

Introduction Document For:

System Builder

“Build for your future”

Written By Brendan McGuinness (B00252443), Daniel Taylor (B00240424) & Martin Wojtczak (B00253233)

Version #2.6

Tuesday, January 05th, 2016

**Table of Contents**

1. Introduction 4
2. Roles 4
   1. Team Roles Overview 4
   2. Roles Breakdown 4
      1. Brendan McGuinness 4
      2. Daniel Taylor 4
      3. Martin Wojtczak 5
      4. All Members 5
3. Communication Strategy 5
   1. Meetings 5
   2. Skype 5
   3. Facebook 6
   4. Communication Method Breakdown 6
4. Learning Outcomes 6
   1. Variables 6
      1. Strings 6
      2. Integers 7
      3. Float 7
   2. If, then, else 7
   3. Vectors 8
   4. Functions 8
   5. Player Input 8
   6. Spawning Objects 9
   7. Collisions 9
5. Project Overview 9
   1. Game Idea 9
   2. Story 9
   3. SWOT 10
   4. Target Audience 10
   5. Game Structure 11
   6. Player Assessment 12
   7. Player Goals 12
   8. Concept Art 13
6. Integrating Content 13
   1. Content Segmentation 13
   2. Intrinsic and Extrinsic Game Progression 13
   3. Repetition 13
7. Development Information 14
   1. Game Engine & IDE 14
   2. Programming Language 14
   3. Asset Creation 14
   4. Development Approach 14
      1. Agile Methodology 14
      2. Explanation 15
      3. Comparison with RAD Methodology 15
8. Risk Analysis 16
9. Evaluation 17
   1. Experience Programmer Test 18
   2. No Programming Experience Test 18
   3. Conclusion 18
   4. Improvements 19
      1. Implemented 19
      2. Future Improvements 19
10. Critical Appraisals 19
11. References 19
12. Appendix 20
    1. Concept Art 20
    2. Game Flow 24
    3. B00252443 Critical Appraisal 25
    4. B00240424 Critical Appraisal
    5. B00253233 Critical Appraisal
    6. Gantt chart

# Introduction

A variety of industries have made use of Serious Games over the years for example medical and military institutions. Serious Games can be used as a way to help increase engagement in lessons, take the risk out of dangerous training situations or just offer ways for people to help practice a particular skill that may not be feasible in the real world such as certain surgeries. Choi et al (2010) designed and developed a serious game in order to help teach orthopaedic blood management to orthopaedic surgeons that treat the musculoskeletal system where bleeding is common and can be fatal if not managed properly.

Another industry that has made use of serious games over the years to try and make students more engaged in certain lessons is the educational industry. One specific are that has made use of serious games is the subject area of teaching programming, this is down to traditional methods of teaching this subject can be seen as boring especially in students at a high school level of education. “Furthermore, Programming Subject is found as a difficult and boring subject; thus, learning with games seems to develop students’ motivation on the subject.” (Ibrahim et al 2011, p.209).

This paper discusses the design and development of a serious games targeted primarily at introducing high school students from the ages of 14-18 to programming in C# by having them follow step by step instructions to create a Shoot ‘Em Up genre of game. The game could also be used by higher educational institutions such as Colleges or University’s that provide introduction to programming/ games programming modules using other methods such as game maker or traditional programming approaches. Finally the resulting game will be evaluated as well as each individual member’s contribution and performance to the project.

# Roles

## Team Roles Overview

The main coding for the project will be carried out by Daniel Taylor, Brendan McGuinness will be in charge of creating and acquiring the majority of the assets required for the game. Martin Wojtczak will be required to help across both coding and resource creation and acquiring areas for the project. The entire team will also be responsible for the games design, completing any documentation required and carrying out any testing before submitting the project.

## Roles Breakdown

### Brendan McGuinness

* Asset Manager
* Programming

### Daniel Taylor

* Project Lead
* Lead Programmer

### Martin Wojtczak

* + Programming
  + Asset Creation

### All Members

* Design
* Documentation
* Testing/ QA

**Green Tasks**: responsible for the final output of the team for the final project, assinging tasks for team members, ensure all work is properly formatted, enusre team members are keeping up with assigned tasks

**Orange Tasks**: responsible for meeting deadlines, project progress

**Red Tasks**: responsible for completing both individual and team based work, design, research, programming, assets

# Communication Strategy

## Meetings

The team plans to have regular face to face meetings once a week either in the lab before or after the lecture takes place. This will give the team on opportunity to see what stage each person’s work is at as well as allow discussion on any issues that have arisen during the project, since there is only 10 weeks to complete the project the team feels getting as much face to face time would be best for the overall outcome as they can easily track and discuss the work.

## Skype

The free version of the online chat software Skype will be the teams primary method of communication outside of the university, this is due to being able to host group chats, send files and messages instantly as well as the ability to share screens between callers to enable us to see each other’s work in real time.

## Facebook

A private team Facebook page has also been set up for the team to use, doing this allows team members to post questions in which the other members can answer and then these answers can be accessed at any time unless removed. This communication method is especially helpful due to the fact it does not require all team members to be online at once but members can still receive answers/feedback as soon as possible, another feature of the Facebook group is that the project documentation can be uploaded and accessed by each of the team members at any time.

## Communication Method Breakdown

|  |  |  |
| --- | --- | --- |
| Communication Method | Pro’s | Con’s |
| Meetings | * Can keep track of each members progress * Face to face communication * Easy to help one another with work * Can get instant feedback | * Members not showing up * Members running late * May not be able to find quite work area * Members could spend more time chatting than doing work * Members schedules could change last minute with work or family matters |
| Skype | * Voice chat * Free * Can share desktop through video * Ability to send files instantly | * All members must have a microphone to make use of voice chat * Screen share with free version does not support more than two users * Files cannot be re-downloaded * Requires internet connection |
| Facebook | * Private group feature * Can store files online * Posts are saved and can be referenced at any time * Has instant messaging available | * Text only communications could make explaining ideas more difficult * Might not receive instant reply * Requires internet connection |

Table 1: Pros and cons of team communication strategy

# Learning Outcomes

Listed below are the programming concepts we will be teaching the users and how we will be teaching the users these concepts.

