**University of Derby**

**School of Electronics, Computing & Mathematics**

**A project completed as part of the requirements for BSc (Hons) Computer Games Programming**

**entitled**

**The impact of degradation as a game feature and how it can affect gameplay in the Unreal Engine 4**

**by**

**Aidan James Feather**

**100401863@unimail.derby.ac.uk**

**May 2020**

Abstract

Acknowledgements

Table of Contents

[Abstract 2](#_Toc40130771)

[Acknowledgements 3](#_Toc40130772)

[Table of Contents 4](#_Toc40130773)

[1. Introduction 7](#_Toc40130774)

[1.1. Project Rationale 7](#_Toc40130775)

[1.2. Project Aim and Objectives 7](#_Toc40130776)

[1.2.1. Project Aim 7](#_Toc40130777)

[1.2.2. Project Objectives 7](#_Toc40130778)

[2. Literature Review 8](#_Toc40130779)

[2.1. Introduction 8](#_Toc40130780)

[2.2. Item Degradation 8](#_Toc40130781)

[2.2.1. Balance 8](#_Toc40130782)

[2.2.2. Usage in different genres 8](#_Toc40130783)

[2.3. Environment degradation 9](#_Toc40130784)

[2.3.1. The effect on gameplay 9](#_Toc40130785)

[2.3.2. Achieving environmental degradation 9](#_Toc40130786)

[2.4. Vehicle degradation 9](#_Toc40130787)

[2.4.1. Comparison of different types 9](#_Toc40130788)

[2.4.2. Using the correct type for your game 10](#_Toc40130789)

[2.5. Conclusions 10](#_Toc40130790)

[2.5.1. Key Issues 10](#_Toc40130791)

[2.5.2. Refined Research Questions 11](#_Toc40130792)

[3. Research Methodology 12](#_Toc40130793)

[3.1. Introduction 12](#_Toc40130794)

[3.2. Research Strategy 12](#_Toc40130795)

[3.3. Data Generation Methods 12](#_Toc40130796)

[3.4. Data Analysis 12](#_Toc40130797)

[3.5. Sampling 12](#_Toc40130798)

[3.6. Ethics 12](#_Toc40130799)

[3.7. Limitations 12](#_Toc40130800)

[3.8. Conclusions 12](#_Toc40130801)

[4. Project Design 13](#_Toc40130802)

[4.1. Introduction 13](#_Toc40130803)

[4.2. Requirements Specification 13](#_Toc40130804)

[4.3. Experiment Design 13](#_Toc40130805)

[4.4. Test Schedules 14](#_Toc40130806)

[4.5. Conclusion 14](#_Toc40130807)

[5. Findings and Analysis 15](#_Toc40130808)

[5.1. Introduction 15](#_Toc40130809)

[5.2. Analysis 15](#_Toc40130810)

[5.3. Conclusions 15](#_Toc40130811)

[6. Discussion 16](#_Toc40130812)

[6.1. Introduction 16](#_Toc40130813)

[6.2. Issue 1 16](#_Toc40130814)

[6.3. Issue 2 16](#_Toc40130815)

[6.4. Conclusion 16](#_Toc40130816)

[7. Conclusions and Recommendations 17](#_Toc40130817)

[7.1. Conclusions 17](#_Toc40130818)

[7.1.1. Conclusion 1 17](#_Toc40130819)

[7.2. Recommendations 17](#_Toc40130820)

[7.2.1. Recommendation 1 17](#_Toc40130821)

[8. References 17](#_Toc40130822)

[9. Appendices 18](#_Toc40130823)

# Introduction

## Project Rationale

This project is to be developing a new degradation system for a first person shooter game that I am developing. I have chosen this as the system that are currently in place for these types of games are used for at a plateau and I wish to change this. This system will be fully interactive with a basic weather system, that will affect weapons, armour and enemies in different ways.   
  
This needs to be addressed as the games that implement this system have a very basic version that is almost the same, apart from a few stand out games. This system can either enhance or destroy a game, so I aim to enhance the playability of my game by making this feature stand out but not be too intrusive, as a new system can enhance the survival genre of games, where this feature thrives.

## Project Aim and Objectives

### Project Aim

To develop a basic first-person shooter in the Unreal 4 game engine, as well as implementing a degradation system. I will also be comparing this to other games on the market, discussing the benefits of my system. This game will have a basic story as well as weather effects, multiple weapons and different types of enemies. This system will also have the end goal of include a vehicle degradation system, simulating damage and weather effects on a vehicle.

