**Maces and Talons: Hnefatafl**

|  |  |
| --- | --- |
| Report Name | Outline Project Specification |
| Author (User Id) | Callum Gwynedd Hay Hutchinson (cgh2) |
| Supervisor | Bernie Tiddeman (bpt) |
|  |  |
| Module | CS39440 |
| Degree Scheme | G401 Computer Science (inc Integrated Industrial and Professional Training) |
|  |  |
| Date | February 07, 2016 |
| Revision | 0.2 |
| Status | Final |

# Project description

The project has the purpose of producing an interactive video game, on mobile and/or other platforms, that allow users to play the Medieval Scandinavian board game ‘Hnefatafl’ [1]. The game functions by having two opposing players controlling two different sets of pieces, one set is the king and knights and the other controls the barbarians. Unlike a comparable game Chess, the two sets of pieces are not equal and both sides have different goals to achieve victory.

The setup of the game depends on the variety of the game chosen, many different versions of the game have been documented and vary depending on geographical location of origin. The version that the project aims to replicate uses a board of 11x11. The pieces of the game will need their own unique 3D models, which I will either must create using modelling software or find online with suitable licensing that allow their use in the project.

The basic feature of the game should allow two player hot seat mode using a single device, by passing the device from player to player taking turns to move their desired pieces. A more complicated version of the project will then use a server application to provide multiplayer support and allow two different devices to connect and player together.

The game application itself will be created in Unity 3D engine, which will allow deployment to multiple platforms namely Android and Windows. Due to having a mobile platform as a target the interaction in the game must accommodate the touch screen controls of those devices, while also still allow regular desktop controls.

The server application could be programmed in either C# or Java both allow for a variety of different pros and cons, C# allows much easier integration with Unity 3D since that will be the language used for that, but the application will only be executable on Windows servers. Java while natively more accessible on many devices it will require more work to create communication between the server and the game client.

# Proposed tasks

**Development** –

**Game application** – the game will allow two player gameplay on a single device with implemented movement and ruleset, the ability to connect to a multiplayer server will allow a single player to connect and play against another remote player.

**Server application** – the server should allow devices to connect to it and provide them with multiplayer functionality. This includes matchmaking lobby, social messaging, as well as multiplayer gameplay. The ruleset will need to be implemented just like the offline version.

**Feature stories** – the features of the different applications will be planned out using story cards and progressively getting more detailed as a feature enters development.

**Regular backups using version control system** – using a private repository hosted on github the project will have its data backed up on a remote server. All parts of the project will be stored in this repository, including code and documentation, to allow continuous backup as well as easy access across multiple development machines.

**Weekly Diary** – to document the progress of the project on a weekly basis a diary will be compiled to allow for review and comment on the project. Challenges encountered should be recorded along with any technical breakthroughs and features worked on.

# Project deliverables

**Android and PC Client application** – These applications should allow the user to either play a two-player game on that device ‘hot seat’ mode where the device is passed between two players. Or allow the user to search for remote/multiplayer games by connecting to the server.

**C# or Java based server application** – This server will be designed to run on either Windows or any JRE capable machine. It will accept connections from the game client and allow a series of multiplayer features, such as player matchmaking and online gameplay.

**Mid-Project Demonstration** – A version of the project must be presentable for a demonstration mid-way through the project time, this demonstration must be able to show off the project in a working form. It will not be complete but needs to show large progress on vital system features.

**Final Demonstration** – This demonstration should be used to show off the completed software that has been produced over the course of the project. All features should be working and permitting enough time should be demonstrated to the marker.

**Final Report** – This will document the process that was taken to complete the project, assess how this was achieve and where improvements could be made. In the report references to software used as well as any project influences should be noted.

# Initial annotated bibliography

1. Various. Game Cabinet. [http://www.gamecabinet.com/history/Hnef.html . Accessed February 2018](http://www.gamecabinet.com/history/Hnef.html%20.%20Accessed%20February%202018).

*This website outlines the rules of many varieties of medieval board game, including the one that is relevant to the project.*

1. Unity 3D. <https://unity3d.com/> Accessed February 2018.

*This game engine will serve as the main development environment for the mobile and desktop client application.*