## Variables

Variables are usually one of the first if not the first programming concept thought to new students, this is because any and all programs written will contain variables at it.

### Strings

Strings are text based variables used in programming at can be used to store information such as user name, address or a message. In order to teach the user about strings we will provide them with the information to declare a string and how to write a string, they will then be required to define a string for their name.

string name = "User Name";

To add additional value to learning about strings we will be using strings again later to benefit the design of the game. In order to achieve this design idea we will also teach the user about string.length() and what it is used for.

string name = "User Name";

name.length();

### Integers

Integers are numerical based variables which hold negative or positive whole numbers. Which can be used to hold information such as a user’s age, the speed of a spaceship or the amount of coffees ordered. To teach them about integers they will be required to define and age integer and define it as their age.

int age = 24;

To go along with the game design idea with the string.legnth() property we will also be teaching them some math functionality using some basic integers.

3 + 4;

int myMumsAge = 52;

int myAge = 24;

myMumsAge – myAge;

### Float

In order to prepare the user for later in the program for moving their ship they will be introduced to floats at the beginning of the program to keep their learning simple and straight forward. Floats are similar to integers where they store numbers however floats are used for numbers with a fractional value.

float myMoney = 10.54;

We will also again provide and exercise with maths and floats to show how to update a current float value with a new value for later in the game.

## If, then, else

If, then, else statements are used in programming to allow the program to make a decision based on information. For example if someone browsing online tries to access mature content a website will ask the user to enter their age and then if they are not old enough they will then be redirected away from the content. These statements are also valuable to learn for creating games as they can be used in numerous ways such as checking if a button is being pressed to do something in game or if there is a collision between a character and a weapon.

int damage = name.lenght() \* age;

if(damage > 15){

damage = 15;

}

else if(damage < 15){

damage = 15;

}

## Vectors

In order to move objects in certain video game engines like Unity the user will need to create a vector and use this to update the position of the objects. In our game we will be teaching the user about vectors specifically 2D vectors due to the game, in order for their ship and projectiles in the game will be able to move.

float movementSpeed = 5;

Vector2 playerPos = this.transform.position;

playerPos.x += movementSpeed \* Time.deltaTime;

this.transform.position = playerPos;

## Functions

A section covering functions and why the user should use them as well as where to use them will also be covered. This will help show the user that by breaking up pieces code compared to putting lines upon lines of code into one function can help them keep their code simple to read and understand.

float movementSpeed = 5;

void moveShipUp(){

Vector2 playerPos = this.transform.position;

playerPos.y -= movementSpeed \* Time.deltaTime;

this.transform.position = playerPos;

}

## Player Input

Users will also be taught how to handle input from a keyboard in order to move their ship and fire their weapons. The input will also make use of the previous sections taught containing If statements, vectors and functions.

if(Input.GetKey(“up”)){

moveShipUp();

}

## Spawning Objects

Another important part of video games are objects and certain programs are objects, objects typically have two characteristics “state” and “behaviour”, for example a TV objects state could be the TV’s colour and a behaviour of the TV would be to display a show. However to keep it simple we will be using a more simplified example and explanation where objects will be explained as things such as the players ship and the players bullets. After this we will then get the player to create an instance of a bullet object using the code below.

Instantiate(obj\_bullet, new Vector 2(obj\_ship.x, obj\_ship.y), transform.rotation);

## Collisions

Collision detection is used in most video games, collision detection can check if two objects or touching or have intersected with each other. We think this is important to teach and users would definitely use this knowledge to make their own game, it may seem a hard concept for beginners to learn but the team feels with the right lesson it should be easy for anyone to learn.

Void OnTriggerEnter2D(Collider2D objectHit){

if(objectHit.name == “obj\_bullet(Clone)”){

obj\_enemyHealth -= 10;

}

}

# Project Overview

## Game Idea

The idea behind System Builder came about through the experience that our team members have gained from learned C# in college. We began discussing what aspects and teaching methods led us to practise our coding and develop a better interest into learning it. We have learned that using online guides such as Codecadamy who won: People’s choice the GREAT Tech Awards 2013 (UK Trade & Investments, 2014) and Ruby Warrior were popular to individuals’ interest in programming. So, we took this area of learning into consideration and began the process of creating the design for System Builder. We decided on the idea of developing a tutorial that will lead the user through a process of a step by step guide that will teach them basic learning outcomes in an introduction for the language of c#. The game within this tutorial is that the step by step guide will teach the user on how to build their own spaceship which they can use at the end of the tutorial in the game that they have just built! Each section the tutorial will introduce new aspects related to beginning of coding. Learning topics such as Variables, Strings, Integers and popular statements.

## Story

System Builder is a programming serious game that introduces students in the learning areas of the language C#. The students will follow a tutorial that will give a step by step guide to rebuild a spaceship. The instructions will be displayed onscreen for the users and this will be provided by the spaceship itself, telling the user how to fix it up. This will be through the use of learning outcomes relevant to learning basic programming skills. This allows the progression of the players learning through a fun and exciting experience due to not allowing them to move on until it is entered correctly for each certain section. After each section has been successfully completed, the user will then move on to the next section and be instructed of new tasks needed to be carried out in relation to the development of the spaceships rebuild. At end of tutorial when the guide has been followed correctly, the user will then be able to control and use the spaceship that they have rebuilt through the use of the tutorial in a mini game. This will give the user a relevant idea of the work they have learned and produced that’s been put into practice for the creation of a game.

## SWOT

After the team had agreed on the game idea and backstory for the game the team then sat down and completed a SWOT analysis that would help highlight both the advantages and disadvantages of the game and the games development.

|  |  |
| --- | --- |
| **Strong** | **Weak** |
| * Compatible team with similar ideas. * Full time likes the game idea and design. * Basic shoot em game is simple to make. * Team members have experience with C#. | * No previous experience creating games to either teach or specifically teach programming. * Only one team member has experience with Unity, that experience is also very brief. * No team member has experience with assessment engine. * Only one team member was able to complete assessment engine tutorial. |
| **Opportunities** | **Threats** |
| * Team members will gain new experience and abilities from creating the game. * Game could be updated to include more areas of programming which could target new audiences. * Chance to promote team members. | * If game gathers a lot of success it could be easily cloned. * Extremely short development time available. * One team member will have to do majority of the programming. |

Table 3: Game Idea SWOT

## Target Audience

The audience the game will primarily target will be high schools/ high school students from the ages of 14-18 who take a computing class that wants to teach an introduction to programming. Although with the flexibility of our game and higher educational institutions doing introduction to programming/games programming in their course, this game could also be targeted to these types of institutions. The final type of audience the game could be targeted towards would be educational clubs such as Code Club which teach children how to program by making video games, so this type of application would easily fit in with their aims and educational program.

