Settle And Destroy (SAD)
Group 13
Jonas Wikberg
Christofer Hjalmarsson
Daniel Westerberg
Saul Amram
André Sikborn Erixon

Project Overview Document (POD)

1. Who are the users and what problem does the system solve for them?

The probable user is a male between ten to thirty years old. The youngest user could be computer-using ten-year-olds or a young person that is able to read while the older users might be persons with an interest in strategy gaming. The common characteristic between two unique users of our game will probably be the interest for casual computer gaming.

The aim of our game and the incentive for a user to play our game is to satisfy the users need for entertainment and strategic challenges.

2. The main uses of the system.

2.1 General information

This is a real-time strategy game with opportunities to play both multilayer and a special training ground. The first thing a player has to do is to choose an alias to play with. Then you can choose to start your own game, or join another player's game that's waiting for more players. The first thing you have to do when you start your own game is to enter the number of players that will participate. In theory the number of players in a game is unlimited, but since all the slots have to be filled a maximum of eight is a rule of thumb. A player that chooses to join an already existing game has to enter the host ip-address to be able to connect. When players join a game they must press the "ready" button to inform the host that they are ready to start. The host can not start the game until everyone joined are ready. The game starts, and due to its nature players will die as time passes. When this happens you can either choose to stay in the game and observe, or simply leave and join another game. The game ends when all players except one are dead, this one survivor is therefore named the winner. As soon as the game has a winner, people will disconnect and then either decide whether they want to play more or perhaps get back to things they did before they started playing.

2.2 Use Case A

Lasse is at home studying for his exam in two weeks. His computer is on and suddenly he hears a beep from his e-mail program saying he has new mail. He quickly checks if it is important and sees that it's from a friend telling him to try this new game. The download link is provided i the mail and Lasse downloads the game to his computer. When the download finishes he starts the game. The first thing that shows in the game is a window where the user can choose to train (single player mode) or to play a multi player game (multiplayer mode). Lasse chooses to train because his friends don't have time to play right now. The game loads a map and starts the main interface. An information window is shown telling the user that training mode is not winnable, it will go on forever until the user decides to quit. Lasse chooses a place to settle his village and now he can start to play. Lasse learns how to build and expand his village by following the on screen tutorial. The purpose of training mode is for the user to learn to optimize the playing time in order to advance as quickly as possible. When Lasse feels that he knows the concept he quits the game by clicking the exit button. He shuts down the game and continues to study. Unfortunately he can't stop thinking about the game and he looks forward to when he can play the game against his friends...

2.3 Use Case B

Bill comes home from school fairly late in the evening, still having a bit of extra time to spend on something. He turns on his computer and starts searching the Internet for a strategy game to play. He finds the game on a web site, downloads it and realizes that it supports multiplayer mode! So he calls five of his friends that also likes computer games and helps them download the game as well. He tells them to all get on MSN to make communication easier. He starts the game and tells his friends to do the same. In the game he sees, that to start a multiplayer game, somebody has to be the host. As he was the one to find the game in the first place, he naturally chooses the role of host. He chooses an alias, configures the specific game settings and opens up a pre-game screen which his friends can join before actually starting the game. Since his friends are starting the game in client mode, he gives them his IP to connect to, which is specified on his current game screen. He can see the player slots filling up on his pre-game screen, and when everybody are connected and have pushed "Ready", he starts the game. The game is up and running and everybody are competing against each other. He starts building up his village, choosing strategically which buildings to build first. Exploring the map he sees that his opponents do not seem to have any military strength, so he starts producing soldiers. He then attacks his friends one by one, destroying their villages, making each of them lose the game when their village is lost. They are not kicked out of the game but can stay in "Observe Mode", seeing what takes place with the remaining players but not actually doing any useful. When he has conquered all of his friends' villages, the game ends, leaving him as the sole winner. He then shuts down the game and turns off the computer, tired but with a wicked smile on his face

3. The context/environment in which the system is to be used.

The game could be used in most environments and a large variety of contexts. Still our game requires attention why the user must have a certain amount of time to spend playing the game. For example the game could be played at home, in school, during breaks or while traveling.

The game doesn't require any special hardware or operative system since the game will be written in the Java programming language. The game does though require one computer per player with free hard drive space since the game will be downloaded from a website. Furthermore the game requires a network connection for every player when playing multiplayer. Depending on the type of game that is setup an internet connection might be required. A game could be setup within a private network but could at same time be played by players both within and outside that private network if the game host does have an internet connection.

4. The scope of the system

Topic	In	Out
Multiplayer mode	X	
Save functionality		X
Turn based		X
Real time	X	
Training mode (Single player)	X	
AI		X
2D graphical view	X	

Build and expand village	X	
More than one village per player		X
Build and expand troops	X	
Move and attack using troops	X	
Grid based map with obstacles and special cells	X	
Different playable classes	X	
Sound effects		X
Mouse and keyboard control	X	
Identification of any preexisting software that can be used	X	
Simple game chat	X	

5. The main factors that need to be taken in to account when designing and building the system

If a player is disconnected because of bad network connection his village will be destroyed and removed from the game.

One player class can become to strong in relation to others, the game needs to be balanced.

Bad network connection will result in latency and we need to compensate for that by good synchronization.

6. Technologies and Risks

6.1 Technology: Programming language

We choose to work with Java because everyone in the group is familiar with this programming language. Java is also a good language to work with because of its portability.

Risk: It would have been a risk to choose a language that not every member of the group was familiar with. We would spend a lot of time learning it and mastering it.

Solution: We choose Java because we are familiar with it and because the game will be playable on different platforms.

6.2 Technology: Network

The game requires a working network solution because of the multi player mode.

Risk: Complicated and time consuming to develop and debug.

Solution: We will use an existing open source solution called MINA from Apache. MINA is known to be very stable and we have used it ourselves before and can therefore confirm it. It is very stable and satisfies the network needs in our game.

6.3 Technology: Graphics

We have two options: Java 2D or OpenGL

Risk: The group members are not equally competent in graphics programming. Generally OpenGL is much faster and more capable of drawing complex graphics, but basic operations like drawing circles and such is easier performed in Java 2D.

Solution: We will use Java 2D because of easier implementation and due to our general group skills. The game will not require advanced graphics and therefore OpenGL is total overkill.