Developing Collision Physics in a Simulated Rugby Video Game

A dissertation submitted in partial fulfilment of the requirements for the degree of Bachelor of Science (Honours) in Computing

by Daniel Curtis

Department of Computing & Information Systems
Cardiff School of Management

Cardiff Metropolitan University

April 2017
Declaration

I hereby declare that this dissertation entitled Developing Collision Physics in a Simulated Rugby Video Game is entirely my own work, and it has never been submitted nor is it currently being submitted for any other degree.

Candidate: Daniel Curtis

Signature:

Date:

Supervisor: Dr. Paul Angel

Signature:

Date:
Abstract

This project aims to develop ball physics within a simulation for a rugby ball using the Unity game engine. Whilst there is research already done on physics related to a normal spherical ball, there is much less completed on the egg-shaped ball of rugby.

Firstly, research will be carried out around the subject of ball and collision physics and games simulations. After this is completed, a simulation will be created to replicate the movement of the ball when hit in different locations to show what happens to the ball as if it was in real life.

Participants will be able to test the product created for opinions on the realism of the simulation and improvements that can be made in future editions.
Acknowledgements

I would like to thank several people who helped me with the completion of this project. Firstly, I would like to thank Dr. Paul Angel who helped guide this project and answer any questions I had.

I would also like to thank my friends, especially Becky and Fliss who provided me food, energy drinks and, most importantly, encouragement throughout the process of creating this project.

Finally, I would like to thank my family who may not have understood what I was doing but nonetheless were there to support me.
# Table of Contents

## Contents

**Abstract** .......................................................................................................................... iii

**Acknowledgements** ......................................................................................................... iv

**Table of Contents** ............................................................................................................. v

**List of Figures** ................................................................................................................. vii

**Chapter 1: Introduction** ................................................................................................. 8

  1.1 Context ......................................................................................................................... 8

  1.2 Project Background ...................................................................................................... 8

  1.3 Project Objectives ........................................................................................................ 8

  1.4 Deliverables .................................................................................................................. 9

  1.5 Timeline ....................................................................................................................... 9

**Chapter 2: Literature Review and Research** .................................................................. 10

  2.1 Overview ...................................................................................................................... 10

  2.2 Video Games Industry ............................................................................................... 10

  2.2.1 Sports Games .......................................................................................................... 10

  2.2.2 Rugby Games .......................................................................................................... 11

  2.3 Physics within Games ................................................................................................. 12

  2.3.1 Physics within Games ............................................................................................ 12

  2.3.2 Games Engines ....................................................................................................... 13

  2.3.3 Unity Game Engine ............................................................................................... 13

  2.4 Collision Physics ......................................................................................................... 15

  2.4.1 Collision Physics .................................................................................................... 15

  2.4.2 Projectile Physics ................................................................................................. 17

  2.4.3 Rugby Ball Physics ............................................................................................... 19

**Chapter 3: Methodology** .............................................................................................. 20

  3.1 Overview ...................................................................................................................... 20

  3.2 Research Philosophy ................................................................................................. 20
# Table of Contents

3.3 Data Collection and Research ................................................................. 21
3.4 Research Methods .................................................................................... 22
3.5 Risks and Limitations ............................................................................... 24
3.6 Design Methodology ............................................................................... 25
3.7 Development ......................................................................................... 26

Chapter 4: Results ........................................................................................... 31
4.1 Overview ................................................................................................. 31
4.2 Development Reasoning ......................................................................... 31
4.3 Functional Testing .................................................................................. 33
4.4 End User Testing Results ........................................................................ 34

Chapter 5: Conclusion .................................................................................... 37
5.1 Completion of Objectives ....................................................................... 37
5.2 Future Work ............................................................................................ 38

Chapter 6: References .................................................................................... 40

Appendix A: Ethics Form and Approval Number ......................................... 46
Appendix B: Participant Information Sheet .................................................. 51
Appendix C: Participant Questionnaire ......................................................... 52
Appendix D: Questionnaire Response ........................................................... 53
Appendix E: Code for Ball Script .................................................................. 55
Appendix F: External Materials ..................................................................... 57
List of Figures

Figure 1 .................................................................................................................................pg.9
Figure 2 .................................................................................................................................pg.20
Figure 3 .................................................................................................................................pg.27
Figure 4 .................................................................................................................................pg.28
Figure 5 .................................................................................................................................pg.28
Figure 6 .................................................................................................................................pg.28
Chapter 1: Introduction

1.1 Context
In-game physics are of great importance to a computer game and are fundamental to how it performs, and ultimately help it with its success in sales and reviews. 3D physics take on many forms within games, from environmental movement of trees and plants to human ragdoll physics. One of the main uses of physics within a game is the use of projectiles, such as ball physics. However, whilst spherical ball physics are commonly used, balls in different shapes such as oval are less used due to their complexity.

1.2 Project Background
Ball and collision physics occur throughout a wide variety of games, whether it be a first-person shooter where a projectile hitting a target propels it a certain direction depending on where it is hit, to role playing games where the player interacts with objects and their environment to complete tasks. The sports genre of games uses these types of physics to replicate what happens in the game, such as ball movement in football, tennis or golf. These types of games are commonplace in games stores and online, with new editions coming out every year for the masses of gamers to enjoy. However, one of the biggest sports in the United Kingdom is often missing from this list, the sport being the game of rugby. These games have often struggled in the marketplace, with users citing reasons such as poor gameplay and lack of realism. One aspect of this is the movement of the ball, which is different from most other sports due to its egg shape which affects the flight and movement of the ball when hit, making it unpredictable. Previous representations have often been inaccurate due to this. This project will try to create a simulation which will represent this better.

1.3 Project Objectives
1.) Research into the topic of game simulations and physics, specifically looking at ball and collision physics to help justify the creation of this project. Also, research to be carried out into previous iterations of rugby simulations for insight into how to improve.
2.) Design a simulation of the physics of the rugby ball which replicates the movement of the ball.

3.) Creation of the simulation within a suitable game engine.

4.) Test the simulation amongst a group of participants on its success and where improvements can be made.

5.) Evaluate the simulation on its success and where improvements can be made for future iterations.

1.4 Deliverables
1.) Code for scripts that run the simulation

2.) Project report which explains the reasoning behind the simulation and results from the testing.

1.5 Timeline
To complete this project in the time available, a timeline must be followed. To visualise this, a Gantt chart has been completed to show how the project has been planned.
Chapter 2: Literature Review and Research

2.1 Overview
This section will review various sources such as journals, books and websites which are relevant to the subject to provide understanding and reasoning behind the topic of interest and give a better background on how to accomplish the task at hand. Research will also be carried out into how to complete the project.

2.2 Video Games Industry
The video games industry has been consistently growing in value across the globe, with the revenue of the industry being stated as just under $100 billion in 2016 and set to rise to $118.6 billion by 2019 (UKIE, 2017). This compares to other entertainment markets such as the movie industry where global revenue in 2016 reach $38.3 billion (Romero, 2016). The market tends to aim at the teenage to young adult age groups, with Lenhart and Macgill (2008) showing that approximately 81% of 18-29 year olds in the USA play video games in one form or another. In terms of numbers of different video games produced, the latest numbers from Steam for 2016 showed that over 4000 different titles were released on their platform alone (Plunkett, 2016), with the potential for many more to be released on consoles and other platforms. Also, adding to this, extra products which add to existing games such as downloadable content packages (DLC) are not counted among this. From these numbers, it can be seen that entering the video game industry at this time with a game that meets the expectations of the consumers could be a profitable one but it has the potential to be risk. Large studios with successful brands of games such as Maxis, Bullfrog and Black Box games have all disappeared due to low sales, sometimes because of one game (Schreier, 2015).