## Game Structure

See appendix two for a flowchart on the game flow.

**Step1 – main menu**

When the game is started the player will be on the Main Menu. In the main menu the player will see the Start button, Controls button and High scores button. The main menu will also have a small description of what the game purpose and goals are. This menu will have a simple space like background and have a plain white colour scheme.

**Step2 – user info**

List of questions

When the player clicks the start button they will be taken to the User Information page. This is a page where the player will enter some details about their self and their coding background so we can properly evaluate them after playing our game.

* Enter your name?
* What age are you?
* Gender? B/G
* Have you ever programmed before? Y/N

These simple questions lets the game put the player in a category to see where they compare to others of the same qualities.

**Step3 – Game level**

This is a list of all the levels we intend to add to our game. The player will have to fully complete each level to progress to the next level. If the player has made a mistake they will be made to correct the problems to enable them to progress.

* Level 1 - Variables
* Level 2 - Strings
* Level 3 - Integers
* Level 4 - Float
* Level 5 - If, then, else
* Level 6 - Vectors
* Level 7 - Functions
* Level 8 - Player Input
* Level 9 -Spawning Objects
* Level 10 – Collisions

Once the player has completed all 10 levels they will then have a fully functional Shoot em up to play. The player will then have to complete the game and beat the final boss to finish the game. The player will then be taken to then End Game screen where they will be shown the statistics of their time playing the game.

**Step4 - End of game**

At the end of the game the player will be taken to an end of game screen. This will show the player all of the statistics of their progress. These stats will be compared to others of the same qualities. The game will inform the player what they have done right and wrong. The game will then suggest what parts of the game the player might want to work on to improve. This kind of feedback is very useful for novice programmers. They will learn from their mistakes instead of struggling to understand the technical references seen in normal compilers.

## User Assessment

We plan to assess our users’ results by using two methods. The first method is during gameplay there will be checks to ensure the users answers or code for each question is correct before allowing them to move on to the next question. If the users answer or code is not correct they will be informed by a pop up message that something in their code is incorrect and they need to fix it before moving on, if their answer or code is correct however a pop up message will appear saying that it is correct and then they can move onto the next question.

The second form of assessment we will be using is to keep track of the amount of mistakes made by the user with each question. This information from each user could be analysed by the user or a teacher to see the areas of the game where they can improve on, this form of assessing is like a high score table in reverse. Where instead of getting the highest number the users will want to get the lost number to be the best. This could help with motivating younger players to want to go back and practise in order to beat their friend’s scores which would help improve their knowledge and performance in the subject.

## Player Goals

The goals of the game are simple. The player will follow instructions that each level sets. The player will then have to follow the instructions correctly and within the rules set to progress through the levels. Each level the player will create a part of a Shoot em up. The player’s goal is to complete each level, this will then lead to a fully functional Shoot em up at the end of the game.

One of the player goals is to repair your space ship systems. In the story line the player’s ships systems have been damaged. Each level in the game will enable the player to code a new function that repairs a part of the damaged ship.

With each level the player will learn to create Variables, Strings, Integers, Float, If, then, else, Vectors, Functions, Player Input, Spawning Objects, Collisions. This will lead to the ships systems being fully operational and a have the player playing a fully functional game.

The main objective of the game that the player is making is to kill the main boss and win the game. After the player has progressed through each level and the player has a fully operational game. They will then play the Shoot em up they have made and eventually meet the main boss. The Boss will be difficult to beat, but the game wouldn’t be fun if it wasn’t a challenge. Once the player has completed the game they will then be taken to the end game screen.

## Concept Art

See appendix one for game concept art.

# Integrating Content

## Content Segmentation

The learning outcomes describe above in their own individual sections will be incorporated into the game in the same flow. The team feels that the order of each learning outcome above will ensure that the user will have an easier time getting to grips with and learning each section as there is a gradual increase in difficulty. Putting the content into sections where the player will have to have an understanding of that area of programming should stop students from progressing too far above their ability.

“The results of the present study support the previous findings that the segmentation of multimedia instruction facilitates basic (recall) and deep (application) knowledge acquisition.” (Doolittle et al 2009, p.648).

## Intrinsic and Extrinsic Game Progression

To deliver the content in the game to the users the teams feels that going with a mixture of intrinsic and extrinsic progression would help keep students engaged more in the game. “*Furthermore, classes taught at school often have recourse to extrinsic motivational factors such as rewards, praise, and punishment, while computer games seem to resort to intrinsic motivational factors.”* (Raya et al 2010, p.178).

The team feels adopting a mixed method approach would be the best method for progression in the game. By only allowing users to play the game after they have completed a task will help keep the user focused on the learning aspect of the serious game and want to focus during each task which will hopefully translate into them gaining more knowledge and skill. This takes and extrinsic approach to our game progression as the user is rewarded for completing their task.

However also using an intrinsic motivation method were the game that the users play is enjoyable and fun could help benefit students that may not think or do not enjoy programming. By creating something they enjoy could make them go back and look at programming again and in fact enjoy learning the subject area.

## Repetition

As mentioned in the user assessment section of the design document the team plans to keep track of the amount of mistakes that the user makes for each section of the game. In doing this the team hopes it will give students and teachers a clear and precise indication of the areas that the student requires additional work to help increase their knowledge and abilities. This could be incorporated into a scoreboard type situation in the class room where students try and complete the game with no mistakes, which similar to scoring systems in games could increase engagement and enjoyment.

# Development Information

## Gantt chart

See appendix six for a Gantt chart showing the timeframe for the development of the game.

## Game Engine & IDE

The game will developed using the Unity game engine and the Unity integrated development environment, the team has decided to use Unity as it is an extremely powerful engine for both 2D and 3D applications. It can also be used to export application created with it to various different platforms which means the game can be accessed on a verity of hardware.

