Ballistics Simulation Within Arma 3

Interim Report

Submitted for the BSc in
Computer Science

January 2017

by

Sean Andrew Sidney Phillips

Word Count: 5193

Table of Contents

1 Introduction

Arma 3 is a military tactical shooter video game developed by Bohemia Interactive. The game takes place on the fictional Aegean islands of Altis and Stratis, with the former featuring 270 Km^2 of landmass and the latter 20 Km^2. Arma 3 consists of over 40 weapons, 20 vehicles and a multitude of other miscellaneous objects, such as clothes, buildings and ammunition crates – all contributing to generating an immersive and authentic experience. This, along with the realistic gameplay allowed by the use of the Real Virtuality 4 engine, creates a gaming environment closely following that of real modern combat.

In the modern age, tanks and other armoured vehicles play an essential role in the dynamics of combat and, due to Arma 3's aim of simulating modern combat, this follows suit in the gameplay of Arma 3. However, Arma 3's model of simulating armoured warfare is currently lacking in the accuracy of its simulation of projectile ballistics, adversely affecting the gameplay. This leaves a jarring experience when switching from more well fleshed out aspects of the game, such as the infantry combat. As Arma 3 takes pride in it authentic gameplay, "Experience true combat gameplay in a massive military sandbox. Authentic, diverse, open - Arma 3 sends you to war" (Bohemia Interactive, 2017), this only further exasperates the problem for the gaming community that regularly play Arma 3.

There is a considerable following of organised groups of players who get together and play Arma 3. This is how Arma 3 is reputed, by many, to be best played, "Join one of 1911 units and experience Arma 3 at its finest" (Bohemia Interactive, 2017). These groups usually play the game by mimicking real life combat tactics. Players are often split into teams with leaders and a hierarchy at play, all collaborating in an organised manner to defeat a common enemy, be that another player team or AI bots in user made missions. Many of these groups take the game very seriously - even calling themselves "MilSim" (military simulation) groups and naming themselves after regiments that exist in today's armies. They enjoy a high level of simulation and use mods such as "Advanced Combat Environment 3" and "Advanced Combat Radio Environment" to enhance the detail of simulation within Arma 3 to fit and mold the gameplay of Arma 3 to their own needs.

Arma 3 also has a substantial modding community that adds extra content, new features to compliment the realistic gameplay that Arma 3 provides and expand the existing features to a higher level of simulation. For example, the popular mod "Advanced Combat Environment 3", augments the infantry combat by revamping the medical system to a higher level of realism. However, within the modding community, there is a significant lack of focus on the ballistics simlation of projectiles hitting armoured targets when compared to other aspects of the game. This project will attempt to fill that gap and provide a package that brings the level of simulation at least on par with the rest of the game, all the while remaining within the original vision that the developers had for Arma 3.

The overall goal for this project is not too provide an exact physics simulation of projectile ballistics, but to improve upon the existing model, bringing up the quality of simulation to the rest of the core features that exist within Arma 3.

2 Aims and Objectives

To help manage the progress of this project, several objectives will be created to provide a criterion upon which the final completed package can be compared against. This will allow for an easy evaluation to determine if the final product has fulfilled the original goal.

The overall aim for the project is “To improve the level of simulation of projectile ballistics in Arma 3”. To achieve this goal, the project will have to complete the following objectives and meet their criteria.

1. Retrieve from the game engine the properties of a projectile and its target after impact.
2. Interface with an external library and pass data between said external library and the game engine.
3. Parse the data outputted into an external library and store as variables for use within the library.
4. Calculate the penetrating ability of the projectile after impact and update the game engine appropriately.
5. Split armour values of the target vehicle based on the position of impact.
6. Create a module-based damage system where individual modules of the target vehicle can be damaged.

2.1 Objective 1 – Retrieve from the game engine the properties of a projectile and its target after impact.

When a projectile impacts any vehicle in the game an event is required which returns the properties of the target vehicle and projectile, this will be needed for later calculations to determine the post-impact properties of the vehicle and the projectile. These properties will then have to be stored as variables for later use. These will include, among others, velocity of projectile, angle of incidence between projectile flight path and armour plating, type of projectile fired, type of target vehicle and position of impact.

