Cardiff University

Computer Science

CM3203: One Semester Individual Project, 40 credits

Initial Plan

Real-Time Networking Game

Author:
Braden Marshall

Supervisor:
Frank C. Langbein

30th January 2017

Contents

1 Project Description

The overall aim of this project is to implement a game, played by at least two people, which uses real-time networking to communicate between all players. The game is to also feature a computer player, which makes decisions using some form of artificial intelligence.

I have chosen to design and implement a game inspired by the light cycles featured in the 1982 film Tron (Walt Disney Productions, 2014). The concept of the game is fairly simple: the involved players spawn uniformly around the outskirts of a square arena. The players then move at a constant speed, without the ability to stop, and leave a trail - known as a *jet wall* - as they go. If a player hits the arena's border or another players' jet wall, they *die*. The last player left *alive* in the arena is crowned the winner.

For the sake of portability, the game is to be developed for use in a web-browser. This is primarily because it requires an absolute minimal amount of effort, from the human user, to get playing. It will also give me with the opportunity to further learn the various technologies introduced within HTML5, along with how they can collectively be used to create a working application.

Because the game is browser-based, all the client-side code for this project is to be developed in the three primary web-languages: HTML, CSS, and JavaScript. However, the JavaScript will be developed to the next-generation ECMAScript 2015 (also known as *ES6*) standards, but for usability; shall be transpiled into current JavaScript using Babel (Babel (Open Source), 2016). This is because ES6 adds a significant number of improvements to the former JavaScript specifications, which I believe will aid development speed and code quality.

To avoid having to add another programming language to this project, I have chosen to use JavaScript via NodeJS to handle the server-side processing. I have never really touched NodeJS before, but have heard good things regarding its use for dynamic web-applications.

2 Aims and Objectives

In order for this project to be a success, it is important that I first break down the project into a set of explicitly stated aims and objectives. This will serve as an explicit vision of what should be achieved by the end of the project.

Below is such a list of objectives, and the tasks encompassed within each:

- Base game:
  - User-interface: chosen graphics library, and got keyboard input from the user working.
  - Game arena: create the arena, along with its internal representation. Player should now be able to move around
  - Game logic: Remaining game logic, such as the *killing* and *winning* conditions.
- Networking:
  - Initial Synchronisation: primitive synchronisation between competing players.
  - Optimisation: implement more optimal synchronisation techniques, to improve performance.
- Artificial intelligence:
  - Controller: an interface in which an AI could attach itself and become a player in the game.
  - Player: consider, then implement the various AI techniques which would result in an ideal computer player.

2.1 Optional

Should the project be completed with plenty of time remaining, we can consider implementing the following list of objectives:

- Arbitrary players:
  - Two players: ensure that the game first works exceptionally with just two players.
  - Multiple players: implement modifications to stretch the game to support an arbitrarily low (say <10) players in a single game.
  - Optimisation: should the game begin to run poorly, begin to consider possible solutions.

- Multiple servers:
  - One server: ensure that the game first works exceptionally with just a single server.
  - Interface: modify user interface to allow users to set up their own servers and connect with friends.
  - Multiple servers: implement modifications which will allow the server to handle multiple concurrent servers.
  - Optimisations: add optimisations to the server, to better handle this new higher amount of servers.
- Polish:
  - Compatibility: ensuring that the game works for most popular browsers and devices.
  - Aesthetics: the game looks visual appealing.
  - Security: protection, or at least the consideration, against malicious users.

3 Work Plan

Now that we have briefly discussed the main objectives and aspects that make up this project, it becomes possible to begin construction of a work plan which we can adhere to remain on track in reaching completion before the project's deadline.

In the following section, we shall discuss the aims and objectives which are listed above to help form a justification regarding the work plan. We shall consider a few other aspects of the project.

3.1 Base Game

The first major objective of this project is to develop the game engine and gameplay mechanics. This will result in a playable game, which we can then begin to confidently extend upon with additional functionality such as networking and an AI computer player.

3.1.1 User Interface

Development of this project will kick-off by identifying a library to handle all our graphical requirements. This library will play a crucial role in the game, so it is of critical importance that it is capable of high frame-rates and isn't too restrictive; so we are able to implement the remaining functionality without too much friction.

I consider this task to be a combination of both background research and implementation, as I will have to consider methods of implementation in deciding whether a particular graphics library is suitable. It is also likely that I will create quick experiments in these libraries, to aid in my determination of the previously mentioned consideration.

3.1.2 Game Arena

Another task is the creation of the game's arena. It is highly likely that the arena will internally be a represented as a grid, where a player is to occupy a single cell and mark previously visited cells as being a 'jet wall'. This arena is to be made easily interpretable by the user, that is they're able to determine where is safe to move etc...