2.2.1 Sports Games
The sports genre is one in a number of different type of games. The genre covers mainstream sports such as football, tennis and golf and some of the more extreme sports like skateboarding and snowboarding. Whilst racing is considered a sport, it is considered in the games industry as its own separate genre. The genre of sports makes up a significant proportion of games sales, with the ESA (2016) stating that approximately 14% of total
games sales in 2015 were made up of sports, compared to other genres such as family-orientated, adventure and strategy games selling less.

2.2.2 Rugby Games
Rugby video games fall under the sports genre of games, yet unlike other sports, it often struggles to captivate an audience and sales are usually much lower than the rest. Despite worldwide interest in the sport in its various forms such as union, league and 7’s growing, with the 2015 World Cup Final being viewed by over 120 million people across the globe (RWC, 2015), the sales compared to this do not match. The most recent iteration of a rugby video game Rugby Challenge 3 (Tru Blu Entertainment, 2016) appears to have struggled, with only 40 user reviews on the games store Steam (2003) being made. In terms of critic reviews, the game has been set back by a number of different issues, ranging from visuals to gameplay (Unknown, 2016). Reviews like this can be damaging to the amount of games which are sold, with Zhu and Zhang (2010) showing that critical reviews are of great importance to games which are less popular to the wider audience.

One of the noticeable issues with the rugby video game genre is that there is no brand series currently in production. In comparison, the football market is largely controlled by Electronic Art’s (EA) FIFA and Konami’s Pro Evolution Soccer. In these 2 series, each brand is able to produce a new product each year with more updates to gameplay to make it seem more like the real sport. Rugby does not have this in any form, with the last series to be made for the sport made by EA concluding in 2008. This means that the genre has not been able to improve on itself yearly, reducing the progress made in gameplay and visuals which affect the sales.

Another issue with creating a rugby video game is the complexity of the sport itself. Other sports like football have 1 objective, such as score more goals than the other team with other rules to balance play, and can be picked up by most people easily. People are able to create games on a whim in the park or make teams to play local games. Rugby’s issue is that there are a lot more intricate rules to the game. There are rules to do with; how many how points are scored for scoring a try, a penalty or a conversion, the tackle and breakdown area of play and set pieces to name just a few. To replicate these in a virtual environment is a
difficult task which requires the process of many variables which may seem daunting to game developers. The complexity of the game can be linked to American Football, where there are a large number of rules to govern an intense and challenging game. However, American Football has an advantage because the game has an array of different stoppages which allow the viewers and gamers to process what has happened and plan their next move, whereas rugby is a freer flowing game with stoppages only for infringements. This means the player must be able to think on the go when making their next move. This may put players off from purchasing the game due to the difficulty of the sport.

2.3.1 Physics within Games
Physics within video games often hold the key to the success of the game. Many modern games on consoles and computers rely on the effect they have to carry out what is intended. However, it is only in recent times that physics development has expanded, thanks to research by Hertzmann and Zordan (2011). They showed that whilst graphics improvements have always been in the front of game developers minds, physics has been overlooked. This could have been due to performance issues of computers at the time, as Geijtenbeek (2013) said that compromises had to be made between the accuracy of the models of physics compared to the performance of the video game. With computers becoming ever more powerful, physics engines are better able to replicate real world actions in a virtual simulation thanks to their ability to calculate the large amount of variables used to produce meaningful results. Using physics within games is also complicated for the developers at hand. Many developers may only come in with a basic understanding of physics and Coleman et al (2005) says that most underestimate the complexity that comes with designing the physics within a game. Game developer jobs and courses often now require a good understanding of mathematics to be able to cope with the number of different and difficult calculations needed to make the game engine perform correctly. Schools are also adapting to the need to train game developers in physics before reaching this level. Coleman et al (2005) shows this with his curriculum including extra game modules to help teach students. However, his idea of extra modules are only optional for students and also fall in the autumn of the year, where many students will not be around to take up the opportunity.
### 2.3.2 Games Engines

To build the project, an appropriate game engine will need to be used. A game engine is defined by Ward (2008) as a program which is able to do tasks such as rendering, physics and input which are common amongst all games so that developers may add their own additions and make changes to make their games unique. Individual objects and textures do not come under the engine itself, and are instead added by the developer who adds the game engines scripts and functions to the objects to make them perform the way that they like. Many game companies will have their own game engine that they are then able to license to others at a cost. In recent years, however, more game engines have been released as open source projects to allow the masses to create their own games easier. Munro et al’s (2009) research into the Quake series and the engines that have been used to develop the games has shown how the game engines have developed over time from disorganized pieces of software to artefacts which have distinct modules which can be transferred across engines. However, this is analysis of only one particular game engine series which is devoted to a particular game. Other game engines which may be used for other genres may be more or less confusing in terms of architecture. Mishra and Shrawankar (2016) agree with how game engines have progressed over the years, and they also add that game engines now provide a solid ground for game development in terms of sounds, graphics and AI.

### 2.3.3 Unity Game Engine

One game engine that is open to the general public as open source software is the Unity game engine. First created in 2005, Unity is a game engine developed by Unity Technologies which is able to create games for PC, consoles and mobile devices and allows for ports across consoles easily. It also is available for free with basic levels of ability to create games, meaning that anyone can download the engine and build their own software. Dickson (2015) discusses how Unity is not currently regarded the same as other game engines such as Unreal by the industry but it stands out for newer game developers as it has a shallow learning curve. He was able to show this by developing a games development course using Unity as the core engine. All the students in the class were able to create a game from scratch with no previous experience of the engine beforehand compared to other courses where Unity was not used where less than 50% of the course was able to complete the task.
Unity can be broken down into different sections (Walsh, 2016):

- **Scenes**: Thought of as game levels, these contain a 3D world where all the other objects can be placed and commands can be issued.
- **Game Objects**: the building blocks of the game, these are formed from various components. They can have features such as textures, sounds and physics applied to them and shared across other people’s games.
- **Scripts**: Component of game objects, they control the behaviour of the object. These can be written by the developer in either C# or in Unity’s own language UnityScript.
- **Materials**: determine how the object will be shown to the user and other features of the object such as bounciness.
- **Shaders**: scripts that show how the object will be coloured.

There are other game engines available for use as well. One example of this is the Unreal game engine (1998), created by Epic Games in 1998 for its Unreal games series. Since then, the engine has been released as open source software like Unity for indie game developers to use on their own software. However, as the background of the game engine is from the first-person shooter genre, it is better suited to create those type of games. Other genres can be made but it is more difficult compared to Unity (Pluralsight, 2014). A suggestion by Pachoulakis and Pontikakis (2015) is that potentially Unreal and Unity engines could be used as a combination for teaching game development students but currently it is up to the user on which one they prefer.

A key feature of Unity that will be crucial to the success of the project will be the use of components on objects. As described before, components make up the features of an object within the game such as the scripts attached to them and the physics that are applied to the object. One of the most important features is the use of colliders on the object. Colliders are defined by Unity Technologies (2017) in its documentation as “components that define the shape of an object for the purposes of physical collisions.”. The colliders provided can come in many different shapes and sizes depending on the needs of the creator and the shape of the object that the component is attached to. One of the more popular colliders is the use of a rigid body collider, which allows and object within the game
to be affected by forces applied to it and react accordingly when it comes into contact with another object with a collider attached to it. Colliders are constantly evolving in their complexity as computer systems are able to handle more intense calculations and complete them at quicker speeds. One area in where they are progressing is through the contact with water, which is currently a complex matter due to the unique features that water has compared to other objects. Kellomäki and Oliveira completed work on using rigid bodied objects within a water based environment and were able to configure them in such a way that water was either blocked by the object, causing them to sink, or to float depending on their previous properties. This was also completed on machines with limited graphics processing unit (GPU) power meaning that older computers would be able to work with this new method.

