Ballistics Simulation within Arma 3

Initial Report

Submitted for the BSc in
Computer Science

October 2016

by

Sean Andrew Sidney Phillips

Word Count: 1507

Table of Contents

1 Introduction

Arma 3 is a military tactical shooter video game developed by Bohemia Interactive. It features an open world environment set on a fictional Mediterranean island named Altis spanning over 290 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 environment. 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.

Arma 3 takes pride in its authentic gameplay, “Experience true combat gameplay in a massive military sandbox. Authentic, diverse, open - Arma 3 sends you to war” (Bohemia Interactive, 2016). While many aspects of the game stand true to this dogma, some features of the game fail to maintain this standard, the most obvious at first glance is the ballistics simulation of projectiles hitting armoured targets – most notably tank on tank warfare, of which this project intends to improve upon and bring up to the standard of the rest of the game.

The overall goal of this project is to provide a major overhaul of the armour damage system in Arma 3. The main tasks to provide a solution to, are projectile flight simulation, penetration simulation, projectile type simulation such as High-Explosive Anti-Tank (HEAT) and Armour Piercing Discarding Sabot (APDS) tanks shells and finally, damage done to the modules and crew inside the vehicle.

2 Background

Arma 3's simulation of projectile ballistics is lacking in comparison to other areas of the game. There are no detailed physics models, amongs others, on the penetration of armoured vehicles, the different projectile types or the damage done by a projectile once it penetrates.

2.1 Current Model in usage by Arma 3

In Arma 3, the damage system penetration and damage are not one and the same. A vehicle has both fire geometry and hit points modelled in. Fire geometry is an invisible physical model acting as armour – used to determine where projectiles will collide with the vehicle. Hit-points are used to determine the health of the vehicle; effectively how much damage is required until it is destroyed.

2.2 Typical Damage Process

To further explain how Arma 3 damage system works, this is a step-by-step process of how damage usually transpires in Arma 3.

2.2.1 Initial Impact

Firstly, a projectile has impacted the armour of a vehicle but has not penetrated. Regardless of whether the round has penetrated the armour, the vehicle takes some initial damage. Locations nearest the impact will take the most damage with the amount falling off geometrically. The vehicle's global hit-points will also take damage, in accordance to the physical properties of the projectile, even allowing the vehicle to be destroyed if the damage taken is significant enough and all these events happening without the projectile penetrating the vehicles armour.

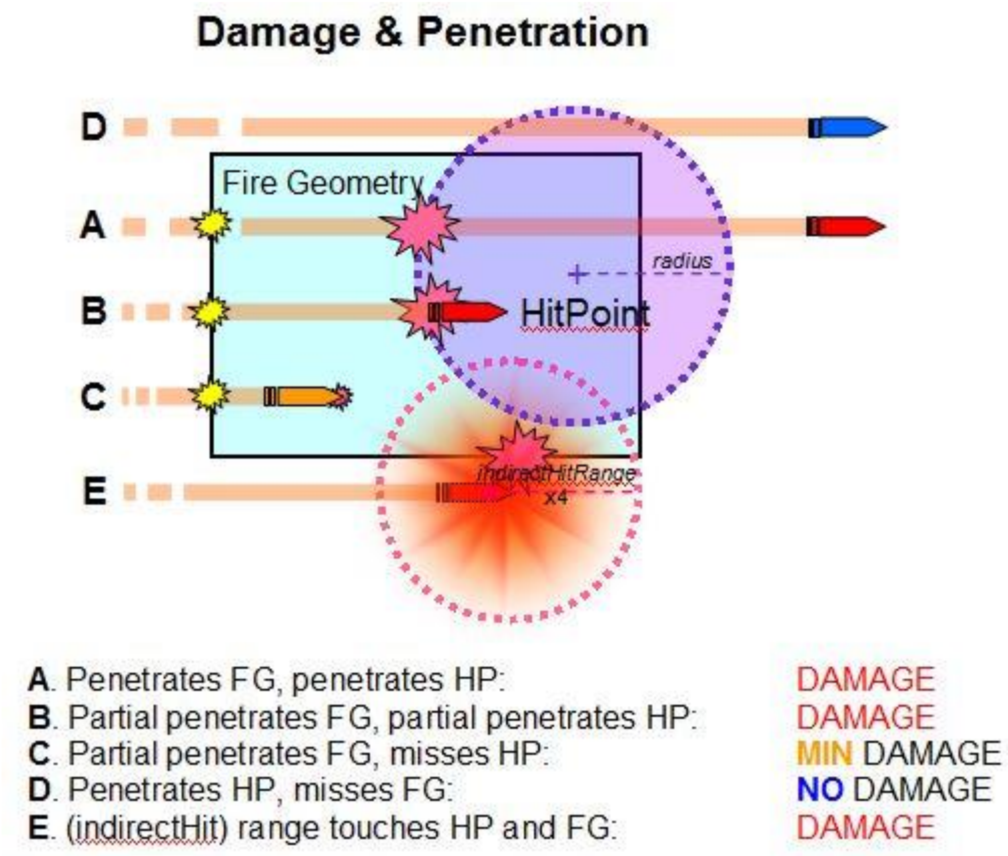
2.2.2 Penetrating Damage

Once the projectile has been determined to penetrate the armour of the vehicle, it will continue its flight path through the insides of the tank, damaging the vehicles modules as it goes. As the projectile travels through the tank it will significantly generate more damage to the surrounding modules, all the while still damaging the global hit-points.

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 needed to have a high standard of simulation.