## Programming language

To get the most use of the Unity game engine and its features the team will be using the C# programming language, by comparing its benefits compared to Unity’s version of JavaScript. There is a lot of talk that applications written in C# has better performance and that C# allows users to make use of more feature, so due to this and the team preference of C# that will be the language used.

## Asset Creation

To create and edited the assets used within the game the team will also be using various other software packages. Adobe Flash and Photoshop will be used to create images for the game and make any adjustments or changes to assets obtained from another source to help speed up development. Similarly to make any adjustments to audio files or create any audio the team will be using Audacity.

## Development Approach

### Agile Methodology

Agile software development (ASD) is a methodology in creative software development that is flexible, promotes testing at each mile stone and rapidly builds each piece of a software project. The ASD methodologies rapid development has the flexibility to evolve to the requirements of a small scale, short and unpredictable project.

The ASD starts with the Project requirements being defined and design process being completed. Once the design process is complete, the project will move onto the second stage. The second stage is the development stage. In the development stage the team will work on a piece of software until they reach a mile stone, once complete, the team will test what they have created and add functionality to it. The team will then move on to the next milestone in the project and will cycle through development process until the project is complete. Once the project is complete the team will then release the software to be publicly tested. If the feedback and reviews are satisfactory and the team are happy with the product, the software will then be released to the market.

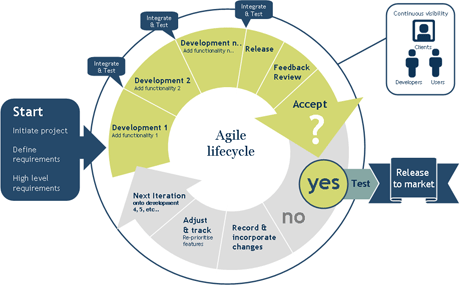


Fig 1: (2010) Agile Software Development Cycle [Online] <http://www.optimusinfo.com/blog/agile-software-testing-and-automation/> [Accessed: 27/09/2015]

### Explanation

When discussing the project at hand, the team came to agree that the agile software development method was the correct method for our project. The ASD method suits the way our team personally work and the team thinks it would complement the small time scale the team will have to complete the project.

The ASD will let our team plan the project at the very start. This design process will include having development milestones and development timescales. This gives us game development students a chance to plan our project out and then execute the plan, with all the planning and design done at the start of the project. When complete, it will give us a clear view of what the team has to achieve when developing the project.

The development process will let the team set weekly milestones and improve upon the project on a week to week basis. If the team is unhappy with what he had delivered after a milestone the team will be able to review the work and make plans to change the part the team members are unhappy with. This makes the project very flexible and will give us confidence when creating our software.

### Comparison with RAD Methodology

When deciding what methodology the team would use to develop our project the team considered the Rapid Application Development method (RAD). This method is a very good and would suit our development style. But it lacked the freedom the team wanted. The RAD method is similar to the waterfall methodology but puts less emphasis on planning and more on development. The RAD has a quick design process and lets the developer create the whole project over the development stage. This lets the developers plan the project as they go along.

This method would have worked for us if the planning section wasn’t so rushed. The team preferred to have a meticulous plan before starting to develop our software. This leaves the team without any misinterpretation of what they have to do and helps less experienced developers to not get lost in the mass of the project.

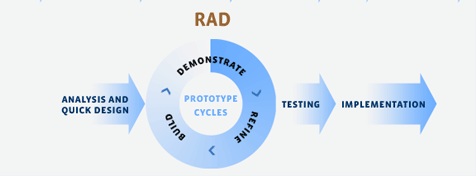


Fig 2: Rapid Application Development Cycle [Online] <http://www.indosakura.com/en/index.php/sample-page/it-services/development-model/rapid-application-development/?lang=en> [Accessed: 27/09/2015]

Table 4 highlights the pros and cons between the agile development cycle and the rapid application development cycle.

|  |  |  |
| --- | --- | --- |
| Methodology | Pros | Cons |
| Agile Method | * Flexible * Meticulous planning stage * Thorough Development and testing process * Works well in small scale * Suitable to less experienced Developers * Works well in short timescale | * Chance of plan not going to schedule and being late |
| RAD Method | * Flexible * Thorough Development and Testing process * Works well in short timescale | * Doesn’t suit less experienced developers * Has a small planning section |

*Table 4: Pros and cons of the compared development methods*

# Risk Analysis

Table 2 describes the risks that the team has analysed and feel could pose a threat to the games development and the goals the game is aimed to accomplish, along with this the team has compiled a key to show the probability of the risk happening and the impact it would have on the final product. Once the probability and impact were both calculated the team came up with solutions on how they could stop this from happening during the games development.

*Key :( very low=1, low=2, medium=3, high=4, very high=5)*

*Estimated Probability (1-5)*

*Estimated impact (1-5)*

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Score | Recognition | Resolution |
| Allowing the user to advance to far beyond their ability level. | *Estimated Probability (2)*  *Estimated impact (5)* | Allowing uses to move to harder challenges before that have a good understanding of previous easier tasks could see them stop playing the game. | The team will have to ensure that the in game assessment does not produce any errors that could allow users to move on before finished the easier tasks properly. |
| Task instructions are not clear. | *Estimated Probability (2)*  *Estimated impact (5)* | If the instructions for the tasks are not clear that no matter the person experience with programming can understand them or them will not be able to or even want to complete the game. | The team will have to ensure that all of the tasks are explained clearly and toughly, to help with this the team plans on asking a friend with no or little programming experience if they understand what to do in each task. |
| Tasks are too similar. | *Estimated Probability (1)*  *Estimated impact (3)* | If the tasks are too similar players could stop playing due to boredom. | The team will have to design the task to ensure they are all varied and serve a purpose to the creation of the game. |
| Not enough gamification. | *Estimated Probability (2)*  *Estimated impact (3)* | If the team does not put enough emphasis on the gaming part of the application players may not play it or stop playing it due to having no enjoyment. | When designing the tasks the team will need to ensure there is a good mix of gameplay and game elements to increase the amount of gamification present. |
| Not teaching a currently used programming language. | *Estimated Probability (1)*  *Estimated impact (4)* | Not teaching a language that is used in our game could have a large impact on the amount of people that play it dues to the fact they may think it would be useless to learn an outdated or hardly used language. | The team is planning to use C# as the programming language taught in the game due to it still being widely in use by many companies and it is the language the game will be created in also. |
| The game they are creating not being fun. | *Estimated Probability (3)*  *Estimated impact (2)* | When it comes to the user testing the game if they don’t find it even a little fun it may stop them from completing the rest of the game. | The team will have to ensure the game that the user is creating is as fun as a basic shoot em up they would play online at the least. |