2.2 Objective 2 - Interface with an external library and pass data between said external library and the game engine.

To be able to use an extension for the bulk of calculations needed, an interface is required between the game engine and an external library. Communication between the extension and the game engine needs to be fast and efficient, with the minimal amount of time between projectile contact and possible penetration, of which the later calculations need to occur in.

2.3 Objective 3 – Parse the data outputted into an external library and store as variable for use within the library.

Data passed into the external library needs to be parsed into a readable form. Setting up this will require string manipulation to store the data passed into variables. These variables will then have to be stored as the relevant type corresponding to the nature of the data passed. This will provide a base for the calculations of the impact of a projectile that will occur in the later stages of this project.

2.4 Objective 4 - Calculate the penetrating ability of the projectile after impact and update the game engine appropriately.

Using the data stored within the external library, a calculation will determine whether the projectile is able to penetrate the target vehicle's armour. This will be done via a set of mathematical equations simulating the real-world mechanics of armour penetration after which, the game engine will appropriately update with the projectile's post impact values and those of the vehicle. These values will include, among others, the damage caused to the vehicle, projectile velocity and direction of projectile (if deflection has occurred).

2.5 Objective 5 - Split armour thickness values of the target vehicle based on the position of impact.

To properly simulate a real-world engagement, the armour thickness values of the target vehicle cannot be the same throughout the vehicle. The armour thickness of a combat vehicles is not often homogenous, generally a vehicle will have its thickest armour at the front and the weakest at the rear, with the side armour being somewhere in the middle. The same assumption can be applied to the turret of the target vehicle, if present, as well.

2.6 Objective 6 - Create a module-based damage system where individual modules of the target vehicle can be damaged.

To improve the level of simulation of the damage model of armoured vehicles, it is insufficient to only have the proper penetrations of the outside armour. In addition to the exterior penetration calculations an interior damage model, where individual key components of the vehicle can be damaged, is required. This will allow a more realistic approach to the damage caused upon a target vehicle after impact. These damageable components will include, but not limited to, ammunition stowage, engine, crew members and fuel tanks.

3 Background

3.1 Problem Context

3.1.1 Current Model of Projectile Ballistics Within Arma 3

Arma 3 uses a basic hit point based damage system to model the vehicle warfare, implicating vehicles and infantry firing projectiles at armoured target vehicles. The game uses two systems to determine the damage done to a vehicle, these are fire geometry and hit points. Fire geometry is an invisible physical model acting as the armour plating for the vehicle – this is used to calculate if a projectile has intersected with the vehicle model. Hit points are used to establish the global and individual modules health pool for a vehicle, also determining the location of damage, damage done and to what component of the vehicle.

Once an impact has occurred, the penetration and damage are not one and the same within the model that exists. An attack must simultaneously penetrate fire geometry and a hit point radius or an explosive radius from a projectile will intersect with a hit point spheres radius to cause damage. This is where problems can occur, as if the hit point spheres are laid out to where they extend outwards past the fire geometry, damage can be dealt without a projectile penetrating or even impacting a section of fire geometry. This will mainly occur with the use of explosive rounds as the explosive radius can intersect with a hit point sphere causing damage.

When the game engine comes to calculating the penetration of a shell against a piece of armour, it is very simplified – only using the shell velocity, caliber and the armour thickness. There is no usage of other material properties (such as density of the projectile / armour, shape of the projectile etc.) needed to fully represent the real world mechanics of projectile ballistics.

