Project: Jarl
Project Overview Document

Group Number: 18

Magnus Andermo
Mattias Frånberg
Carl Johan Gustavsson
Pontus Stenetorp

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

The users are male and 10 to 25 years of age. They have previously played games and play about two hours a day. Their problem is that there is no modern mix between action, roleplaying and strategy. Games such as these are too old (six years old or more) and no new games which support the same kind of fast mod-maps have been released. There is a gap in the genre which we intend to fill.

2 The main uses of the system.

"Bob is bored, he needs something to do and cheer himself up but he is not interested in playing an advanced online which takes hours of time per day and a monthly fee. Instead he is looking for a quick action and starts up Jarl, which he previously downloaded from the Jarl website. He starts the game and presses "Quick game", he is the automaticly set up with other players also looking for a quick game and is roughly at the same skill level. He plays a game and when he is finished his statistics and his ranking change is displayed. Bob is pleased with his ranking and having bashed some other players, shuts down the game and returns to doing some school work."

"Jim is bored with his current set of games and is looking for a new one. He wonders if there is a free game out there that might be fun and stumbles upon Jarl, some sort of strategy game. At the front page he sees a set of screenshots but he isn't sure if he wourld like to download it. Then he notices a nice "Play now! No download required.". He clicks the button and since he has Java installed an animated splash-screen loads. The game starts and he is asked if he wants to have hints shown during game play. Jim answers yes, since it's his first time. He then chooses quick game. After the game he is shown his ranking and asked if he would like to register in order to save it. He enters his email address and a username and is registered and then shuts down the game."

"Alice and Ethan talk to each other on the IM and Alice invites Ethan to a game of Jarl. They both start up Jarl and Alice uses the "Start game" function to create the game, she also sets a password in order to keep the game private. Ethan then simply uses the "Search game" function to find any game Alice has created or is currently playing. He joins the game after entering their password. After playing they both shut down the game and "

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

The system will be used at home, at school and at internet cafes. Since we will be using Java the game will be portable and run on most systems that supports Java.

4 The scope of the system.

Topic	In	Out
Clean and simple vector styled graphics	X	
Quick games	X	
3D engine development		X
Support for the three major OS;es	X	
Support for up to 8 players per game	X	
Cell phone support		X
Player ranking system	X	
Extensive use of preexisting framworks and technologies	X	
Game lobby	X	
Discrete Z-axis, Continous X,Y-axis	X	
Easy to learn, game system	X	
Optional installation	X	
Anime/Manga;ish design	X	
Forum to keep in contact with the player base	X	

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

A game developed by small devlopers need to be easy to learn and contain enough depth in order to keep a player hooked. Usually a game is marketed extensively and a player base is already established when the game is released. We don't have those kind of advantages. We will need to strike from below, using what little means we have and compensate by using our combined strengths.

- We don't have more than 15 seconds to hook a potential player, this considered, we need one amazing application and an even more amazing webpage to present it. We always have to keep in mind, that no matter how many amazing features we have, the first look has to be stunning. Java webstart is a must, users are not willing to download some game they see for the first time. We need to get them hooked without a download and then hopefully, get them to download it after a game or two since the downloaded version will load much quicker which will be pointed out in the webstart splash screen.

- We need keep players hooked, in order to make a succesful game you need people to play it over and over again. People are competitive by nature, and love to rank themselves among others. By presenting their current ranking and their ranking relative to others we can encourage them to play more and play a few times a day. This is essential for a long term success.

- Another factor that needs to be considered is how we connect back to our player base. Players love to brag, disuss and so on about a game. If we provide a discussion forum players are likely to talk about things they like and things they dislike. Using this, we can let our game evolve and keep

it interesting simply by giving the player base a forum where they can relate.

- Keeping our limited resources in mind, we need to use an extensive amount of API;s written by others. The more we can borrow, the more we will be able to focus on what we really care about, the success of the project. There exist 3D engines, off-the-rack game servers and so on. We need to glue them together.
- Japanese popular culture, primarily manga and anime, has hit the west and is gaining popularity among young players. Clearly, this style of art appeals to our target player base. The best thing about it is that it reduces detail to a minimum and is easily presented on the screen.
- Using a new and innovative way of presenting graphics we may reduce the amount of artwork needed, the amount of game logic and give the game a unique look. We will present the graphic using 2D layers in a full 3D enviroment using a discrete Z-axis. While giving the user a sensation of 3D it allows us to use 2D textures for animation which greatly eases the burden on the artist and also simplifies the game logic.

6 Technologies and Risks.

Risk	**Probability**	**Effects**
Third party libraries can't deliver the performance expected	Medium	Rework of design and the need to spend additional time on work arounds.
Third party libraries prove to be more complex than estimated and take more time to learn	High	Additional time need to be spent learning the API;s
Java has performance issues	Low	Java performance is too low to handle the game logic and additional time need to be spent reworking the code in order to remove bottlenecks.
Webstart is too slow	Medium	Might result in webstart not being used at all or additional time spent on improving performance and looking into webstart workarounds.