Table 2: Project risky analysis

# Evaluation

In order for the team to fully evaluate our game the team decided instead of just going through the documentation and game then looking at what we originally wanted to accomplish compared with what we had achieved, we decided to also get two people outside of the team to test and give feedback on the game. We decided to do this so we could have a more in depth and less biased evaluation of the game itself, by having someone who was not part of the development processes provide feedback. For the testing we wanted to target two specific types of people that were on either side of the programming scale in terms of programming ability and knowledge, in order to get genuine and helpful feedback that could be used to improve the game as well as highlight the teams goals and if the team reached them or not.

Each of the participants in the test were given an executable version of the game and received no help from any of the team in order to gain true results from each test. The testers were then asked to provide feedback by one of each method dependant on the situation, the methods included:

* Instant messaging though Facebook Messenger
* Email text file with experience and thoughts
* Feedback though Skype call

A questionnaire was not used to gain the feedback so that the tester’s results could not be manipulated into specific and predetermined answer, a qualitative research method also seemed more beneficial for the team so they testers were encouraged to speak freely and in as much detail as possible.

## Experienced Programmer Test

The first tester we asked to provide the team for feedback was an experience programmer who was working 4 days a week at a software company in Glasgow as a C++ programmer and attended a software development degree at the University of The West of Scotland and was friends with one of the team members.

The feedback we received noted:

* Ability to tab between textboxes for user information.
* Background music in question levels was distracting.
* Some of the questions and information to the questions was a bit vague.
* Question 6 introduces a lot of concepts with no real information.
* Be able to quickly clear all text from input box.
* On the questions that are word heavy try and make them more concise.
* Having obj\_ at the start of each variable is confusing.
* Add in a lexer to have more possible correct user code options.

## No Programming Experience Test

This tester had no programming experience what so ever but wanted to learn how to program in order to make video games, this tester was friends with all of the team members.

The feedback we received noted:

* Add in more questions.
* Give a more in-depth explanation on what is thought.
* Word the questions and instructions better.
* Add in more gameplay for the game being taught.
* Add in questions to teach how to make the enemies move and shoot.

## Team Conclusion

Overall the game meets the goals set out by the team as it includes all of the learning outcomes that were planned to be added, however the team feels that the overall quality of how the question and instructions are written could be improved to up the quality of the games teaching aspect. For the gameplay part of the game that is currently included were the player controls the ship the team feels that this does need to be improved upon as it is very basic at this stage. One area of the game that was not incorporated fully as the team originally wanted was the assessment engine, the team originally wanted to use the engine to check the input text for the users code matched the answer and if it did not it would return a message stating the answer was missing a piece of code. However some of the checks such as if the user was missing any syntax were not possible, the team also wanted user to be able to create accounts and log in to the game in order to save their progress but creating user account was not shown in the tutorial or on the documentation on the assessment engine website. The team did manage to include the badge functionality of the game for each stage being complete which could be shown to teachers through the assessment engine website, the team feels if the assessment engine was to be update to include more functionality we would include it more in the game as it is potentially a very valuable and helpful asset.

## Improvements

In the below sections you can find lists on changes we have or plan to implement into the game, the lists were created from the feedback of the testers we used to evaluate the game and the teams own thoughts.

### Implemented

* Font changed as it was hard to read.
* Lowered background music to make it less prominent.

### Future Improvements

* Add in more questions and sections to give more lessons on how to make the game.
* Improve quality of questions and instructions.
* Allow for more possible correct user code.
* Add in more gaming type content to the gameplay sections of the game.
* User log in.
* Save user progress and data for using later.
* Better fitting audio.
* Improve visuals.

# Critical Appraisals

See appendix three to five for each team member’s individual critical appraisals.

# References

Choi, S.K., Chui, P.Y., Heng, A.P., Pang, M.W. and Qin, J. (2010) Learning Blood Management in Orthopaedic Surgery through Gameplay. Computer Graphics and Applications, Vol.30 (2), pp.45-57.

Doolittle, E.P., Evans, D.A., Jeffrey, R.T., Lusk, L.D., Palmer, R.K. and Wikstrom, S.C. (2009) Multimedia learning and individual differences: Mediating the effects of working memory capacity with segmentation. British Journal of Educational Technology, Vol.40 (4), pp.636-651.

Ibrahim, R., Jaafar, A., Omar, M.H. and Yusoff, C.M.R. (2011) Students Perceptions of Using Games to Learn Introductory Programming. Computer and Information Science, Vol.4 (1), pp.205-216.

Raya, A.M. and Wrzesien, M. (2010) Learning in serious virtual worlds: Evaluation of learning effectiveness and appeal to students in the E-junior project. Computers and Education, Vol.55, pp.178-187.

UK Trade & Investments (2014) The GREAT Tech Awards [Accessed from the UK Government Website at <https://www.gov.uk/guidance/great-tech-awards>, 9th November 2015]