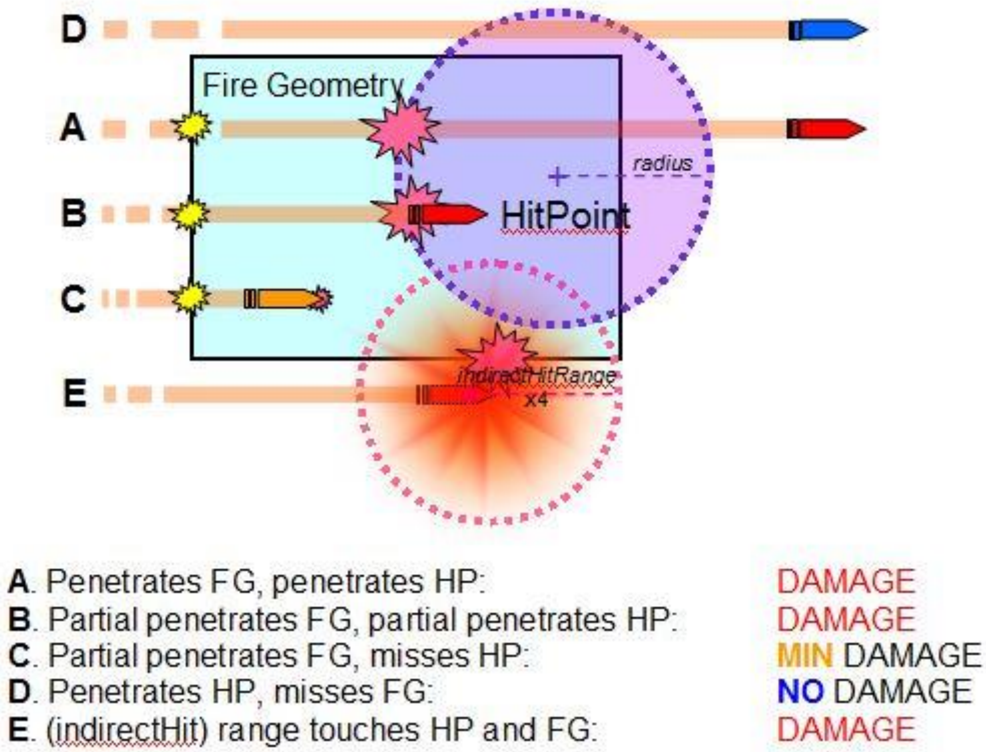

Figure 1. A diagram of a small magnified cutout of a vehicles armour and interior further explaining damage and penetration to fire geometry and hit points and how those damage the vehicle – (FG = Fire Geometry / HP = Hit Point), (Olds, 2015)

In figure 1 attacks A-D are direct hits and attack E is an example of an indirect hit. This is where problems can occur and the damage system does not simulate real life closely – a shell with a large indirect hit range can bypass the fire geometry and damage the hit point radius. The diagram above also shows that the fire geometry acts as a sort of armour, protecting the vehicle from direct attacks, however (as stated above) indirect fire can bypass the fire geometry and still cause significant damage to the vehicle.

3.1.2 Mechanics of Real-World Projectile Ballistics

To properly develop a solution for this project, one must first understand the real word mechanics of projectile ballistics and what can happen to a target vehicle that has been impacted by a projectile. The types of projectile that can threaten an armoured vehicle are usually separated into two main groups, specifically kinetic and chemical energy weapons.

Kinetic Energy Weapons

The most common category of kinetic energy weapons that exist within Arma 3 are long-rod penetrators, namely an APFSDS (Amour-Piercing-Fin-Stabilised-Discarding-Sabot) shell. This type of shell is mainly fired from a gun mounted on a main battle tank and rely upon kinetic energy alone to destroy an armoured target. An APFSDS shell is a long, thin rod of a dense material, usually a tungsten-carbide or depleted uranium alloy, with a larger, thicker and soft jacket that surrounds the rod core – allowing the shell to be fired from a barrel of a much larger diameter than the core. After leaving the barrel of a gun the dense core will eject the surrounding jacket and continue on it flight path towards the target.

Upon impact with an armoured vehicle, the shell will rely upon it kinetic energy to force its way through the armour. If it is successful at defeating the armour then the shell will generate heat and spalling, creating a pressure wave that will travel through the vehicle, destroying the target. If the projectile does not have sufficient kinetic energy to defeat the armour then the shell will either shatter upon impact or deflect away harmlessly.