2.3 Examples of projectiles fired at a vehicle

Below, in figure 1, is an example of different scenarios that can occur when a projectile is fired at an armoured target in Arma 3. The diagram shows a cutout of an entire vehicle showing the fire geometry and the hit-point circle in action. The whole picture of an armoured target would have the fire geometry taking on the outline of the armoured vehicle with multiple hit-point circles packing the insides of said vehicle.



A. Penetrates FG, penetrates HP:	DAMAGE
B. Partial penetrates FG, partial penetrates HP:	DAMAGE
C. Partial penetrates FG, misses HP:	MIN DAMAGE
D. Penetrates HP, misses FG:	NO DAMAGE
E. (indirectHit) range touches HP and FG:	DAMAGE

Figure 1. A diagram 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)

As shown above in figure 1, a projectile fired has to penetrate both the fire geometry alone or the fire geometry and hit-point circle radius to cause damage. For example, projectile D penetrates the hit-point circle radius but misses the fire geometry therefore causing zero damage. Projectile A, however, penetrates both the fire geometry and the hit-point circle radius causing damage to the target vehicle. Projectile E is where the system breaks down and acts unrealistically. This projectile explodes outside of the fire-geometry but because the range of the explosion intersects with the hit-point circle radius it will cause damage to the vehicle.

3 Aim and Objectives

The overall goal 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 the external library and store as variables for use within the library
4. Calculate 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

3.1 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 vehicle and projectile, this will be needed for latter calculations to determine penetrating power, damage taken from projectile etc. 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.

3.2 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 which the later calculations need to occur in.

3.3 Objective 3 – Parse the data outputted into the 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 by said library. 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.

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

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

3.5 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 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.

3.6 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. An interior damage model where individual key components of the vehicle can be damaged is required. These components will include, but not limited to, ammunition stowage, engine, crew members and fuel tanks.

4 Task List

#	Task Name	Description	Duration (days)
1	SQF Learning	Become familiar with the SQF scripting used within the Arma 3 engine	14
2	Store Projectile Properties	Using the SQF scripting engine, store, as variables, the projectiles properties and become accustomed to what each of these variables are and how they affect the projectile	14
3	Create an external library	To be able to make calculations needed to complete some of the later objectives, an external library has been chosen to complete said objectives. Therefore, attempt to create a .dll that will be able to interface information between the external library and the Arma 3 engine	14
4	Interface external library with Arma 3 engine	Create a .dll that will be able to pass information between the external library and the Arma 3 engine. This will need to pass information correctly and within a minimal amount of time. Also tests will be necessary to ensure that these requirements are met.	14
5	Parse data into variables in a readable form by the external library	For code within the external library, the data passed into the external library needs to be parsed into different variables all with the relevant type so that calculation can be written.	14
6	Create a simple ballistics model for the penetration of shells into armour	Using simplified equations, create a simple temporary base for the ballistics model. Over time gradually developing the features and create a more advanced system over time. The simplified ballistics model will only include the kinetic energy of the shell to take into account whether it will penetrate the target or not.	14
7	Create a more advanced ballistics model for the penetration of shells into armour	After the simplified model has been created, a more advanced system can be developed. This is will include the final variables needed (apart from armour thickness varying from location on the tank) to make an accurate penetration calculation. These variables will include, among others, angle of incidence of the impact, velocity of the shell, density of the shell and also the density of the armour.	14
8	Interim report		21

9	Create a database of armour locations and their thickness	Due to not being able to retrieve armour thickness with a scripting command within SQF, a database of armour thickness and their locations will be needed. This will allow calculation to compare the location of the shell and determine the thickness of the armour impacted.	14
10	Using armour thickness database, develop ballistics model further	Finally, to finish the penetration calculation, armour thickness will be taken into account. This will be done by comparing the location of the impact and comparing that to the database, where it should show the armour thickness value for the impact, allowing the penetration calculation to be fully accurate.	14
11	Create a modules-based damage system within the Tank	Create a set of modules within the tank, such as fuel tank, crew locations, ammunition, engine, etc. These will be able to be damaged when a shell passes through the insides of the tank after penetration. Calculate whether the shell will hit the modules, what damage they take and update the projectile properties. Once the shell reaches the inside wall of the tank make another penetration calculation.	14
12	Final report		x

5 Time Plan

		University Calendar Weeks																																
#	**Task Name**	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1		▨																																
2			▨	▨																														
3					▨	▨																												
4							▨	▨																										
5									▨	▨																								
6											▨	▨																						
7													▨	▨																				
8	Interim														▨	▨	D																	
9																		▨	▨															
10																				▨	▨													
11																						▨	▨											
12	Final																							▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	D

6 Risk Analysis

Risk	Severity (L/M/H)	Likelihood (L/M/H)	Significance (Sev. x Like.)	How to Avoid	How to Recover
Data loss	H	M	HM	Keep Backups	Reinstate from backups
Loss of backups	H	L	HL	Multiple Backups	Use alternate
Updates of Arma 3 engine causing incompatibility with written code	L	H	LH	Keep up to date with the Arma 3 dev blog to pre-empt and changes that may break the code written	Re-write code that has been affected by update to conform to the new update
Interface between game engine and external library not working	M	M	MM	Research external library usage within Arma 3	Use SQF scripting only to complete project
Personal PC capable of running Arma 3 breaking	M	L	ML	Be careful with fluids around PC and do not overclock too much	Use university PCs capable of running Arma 3 to complete project

References

Franze, K.J, Waffle SS & Tom 48 97. (2015) *LOD*. Available online: https://community.bistudio.com/wiki/LOD#Fire_Geometry

Olds, Tom 48 97, K.J & Strangepete. (2015) *Arma 3 Damage System*. Available online: https://community.bistudio.com/wiki/Arma_3_Damage_Description

Bohemia Interactive. (2013) *Arma 3*. Available online: https://arma3.com/