3.1.3 Game Logic

This task of the base game involves the implementation of any remaining logic required by the game, for example *killing* a player and detecting when the game has ended and the winner is to be chosen and glorified for their incredible triumph!

3.2 Networking

The second major objective of this project is to design and develop all the network communication aspects. Simply put, when this objective is complete, multiple players should be able to connect remotely and play the game against one another. The main challenge of this objective is to construct a system of well-thought-out techniques to minimise latency and make the game as fair, and synchronous, as possible for all players involved.

3.2.1 Initial Synchronisation

The first task to be completed within this objective is to broadcast, and somewhat synchronise, the game's state to all connected players. I predict that this will take quite a large amount of work on both the client-side and server-side. I am unable to further expand upon this objective, as I am not that familiar with HTML 5's real-time technologies.

3.2.2 Optimisation

Once an initial form of synchronisation has been established, among all the players, I shall then begin to run various tests to see how well the connection performs under stressed conditions etc. If applicable, and possible, work shall then be focussed on how to remedy any weak-points that may have been discovered.

3.3 Artificial Intelligence

Once the base game exists and multiple players are capable of playing against one another, it then becomes relevant to begin development of a computer player powered by some artificial intelligence technique(s).

Due to the fact the user is to compete against the computer, it would not be ideal to implement a near perfect and unbeatable AI players. Thus, the focus of this objective is to develop an AI which is enjoyable to play against, that is it must be within reason to beat.

3.3.1 Controller

It must be considered that a computer player is not equivalent in implementation to a human player. For example, it will most likely exist within the server - making it less susceptible to network-related latency. So, the first task towards achieving this goal is to develop some controller in which an AI could be *plugged* in and participate within the game.

3.3.2 Player

From the start of this project, we can pretty much immediately begin to consider the various AI techniques which can be utilised to control our computer player. So, I plan on carrying out background research somewhat consistently over the coming weeks. However, it only really becomes possible to implement and test our AI once the game is sufficiently developed. Most notably, the AI controller must be pretty much completed.

3.4 Extra Features and Polish

If the above objectives have been completed successfully, and if there is sufficient time remaining, I *may* start to consider any remaining features that the game lacks. However, all tasks described within this object are purely considerations, and I hold no guarantee that any of it will be implemented. I shall discuss these throughout the remainder of this subsection.

3.4.1 Arbitrary Players

The game thus far has been rather unspecific regarding the number of players participating, but by default, it has assumed to be only two. However, it could be interesting to see how easily scalable the game is to an arbitrarily low (such as less than 10) player count.

3.4.2 Multiple Servers

By default, the game will feature just a single server, that is only a single game can be played at any one time on the web-server, and most commercial multi-player game servers support an arbitrary amount of concurrent games. Hence, this could be an interesting task to explore.

3.4.3 Polish

The extent of this task is fairly open-ended. It does not refer to any particular non-existent feature, but rather to how the existing features, and the game as a whole, could be improved upon. Some improvements that come to mind are as follows:

- Browser and device compatibility.
- Aesthetics of the game and its user-interface.
- Security and prevention against hacks/exploits.

3.5 General Efforts

There are some aspects that exist throughout the project's development whose existence don't strictly change the final outcome. However, these aspects can be considered *good practice* and will almost aid development speed and the underlying quality of the final product.

The following are ideologies that are to be maintained throughout the course of the project.

3.5.1 Development Methodology

As this project consists of several clearly distinct components, it shall be developed in an iterative, incremental and evolutionary manner. The reasoning for this is that the game will almost constantly remain at a somewhat playable state, which will demonstrate progression. Also, it allows me to work comfortably whilst remaining productive.

3.5.2 Version Control

A *git* repository will be maintained for the development of the project's source code. It grants us the ability to keep a log of all changes and remain flexible whilst working on different features etc...

3.5.3 Documentation and Standards

Wherever possible, the code-base should be well-commented, with useful variable names etc. Also, time should be invested in learning a generally accepted set of standards that we can adhere to. A linting program can often be used to enforce these.

3.6 Milestones

As touched upon within the Aims and Objectives section, this project consists of three main objectives: base game, networking, and the artificial intelligence. I propose to have a review meeting, with my supervisor, upon the completion for each of those three objectives. I shall consider these meetings to be *milestones* of the project.

Yet I do believe that they will help to determine whether progress is on track and that the project is indeed heading in the desired direction. I predict they will consist of a total review of everything that has thus far been accomplished, along with what's currently in the pipeline, as well as what might still be missing. Ideally, the discussion would then turn towards the future of the project, such as the next main objective.

On top of this, my supervisor and I have arranged to meet once a week, in order for to try to resolve any questions or confusion I may have regarding the project.

3.7 Report

Throughout the duration of the project, alongside working on everything else, I plan to consistently make some progress on my report. But, to remain safe, I have set aside the final week to solely be used for finalising the report and to possibly work towards some remaining extra features and polish.