The Unity Engine is made accessible to all types of developers from beginners to fully fledged companies thanks to it providing access to what is known as the Asset Store. This section of the software allows users to put objects, animations and other types of components on the company’s website for other users to use in their own projects. These objects can either be put up for a cost to help fund the developers time or they can be released for free. As of writing this, currently the Asset Store contains over 15000 different objects that are ready to be used to improve people’s game development.

2.4.1 Collision Physics
Within physics in games, collision physics are sometimes considered the heart of the game as without them the game would simply not function properly. Collision physics is defined by Bourg (2002a) as the calculations and actions of objects when they collide into each other. In the games industry, collision physics is split into two different sections; collision detection and collision response. Collision detection is the issue of whether two objects have actually collided together whilst the response is what happens to them after they have collided together. In terms of this project, this means that there needs to be detection of whether the foot has hit the ball and where, and then the response needs to be where the ball goes after being hit, how fast it goes and other variables that may come into effect.
In terms of collision detection within a system, Ericson (2005a) showed there are six factors which determine how well a physics simulation works to analysing the connection; application domain representation, queries, environment simulation parameters, performance, robustness and ease of implementation. Application domain representation refers to the shape of the object, such as a sphere or cube. In the case of this project, the ellipsoid shape of the ball makes it more difficult to replicate. To counter this, Ericson (2005b) states that objects which are more complicated than these standard shapes can be made up of triangles to form a polygon mesh. While this will mean that a rugby ball can be built properly, it will be less efficient than using a standard shape as the simulation may not understand what is inside the ball and therefore effect the results of the collision response. Collision detection is also affected by the number of objects in the scene at any one time. Liljeby, Seipel and Jenke (2011) showed that the complexity of the model is $O(n^2)$, meaning that if there are multiple objects in one scene and the system is attempting to detect a collision of object A, it has to calculate if there have been collisions with all the objects to object A. To make the simulation efficient for the this project, the number of objects in any one scene will be reduced as much as possible to reduce the complexity and the computing power needed to run.

Collision response has its own set of problems which need to be addressed. One of the first problems that has to be addressed is the immediate impact of the collision between the foot and ball and also ball to ground. The ball will have to follow the impulse-momentum principle (Bourg, 2002a), where impulse is defined as the force that acts on an object in a very short time. To calculate the impulse, the following equations can be followed:

$$Ft = m\Delta v$$

This is where force ($F$) multiplied by time ($t$) equals the mass ($m$) multiplied by the change in velocity ($\Delta v$).

Another consideration that needs to be made is following Newton’s principle of conservation of momentum. This principle is stated by Feynman (1964) that the momentum
of both objects before the collision should equal the momentum of both objects after the collision, providing there are no other external forces in play. This is shown by the equation here where u is the velocity before and v is the velocity after:

\[ m_1u_1 + m_2u_2 = m_1v_1 + m_2v_2 \]

However, as mentioned, this only accounts for if two forces are at play. In a normal rugby game, however, there are more forces such as deformity of the objects. To account for these extra factors would take even more computing power than before, further effecting the speed of the simulation. The solution to countering this is to make the object a rigid body structure. Thakur and Gupta (2012) define rigid body structures as objects in computer simulations which are made up of contact points. These contact points can then be used by collision detection algorithms to produce a reaction to the force. This comes with its own problems as this means other forces have to be disregarded. Thakur and Gupta also say that more complicated rigid body structures with more contact points need more computational power. For this project, as the shape of the ball will be an oval shape so this should not be an issue.

### 2.4.2 Projectile Physics

Once the collision has been detected and a response has been calculated via the impulse-momentum principle, the ball will then become what is known as a projectile (Bourg, 2002b). The ball will have no force acting on it to speed it up in the direction that is heading but other forces will be in play.

The most important force that will act upon the ball which needs to be taken into account is the force of gravity. Following Newton’s law of universal gravitation (1729), the following equation can be used to calculate the force applied to the ball when it is in flight:

\[ F = mg \]

where F stands for the force of gravity, m for the mass of the object and g for the gravitational constant which has a value of 9.81 m/s². Using this simple equation and from
other variables that are already known, the distance that the ball travels can be calculated. Grimshaw et al (2006) shows that the range can be calculated using:

\[ R = \frac{v^2 \sin \theta}{g} \]

where \( v \) equals the velocity that the ball is launched at and \( \theta \) is the angle at which the ball is launched at. The height of the ball can also be calculated as well to produce a more accurate simulation. Grimshaw et al (2006) again shows that this can be calculated by the following two equations:

\[ H = \frac{(v \sin \theta)^2}{T} \]

\[ T = \frac{2v \sin \theta}{g} \]

The top equation calculate the height of the projectile in the air whilst the bottom equation calculates the time the projectile is in the air. With all of the above calculations, the motion of a projectile such a ball travelling in the air can be calculated with only the force of gravity acting on it. However, in real life there are other forces acting upon it. One of the key factors is the aerodynamic drag on the ball due to air resistance. Elert (1998) defines aerodynamic drag as the force that acts on an object the resists its motion when travelling through a fluid. Whilst the air is not a typical fluid as in liquid form, it is described as such it is not able to resist another force for a long amount of time. Drag is a complicated force to calculate, however, due to the increased number of variables which affect the true value of the force. This is shown in Alam et al’s (2011) study into the aerodynamics of golf balls. They observed that other factors such as the golf balls dimpled structure made calculating the drag of the ball much more difficult. Other factors include the amount of spin applied to the ball which also has the ability to affect the trajectory of the projectile in flight and the differences in production of the gold balls by separate manufacturers which delicately alter the design of the ball.
2.4.3 Rugby Ball Physics
As seen previously, calculations can be done to work out the speed and trajectory of a projectile when a force is acted on it, such as someone kicking a ball. However, the difficulty with the trajectory of a rugby ball come with the shape of the ball. A standard rugby ball in an official Rugby Union game must meet the laws of the game which are as follows; a circumference end to end between 740 and 770 millimetres, a circumference in width of between 560 and 620 millimetres, a weight of between 410 and 460 grams, and, most importantly, made up of 4 panels into an oval or ellipsoidal shape (2003). There have been a number of studies into the aerodynamics of a rugby ball and how the ball travels in the air. Alam et al (2006) studied the aerodynamics via wind tunnel experiments and computational experiments and found the drag coefficients to be around 0.18 in both methods. Further research by Djamovski et al (2012) also agreed with the research completed by Alam et al but also said that the calculations were incredibly complicated on the shape of the ball when it is in one ball. The spin of the ball in the air increases the complexity of the work even more. Another factor that they mentioned that was also previously mentioned in the section before was the material and texture of the ball affecting the flight of the projectile. As rugby balls often have a dimpled surface to improve grip of the ball for players, this increases the amount of drag further on the projectile. For further research into how much this affects the ball, they suggest using detailed 3D images of the ball in computational experiments to explore this further. Another piece of research was carried out by Seo et al (2010) to look at the affect the seams of the rugby ball had on the trajectory of the ball in flight. They found that the seams cause asymmetrical pressure distribution across the ball, making a spinning ball in flight shift from side to side. In the simulation that will be built, this may be disregarded as the shift in flight is minimal and would add extra computational work to the simulation.

From the research of the literature reviewed here, creating a simulation of rugby ball will require a large amount of complicated calculations to recreate the true flight of a rugby ball.
Chapter 3: Methodology

3.1 Overview
This section of the project will cover the research methods and design process of the software to show the development of the solution. It will show the use of primary and secondary data collection to help show how successful the project is. It will also explain the ethics behind it to show that no bias has been involved in the production of the solution, and that any problems that arise may be solved.