Chemical Energy Weapons

Unlike kinetic energy weapons explained above, chemical energy weapons rely upon the release of energy from an explosion and the resulting affects to inflict damage on armour. These are represented within Arma 3 by the use of HEAT (High-Explosive-Anti-Tank) and HE (High Explosive) shells. Both of which are not only fired from armoured vehicles, but employed also by infantry, helicopters and airplanes.

HEAT shells operate on the fundamental basis that the Munroe Effect provides. This occurs when a high-explosive shell has a cavity and a metallic liner facing towards the target. Upon impact, the explosive charge within the shell will detonate, collapsing the metallic liner forming a high-velocity jet of molten metal (typically copper) that will penetrate the armour.

HE rounds rely solely upon the blast wave formed from the impact explosion and the subsequent flying fragments of the shell. Detonation will occur upon impact with a hard surface, due to an impact fuse located at the tip of the shell's body. The blast wave and fragments formed upon impact are often ineffective against well-armoured targets and will often fail to penetrate a vehicle armour. However, the shell will still generate a pressure wave within the target vehicle, even if no penetration has occurred, that can inflict damage upon the crew and essential components.

3.2 Comparison of Technologies

To create a simulation of projectile ballistics and their interactions with armoured vehicles there are a number of technologies that can be used to achieve this. This is in reference to the game engine that can be used, as there are alternatives with sufficient resources in addition to Arma 3. These technologies with their advantages and disadvantages from their implementation will be assessed as to confirm the right solution for this project.

3.2.1 Arma 3

The technology that is the most developed in the area of this project would be the Arma 3 engine. This encompasses a package that is already fully developed and incorporates a simplified simulation of projectile ballistics and vehicle combat along with the other features of the game that compliments this. However, having a partially developed solution can pose a number of problems along with the obvious benefits that it provides.

The advantages to using the Arma 3 game engine would include:

- A vehicle combat system is already in place within Arma 3. This implicates less development time spent on modelling in the basics of vehicle combat, such as driving and shooting, which would be out of scope of the project aims and objectives.
- There are various coding languages that can be used for development within Arma 3, offering a number of alternative methods if the software engineer is not familiar with a particular language.
- Arma 3 offers a number of testing functions to assist in black box testing. Vital development time can be spent in other, more important areas that are relevant to the projects objectives.
- Arma 3 is easily accessible being a popular PC video game sold from a multitude of sources.

On the other hand, developing with Arma 3 does not come with its faults. The disadvantages that may occur are:

- Arma 3 lacks any significant documentation and a multitude of experimental investigations into the game itself, to familiarise with the software's functions, will need to occur before any real development can begin.
- There are poor error checking systems in place when conducting white box testing; No stepping through code exists and script errors are often obscure and inaccurate.
- If the current projectile ballistics simulation is not in line with the projects goals, then development time will be wasted on workarounds to bypass the code that already exists.

3.2.2 Third Party Game Engine

A game engine developed by a third party can be considered for use with this project. There are many options to choose from, however the most popular and arguably considered the best to use would be the Unity and Unreal game engines. Using these engines would offer a number of benefits, most notably the ease of development. Multiple error checking systems are in place helping speeding up the development process.

Advantages from developing with a third party game engine include:

- These third party game engines are easily accessible, often being free if development is done without the intent of monetising the final product.
- They are very well documented with numerous tutorials and articles on the systematics of the software – less development time would be spent on learning the new software tools.
- When conducting white box testing there are numerous error checking tools available, making the process quick and efficient.

There are many disadvantages from developing with a third party game engine however:

- A basic vehicle combat system would have to be developed before the work can begin on the projects objectives, meaning vital development time wasted.
- Testing functions would have to be created to assist in black box testing.

3.2.3 VBS 3

VBS 3(Virtual Battlespace 3) is a virtual military battle simulator developed by Bohemia Interactive Simulations that is related to Arma 3 by being derived, at a basic level, from the same engine. Due to this, many of the advantages and disadvantages are the same as Arma 3 with only a few variances.

Advantages from using VBS 3 would be:

- Even more development would be done in relation to the project's goals, implying less time spent on bringing the current model up to par before real project work can commence.
- A multitude of languages are natively supported, not unlike Arma 3, allowing various alternative solutions to the project's objectives.
- A number of functions exists that can be of assistance when conduction black box testing.