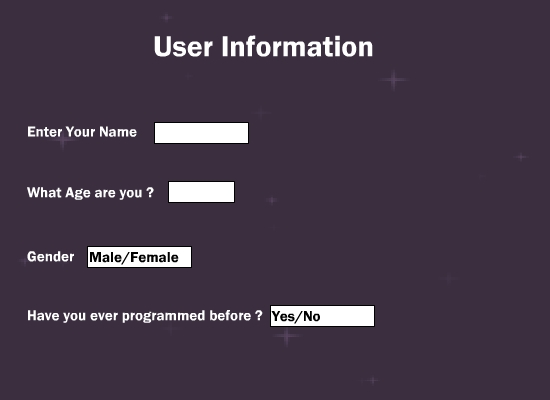
# Appendix

## Concept Art

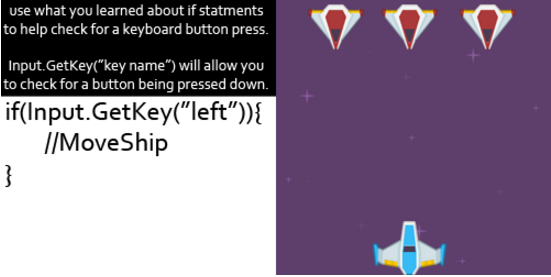
### Main Menu



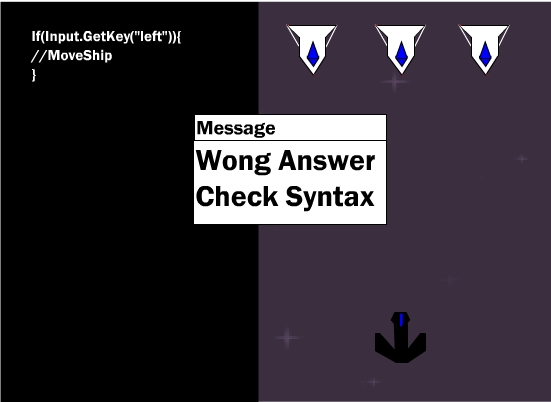
### User Information



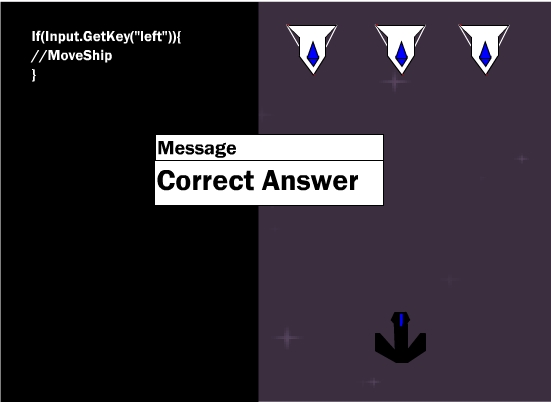
### Gameplay



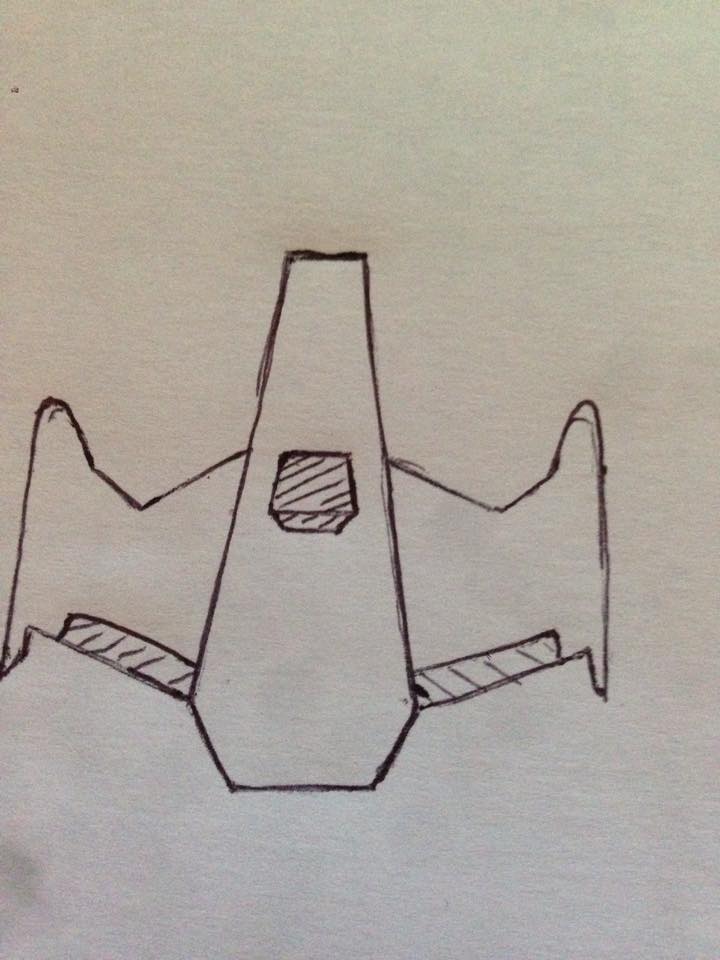
### Wrong Answer



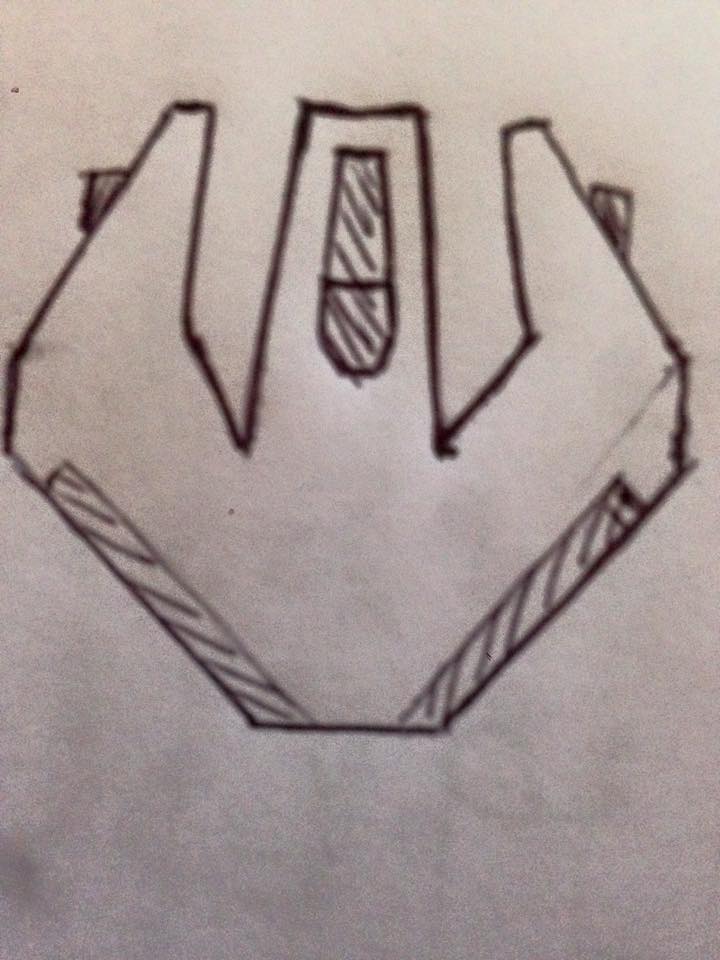
### Correct Answer

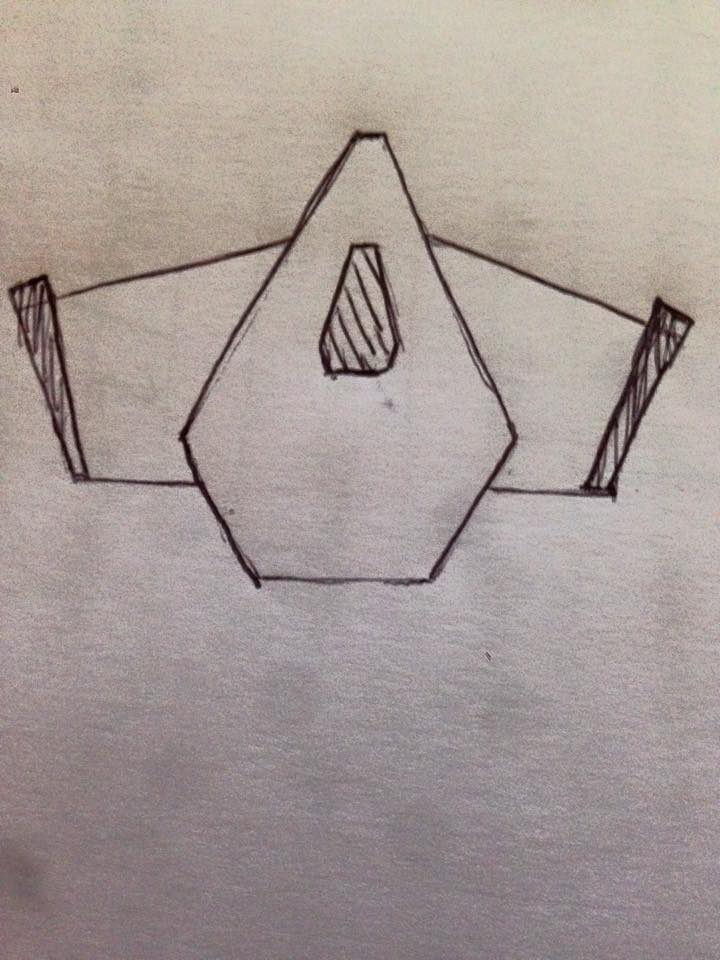


### Player Ship

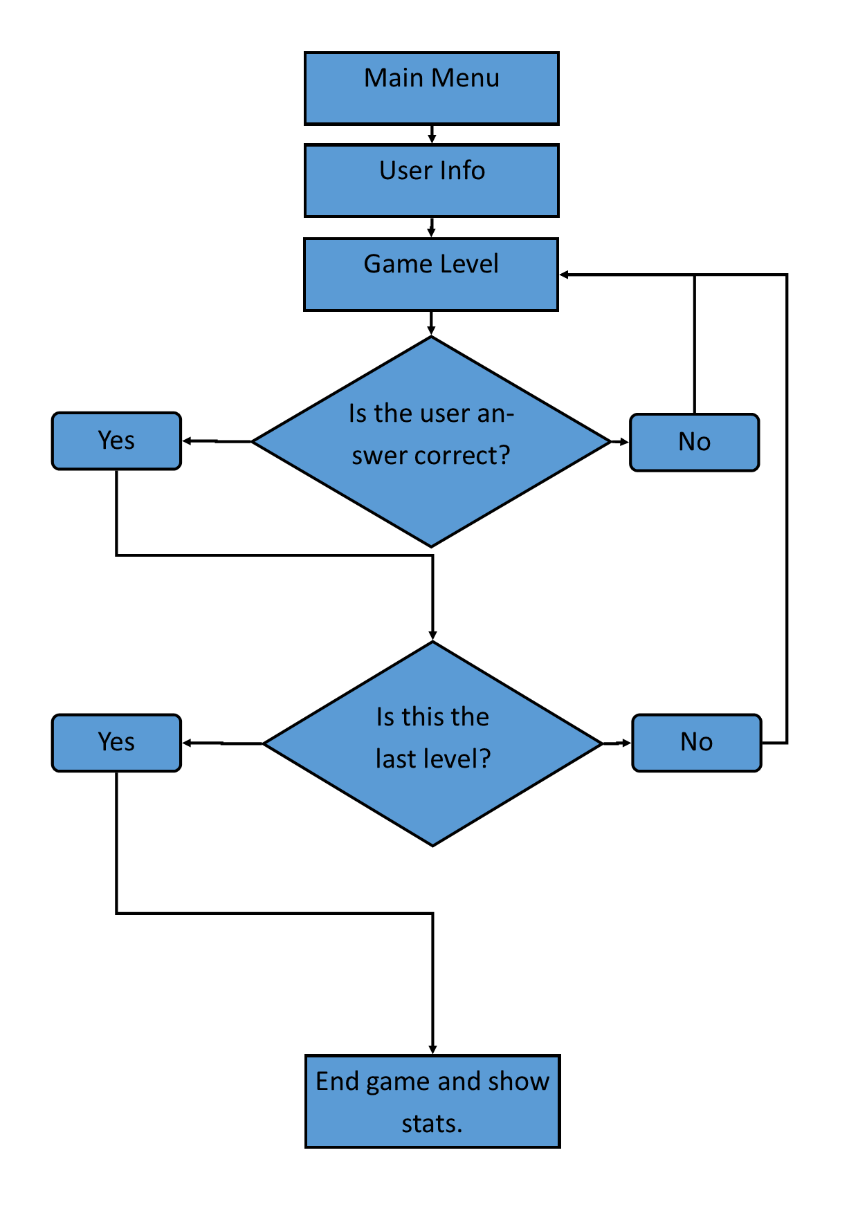


### Enemy Ships





## Game Flow



## B00252443 Critical Appraisal

## B00240424 Critical Appraisal

### Brendan

At the start of the project during the literature review Brendan got his work in at the time agreed by the team, however some of his work did require some editing to fix and improve the quality. After the first task he started to be quite late past the agreed on hand in times agreed by the team as well as complained about the work and about myself for asking him to hurry with getting the music assets for the game. During the project Brendan had the least amount of work to do himself and with most of his I work I helped him in doing it such as finding papers for him to use in his section of the literature review, explaining what to do and giving him tips on what to write for his two headings on the games design document. He also said that he wanted to help out with the games programming and writing out the questions for the game, however when it came to do this I had to be on skype with him to help him to do this work. In the end I done these tasks for him as we were working on these tasks for 4 hours and he could not do them at all. We did know that Brendan was weak with his programming abilities but he could not even manage to write an if statement even when I told him the exact code to write. Overall Brendan’s contributions to the project was extremely lacking compare to the rest of the team, he spend the majority of the project complaining about having to do work and would constantly leave the team waiting on his work in order to continue. By the end of the project myself and Martin worked on the project alone and did not ask Brendan to help as it was a waste of time and I think the project could have been better without him in the team.

### 14.4.2. Martin

During the first few weeks of the project Martin was usually a day or two late when delivering his share of the tasks after the team had agreed upon a deadline. However once the tasks were handed in and checked by the rest of the team his work only required a small piece of editing on a couple of occasions. When it came time to make the game Martin wanted to help with coding and writing out a few questions as he wanted to use Unity for his honours project, the questions he had wrote out were for the most part fine and jut required one or two changes. During the programming part of the project it was agreed Martin would program two of the questions in the game, during the first section he was unsure about what to do and his code required a bit of editing to get it to work but by the time he had finished the second question it only needed fixed with one line of code. Overall during the project Martin was very helpful and willing to help with any of the tasks when needed, he also worked extremely hard and done nearly all of the game testing and document spelling and grammar fixes alone.