3.2 Research Philosophy
When completing research in any field, the research philosophy must be decided on before any data collection or interpretation of results can be carried out. There are many different philosophical approaches that cover the range of how to interpret the world. These can be seen in Figure 1.

Figure 2. Research Onion (Saunders et al, 2007)

The research “onion” created by Saunders et al (2007) describes the way a researcher needs to go about creating an appropriate method for completing research. It covers features such as techniques, approaches and strategies for research. The outside layer of the “onion” covers the different philosophical approaches for research. One of the main research philosophies is the positivism approach. This approach assumes that knowledge gained
through observation through the senses is reliable and accurate. The problem with taking this approach for this project is that the facts are purely objective and external (Wilson, 2007). With this project, the research relies on more interpretation from participants who may have differing views on how the project acts. Another problem is that this approach needs large sample sizes for it work effectively, which this project will not use. Therefore, this makes following the positivism approach not viable.

Another approach that could be taken is the realism approach. This approach can be separated into 2 different sections; direct realism and critical realism (Novikov and Novikov, 2013). In direct realism, the researcher simply states that the results gathered are what they appear to be, whereas critical realism may question the results based on other factors getting in the way, such as the human senses. Whilst this would be a potentially viable way to approach the research as the results are dependent on the participant’s views of what they observe, it does not cover a complete view of the research topic at hand and so this approach is also not viable. The research methods must also be tailored specifically for the subject and must collect either qualitative or quantitative.

The final approach that was looked at was the pragmatic approach (48). This approach states that no one view can give an entire view of the topic and that instead there may be many other possibilities instead. This means it allows the research to take either an objective or subjective view of the topic at hand, which allows it be more flexible as times progresses through. It also allows both qualitative and quantitative data to be attained for use as interpretation. For these reasons, therefore the research will follow the pragmatic approach.

**3.3 Data Collection and Research**

Goddard and Melville (2004) said that research is not just a method of gathering information, but is instead more about either answering unanswered questions or creating new questions where there currently aren’t any. For this project, the solution that is being created is trying to answer the question of whether a working simulation of the collision physics of a rugby ball can be potentially used in further work in the games industry.
To answer this question, two different types of data will need to be collected; primary and secondary data. Primary data is information that has been collected by the researcher themselves. Glass (1976) simplified this to original analysis of data using statistical methods. Secondary data is described by Glass as the re-analysis of data to either answer a previous question with new analysis techniques or to answer a new question with old data. Both types of information have their own advantages to advancing the knowledge on the subject at hand. Primary data collection via methods such as interviews, questionnaires and surveys allow the researcher to gain up to date data on their research topic to keep it relevant. This is especially important in the computer science field as technological advancements are constantly ongoing. Primary data collection can also be tailored by the researcher to specify which data is needed to help answer the question rather than more general collection which may lead to more ambiguous results (Anonymous, 2016). However, primary research is setback by the time and cost it takes to gather the data and the data collection depends entirely on the participants and their responses. Secondary data collection via the use of data previously collected, meanwhile, has the advantage of taking less time to gather and provides the basis for which primary research can be built upon. Its disadvantages however are that it may be out of date by the time the new research is carried out, and that the data may not fit correctly with the new research. It may have to be disregarded due to changes such as boundary changes through the year (Anonymous, 2016).

Both methods of data collection will need to be used together to form the basis of research around the topic. For primary data collection, a questionnaire will be created which will be able to gather data about the simulation from participants. Secondary data will be collected from the simulation itself via the results gained from where the ball travels and how it does this. Separate testing of the functionality and usability of the software will also be done to provide the secondary data.

### 3.4 Research Methods

Once what data needs to be collected has been decided, the next step is to decide on what type of research needs to be completed. There are two types which can be used in this situation; quantitative and qualitative. Quantitative method is the collection of numerical data from a large number of tests or participants which allow trends to be identified.
Examples of this include using questionnaires with fixed set of answers for each question or from a software solution which can be repeated many times over (Babbie, 2010). Qualitative research meanwhile, takes on a more interpretive view of the data collected. It is used to collect data about more “human” answers, such as opinions and beliefs. These are gathered through methods such as focus groups or specific questions in surveys which ask for an explained response (Mack et al, 2005).

For this project, a combination of the two will be used to gather the data necessary to evaluate the project. Quantitative data in the form of a questionnaire of participants will be gathered who will test the solution using questions with the use of both fixed answers on a number scale and space to show their opinions. The solution will also be repeatedly tested against what is expected in a real life situation to assess whether the solution meets the expected result. Qualitative data will also be gathered via the questionnaire as well as there will be open-ended questions which will allow the testers to give their opinion on what improvements can be made to the solution. The project will take on a deductive approach to the research rather than an inductive one as much of the data gathered will proving that the solution works. This compares to an inductive approach which is used for trying to build new theories instead (Stenbacka, 2001).

When selecting the participants for the testing of the software, a suitable method of selection needs to be used to make sure that bias of the results is kept at a minimum. There are many different methods which are suitable for different projects depending on time allowed and the type of software that is being collected. Many take a long period time to calculate who should be picked according to what attributes the participants have whilst not including time taken for participants to carry out the study. As both time is limited in creating the solution and testing of the solution will also take time to complete, the strategy for sampling chosen is the opportunity method, otherwise known as the convenience method. This will select people to test based on their availability at the time of testing. Whilst this will be achievable in the time period available, it will also be likely that it is not representative of the general public’s opinion as only participants from one type of background will be used. Also, participants will have varied knowledge of computer
programming and modelling of software which means that opinions will vary in the scope of how the system could be improved.

### 3.5 Risks and Limitations
When completing any kind of research project, there are associated ethical risks which can cause problems with bias and make the research unusable. There may be other problems that arise throughout the development that may need to be accounted for in case they happen.

One of the first risks towards this particular project is the understanding of the work at hand. As mentioned previously, convenience sampling will be used to determine which participants are used for testing of the software with the potential for the snowball technique to be used if necessary. The problem here is that participants may not be aware of either the sport of rugby and the expected outcome or they may struggle to use the software if they do not have a technical background. To solve this problem, participants will be informed of the process beforehand via an information sheet which will give a guideline to the research being carried out. The questionnaire that they will be asked to fill out will also be tailored for a user with beginner level training in IT so that they will be more able to answer the questions provided. Selection of people from various backgrounds will also reduce bias in the results as their knowledge of both the sport and software will have less of an impact.

Another potential problem is bias when selecting which primary research to use. For primary research, it could be seen that specific questions will be made to get an answer that the researcher wants. The way to solve this is to make every question that is made to collect primary research will be left open ended to ask for opinions only to prevent this problem from occurring. There is still the risk, however, that the questions asked are what the researcher is looking for and not what is needed to make the data gathered useful to the wider reading audience. This is a risk that occurs when creating any type of questionnaire for research and must be accounted for as part of the evaluation of the project. Questionnaires also face the problem of honesty in the answers provided as participants may want to fulfil the social status quo of leaving positive or negative results so as to not
stand out from the crowd. For this issue, it is imperative that the respondent knows that the information provided is private and cannot be linked back to them in any way.

3.6 Design Methodology
With the methods of data collection and research completed, it is now possible to decide on an appropriate software design method to complete the solution. There are many different possibilities when deciding on a design method which depends on what is being created, the time needed to create the artefact and the accuracy needed on the project. One method that can be used is the Waterfall method. This method is a sequential step by step method of design organising the requirements, design, implementation and maintenance into a logical order (Rouse, 2007). Waterfalls downfall is that the process only contains one iteration of the production of the software, meaning that improvements and other features are not allowed to be implemented. This will not be satisfactory for this type of solution as other features may be implemented at a later date or problems with previous iterations may need to be fixed.