### Project Objectives

1. To develop a first-person shooter with a basic story and multiple levels

2. To implement a degradation system that can affect weapons and the environment. This will also include the weather affecting weapons, with the possibility of expanding into a vehicle degradation system.

3. To compare this system to ones available in other games

4. To discuss the benefits of this system over already established ones

# Literature Review

## Introduction

In this literature review, I will be reviewing the core concepts that are the focus of my dissertation. These concepts include item degradation, environmental degradation and vehicle degradation.

## Item Degradation

### Balance

Balance is a key part of a degradation mechanic. When these mechanics are balanced, you need to take into consideration whether they fit with the genre that the game is being developed for. This can make the difference between making the game appealing and non-appealing for the audience. For example, some professional reviewers feel that the degradation system in Legend of Zelda: Breath of the Wild is too impairing to the flow of combat, when paired with the User Interface in that game (Sterling, 2017).

### Usage in different genres

Item degradation has been used in many different genre of games, however it has been used most commonly in the survival and roleplaying game genres. Examples of this feature include Fallout 3, Fallout New Vegas as well as game such as the Witcher 3 and Days Gone. These have been used to various effects and are especially as prevalent in the games that have a post apocalyptic style as it is used not only to create different kinds of gameplay but to emphasis the setting.   
When this feature is used in Breath of The Wild, the weapons break quite frequently. The game also does not allow you to repair any of the weapons that break, which puts an emphasis on frequently picking up new weapons (Sterling, 2017). This is in stark contrast to a game such as Fallout New Vegas, which allows you to repair weapons with resources found in the game. This allows you to keep the same weapon throughout the game, if you have the required resources to repair it. This means that if you find a rare weapon, it can be kept whereas Breath of the Wild doesn’t allow you to keep these weapons.

## Environment degradation

### The effect on gameplay

Environmental degradation, which can also be referred to as environment destruction, is an expansion of the previously mentioned game mechanic. This allows for am effect on gameplay which can change the game dramatically. A key game that emphasises this is Red Faction: Guerrilla and the Battlefield series of games (Veloria, 2017). Red Faction: Guerrilla allows for the destruction of nearly everything, apart from the terrain. This allows for different tactics that give the game the appeal to replay. You can play it like a 3rd person shooter or you can drop a build on the target with explosives. This makes the game appealing and more fun as it gives the player more choice. (Rock Paper Shotgun, 2019)

### Achieving environmental degradation

One of the ways to achieve environmental degradation is to use destructible meshes for objects (Docs.unrealengine.com, 2019). If you use a variable to give the object a health rating, once this has depleted, it can trigger an event that will destroy the mesh. This can give a realistic feel to the game, as well as make the game more enjoyable, as you find new ways to approach a level. These objects can be set to destroy on contact, when attacked enough times, as well as making the hurt enemies, or yourself, if they ragdoll from an explosion. (Wiki.unrealengine.com, 2019)

## Vehicle degradation

### Comparison of different types

Vehicle degradation comes in many different types. Vehicle simulation has been used in many different driving games. These can simpler physics that simulate the effect of racing on tires, such as heat and tire damage, as well the effect of crash on the chassis and the overall performance of the car.

Crash damage on vehicle is present in nearly every game that has a driving mechanic. If I were to compare three video games, these would be Grand Theft Auto 5 and Forza Horizon 4. These games all have crash physics but approach them in different ways due to the kind of games they are, as well as different limitations to the game engine.

Grand Theft Auto 5 uses a very mild form of damage physics as this not the focus of the game. These physics are effect by the damage taken via different ways, as you can incur them from crashing or taking firearms damage. They are fixed, as when the car takes enough damage, the performance drops, not matter where the damage is taken, although the cosmetic damage is dependant on where the damage is coming from.