3.8 Gannt Chart

In the following page is such a work-plan, presented in the form of a Gannt chart(Wikipedia, 2016):

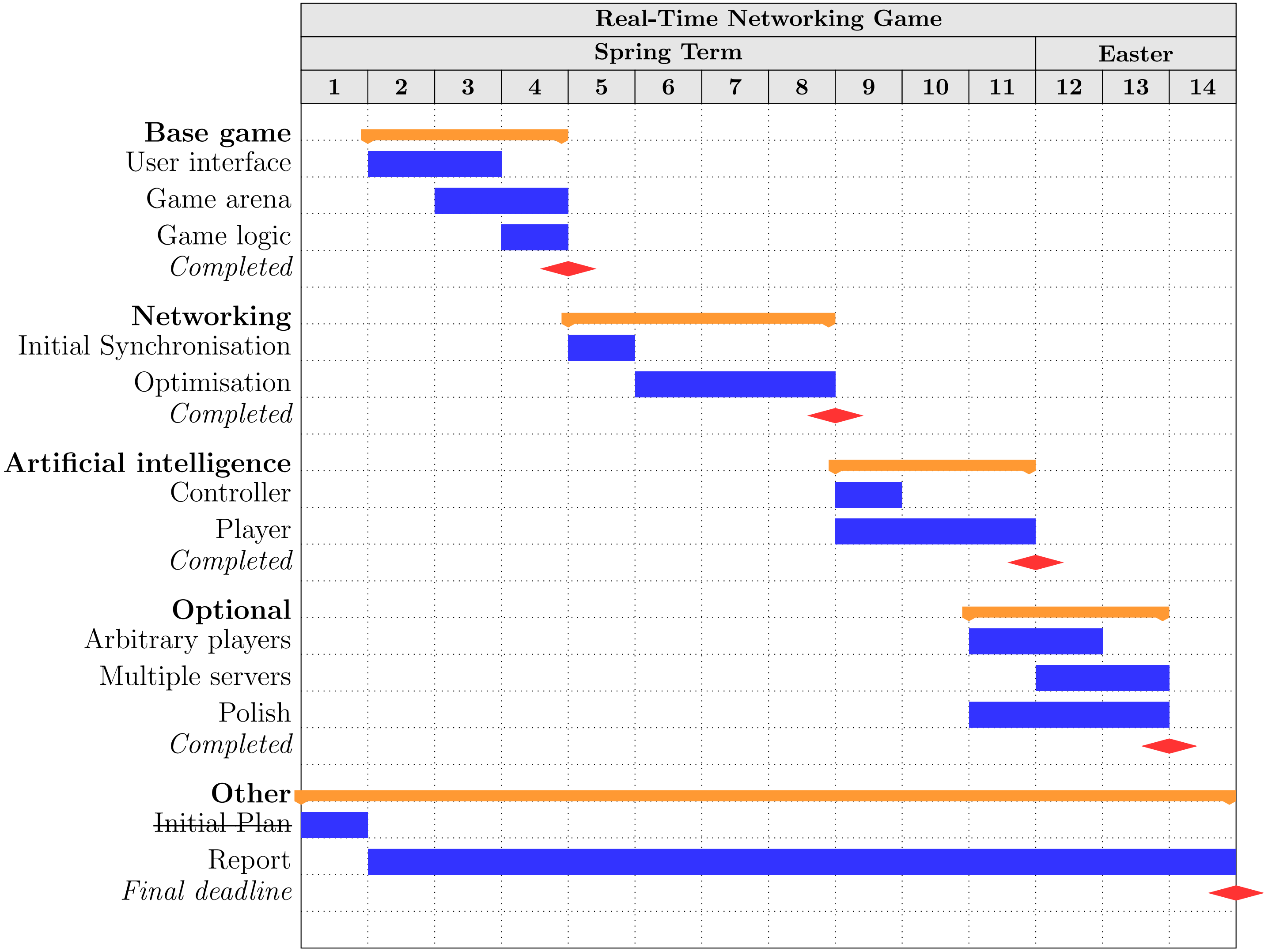

Figure 1: Gantt chart for Real-Time Networking Game's Work Plan

References

[Bab16] Babel (Open Source). *Babel · The compiler for writing next generation JavaScript*. 2016. URL: `https://babeljs.io/` (visited on 26/01/2017).

[Wal14] Walt Disney Productions. *Tron (1982) - "Light Cycle Battle"*. 2014. URL: `https://www.youtube.com/watch?v=7DgL_w5qwIw` (visited on 26/01/2017).

[Wik16] Wikipedia. *Gantt chart - Wikipedia*. 2016. URL: `https://en.wikipedia.org/wiki/Gantt_chart` (visited on 28/01/2017).