Disadvantages would include:

- A significant lack of documentations, even when compared to Arma 3, due to VBS 3 being a closed piece of software only available to large institutions.
- Development tools are limited with few error checking systems in place, white box testing would be time consuming as a result.
- Limited access to software as it is only available to certain customers such as universities and those with military related backgrounds.

3.2.4 Decision

To decide the final technology to be used for development of this project a multitude of factors will be taken into consideration, such as performance and efficiency, ease of development and extent of documentation available. The engineers experience will also have to be taken into account along with those outlined in the above.

After considering the factors evaluated, the technology to be used with the development of this project will be Arma 3. The software engineer is familiar with this piece of software and has vital experience in development related to Arma 3. Arma 3 does come with its disadvantages however, such as a lack of documentation and error checking systems, but these are out-weighed by the engineer's familiarity with the software and development time that would be saved from not having to create a basic vehicle combat model – out of scope with the projects aims and objectives.

3.3 Comparison of Algorithms

The most notable algorithms to be used within this project will be the penetration probability at impact and the post-impact values of a projectile and its target. There has been significant research in the academic world within this field, all of which, are vastly different in the complexity and accuracy of the equations derived from this research.

The algorithms that have been chosen are equations derived from academic research. These include the DeMarre equation, an forumla for impact depth conceived by Sir Issac Newton and Project Thor, a technical report into projectile ballistics done by U.S Army contractors.

3.3.1 DeMarre Equation

The DeMarre equation was used by a multitude of countries militaries during and partially after World War 2. It estimates the penetration of projectiles against homogenous steel armour and is assumed that the impact angle is 0° and the projectile does not deform. The DeMarre equation only applies to kinetic energy projectiles and those with no explosive nature. The DeMarre formula is as follows:

$$t = d^n \sqrt{\frac{wv^2}{kd^3}}$$

Where w = weight of projectile, lbf

V = velocity of projectile, ft/s

d = diameter of projectile, in

t = thickness of armour penetrated, in

k = constant, conforming to the projectile and target materials properties

n = 1.4

The use of this equation would provide a relatively accurate simulation of projectile ballistics and yet remaining simple enough to be within the project scope, however it does come with a number of shortcomings. Firstly, there is no consideration for the angle of impact and crude calculations would be needed to apply this affect to the equation. Second, there are no values given for the residual velocity after impact. Finally, the shape of the projectile is not taken into account. However, the formula would be simple to transcribe to code, with minimal mathematical functions needed.

3.3.2 Impact Depth

Impact depth is a rough approximation for the depth of penetration by a projectile originally theorised by Sir Issac Newton. As this equation is only a rough approximation for penetration it does have numerous drawbacks and would be generally unsuitable for anything but the simplest of calculations. Only a limited set of properties are considered - material density of the target, area of the impact and length of the projectile. The equation does not consider vital properties important to the performance of a projectile such as the angle of impact, shape and velocity of the projectile. Most notably, this equation only holds true for blunt projectiles and target materials with no cohesion.

$$P \approx L \frac{Dp}{Dt}$$

Equation to estimate a projectiles penetration depth.

Where P = Penetration Depth, in

L = Length of Projectile, in

Dp = Density of Projectile, in

Dt = Density of Target material, in

3.3.3 Project Thor

Project Thor is a technical report conducted by U.S Army contractors in 1961. The report gathered and analysed perforation data for steel fragments impacting on metallic targets. The fragments used were small (less than 10cm in diameter) and had a length/diameter ratio of less than 3. Three equations were derived from the research, these were the striking velocity to just penetrate, the residual velocity and residual mass after penetration. The equations described are as follows.

$$Vr = Vs - 10^{c}\ (hA)^{\alpha}\ (Ms)^{\beta}\ (\sec\theta)^{\gamma}\ Vs^{\lambda}$$

$$Vo = 10^{c1}\ (hA)^{\alpha 1}\ (Ms)^{\beta 1}\ (\sec\theta)^{\gamma 1}\ Vs^{\lambda 1}$$

$$Ms - Mr = 10^{c}\ (hA)^{\alpha}\ (Ms)^{\beta}\ (\sec\theta)^{\gamma}\ Vs^{\lambda}$$

