing the shortest path available.

Rationale: The monsters need to move to get to their goal.

1.2 Medium priority

- **Host multiplayer game.**

The user makes his computer the host computer of a multiplayer game that supports 2-4 players.

Rationale: A host is required to allow other users to connect to it.

- **Join multiplayer game.**

The user connects to a host computer of a multiplayer game.

Rationale: To play a multiplayer game, at least one user must join a host computer.

- **Remove towers.**

It has to be possible to remove a built tower.

Rationale: If the user is unhappy with the placement of one of his towers he needs to be able to remove it.

- **Upgrade towers.**

All tower types have upgrades that make them more powerful.

Rationale: Towers need to be upgraded to handle more powerful monsters.

1.3 Low priority

- **Receive information about monster's health.**

The health of each individual monsters should be available to view.

Rationale: The monsters health gives the user an idea of how powerful his towers need to be to kill the monsters in time.

- **Send monsters to other players.**

When sending monsters to another player, additional monsters will appear on an opponent users playing field.

Rationale: Being able to send monsters to opponents gives the users a mean to defeat each other.

- **Chat with other players**

The users can send text messages to each other.

Rationale: Being able to talk to the opponents is a fun addition to the game.

- **Configure game settings.**

Alter settings such as player name.

Rationale: Every user prefers their own individual configuration.

- **View individual tower statistics.**

See information about the selected tower such as damage per shot, rate of fire etc.

Rationale: Statistics about the different towers gives the user information about how efficient they are and this will help him to select the best towers to build.

2 Non-functional requirements

- **Performance**

The computer the game is meant to be run on does at least have a 1.4 GHz processor and 512 MB ram. This is the minimum requirement because it is the slowest system we will be testing the game on.

- **Implementation**

The code has to be written in C++ and the graphics must use OpenGL.

- **Space**

The system should be no larger than 32 MB before installing because it should be easy and fast to download.

- **Reliability**

The system will have to sustain the connection between host and clients.

- **Safety**

No illegitimate access to host- and client-computer should be possible.

3 Use Cases

All types of user inherits the Generic-users goal.

Actors	Goals
Generic-user	Play the game
Host-user	Host a game and play it
Client-user	Join a game
Singleplayer-user	Start a Singleplayer-game
Host-computer	Allow Client-computers to connect to a game
Client-computer	Connect to a game
Web-server	Distribute the game

3.1 Installation

Primary actor: Generic-user

Secondary actor: Web-server

Goal: Install the game on the users computer.

Preconditions: Internet connection.

Main path:

1. The user opens a web-browser of his choice.
2. The user enters the address of our website.
3. He presses the download button to download the game to his computer.
4. He runs the executable file that installs the game.
5. The game is installed.

Alternate path:

1. Web-server unreachable, user cant download the game.

3.2 Start singleplayer

Primary actor: Singleplayer-user.

Secondary actor: None.

Goal: To start a game in singleplayer mode.

Preconditions: Game installed on the computer.

Main path:

1. He starts the game on his computer.

2. In the Main Menu he chooses “Singleplayer Game”.
3. The singleplayer game starts.

3.3 Host multiplayer server

Primary actor: Host-user.

Secondary actor: Host-computer, one or more Client-users and Client-computers.

Goal: To host and start a multiplayer-game.

Preconditions: Game installed on the computer. Internet connection.

Main path:

1. He starts the game on his computer.
2. In the Main Menu he chooses “Host Multiplayer Game”.
3. The system now makes his computer the Host-computer and allows Client-users to connect.
4. A screen with a list shows. This list contains the host-user and the Client-users, when Client-users connect to his server.
5. When one to three Client-users has joined, he can start the game by pressing the “Start Game”-button.
6. The game starts.

Alternate path:

1. No Client-users connect and the Host-user can not start the game.

3.4 Join multiplayer server

Primary actor: Client-user.

Secondary actor: Host-user, Host-computer, other Client-users and Client-computers.

Goal: Join and play a multiplayer-game.

Preconditions: Game installed on the computer. Internet connection. An existing server.

Main path:

1. He starts the game on his computer.

2. In the Main Menu he chooses “Join Multiplayer Game”.
3. He is prompted for an IP-address and he enters the IP-address of the Host-computer of the server he wants to join.
4. A screen with a list shows. This list contains the host-user, himself and other Client-users, if other Client-users are connected.
5. He waits for the Host-user to start the game. After a while, the Host-user starts the game and the Client-user gets to play.

Alternate paths:

1. No Host-server exist with the entered IP-address and the Client-user is prompted to reenter the IP-address.
2. The Host-server already has four players connected. The Client-user gets a message that the server is full.

3.5 Build towers

Primary actor: Generic-user.

Secondary actor: None.

Goal: Build a tower on the playing field.

Preconditions: The Generic-user is currently playing in an active game.

Main path:

1. The Generic-user selects the type of tower he wishes to build from a menu on the right hand side of the screen.
2. He moves the cursor to where on the playing field he wants the tower to be built.
3. He clicks the left mouse button to build the tower.
4. The gold needed to build the tower is debited and the tower is built.

Alternate paths:

1. The chosen spot is occupied.
2. The user can't afford the chosen tower.
3. The chosen spot is blocking, meaning the monsters have no path to the exit.

All the alternate paths above lead to the tower not being built and no money being debited.

3.6 Upgrade towers

Primary actor: Generic-user.

Secondary actor: None.

Goal: Upgrade a pre-existing tower on the playing field.

Preconditions: The user has built a tower on the playing field.

Main path:

1. The user selects the tower on the playing field that he wishes to upgrade.
2. The user performs the task for upgrading the tower.
3. The gold needed to upgrade the tower is debited and the upgrade process starts.
4. While the tower is upgrading, it does not function.
5. After a set amount of time, the tower is upgraded and functions again.

Alternate path:

1. The user can't afford to upgrade the chosen tower. The tower does not start the upgrade process and no money is

3.7 Send monsters

Primary actor: Generic-user.

Secondary actor: Other Generic-users.

Goal: Send monster to an opposing player.

Preconditions: The Generic-user is playing in a multiplayer-game.

Main path:

1. The user performs the task for upgrading the tower.
2. A menu appears where the player is asked to choose which player to send the monsters to.
3. The gold needed to send monsters is debited and monsters spawn on the selected players playing field.

Alternate path:

1. The user can't afford to send monsters. No monsters are sent and no money is debited.

3.8 Settings

Primary actor: Generic-user.

Secondary actor: None.

Goal: To change the settings of the game.

Preconditions: The game is installed on the computer.

Main path:

1. The user starts the game.
2. The user enters the “Settings” menu.
3. The user sees the different settings - change name, etc.
4. The user changes his in-game name, and exits the menu by pressing the “Done”-button.

3.9 Sell towers

Primary actor: Generic-user.

Secondary actor: None.

Goal: Sell an existing tower on the playing field.

Preconditions: The user has built a tower on the playing field.

Main path:

1. The user selects the tower on the playing field that he wishes to sell.
2. The user performs the task for selling the tower.
3. The gold earned for selling the tower is received and the selling process starts.
4. While the tower is selling, it does not function.
5. After a set amount of time, the tower is sold and the spot empty.

3.10 In-game chat

Primary actor: Generic-user.

Secondary actor: None.

Goal: Write messages to the other players.

Preconditions: Users must be in a Multiplayer-game.

Main path:

1. The user presses the chat key.

2. The user inputs his message.
3. The message is displayed on all the other users screens.