

# FinalYearProject

By Cormac Raftery G00348802

## Abstract

This dissertation is based on the project 'FinalYearProject', which was completed as part of the module 'Applied Project and Minor Dissertation' by Cormac Raftery. This project is a server-client-database application, using MAMP as the data tier and unity as the logic and presentation tier. An objective of this project was to gain an understanding of the complexities behind connecting users that are not using the same system in real time and to further my own knowledge in games development.

An agile approach was taken to develop a system that uses a SQL database that can be accessed by a Unity application. Throughout this investigation I found that unity can be very easy to work with but with its constant updates, some of which can render methods obsolete can also be very tricky to find up-to-date information in areas.

This project has been developed fully by Cormac Raftery with ID number G00348802. The project source code may be found on the GitHub Repository:

<https://github.com/CormacRaftery/FinalYearProject>

Video for the project:

[https://www.youtube.com/watch?v=3\\_VWhPnR4ck](https://www.youtube.com/watch?v=3_VWhPnR4ck)

## Project Objectives

1. Gain a better understanding of gaming development.
2. Create a system capable of hosting an online lobby.
3. Create a user-friendly interface.
4. Integrate a feature to allow a player to join from a different system.
5. Authentication that's simple to use but secure.
6. Develop skills working with agile methodology.
7. Have all statistics for each player recorded across games.

## Methodology

This project was developed using the waterfall methodology. The developer decided what they wanted done. Researched into which technologies would best suit the specification. Then a plan was set with a goal for each week. After each objective was completed and the developer was satisfied with how it worked, the next step was started.

## Database Design

Once the base idea for the project was selected; a database could be designed. It was looked at what information would need to be saved in the database that could potentially apply to a player.

Once this simple database wireframe was set up, it was possible to write queries against the mock data to ensure that the database server was working as expected and that the data could be interacted with.

With a back-end wireframe up and running, and that it could perform CRUD operations on very simple data using all of the required technologies, work could begin on the interaction with the unity project.

Now that it was verified that a front to back end wireframe was in place and working correctly it was felt that sufficient research and experimentation had been conducted and now sprints to begin building on this wireframe could begin.

## Hosting and Navigation

The application was designed to be hosted locally with an access point opened at port 6321 for a player to connect using TCP.

The application is navigated using the Unity SceneManager scripting API.

Access to the database was done using cs scripts in Unity and php files in the MAMP program.

## Front End Configuration

To begin front-end development, the view of the Main menu is displayed. On the main menu there are 3 buttons and a username. The 3 buttons each push to separate pages. The register button brings the user to the register page where the user is prompted to register an account. The Log In button will bring the user to the log in page where the user will be prompted to log in using an existing account in the database. The play button only becomes available after the user is logged in and will start the game.

## Front End and Back End Connectivity

With the initial layout of the game completed using the unity UI the program can then be controlled using cs scripts. The cs scripts for the database side of the application are then connected to the php files in the MAMP sqlconnect folder.

All controls for the gameplay itself are done within the cs scripts, mainly GridSpace.cs and GameController.cs.

The Server-Client portion of the project is contained within the Client.cs, Server.cs, GameManager.cs scripts.

The database functionality of the project is contained within the scripts Login.cs, DBManager.cs, Registration.cs, Game.cs and Webtest.cs.

## Styling

The unity UI functionality was used heavily to achieve the simple user interface with elements of cs used to make buttons active or inactive.

## Technology Review

This section is comprised of an in-depth view of all the technologies used throughout the project.

### Data Tier

#### *MAMP*

MAMP is a free, local server environment that provides high performance and is easily scalable. MAMP is perfect for applications that involve working with smaller projects as it is very easy to install and very intuitive to use. MAMP includes all the necessary tools for local testing purposes such as Mac OS X, Apache, MySQL and php which make the abbreviation MAMP. MAMP tops any list for the best localhost server.

#### *PhpMyAdmin*

phpMyAdmin is a popular and free open source tool used for administering MySQL with a web browser. Typical operations such as the management of databases, tables, indexes, permissions, and so on are executed with the user interface. Administrators can also use phpMyAdmin to directly execute any SQL statement.