Where Vr = projectile residual velocity after penetration, in ft/s

Vs = projectile striking velocity, in ft/s

Vo = projectile striking velocity, just to penetration, in ft/s

h = target material thickness, in inches

A = impact area, in inches

Ms = weight of projectile, in grains

Mr = residual weight of projectile after penetration, in grains

Θ = angle of incidence between projectile flight path and target material normal, in degrees

α, β, γ, λ = constants depending on projectile and target material

Project Thor encompasses three highly detailed equations for determining the projectile and target post-impact properties. It considers the angle of impact, coefficients unique to specific materials and even giving the velocity and mass after penetration has occurred. However, among these advantages there are a number of disadvantages that Project Thor possess. First, only projectiles with a length / diameter ratio less than 3 were used. Second, only projectiles were analysed if they did not deform or break up on contact. Third, notably the greatest disadvantage, is that small bullet-sized fragments were used. As this project development consider projectiles much larger in nature, it may produce inaccurate results when Project Thor's equations are applied to larger projectiles.

3.3.4 Decision

The algorithm to be used for this project will be the DeMarre equation. This equation provides a more accurate simulation when compared to impact depth, it considers the velocity of a projectile and is based on projectiles with an ogival head, on the contrary, impact depth only considers projectile with a blunt nose. It is also derived from research done in the same area as this project, unlike that of Project Thor which is conducted on much smaller projectiles, further confirming that results gained from it will be accurate. The DeMarre equation maintains these advantages while also remaining simple and within the projects aim to bring the level of simulation up to par. It is not without its disadvantages however, as there are no considerations for the angle on impact, therefore further calculations will be needed to compensate for this deficit.

4 Designs

4.1 Experimental Design

As Arma 3 is currently lacking in documentation, a series of experimental tests will be developed to acquire a further understanding of the game engine and how it will integrate with the projects development. To conduct said experiments, the following tests have been designed:

1. Investigate if a projectile / target properties can be modified during and after impact.
2. Determine the effectiveness of a C++ extension and if it can be used to change the projectile / target properties.
3. Examine the projectile / target properties retrieved and if they are accurate enough to be used for the development of the project.

4.1.1 Experimental Test 1 - Investigate if a projectile / target properties can be modified during and after impact

This test is integral to the success of the project, as without being able change the properties of the projectile, more specifically the vector, velocity and the damage done to the target, this project will be impossible to complete.

To setup this experiment a test environment will be created within the game with a tank firing a projectile at another armoured vehicle. Using the in-game scripting language and the popular mod CBA (Community Based Add-ons), an event handler will fire when the target tank is hit. This event handler will then return the projectile properties which will be used to change the projectile flight path and the damage done to the vehicle. The projectile flight properties will be recorded to a text file at set time intervals to determine if the experiment has been a success.

4.1.2 Experimental Test 2 - Determine the effectiveness of C++ extension and if it can be used to change the projectile / target properties

Using a C++ extension to conduct the majority of development will give a multitude of advantages. Being able to pass to and from a C++ extension will bring better performance and a more efficient development environment.

This experiment will test whether data can be transferred between a C++ extension and the Arma 3 engine. It will also investigate what the form the data passed is in and the performance. Performance is of particular importance because of the rapidity of time with projectile ballistics.

To setup this experiment, a test environment will be built upon the already existing experimental test 1. A C++ extension will be integrated with the mission and be used to change the projectile / target properties. Once the test has concluded, the projectile / target properties will be analysed and compared to that of the modified values.

4.1.3 Experimental Test 3 – Examine the projectile / target properties retrieved and if they are accurate enough to be used for the development of the project.

There are various properties of a projectile / target when impact has occurred, most notably are the surface normal vector and position at the point of collision between projectile and its target. The format of both properties will be investigated further. If development proceeds without knowing these values, then any calculations that occur will not be valid.

