Title

Project Summary

This project is an advanced version of the classic game Space Invaders. It will retain some of the core mechanics and add some features that will provide a challenge for the player.

There are two parts to this project, the game and the supporting website. The website is used to create accounts for players as well as having a leader board which displays data about the player (for example, their high score, time played and highest level achieved).

This project uses the same perspective as the original game (top down, two dimensional). The difference between this version and the original is that it will become harder to beat as the players score increases. The objective of the game is to survive as long as possible and achieving the highest score possible.

This project will use modern tools and techniques as this was a limiting factor in the original implementation. These factors are discussed in more detail in the Background and Research portion of this report.

Acknowledgments

Declaration

Contents

1 Introduction

1.1 General Introduction

This chapter will give a brief description of the chapters that follow. Chapter two discusses that background and research that went into this project. Chapter three will show case some of the design that went into this project. Chapter four will address some of the implementation and issues encountered. Chapter five will showcase the testing and evaluation used in the project. The final chapter discusses the conclusions and possible further development of this project.

1.2 Motivating Factors

The main motivation for undertaking this project was to improve the Space Invaders game. From the research completed (this is discussed in Chapter 3), many clones have a set difficulty. This results in an imbalance in the difficulty. A highly skilled player will be able to get a higher score quite easily. The result of this project has created a game that uses information collected by the player to make it more difficult for them to beat.

A secondary motivation to complete this project was to become familiar with using a modern game engine. It would be a useful skill to add to my repertoire, especially if I want to apply for a job in a game company after college.

Another motivation for doing this project is to increase my knowledge in web based technologies. As most companies have a web presence it would be useful to know about web technologies, especially those based around JavaScript.

The final motivation for completing this project is to combine the knowledge I have learned from various modules. This project incorporates aspects of Programming, Object Orientated Programming, Database Systems, Distributed Systems and Systems Analysis and Design. Principles learned from these modules will be applied in the project

1.3 Objectives of Proposed Work

This project has two main objectives with multiple elements to be completed. The two primary objectives are the game and the website.

The following list details the objectives to be completed for the game.

1. Programming the initial game using simple graphics.
2. Refining the game mechanics.
3. Program a login screen for the game to allow a player to login so their details can be recorded.

The next list details the objectives to be completed for the website.

1. Programming the website.
2. Programming the server.
3. Responding to any GET, POST, PUT and DELETE commands to make the website RESTful.
4. Creating the database and populate it with test data.
5. Use a testing framework to test if the correct data is being transferred to the various parts of the project.

2 Background and Research

The background and research section discusses the previous implementations of Space Invaders and the considerations that the original designer had to consider. It will also analyse the technologies that will power this project.

2.1 Analysis of Existing Products

The original Space Invaders was very primitive in its implementation. It was so primitive that it was unable to render many colours and had to rely on a colour overlay. (arcade-museum.com, 2016). The specification for some of the system is given below

- Processor: Intel 8080
- Raster graphics on a CRT monitor
- Texas Instruments SN76477

An interesting point to note is that the graphics were drawn using the processor. This was due to the lack of dedicated graphics processing units at the time. It also explains why the graphics were primitive by today's standards (for example, pixel based with little colouring).

In 1980 a version of Space Invaders was ported to the Atari 2600 (gamespy.com, 2016). This had a number of advantages over the original. The biggest advantage was that the colour palette was increased. Other hardware improvements included the addition of RAM and cartridges that could have memory included. (problemkaputt.de/, 2016)

At the time of writing this report, there are many clones that exist. These clones are available on many platforms from the original game to iOS (kotaku.com, 2016). These clones can differ in many ways from the original. Some will have different aesthetics while others will have different mechanics. The table below is a compiled list of some of the data available about space invaders clones (bandainamcoent.eu, gamespy.com, ign.com, kotaku.com, theisozone.com, 2017).

Name	Platform(s)	Released	Features
Space Invaders (Original)	Arcade, Atari 2600, Nintendo Entertainment System, iOS	1978, 1980, 1985, 2009	Raster graphics
Space Invaders Part II, Space Invaders Deluxe (USA)	Arcade, Game Boy	1979, 1990	Gameboy version allowed multiplayer, USA version had updated graphics
Space Invaders '95	Arcade	1995	Updated graphics
Space Invaders 1999, Space Invaders X (Japan)	Playstation, PC, Nintendo 64, Game Boy Color	1999	2D and 3D graphics, competitive mode, cooperative mode,
Space Invaders Evolution, Space Invaders: Galaxy Beat (Japan)	Playstation Portable	2005	Updated graphics, sounds and gameplay, multiplayer mode, rhythm action gameplay
Space Invaders Extreme	Nintendo DS, Playstation Portable, Xbox 360	2008, 2009	Integrated musical elements
Space Invaders Infinity Gene	iOS, Playstation 3, Xbox 360, Android	2009, 2010, 2013	Updated graphics

2.2 Unity Game Engine

The Unity engine is a multiplatform game engine. It allows developers to target a range of devices for example PC, Android, iOS and Xbox, to name a few. It uses one click deployment to build solutions for as many devices that are required. (https://unity3d.com/, 2016).