Forza Horizon 4 is a more realistic version of the damage model (Johnson and Frederiksen, 2019). This model uses a physics model that can affect the wear on the tyres, as well as crash damage from collisions. This is on the visible model. The tyre can increase and decrease in temperature, which can change the level of grip the tyres have, as well as simulate damage to the various components of the car, such as the engine, clutch, gearbox, suspension, etc. This can make for a more challenging race as the player would have to be careful not to damage the car too much as it could lose them the race. This forces a change in driving style and more tactical racing. A critical point is this degradation mechanic can be turned off as well, which makes the game appeal to people who want a more genuine driving experience and to people who want a game that has a more arcade style. (Towell, 2019)

### Using the correct type for your game

The correct type of vehicle degradation to use is a critical choice. It has to be the correct type for the genre of game that you are creating. If it is a racing game, then a more detailed and simulator-like degradation model should be used. However, as I am creating a first-person shooter it would be more appropriate to use a more simplistic model. This would give the vehicle a more simplistic health rating and would reduce performance and give cosmetic damage where appropriate. This means that there is less code and calculations being devoted to a less important part of the game.

## Conclusions

### Key Issues

One of the key issues that I am encountering is how I can implement a degradation system that keeps gameplay balance and fair without making it a background feature that is not important. One of the many was that I would do this is to implement a scaling system that would allow the players to modify how much degradation there is, without switching it off. This means it would be appealing to a wider audience, as if people wanted a hyper-realistic game they could have it, but if they wanted a nicer experience then they could turn the degradation system down.   
  
To implement it on things like weapons and armour, then I could have each equip-able item have a “health” bar that would wear down with either usage (in the case of weapons) or damage taken (such as armour and the environment. This system could be further modified. So that if I added different, rarer ammunition for the weapons, they could do significant more damage to different armour or enemies, at the cost of increase degradation.   
  
Another key issue is detailed damage models and if I would implement them. These damage models would require potential modifying and take a lot of time to implement. These may require things such as destructible meshes, which require some work to implement. In the case of the environment, I would have to decide whether I wanted the whole environment destructible or just key parts.

Optimisation is a key issue. If this game is to implement an environment degrading system, then how sever will the impact on performance be? The optimising of the game is key as it needs to be able to run on lower end systems to make sure that it can have the best reach to customers.

### Refined Research Questions

The main questions that now need addressing are:

How destructible is the environment?

How severe is the degrading system on the items?

How demanding would this feature make the game?

# Research Methodology

## Introduction

This section discusses the methodology that I have used to research different variants of weapon degradation in video games.

## Research Strategy

I have search through articles, journals and forum posts on the internet.

## Data Generation Methods

I have compared different articles and forums across the internet to research methods of implementation as well as ideas on what to include in the durability system. This a quick way to gather opinions on these systems as they are not as common in video games nor do they tend to be a focus.

## Data Analysis

The data that I have found shows that most systems are used to create a new function

## Conclusions

# Project Design

## Introduction

For this dissertation, I have created a durability mechanic that has been the focus of the technical demonstration. This mechanic focuses on bringing a health mechanic for the weapons that are implemented, which will be useful for survival simulator/video games. This has been based of a template that is included with the Unreal 4 Engine.

## Requirements Specification

This project must include a basic firing system, targets and the weapon durability system. The durability system must be proven to work on at least one weapon, with the configuration to work in and out of the biome boxes that are included. The biomes are used to modify the durability mechanic for the weapons. The biomes will modify the rate in which the durability will drop, usually increasing it. The design must have the expandability to add numerous weapons into the mechanic, with compromising it. This all must run smoothly in the Unreal 4 Engine.

## Experiment Design

To start with the project, I based it on the first-person C++ template that comes with the Unreal Engine. I did this as it gives a fully functional and understandable basis for the durability system to run from. I decided on this as the focus of the project is the durability mechanic, not the shooting mechanics. This saves valuable time, as well as being a template a lot of people know, due to it being included with Unreal.

In the first version of this project, I set up a new basis for the weapon degradation system, this was done by setting the contents OnFire() method in a while loop. To control this while loop, I declared three new variables, WeaponHealth, WeaponDegrade and WeaponTotal. Weapon Health was the initial weapon health and WeaponDegrade was the rate in which the weapon would degrade. The WeaponTotal was the final value of the weapons health, when this runs out, the OnFire() method would cease to function.

The biomes are set up as Trigger Boxes. When walking into a specific biome, the trigger box runs a small bit of code that overwrites the current WeaponDegrade value, allowing for a higher or lower rate of degradation, depending on the biome that you have entered/exited.

In the second version of the project, I changed the degradation system dramatically due to problems that I was having getting a Heads Up Display (HUD) to work with former layout of code.