Another method that could be used as a design methodology is the use of Rapid Application Development (RAD). RAD involves the process of developing versions of software quickly whilst not attempting much in the way of planning and designing of the software (Rouse, 2016). Instead, any requirements of the software may need are developed whilst creating the software and are built on with each iteration of the solution until a fully working piece of software is finished. As with the Waterfall model, this will not appropriate with this type of software production. This is because the solution needs appropriate research and planning to identify where weaknesses are in previous builds before the solution can be developed.

The final method that has been approached is the Agile method (2008). Created in 2001, Agile was developed by 17 different independent software designers to create a design methodology that would focus on producing software quickly for the users needs. Agile is similar to RAD in that it follows an iterative process to develop the software. However, it differs from RAD in the requirements and improvements needed are constantly taken into account and changed with every iteration. This allows the software to evolve according to
what a client wants and allows the software to be adapted if any problems arise further down the line of development. This is suitable for this project as throughout the development cycle, there may be other features that could be added to help improve the simulation and as the researcher is using this type of software for the first time, problems are expected to occur throughout the process of development. Therefore, it is the Agile method that will be used to develop this software project.

The design had to encompass both the creation of code which would control how the system would operate and also the design of the scene on which the system would run. Firstly, the objects were visualised onto the screen, taking into account where there placed in the field of vision so that it was easy to see all aspects of the game. When the researcher was happy with the layout of the scene and it was made simple to duplicate across different levels, the code design followed after. This involved looking at what functionality was needed make the system interactive with the user and the intended results from these actions. Different functions were written out with what their intentions were in pseudocode for so that they could be referred to throughout the development phase. As part of the design process, variables such as computing power needed to be taken into account to prevent problems from running the simulation on most personal computers. To meet these needs, a set of objectives were set up that needed to be met for the simulation to be successful:

- The layout must be easy for a new user to understand
- The user interface and controls must be easy to understand for a new user
- The simulation must run at a quick speed
- The movement of the ball should replicate the movement of a real life rugby ball

These features could then be tested through the use of functionality and usability testing.

3.7 Development
The design of what was intended to happen allowed the researcher to develop the solution appropriately. Unity version 5.6 was downloaded and the original scene was created to act as the platform for the different levels of the game. A plane could then be set up which would act as the field and the components of the field were set up. A box collider which
would surround the object was created to act against any objects that would come into contact with it, namely the rugby ball upon landing. A set of goalposts were also created from a set of three cylinders under one empty game object. The cylinders were transformed in scale, rotation and position to form the shape in relation to the size of the field. A plane object was also added into the gap between the two upright posts which had no mass or physical structure to them. This would act as a detection collider for when the ball travels through the post. The plane also stretched higher than the posts as due to rugby rules, as long as the ball travels through where the posts should be, it still counts as a successful kick. With the participant having the opportunity to kick the ball as high as they would like, this needed to be included in case of the eventuality of this occurring. When the posts were set up correctly, the group of objects was duplicated and placed on the other side of the pitch to allow for improvements and additions to be made at a later date.

Once the setup for the field was complete, the researcher added the ball object to the field. The ball was added centrally to the field to begin with, with the option to translate its position across the field in different levels. The object was also transformed to been in scale for the rest of the other objects on the scene, and the weight of the object was set at 0.416, the same as a real ball. The camera was then positioned behind the ball so that the user could clearly see ball and the posts. The camera also had to be positioned so that the user could see the direction and angle that the ball was facing to make it easier for changing the positioning when in use.
The final objects that needed to be created for the simulation was to create a user interface (UI). The UI needed to be able to show a power level bar which the user could move to change the speed of impulse of the ball when hit, a button to hit when the ball was ready to be hit and a reset button in case the ball missed. The UI also contained a score count for progress through the different level designs.
With the environment for the simulation set up, the next step was to look at the scripts that would control how the ball would interact with the world. The script also needed to be able to interact with the UI to control the power of the ball and when it needed to be hit and reset. The first interaction that was scripted was the controls of the ball. The researcher first attempted using sliders in the UI along with the power bar to control the pitch and direction of the ball. However, this proved difficult to control at a rate that easy to control for the user whilst also not being too slow that the user would have to move it in large amounts to get any form of movement. To stop this from occurring, it was decided by the researcher that the controls would move to the keyboard, namely the arrow keys, with the up and down keys controlling the pitch of the ball and the left and right keys controlling the direction. These controls are controlled through and update function which is continuously running through with each frame to check if a button has been pushed. Each key press was calibrated to change the angle by 5 degrees to allow for a large amount of freedom in the movement of the ball.

The goalposts and how they acted upon the ball being kicked between them also needed to be scripted for them to work effectively. The detection collider plane had to be scripted in such a way that when the ball collided with it, it needed to trigger a set of events. It needed to be able to add to a score marker throughout the game and also proceed onto the next level. This was scripted as an on trigger event, meaning that any object that collided with it would start the event. As there was only ever going to be one object colliding with it, this meant that the event would only trigger once per level.

The ball object was the final part to be scripted. There were two features that needed to be taken care of; firstly, the ball had to made available to be hit. When creating the scene, the kinematic feature of the ball which controls whether the ball is affected by forces and collisions, is configured to be false, meaning that the ball stays in one place. In the script, this feature gets changed back to normal which allows the ball to be moved by the impending force. The second feature translated the force that the user set to a scaled force
that would move the ball in the direction that it was facing. This was assigned to the go button so that these actions would only occur once the user was ready for it to happen.

Finally, with the scripting of the objects sorted, the scene could be replicated a further four times to produce new levels. To make them differ from each other, the ball and cameras position could be translated to a different area of the pitch. This would not affect any of the other objects or its movement as the script was constantly linked to the object throughout the duplication.
Chapter 4: Results

4.1 Overview
This section of the project will cover the results from producing the solution. It will explore why certain choices have been made when developing the project and it will delve into the results of the participant testing.

4.2 Development Reasoning
Throughout the process of development, many decisions needed to be made to construct the simulation appropriately.

One of the key decisions that needed to made is how to design and shape the ball. As mentioned previously in the literature review, the ball is an ellipsoidal shape which is made up of four separate panels. When modelling this in the Unity engine, it is difficult to get this shape as the pre-built shapes are of basic shapes such as cubes, spheres and cylinders. In normal circumstances, these can be stretched and transformed to create other different shapes but, in this situation, this would not produce an accurate representation of the ball. To solve this problem, a model produced in another program which can be imported into the project space was chosen to use. With the researcher’s skills in using modelling software not sufficient enough for this project, a model created by another designer was used (Kokiashivili, 2013). This model crucially had the four panels that were required and was able to be transformed appropriately so that the size of the ball was in scale with the rest of the simulation. Using the prefab model also required changes to the shape of the collider that was provided. As the shape was unusual for the engine’s usual capabilities, the collider needed to be shaped manually to surround the model completely. Unfortunately, the engine’s colliders transformation properties prevented the collider from being moulded around the model. Instead, the collider was shaped into a capsule that covered the majority of the model but left the edges of the model untouched. This prevented the ball from fully connecting with the ground as expected. The end result was that the ball would occasionally cut through the field plane at the points of the ball. To solve this, the object was amended to have a mesh collider which would take the entire object and mould the collider around it. This then stopped the previous problem of the ball cutting into other objects and provided a more realistic outcome. Whilst working with the collider, a material was also added to the
ball called “Bounce”. The feature allowed the ball to have variables for the friction that acted on the ball when it came into contact with an object and the elasticity of the object upon hitting an object. This was especially important when the ball came into contact with the field plane if the user was unsuccessful in scoring a point.

In continuing with colliders, the plane set up for the field also needed to be configurated so it would act similarly to what was intended. In this case, this was much more simple than the ball as it was a simple cube. Therefore, a box collider could be used which would easily surround the object completely. This collider needed configuration in its size though as original collider provided was too large in size, causing the ball to appear to float on top of the surface rather than touch the ground.