### 14.4.3. Myself

#### 14.4.3.1. Game Idea

The game idea that was chosen after we found out the idea was not accepted for the project, was suggested by myself. The other team members ideas was a shoot em up medical game which would have required a lot of research into different medical areas as we were not familiar with them, the idea that was presented for the game also focused more on the game side of the project and was very vague and small on the teaching side. I suggested a game to teach programming after using such sites as codecademy.com and playing a little platformer to teach html tags, I approached the team with the idea to teach programming like on codecademy.com but mix it with a narrative of a space shooter adventure with the piolet (player) recoding the ships systems as well as fighting enemies so it would have a strong narrative but not lack in the teaching aspects. The other members like this idea and we decided to go with it.

#### 14.4.3.2. Pitch

When it come for us to pitch our idea I created the PowerPoint on my own as I knew I could create it faster by myself. Once we started to practice our pitch as a team we decided that I would speak about the design aspects of the game and the idea of it as it was my idea, so the other members would speak about the more generic sections. Even though my other two team members had the more generic sections of the pitch I had to give them notes on what to talk about during it as they could not come up with what to say on their own.

#### 14.4.3.3. Literature Review

For the literature review I laid out the document and added in the headings for the various sections to help the team make a start on it, I also found the papers that both myself and Brendan used writing the previous work section which I wrote slightly more of. I also came up with the idea of doing some research and asking people we knew what they thought of our idea and what they would like within the game, so for this I created a survey on my own and then analysed the results and compiled them in the document. I also wrote the section on the influence to our design, teaching method and the design of the game itself and I contributed the most content.

#### 14.4.3.4. Game Documentation

Similar to the Literature review I done the majority of the work into the games design document with Brendan and Martin covering two sections each, I done the majority of the work alone as I knew certain areas of the document I could do slightly better and more detailed work. I also had to do specific sections of the documentation as the other team members were not able to do it or did not know how to do it either. My other reason for doing most of the work alone was I knew that it would not take much time for me to do it and as my team members were late with their literature review work I did not want to waste time waiting on them.