When modifying the code so I could implement a HUD, I created more UFUNCTIONs for most of the code to be executed by.

To link the variables that display the values for the Health and Durability systems, I have used declared the variables as a UPROPERTY. This allows me to give the variables certain properties. The most common ones that I have used are EditAnywhere, BlueprintReadWrite and Category. The EditAnywhere property allows for these variables to be edited in any property window that they appear in, which allows for these to be edited without opening the code.   
The BlueprintReadWrite property allows for the variable to be both read and modified by any Blueprint in which they are called.  
The Category property will organise these variables in the property tabs in the Unreal Editor. This is beneficial for finding these variables in case you need to edit them.

To create a HUD I have used the Blueprints that are included with Unreal 4. Blueprints are visual scripting system that are used more for beginners. By using Blueprints, it means that I have an easier editing tool the modify the visual side of the HUD.

## Test Schedules

## Conclusion

# Findings and Analysis

## Introduction

In this section I will discuss the findings that I have had over the course of the developing this project. I will be comparing it to different versions in different games on the Unreal 4 engine as well as outside of the Unreal 4 engine.

## Analysis

## Conclusions

# Discussion

Discussion about issues and evolving. Comparing this and current games.

## Introduction

## Issue 1

## Issue 2

## Conclusion

# Conclusions and Recommendations

## Conclusions

### Conclusion

## Recommendations

### Recommendation

# References

TV Tropes. (2019). Breakable Weapons - TV Tropes. [online] Available at: https://tvtropes.org/pmwiki/pmwiki.php/Main/BreakableWeapons [Accessed 18 Nov. 2019].

Sterling, J. (2017). [online] Thejimquisition.com. Available at: https://www.thejimquisition.com/post/the-legend-of-zelda-breath-of-the-wild-review-broken-sword [Accessed 18 Nov. 2019].

ComicsVerse. (2019). Is Equipment Durability Always a Bad Thing? - ComicsVerse. [online] Available at: https://comicsverse.com/equipment-durability/ [Accessed 18 Nov. 2019].

Medium. (2017). Weapon Durability in Breath of the Wild: A Game Design Analysis. [online] Available at: https://medium.com/@JohnnyUzan/weapon-durability-in-breath-of-the-wild-botw-has-been-a-source-of-controversy-among-gamers-ever-675e9c1bdae0 [Accessed 25 Nov. 2019].

Veloria, L. (2017). 12 games that do destructible environments right. [online] gamesradar. Available at: https://www.gamesradar.com/games-incredible-destructible-environments/ [Accessed 5 Dec. 2019].

Wilson, E. (2019). A History Of Environmental Destruction. [online] The Games' Edge. Available at: https://thegamesedge.com/602/a-history-of-environmental-destruction/#.YOGqyQhf8Z [Accessed 5 Dec. 2019].

NVIDIA Developer. (2019). PhysX Destruction. [online] Available at: https://developer.nvidia.com/destruction [Accessed 12 Dec. 2019].

Rock Paper Shotgun. (2019). Red Faction: Guerilla’s destructible scenery makes it still worth playing today. [online] Available at: https://www.rockpapershotgun.com/2017/04/25/red-faction-guerillas-destructible-scenery-makes-it-still-worth-playing-today/ [Accessed 12 Dec. 2019].

Docs.unrealengine.com. (2019). Destructibles Content Examples. [online] Available at: https://docs.unrealengine.com/en-US/Resources/ContentExamples/Destructables/index.html [Accessed 12 Dec. 2019].

Wiki.unrealengine.com. (2019). Destructible Troubleshooting Guide - Epic Wiki. [online] Available at: https://wiki.unrealengine.com/Destructible\_Troubleshooting\_Guide [Accessed 12 Dec. 2019].

Johnson, R. and Frederiksen, E. (2019). Forza Horizon 4 reviewed - The Tech Report. [online] The Tech Report. Available at: https://techreport.com/review/34234/forza-horizon-4-reviewed/ [Accessed 18 Dec. 2019].

Towell, J. (2019). How Forza Horizon 4 addresses the series’ three major problems. [online] gamesradar. Available at: https://www.gamesradar.com/uk/how-forza-horizon-4-addresses-the-series-three-major-problems/ [Accessed 18 Dec. 2019].

# Appendices