Another decision that needed to be made was the initial positioning of the ball when starting the first level. The intention of the first level was to show the basics of how the game ran by giving the user a simple target to aim for before making it more complicated the further they got through. Also, in a real rugby game, one of the more easier kicks is positioned in front of the posts after a try is scored underneath the posts. Therefore, it was decided that the first position was to be placed in front of the posts from close distance, meaning there was only a small chance of the user missing. From this position, the ball could be translated across the field using the initial position as a reference point. With each change of the balls position, the camera’s position also had to be moved so that it remained the same distance from the ball at all times whilst allowing the user to clearly see the target of the posts. An improvement to this system could be to allow the user to move the camera to permit greater freedom of kicks as well as more freedom across the pitch. A current issue is that the ball also appears to be starting in a floating position. This was set as this to allow the ball to be fully rotated 180 degrees in the vertical axis but does not look as real as expected. A future edition would need to correct this by possibly adding a tee object which the ball could sit on that would also not influence the balls trajectory once a force is applied to it.

When creating objects in Unity, the components need to be set up to make sure that they are functional and act appropriately with other objects. In the case of the rugby ball, the
physics of the object needed to be set up. To make this happen correctly, the rigid body component was applied to the ball. This allowed the ball to interact with gravity and combine with the collider set up previously so that collisions would react in the correct way. Using this component also allowed the weight of the ball to be set to the standard value in a normal game of rugby. This meant that when working with the scripts, the power could be correlated with the weight of the ball to provide an accurate result when the ball was acted upon. Another feature used as part of the rigid body component was the use of the kinematic feature. As mentioned in the development of the simulation, this feature had to be set to false as starting it on true meant that the ball rolled away under the effect of gravity, meaning that the direction and angle of the ball would always be off for the user. The component also has a feature for angular drag, which set to a value of 1 so that the ball would be acted on by aerodynamic drag when flying through the air.

Writing the script which controlled the system involved making further decisions. Much of this involved around where to put the functions. In Unity scripting, there are two main functions that occur in most scripts; a Start function which controls what happens when starting the program, and an Update function which runs repeatedly with each frame of the game that passes. When coding the script, it was decided by the researcher that nothing need to be enabled or disabled upon starting the program, so the Start function was removed. However, the program needed to be able to check for user input on each frame so the Update function was kept and altered to transform the direction and angle of the ball when the user pressed the arrow keys. The individual features of the user interface were then coded independent of each other to alter the variables needed to control the ball. The reset button was also coded to reset the current scene as coding it to put the ball back into its original position would involve writing separate scripts for each level.

4.3 Functional Testing
As part of testing whether the system was working as intended, a set of tests were set up to test the different features of the environment and whether they worked correctly. This is known as functional testing and it is designed in such a way that it is from the user’s perspective to see that the outcome is as expected (Guru 99, 2016). This compares to non-
functional testing which is used to evaluate the reliability and performance of the system and requires separate tools to provide accurate results.

The following features were tested by the researcher to make sure that the program was functional;

- The user could control the direction of the ball using the left and right arrow keys.
- The user could control the pitch of the ball using the up and down keys.
- The user could set the power of the force applied to the ball using the power slider.
- The user could start the simulation by pressing the Go button.
- If the kick was successful, the system would move on to the next level.
- If the kick was unsuccessful, the user would be able to reset the scene using the Reset button.

All of these features were successful upon testing.

4.4 End User Testing Results

Once the simulation was completed sufficiently enough, it was released to participants to be able to test. Users were given 5 minutes to test the product once the researcher had instructed them on how to control the program and what the aim of the game was. When either their time was finished or they had completed the game, they were asked to fill out an anonymous questionnaire which would rate the game based on its realism and usability and asked for opinions on improvements. A total of 7 different participants completed the testing and questionnaire.

The following questions were asked of the participants:

- On a scale of 1-10, 1 being completely fictional and 10 being lifelike, how would you rate the realism of the simulation? (Select 1 answer)
- On a scale of 1-10, 1 being very slow and 10 being very quick, how would you rate the speed of this simulation? (Select 1 answer)
- On a scale of 1-10, 1 being unusable and 10 being very easy to use, how would you rate the user interface of this simulation? (Select 1 answer)
• Are there any improvements that you would like to be made to the simulation?
• Any other comments?

The overall reaction to the software was generally very positive. Results for question 1 and 2 were between 7 and 10, and question 3 had results between 8 and 10. This was a good reaction to a product that was the beginning of larger things to come. This also links to potential player retention if the initial outlook of the game is promising.

As the aim of the project was to produce a life like simulation of the ball when being hit, it was a promising sign that question 1 was answered with positive results. With previous rugby game titles having problems with immersing their users due to problems within the game, it is good to see that users believed the simulation met their expectation for realism.

Speed was another crucial factor to the success of the project as slow programs halt the users experience and make it not enjoyable for them to use. With the positive results coming from question 2, the simulation succeeded in this regard. However, in future editions, if any improvements and additions are made, they may affect the speed as the simulation will have to load more objects and their proprieties each time on playing which will use more computing power.

The final rated question also produced very positive answers. This is due to the simplicity of the controls being only arrows keys and a slider to control power. As before, if future additions are made which may complicate the game further, this may make it more difficult to understand.

When looking at question 4, they were varied results on what could be added to further improve the simulation. The key point that stood out from the results was that the participants wanted more external factors that could affect the ball. This could have been in the form of either wind speed and direction or possible weather conditions such as rain and snow. These could have affected the direction and power needed to strike the ball and add an extra level of complexity. Also, potentially related to this, another tester asked that more levels to be introduced into the game. These could potentially combine to make the game
more difficult with more external factors as the user progresses through the game. The final answer from this question responded with the possibility of being able to position the starting place of the ball differently themselves, which could potentially lead on to the possibility of creating a custom game that the user can create within the simulation. The responses provided gave the researcher many options from different users on where the simulation can progress in the future.

For all details of responses to the questionnaire, please see Appendix D.
Chapter 5: Conclusion

5.1 Completion of Objectives
The aim of this project was to be able to complete a working simulation of a rugby ball via the use of a game engine. The end goal was that the ball would travel through the air and land realistically after different forces had been applied to the ball at different positions across the field and on different places on the ball object itself. This was achieved via background research into literature around the subject of rugby ball physics, previous rugby game titles and game engines that would help assist in creating the project. This provided a base for the design and development of the project.

With the literature review complete, a design could be put in place that would best replicate the flight of the ball within the chosen game engine, Unity. This design allowed a game to be created around the subject which would help replicate the balls movement whilst also providing a platform for users to test the simulation in a gaming environment. Once a design using the appropriate methodology was created, the development of the simulation could start. The development of the simulation created a game with five different levels where the user could attempt to kick the ball through a set of goalposts from different positions of increased difficulty. The user was able to change the direction, pitch and power applied to the ball to make the ball fly. A set of participants were able to test the finished product and asked to fill out a questionnaire about their thoughts and opinions related to the product and any improvements that they would like to be made to future editions. This means that all of the objectives set out in the introduction of the project were met.

This project, whilst it met the objectives set out at the start of the project, is still hopefully the first of many different iterations that will improve over time, increasing the functionality and difficulty for the user whilst also better replicating what happens in the real world of a rugby game.
5.2 Future Work

Despite the project being complete, there are still future additions and improvements that could be made to the solution thanks to the responses from the participants when testing and from the researchers own ideas. One of the main improvements that need to be made to the project is the graphical representation of the project. Whilst this simulation focussed on the physics and practicality of the ball, the image of the project let the project down. With improvements to the graphics, the realism of the game for users would increase and it would make the game itself more immersive. Features such as stands and the crowd could also be added as part of this. Also, as mentioned in the literature review, one of the drawbacks to previous rugby titles were the graphics so to improve these would be the next logical step. A potential step to this would be to change the game engine to one with models with a higher resolution or to introduce new models which have been built more fluidly than the ones currently in use.