To conduct this test, a mission will be created based upon experimental test 1. Once the projectile is fired, a third-party script will be used to record the projectile flight path via the drawing of visible lines. The position will be recorded and then a spherical test object will be placed at the recorded coordinates. Further tests will be then conducted upon the test object to determine the format of the position property. The surface normal vector will then be recorded and the projectiles flight path will be modified to this value, clearly showing the surface normal direction.

4.2 Tests Design

For this project, a number of tests will be designed to confirm that the coding is working and correct and meets with the project aims and objectives. The most notable area of development that can be tested is related to objective 4 and 5. To further explain, the calculations done when a projectile has impacted a target and whether the projectile will penetrate along with the post-impact properties of the projectile / target.

To conduct this test, data will be gathered from external research done on the penetration ability of select projectiles. This research involves data gathered from WW2 sources including live firing tests and their results. Noted down during these tests are the type of projectile fired, range of projectile flight, thickness and type of target material impacted and the penetration depth of the projectile. Once assembled, the equation chosen, to complete objective 4 and 5, will use the data gathered and the outcome will be analysed and compared against the external research results.

5 Project Management Review

To review the management of this project and its development so far, the developed work up to now will be compared against the original time plan. If progress is under achieving, then a revised time plan will be set to help quicken the pace of development and allow the project to be completed on time. However, if the development is ahead of time, then new objectives will be set to make best use of the extra time available.

5.1 Project Progress

Project progress up to this point has been exemplary, with objectives 1-5 being completed fully and objective 6 being completed partially. To compensate for the extra time made available by being ahead of the original time plan, new objectives will be set for the extra time left for development of this project. The new objectives to be set will be the following:

7. Further developed the penetration calculation to account for HEAT (High-Explosive-Anti-Tank) Shells.
8. Increate the accuracy of simulation of projectile ballistics for the existing penetration calculations.

5.1.2 Objective 7 – Further develop the penetration calculation to account for HEAT (High-Explosive-Anti-Tank) shells.

HEAT shells are a common round type in use within Arma 3. To implement these rounds, a new algorithm will be developed alongside the already existing penetration calculations for kinetic-energy type penetrators. HEAT shells make use of the munroe-effect, are independent of projectile velocity and will require extra research into this field to devise an equation to calculate the penetration of said shells.

5.1.3 Objective 8 – Increase the accuracy of simulation of projectile ballistics for the existing penetration calculations.

So far for development of this project, the equations in use the various calculations for projectile ballistics have been relatively simple and omit numerous factors that happen in real-world ballistics. Some of these factors that could be considered are, the shape of the projectile head, the slope-effect (how a shells penetration is affected by armour sloping) and if the extra time gained is adequate, the perforation of armour that occurs during impact.

References

Achim Sippel, H.-J. K. (1991). *United States of America Patent No. US5009167 A.*

Ballistic Analysis Laboratory . (1968). *PROJECT THOR TECHNICAL REPORT NO. 47 .* Baltimore: Ordnance Research and Development Project No. TBJ-0238 .

Bohemia Interactives. (2013). Retrieved from Arma 3: https://arma3.com

Brothers, J. (1968). *United States of America Patent No. US3416449 A.*

Buc, S. M. (1994). *United States of America Patent No. US5297492 A.*

Captain Jonathan M. House, U. A. (1968). *Toward Combined Arms Warfare:- A Survey of 20th~Century Tactics, Doctrine, and Organization.* Washington D.C: U.S Goverment Printing Office.

Franze, Waffle SS, & Tom 48 97. (2015). Retrieved from LOD: https://community.bistudio.com/wiki/LOD#Fire Geometry

John S. Foster, J. A. (1983). *United States of America Patent No. US4418622 A.*

Lorrin Rexford Bird, R. D. (2001). *World War II Ballistics, Armor and Gunnery.* New York: Overmatch Press.

Olds, T. 4. (2015). Retrieved from Arma 3 Damage System: https://community.bistudio.com/wiki/Arma_3_Damage_Description

Thomanek, F. R. (1978). *United States of America Patent No. US4111126 A.*