#### 14.4.3.5. Game Creation

I knew from the start if the class I would be doing most if not all of the programming for the game as I was the strongest programmer in the team. Even though my team members wanted to help program some of the game, I ended up doing nearly all of the programming and the two functions I did not write I helped Martin fix the errors with his code. As this was only my second time creating a game within Unity I did learn a lot from the project, such as getting a better understanding on how to scale the UI, using regular expression for the first time and interacting with web services. As for the assessment engine, again I was the only one in the team that knew how to use it and I was the only one that finished the tutorial so again I was the only one to work on it. When implementing the engine at first I did have some troubles as the documentation was not as clear as it could have been, then I found out the code for my project had issues in it. That the engine creator did not catch but I managed to get it fixed and working within the game. We had planned to use the assessment engine more within the project, but due to the lack of documentation, functionality of the engine and as no one else could program either the engine or the game I made the decision to focus more on the game itself.

#### 14.4.3.6. Overall

Overall I am happy with the final outcome of the project as we managed to deliver on all of our targets. I would have liked to have spent more time working on the game itself but as the documentation was a bigger factor in the marking scheme I had to make that a priority especially as some of my team mates were not delivering work at all or on time. I am happy with both the quantity and quality of all the documentation that the team has delivered, however I would have liked to have included a bit more to the literature review and added some more academic references to it. If we were to do the project again I think we could have delivered a better literature review, due to our lack of experience we did not know the difference between a citation and a reference which wasted time for us. I also spent some time procrastinating which wasted a little time that I could have possibly added in an extra reference or two, as well as my bad spelling and grammar would have wasted a bit of time to fix them.

In terms of the game I am happy with the work I have delivered and I included all of the sections that we had planned too. I would have liked to have had a lot more questions added in, possibly around 5 per section with a more gradual teaching slope. However I did find it hard to design the questions and if I could do it again I would have spent some time researching how to create better questions for the game. I would also have liked to add in more gameplay for the game that was created too. As for being the team leader I feel that I did a good job with pushing the team to get work done early so we could accomplish more, where possible with certain team mates. I often helped with other team member’s tasks that they were unsure of or stuck on and I also gave detailed information and tips when assigning tasks to team members to help them get started and ensure they knew what to do.

## B00253233 Critical Appraisal

## Gantt chart

