

Finnish Game Jam Report

Jaakko Manninen LUT School of Energy Systems, 0418653 jaakko.manninen@student.lut.fi

Jere Kaplas LUT School of Business and Management, 0403105 jere.kaplas@student.lut.fi

Sakari Laine LUT School of Business and Management, 0418598 sakari.laine@student.lut.fi

Toni Nevalainen LUT School of Energy Systems, 0418310, toni.nevalainen@student.lut.fi

What type of game were you making, what was the first draft of the game?

We created a 2D multiplayer game developed for browser desktop platforms with JavaScript, using Phaser.io as the engine and Node.js for the client-server communication. The game is placed around the Indian ocean and is based on Somali pirates attacking a cruise ship and then attempting to drown as many people as possible, whilst the second player (a rescuer) attempts to save as many drowning people as possible. Whoever drowns/saves the most people wins the game.

The first draft was more or less the same thing as the end result, except without the moving background and missing some of the sprites as well as no sounds in the game yet.

How did you come up with the idea for the game, how did you "refine" it?

The theme for the game was waves. We contemplated on physics waves, light waves, etc. but though a game involving waves from the ocean with pirates would be more funny.

We brainstormed with the idea of saving drowning people, but decided to also add the element of a bad guy trying to drown them, as then we could make the game a multiplayer game. The element of waves is really just represented as the scenery of the game, rather than as a gameplay mechanic.

The idea was refined constantly as we were developing, if somebody came up with a great idea we would simply implement it. At no point did we have any revolutionary ideas that would force us to delete parts of the code or to start over on anything really.

How much prior experience did your group members have?

The same group members had participated at least in one prior Game Jam.

Sakari had experience in Node.js and JavaScript, so he was the lead developer of the team, focusing mostly on the code. Jere had experience in phaser and graphics so he worked mostly on the basic mechanics of the game code wise and the graphical side such as sprites and animations. Jaakko and Toni had less experience with game development and programming so they focused on creating some of the graphics, as well as finding and

creating resources (ideas, images, sounds, sprite generators, etc.) for the team and took care of the management side of the project.

Sakari also had experience with flash and video editing so he created the amazing cinematic video for the game.

In a few sentences, describe what applicable experience your group had (such as programming skills, 3D modeling skills, how many years of experience?)

Jere: Programming 3-4 years, graphics editing 6-7 years of experience.

Sakari: Programming 5+ years, graphics editing 5+ years

Jaakko: Programming 1 course for Python, C, graphics editing 2 years

Toni: Programming 1 course of Python, graphics editing 4 years

No 3D modelling/textures required for this game.

Which tools and development methods did you apply, why?

The programming was done using a simple text-editor (Notepad ++), but the game engine (for non-server sided parts) was Phaser.io and for the server side we used Node.js. For graphics (editing) we used Gimp and Photoshop. For editing sound we used Sony Vegas. For filesharing (and version control system) we used FTP; members uploaded the files to the game server and Node.js restarted the game every time a file has been modified, so it was very simple. For the video, we used Adobe Flash and Sony Vegas.

We had previous experience with these tools and they are very easy to use and thus allow for fast development (great for game jams). Node.js was used because we wanted a fast way to create an online multiplayer game.

Which were the most important objectives in your development (visual presentation, the game rules, balance etc...), why?

The most important objective was to get the game working smoothly online. Secondly we wanted simple gameplay mechanics that worked well both on the client and server side. Our team didn't have too much experience with creating graphics for ourselves so we stuck

mostly with free resources we could find and edit for our needs, so we didn't focus too much on the graphical side of the game.

How did you test your game demo?

We tested the demo by playing online on Sakari's website <http://sakkee.org/fgj> ourselves and invited other people to play as well. The game mechanics were edited according to the feedback we received.

For the client sided (engine) changes, we could test changes rapidly by running a local http server, without having to upload the files to the server all the time.

How close to the original idea did you get? (draft vs. actual demo)

Our initial plan was to make an online multiplayer game, where you either save or drown people. Our final product includes both of these so we can say that we reached the "goal".

We didn't plan too much ahead on any features, so we didn't feel like we missed out or didn't accomplish anything. If we had more time left, we could have implemented more features, but that was nothing that we had planned and didn't finish.

What were the most difficult parts in your work, why?

On the programming side, the logic side (in example, real-time movement) was quite difficult. Also creating the boat-character sprites were difficult, I think there were about 40 images in a single spritesheet, as it included the movement of the boat in all 8 directions, as well as the attack/save animations for both characters.

On the animation side, getting the background waves to sync properly was a bit of a pain, as it was a lot of replication and you had to be pixel perfect with the position of the sprites.

Did you change your game design during the implementation? i.e. To eliminate some obstacles because of technology difficulty?

Due to the lack of time we didn't implement every functionality that was discussed, such as the movement / turning acceleration of the boats, but we hadn't planned on doing these

necessarely, they were just some ideas we thought we could implement if we had left over time.

Besides that the game was very simple in its mechanics so we didn't feel like we needed to change anything or dumb the game down to make it playable.

Did you use some any ideas from systematic software development methodology (such as Waterfall or Agile) during the game development? if you did, which one did you select? and if not why?

We didn't use any, because our plan was very clear from the beginning and everyone knew exactly what to do in order to make the game work as expected. Because we knew specifically what we wanted and what we were able to execute from the very start, our plans didn't change in the middle of the project and no systematic software development methodology was needed.

As a game developer, do you prefer advanced game engines (such as Unity3D) or traditional tools and libraries for game making? Why?

We prefer both. Advanced game engines do help a lot but the libraries and "traditional tools" give you more power over your game.

For this game we did use a JavaScript "Engine" or framework for the development of the game, and Node.js framework for the server side, but the game logic was really mostly developed by hand in JavaScript. So in the end it's a bit of both that was used.

Did you design the game architecture and write down the design document during game development?

No. We had some basic idea of how to develop the game as we had done something similar in the past, so we didn't feel like we needed to write down a lot of documentation, especially considering how simple the game is in the end. Some practicalities were sketched and discussed before implementing, such as how to detect the "trigger area" for the players depending on the angle.

How large proportion of your project time did you dedicate to the game design?

We brainstormed for about two hours on the first evening before starting to write code and designing graphics.

Mostly the design of the game came together as we were developing, instead of setting everything in stone right from beginning.

It's hard to estimate how big portion of the project was spent on designing versus actually developing the game, as it was a mixed process. If I had to guess I'd say it was about 30% designing and planning.

Should you continue with the demo development, which would be the next activities you would do?

Adding a communication system to the game, either voice chat or emotes or a chat so you can easily communicate with the player you are playing against. We could also apply some sort of a ranking system.

We could also improve the game mechanics (better physics like acceleration, more features, customization of characters, etc.)

Also, the game mechanic could be changed a bit, so that in the start there would be more NPCs to rescue/drown and instead of the spawning time accelerating, it could be slowing.

If you were to redo the demo completely, what would you do differently?

We would start developing directly on the Node.js platform, instead of first building a prototype on the Phaser engine. This would save a lot of time, as often the work done on the client sided engine was just replicated on the server side.

We might also spend a little more time designing the game and the architecture of the code a little more, before starting the coding.

Open feedback on the Game Jam event, this course, or anything related.

Free coke was nice. The event was well organized, and we were able to create a great game. Maybe some other food than chips next time. Some of the tables were not very steady and the chairs caused some serious back pain.