The engine supports a multitude of file formats for images, audio, video and text. The engine abstracts the more difficult aspects of creating a game such as optimisation and graphics rendering. For graphics, it targets APIs depending on which system is selected for the build. Windows can use both Direct3D and OpenGL, Android and iOS use OpenGL ES and consoles use proprietary APIs.

Programming in Unity can be achieved by using a number of languages. The most widely used language for Unity is C sharp. JavaScript can also be used; however, it is not the same JavaScript that is used for web browsers. It is commonly referred to as UnityScript. (answers.unity3d.com/, 2016). A third language called Boo is also available, however it is now deprecated. It is not documented by Unity anymore, although it will still compile. (forum.unity3d.com, 2016).

Due to its better support and documentation, C sharp will be the language used to implement the game portion of this project.

2.3 MongoDB

MongoDB is a high performance, scalable, document orientated NoSQL database solution. (stackoverflow.com, 2017). MongoDB uses a JSON-like document with schemas. JSON is a lightweight data format that is easy for humans to read and write as well as to parse for machines. Figure 3.1 shows an example of some data in the JSON format.

JSON is built upon two structures:

- A collection of key/value pairs that correspoind to an abject, record, dictionary, hash table, keyed list or an associative array, depending on which language is being used.
- A list of ordered values that correspond to an array, vector or similar data structure.

(json.org, 2017)

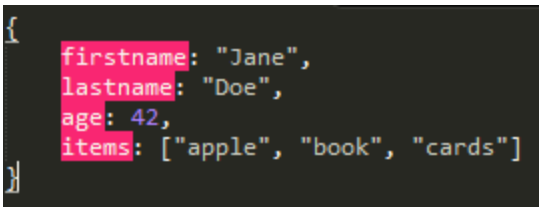

Figure 3.1: Data in JSON format

Because MongoDB is a NoSQL solution the schema is not enforced as strictly as it would in a relational database. The dynamic schema means that each document can have different key/value pairs and can still be inserted into the database. The table shown below summarises the differences between a Relational Database Management System (for example, mySQL) and MongoDB.

Relational Database Management System	MongoDB
Table	Collection
Tuple or Row	Document
Column	Field
Primary Key	Key id provided by MongoDB

The _id provided by MongoDB is a 12 byte hexadecimal number which ensures the uniqueness of every document. This hexadecimal number is construced as follows:

- The first 4 bytes are a timestamp since the ObjectId's creation. This uses the seconds elapsed since the Unix epoch.
- The next three bytes are a machine identifier.
- The following 2 bytres are a process ID
- The final 3 bytes are a counter that starts at a random value

If a document does not specify a unique _id field, MongoDB will generate an ObjectId for this field. This is because the _id is used as a primary key (docs.mongodb.com, 2016). Figure 3.2 shows the same data as Figure 3.1 but now includes an the _id field.

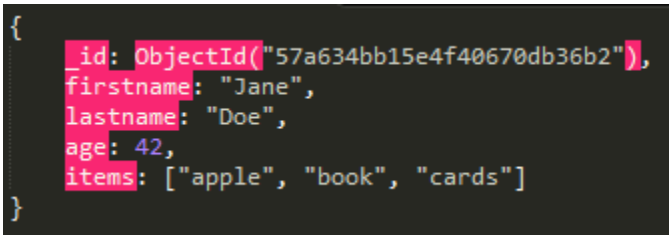

Figure 3.2: Data including the 12 byte hexadecimal _id field. In this example the _id was automatically generated from MongoDB.

2.4 ExpressJS

ExpressJS is a flexible JavaScript web application framework which uses NodeJS to create features for web and mobile applications (expressjs.com, 2017).

It is minimal and provides tools to build applications, with many more plugins available via the Node Package Manager (npm). ExpressJS is similar to Rails for Ruby and Django for Python. Unlike Ruby on Rails or Django there is no documented best way to use the ExpressJS framework. Because it is written in JavaScript and uses NodeJS, express is able to interact with MongoDB in order to retrieve, insert, update and remove documents from a database.