The next addition that could be made to the simulation is looking at the flight of the ball when kicked from hand. When kicking the ball from hand, there are many more different possibilities that can occur such as kicks along the ground, kicks to touch and drop goals. This would also open the opportunity for technical kicks within the rugby game such as box kicks from the back of a contact area, or the spiral kick which is used for increased length and accuracy down the field of play. This is an important feature for any future rugby game titles to look at with the increasing importance of kicking the ball from open play.

Another aim towards improving the simulation is to look at where the ball is being struck to provide different results. For example, hitting the ball at the top on the point rather than in the centre would force the ball to go across the ground, which relates to the previous improvement in allowing more complicated kicks to occur. To make this happen, improvements to the UI system, such as an image of the ball where the user could mark where the force will be applied, will be needed to specify where the force is applied to the ball and the script would need to be amended so that the new UI would be able to interact with the ball object.
An opinion brought up from the responses to testing was the introduction of external forces to make the act of kicking the ball harder. One way this could be introduced is through the use of wind acting in a random direction on each kick. Visual representation would be needed to show the user which way the wind is acting and a randomised way of exacting a constant force could also be applied which would be able to replicate this. The forces of the wind will also need to correlate with the system and strength of wind in the real world so it is not too hard for the user whilst also replicating world conditions.

As the project focuses on collision physics, the final improvement could be the next logical step once the kicking and landing of the ball has been correctly simulated. Looking at the collisions between rugby players themselves would be key to further success as this concept is a major part of the game. However, this would be much more complex than measuring the collisions of rugby balls due to the shape of the human body and the way it reacts when hit in different areas under different forces. The researcher would have to look into another form of game physics called ragdoll, which interprets how the human body acts under different forces and how these could be modelled effectively to the users liking.
Chapter 6: References


Anonymous, (2016) Primary Market Research Available:


Epic Games (1998) Unreal Engine, video game engine, PC.


Kokiashvili, T (2013), *Ball Smasher Starter Kit*, game object models, PC.


Newton, I. (1729)"In [experimental] philosophy particular propositions are inferred from the phenomena and afterwards rendered general by induction": "*Principia*", Book 3(2), General Scholium, p.392.


Tru Blu Entertainment 2016, Rugby Challenge 3, video game, PC.

Valve 2003, Steam, video game management, PC.


Unity Technologies 2005, Unity, video game engine, PC.
Appendix A: Ethics Form and Approval Number

Approval Number:

st20064513    Curtis    Daniel    Comp PA    2016D0197

Ethics Form

CARDIFF METROPOLITAN UNIVERSITY
APPLICATION FOR ETHICS APPROVAL

When undertaking a research or enterprise project, Cardiff Met staff and students are obliged to complete this form in order that the ethics implications of that project may be considered.

If the project requires ethics approval from an external agency (e.g., NHS), you will not need to seek additional ethics approval from Cardiff Met. You should however complete Part One of this form and attach a copy of your ethics letter(s) of approval in order that your School has a record of the project.

The document Ethics application guidance notes will help you complete this form. It is available from the Cardiff Met website. The School or Unit in which you are based may also have produced some guidance documents, please consult your supervisor or School Ethics Coordinator.

Once you have completed the form, sign the declaration and forward to the appropriate person(s) in your School or Unit.

Participant recruitment or data collection MUST NOT commence until ethics approval has been obtained.

PART ONE

<table>
<thead>
<tr>
<th>Name of applicant:</th>
<th>Daniel Curtis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisor (if student project):</td>
<td>Dr. Paul Angel</td>
</tr>
<tr>
<td>School / Unit:</td>
<td>Cardiff School of Management</td>
</tr>
<tr>
<td>Student number (if applicable):</td>
<td>st20064513</td>
</tr>
<tr>
<td>Programme enrolled on (if applicable):</td>
<td>BSc (Hons) Computing</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Developing Collision Physics in a Simulated Rugby Video Game</td>
</tr>
<tr>
<td>Expected start date of data collection:</td>
<td>10/03/2017</td>
</tr>
</tbody>
</table>
Approximate duration of data collection: 6 weeks

Funding Body (if applicable): None

Other researcher(s) working on the project: None

Will the study involve NHS patients or staff? No

Will the study involve human samples and/or human cell lines? No

<table>
<thead>
<tr>
<th>Does your project fall entirely within one of the following categories:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper based, involving only documents in the public domain</td>
<td>No</td>
</tr>
<tr>
<td>Laboratory based, not involving human participants or human samples</td>
<td>No</td>
</tr>
<tr>
<td>Practice based not involving human participants (eg curatorial, practice audit)</td>
<td>No</td>
</tr>
<tr>
<td>Compulsory projects in professional practice (eg Initial Teacher Education)</td>
<td>No</td>
</tr>
<tr>
<td>A project for which external approval has been obtained (e.g., NHS)</td>
<td>No</td>
</tr>
</tbody>
</table>

If you have answered YES to any of these questions, expand on your answer in the non-technical summary. No further information regarding your project is required.

If you have answered NO to all of these questions, you must complete Part 2 of this form

In no more than 150 words, give a non-technical summary of the project

The project will involve developing a physics engine where the physics of a rugby ball and human collisions are displayed through the use of a virtual rugby simulation which a user can interact with. Research will be carried out into previous work surrounding the topic of ragdoll physics within games and current iterations of rugby games. A new physics engine will then be developed from the results of this research and the artefact that is developed will be tested and evaluated by users for its comparisons to real life collisions in the sport of rugby.
DECLARATION:
I confirm that this project conforms with the Cardiff Met Research Governance Framework

I confirm that I will abide by the Cardiff Met requirements regarding confidentiality and anonymity when conducting this project.

STUDENTS: I confirm that I will not disclose any information about this project without the prior approval of my supervisor.

Signature of the applicant: Daniel Curtis
Date: 06/03/2017

FOR STUDENT PROJECTS ONLY

Name of supervisor: Paul Angel
Date: 06/03/2017

Signature of supervisor: Paul Angel

Research Ethics Committee use only

Decision reached: Project approved
Project approved in principle
Decision deferred
Project not approved
Project rejected

Project reference number: Click here to enter text.

Name: Click here to enter text.
Date: Click here to enter a date.

Signature:

Details of any conditions upon which approval is dependant: Click here to enter text.

PART TWO

A RESEARCH DESIGN

A1 Will you be using an approved protocol in your project? No

A2 If yes, please state the name and code of the approved protocol to be used N/A

A3 Describe the research design to be used in your project

The researcher aims to gather data from the system created on the end result of simulations. The results should show whether the testing has replicated the action of a real life rugby ball appropriately. The pragmatism philosophy approach will be used to carry out the testing as testing will occur in a number of different ways.

---

1 An Approved Protocol is one which has been approved by Cardiff Met to be used under supervision of designated members of staff; a list of approved protocols can be found on the Cardiff Met website here
A sample of at least 10 participants who are currently studying at Cardiff Metropolitan University will also be used to analyse the finished project and assess on the realism of the simulation and whether improvements can be made. A questionnaire will be handed out to the participants and it will be completed after testing the software. There will be a minimum of 8 questions for the participants to answer to gain a large amount of valid data. The questionnaire should take no longer than 15 minutes to complete, including testing of the software. The sampling of participants will use the convenience technique as it will depend on the date of finishing the software. The snowballing technique may also be used if the software is recommended by the original sample.