### Logic Tier

#### *Php*

PHP: Hypertext Preprocessor is a scripting language that interacts directly with the MySQL database set up in PhpMyAdmin. Interacting with the database I set up in PhpMyAdmin was very simple through the use of php and connected to the cs scripts very well. The code generation features provided by certain PHP frameworks enable programmers to keep the source code of web applications clean and maintainable.

#### *C#*

C# is an object-oriented programming language that is used to write different types of applications using the .NET Framework. C# is the typical language used for scripting when using unity.

### Presentation Tier

#### *Unity UI*

Unity UI is a UI toolkit for developing user interfaces for games and applications. It is a GameObject-based UI system that uses Components and the Game View to arrange, position, and style user interfaces.

### Development tools

#### *Git/GitHub*

Git is an open-source version control system that was started by the same man who created Linux; Linus Torvalds. Git is considered, by many, to be the most preferred version control system on the market. A Git "Repository" is simply a directory on a machine that is being tracked by Git. Any changes made to any of the files in this directory can be "added" and then "committed" to the repository. The users have control over what files get added, committed or even tracked by Git within that repository. GitHub is a web-based hosting service for version control using Git. Users of Git can create remote versions of their Git repository and "push" changes made in their local repository to the remote repository on the GitHub servers. GitHub has multiple useful features including a nice graphical user interface, in contrast to the

Git command-line tool. Git is also one of the most popular version control systems in industry. Git is used in nearly every module in the software development course and it is obvious why.

## Development environment

### *Visual Studio*

Visual Studio is a very popular IDE developed by Microsoft. Visual Studio provides support for debugging, embedded Git control, syntax highlighting, intelligent code completion, and more. Visual Studio Code was selected as the script editor for the project.

### *Overleaf*

Overleaf is an online latex editor that was used to write this dissertation. It has proved very simple to use for a programmer compared to the likes of MicroSoft word and it is free to use. Overleaf helps accelerate the writing process as it formats the document.

### *Screencast-o-matic*

Screencast-o-matic is a free online screen recorder which allows for voice overlay and the option to either download your recording as a file or upload the recording directly to YouTube. Screencast-o-matic is the simplest form of recording an overview of a project.

## Conclusion

To conclude, the developer was not content with the outcome of the project. While the project did reach most of its original goals, a lot of what the developer had planned did not come together. 1 of the 2 main objectives of the project were to create a server-client application which unfortunately proved very difficult to work successfully although the developer felt they were very close to achieving the goal. The developer is however content with the database functionality of the project as data can be accessed and updated in accord with the outcome of each game. The developer believes that the data is safely secured as the password for each user is hashed using a salt which is an aspect that the developer had little prior knowledge about. The developer also believes that working in a team would prove much more beneficial in terms of learning how to communicate and use methodologies more efficiently. The developer believes that they succeeded in reaching the following goals:

1. Gain a better understanding of gaming development.
2. Create a user-friendly interface.
3. Authentication that's simple to use but secure.
4. Develop skills working with agile methodology.
5. Have all statistics for each player recorded across games.

With the above in mind, it can be concluded that most of the initial objectives for this project were met in some capacity or another.

## Learning Outcomes

The developer senses that work done in the project will not only help them in future gaming industry projects but also with working with databases in the future. The developer had the opportunity to improve their skills in multiple domains of development, such as database management, server configuration, client-side development, database security and system architecture. This experience also exposed the gaps in the developer's knowledge such as server-client integration. Going forward, the developer intends to pursue a more in-depth knowledge about this topic. All aspects of this final year project have challenged the developer in terms of research, development, and self-discipline.

## Appendix

More information about the development of the project can be found in the developer diary.

Source code link to GitHub - <https://github.com/CormacRaftery/FinalYearProject>

Link to video demonstrating the project - [https://www.youtube.com/watch?v=3\\_VWhPnR4ck](https://www.youtube.com/watch?v=3_VWhPnR4ck)

## Bibliography

<https://screencast-o-matic.com/>

<https://www.mamp.info/en/windows/>

<https://www.overleaf.com/>

<https://visualstudio.microsoft.com/>

<https://github.com/>

<https://unity.com/solutions/game>

<https://unity.com/>

<https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

<https://www.php.net/manual/en/intro-what-is.php>

<https://www.phpmyadmin.net/>