2.5 AngularJS

AngularJS is a client-side JavaScript framework that is developed by Google. HTML is useful for declaring static views but fails when trying to declare a dynamic view. Other frameworks deal with this issue by abstracting technologies to manipulate the Document Object Model (DOM). These solutions dont address the root problem which is HTML (angularjs.org/, 2017). AngularJS addresses this issue by extending HTML which is achieved by using directives and binds data to HTML using expressions.

AngularJS is used to develop the frontend elements of a system. Angular uses the Model View Controller (MVC) pattern in order to separate the representations of the data from how it is displayed to the user. Because it is JavaScript based, AngularJS is able to communicate with the other chosen technologies easily.

2.6 NodeJS

The final element to the backend of the project is a NodeJS server. It is not a JavaScript framework but many of the modules it uses are written in JavaScript. It is built upon Google Cromes V8 JavaScript engine. NodeJS uses an event driven, non-blocking I/O model the makes it lightweight and efficient (nodejs.org/en/, 2017).

A non-blocking I/O (also known as an asynchronous I/O) means that an I/O request will be queued straight away and the function returns. The I/O is processed at a later point. A blocking I/O thread cannot process anything else until the I/O is complete. In the case of sockets, this could result in a long waiting time. NodeJS uses only uses one thread to service all requests. (stackoverflow.com/, 2017)

The advantage of using NodeJS is that it has its own package manager, similar to Ubuntu, called Node Package Manager (npm for short). This allows developers to

share libraries of code as well as managing versions of the code. This is how the ExpressJS framework will be integrated into the project.

3 Design

The design portion of this report will deal with the various thought processes behind the design of the many parts in this project.

3.1 Game Design

3.2 Database Design

3.3 Website Design

4 Implementation

5 Testing and Evaluation

6 Conclusions and Further Development

7 Appendicies

8 References

Angularjs.org (2017) *ngController* [online], available: https://docs.angularjs.org/api/ng/directive/ngController [accessed 1 January 2017]

Atari 2600 Specifications (2016) *Technical Data* [online], available: http://problemkaputt.de/2k6specs.htm [accessed 29 December 2016]

Bandai Namco *Space Invaders Evolution* [online], available: https://www.bandainamcoent.eu/product/space-invaders-evolution/psp [accessed 4 January 2017]

Express (2017) *Web Application* [online], available: http://expressjs.com/ [accessed 1 January 2017]

Gamespy (2016) *The Gamespy Hall of Fame* [online], available: https://web.archive.org/web/20080408152913/http://archive.gamespy.com/legacy/halloffame/spaceinvaders.shtm [accessed 30 December 2016]

IGN (2008) *Space Invaders Extreme* [online], available: http://ie.ign.com/games/space-invaders-extreme/nds-965078 [accessed 4 January 2017]

JSON (2017) *Introducing JSON ECMA-404 The JSON Data Interchange Standard* [online], available: http://www.json.org/ [accessed 1 January 2017]

Kotaku (2016) *Space Invaders Infinity Gene Micro-Review: Evolve or Die* [online], available: http://kotaku.com/5328750/space-invaders-infinity-gene-micro-review-evolve-or-die [accessed 30 December 2016]

MongoDB (2017) *BSON Types* [online], available: https://docs.mongodb.com/manual/reference/bson-types/objectid [accessed 1 January 2017]

NodeJS (2017) Home [online], available: https://nodejs.org/en/ [accessed 2 January 2017]

Stack Overflow (2017) *MongoDB* [online], available: http://stackoverflow.com/questions/tagged/mongodb [accessed 1 January 2017]

Stack Overflow (2013) *What is non-blocking or asynchronous I/O in Node.js?* [online], available: http://stackoverflow.com/questions/10570246/what-is-non-blocking-or-asynchronous-i-o-in-node-js [accessed 2 January 2017]

The International Arcade Museum (2016) *Space Invaders* [online], available: http://www.arcade-museum.com/game_detail.php?game_id=9662 [accessed 29 December 2016]

The ISO Zone (2015) *Space Invaders* [online], available: http://www.theisozone.com/downloads/pc/windows-games/space-invaders-12/ [accessed 4 January 2017]

Unity (2016) *Build once, deploy anywhere* [online], available: https://unity3d.com/unity/multiplatform [accessed 30 December 2016]

Unity (2009) *How should I decide if I should use C, JavaScript (UnityScript) or Boo for my project?* [online], available: http://answers.unity3d.com/questions/7528/how-should-i-decide-if-i-should-use-c-javascript-u.html [accessed 30 December 2016]

Unity (2015) *Will Boo ever come back?* [online], available: https://forum.unity3d.com/threads/will-boo-ever-come-back.375988/ [accessed 30 December 2016]

9 Bibliography