The quantitative data gathered from the testing of the software and the questionnaires will be shown on a spreadsheet using tables and charts. Any qualitative data gathered will be assessed by the researcher and used for recommendations for further improvements of the software. A deductive approach to the data will be used as the qualitative data is a smaller part of the testing. Data will be grouped using an exploratory framework and then coding of the data will be used to find similar patterns among the responses.

Participants will be chosen to not discriminate against age, race, gender etc. All participants will be above the age of 18 and will remain anonymous throughout testing. They will be able to pull out of the research at any time without penalty.

Submission of a completed questionnaire will be taken as informed consent for the given data to be used in this study. All questionnaire participants will be made aware of this. In addition, all participants will be notified that the researcher will have sole access to the data collected and that the data will be stored in a safe and secure manner. Data used in the written report will be anonymised and the participants will be coded to safeguard from identification.

<table>
<thead>
<tr>
<th>A4 Will the project involve deceptive or covert research?</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A5 If yes, give a rationale for the use of deceptive or covert research</td>
<td>N/A</td>
</tr>
<tr>
<td>A6 Will the project have security sensitive implications?</td>
<td>No</td>
</tr>
<tr>
<td>A7 If yes, please explain what they are and the measures that are proposed to address them</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**B PREVIOUS EXPERIENCE**

**B1 What previous experience of research involving human participants relevant to this project do you have?**

The researcher has not done research involving human participants before.

**B2 Student project only**

**What previous experience of research involving human participants relevant to this project does your supervisor have?**

Dr Paul Angel has 8+ years of previous research involving human participants.
<table>
<thead>
<tr>
<th>C POTENTIAL RISKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 What potential risks do you foresee?</td>
</tr>
<tr>
<td>1. When using the software, participants may struggle to understand the results obtained and therefore be unable to give thoughts and opinions on the project.</td>
</tr>
<tr>
<td>2. Participants may be concerned about anonymity.</td>
</tr>
<tr>
<td>3. Participants may be concerned about the use of the data procured from the questionnaire.</td>
</tr>
<tr>
<td>4. Concerns over the timescale of development of software, testing and research phases.</td>
</tr>
<tr>
<td>C2 How will you deal with the potential risks?</td>
</tr>
<tr>
<td>1. The researcher will make sure that questions are appropriate for a variety of different participants. In addition, participants will be given a “Participation Information Sheet” prior to completing the questionnaire.</td>
</tr>
<tr>
<td>2. Participants will be informed that the data collected will be anonymised and stored in a safe and secure manner.</td>
</tr>
<tr>
<td>3. Participants will have the option to withdraw from the research at any time without penalty. Any data collected from them will be destroyed.</td>
</tr>
<tr>
<td>4. A detailed timescale plan will be created to give enough time to each section. A Gant chart will be created to keep a record of this.</td>
</tr>
</tbody>
</table>
PARTICIPANT INFORMATION SHEET

Developing Collision Physics in a Simulated Rugby Video Game

Cardiff Metropolitan University Protocol Number: 2016D0197

Project summary
The purpose of this research project is to establish Developing Collision Physics in a Simulated Rugby Video Game. Your participation will enable the collection of data which will form part of a study being undertaken at Cardiff Metropolitan University.

Why have you been asked to participate?
You have been asked to participate because you fit the profile of the population being studied; that is you are between the ages of 18+.

Project risks
The research involves the completion of a questionnaire which will be recorded for later analysis. We are not seeking to collect any sensitive data on you; this study is only concerned with developing collision physics in a simulated rugby video game. We do not think that there are any significant risks associated with this study. However, if you do feel that any of the questions are inappropriate then you can stop at any time. Your participation is entirely voluntary and you may withdraw at any time without penalty.

How we protect your privacy
All the information you provide will be held in confidence. We have taken careful steps to make sure that you cannot be directly identified from the information given by you. Your personal details (e.g. signature on the consent form) will be kept in a secure location by the research team. When we have finished the study and analysed all the information, the documentation used to gather the raw data will be destroyed except your signed consent form which will be held securely for 5 years. All data will also be held in a secure and confidential environment during the study and destroyed after 5 years.

YOU WILL BE OFFERED A COPY OF THIS INFORMATION SHEET TO KEEP

If you require any further information about this project then please contact:

Daniel Curtis
st20064513@outlook.cardiffmet.ac.uk

Dr Paul Angel (Project supervisor)
PNAngel@cardiffmet.ac.uk
Appendix C: Participant Questionnaire

1.) Age:

18-30 □  31-40 □  41-50 □  50+ □

2.) Gender:

Male □  Female □

3.) On a scale of 1-10, 1 being completely fictional and 10 being lifelike, how would you rate the realism of the simulation?

1  2  3  4  5  6  7  8  9  10

4.) On a scale of 1-10, 1 being very slow and 10 being very quick, how would you rate the speed of this simulation?

1  2  3  4  5  6  7  8  9  10

5.) On a scale of 1-10, 1 being unusable and 10 being very easy to use, how would you rate the user interface of this simulation?

1  2  3  4  5  6  7  8  9  10

6.) Are there any improvements that you would like to be made to the simulation?

…………………………………………………………………………………………………………………………………………………………………

7.) Any other comments?

…………………………………………………………………………………………………………………………………………………………………

Thank you for completing this questionnaire. Submission of a completed questionnaire is taken as informed consent.
Appendix D: Questionnaire Response

Q3 - On a scale of 1-10, 1 being completely unrealistic and 10 being lifelike, how would you rate the realism of the simulation?

Q4 - On a scale of 1-10, 1 being very slow and 10 being very quick, how would you rate the speed of the simulation?

Q5 - On a scale of 1-10, 1 being unusable and 10 being very easy to use, how would you rate the usability of the simulation?
Q6 - Are there any improvements that you would like to be made to the simulation?

Are there any improvements that you would like to be made to the simulation...

Maybe introduction of outside factors that could influence the ball making the game more difficult e.g. Wind

More levels

Types of weather

The ability to choose where to shoot the ball from.

Showing Records: 1 - 4 Of 4

Q7 - Any other comments?

Any other comments?

Great game!

A very good game that seems accurate

Showing Records: 1 - 2 Of 2
Appendix E: Code for Ball Script

```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class HitBall : MonoBehaviour {
    public float power = 50f;
    private int score;
    private Text scoreLabel;
    private int level = 1;

    public void kickBall() {
        GetComponent<Rigidbody>().isKinematic = false;
        GetComponent<Rigidbody>().AddForceAtPosition(transform.forward * power, transform.position);
    }

    public void OnPowerChanged (float newPower) {
        power = newPower * 10;
    }

    public void Reset() {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }

    void OnTriggerEnter(Collider ball) {
        score = score + 1;
        level = level + 1;
        scoreLabel = GameObject.Find("ScoreNoLabel").GetComponent<Text>();
        scoreLabel.text = (score.ToString());
        SceneManager.LoadScene("Pitch " + (level.ToString()));
    }

    private void Update() {
        if (Input.GetKeyDown(KeyCode.LeftArrow)) {
            GetComponent<Rigidbody>().transform.Rotate(0, 5, 0);
        }
        if (Input.GetKeyDown(KeyCode.RightArrow)) {
            GetComponent<Rigidbody>().transform.Rotate(0, -5, 0);
        }
        if (Input.GetKeyDown(KeyCode.UpArrow)) {
            GetComponent<Rigidbody>().transform.Rotate(5, 0, 0);
        }
        if (Input.GetKeyDown(KeyCode.DownArrow)) {
            GetComponent<Rigidbody>().transform.Rotate(-5, 0, 0);
        }
    }
```
Appendix F: External Materials

A number of external materials were used to help produce this program

- Unity 5.6.0f3 game engine used to create the simulation
- Visual Studio used to write the script
- Ball Smasher Starter Kit 1.1 used to bring in the working model of a rugby ball