University of Portsmouth
School of Film, Media, and Creative Technologies

Final year Project undertaken in partial fulfilment of the requirements for the BSc (Honours) in Computer Games Technology

Unity Ballistics and Health system

By
Katie Richards
2208979
https://katie.games

Supervisor: Adam Jerrett

Module: M31481
April 2026

Project Type: **Combined**

Abstract

Many flat screen and virtual reality shooter games use simple ballistic mechanics often unchanged from those used in the 80s and 90s. This project aims to change that and improve player immersion by creating a VR shooter game with a variety of ballistic methodologies taken from terminal ballistic physics and engineering fields that have not before been implemented into video games. It will allow a user to compare directly between them and a new advanced model to determine the effect which they have upon realism and player immersion. Furthermore this project will also aim to improve issues with health systems used in shooter games. These too, like the ballistic mechanics, are often lacking in detail and realism and can reduce player immersion especially in VR titles. This project aims to fix that with a highly detailed health system allowing for individual localised damage to different limbs and organs. Though this system is primarily designed for use in a VR health system it could also be implemented in flatscreen games for the player, or for things such as animals in hunting games where a high level of detail is beneficial to the players gameplay experience.

Introduction

Topic

Creating a detailed health and ballistic system for use in VR experiences to improve realism.

Target Audience

The main demographic which this project aims to appeal to is the average PCVR user. Though support for standalone VR platforms would be nice due to their large market volume they often struggle with performance capabilities for even modern rendering techniques, let alone having the additional frametime spare for ballistic calculations. Furthermore in recent years the enthusiast industry is seeing a shift towards PCVR systems again, with recent platforms such as the PSVR2 (Sony, 2023) and Steam Frame (Valve, 2025) proving this. These platforms rely on external computers for their processing and rendering allowing them to have far greater performance in comparison to their standalone counterparts and thus are a more realistic target for this project.

The average VR user tends to be between the ages of 16-49 (Ramirez, 2026) and in the case of PCVR is often willing to spend large amounts of money in exchange for improved immersion and realism as is evidenced by the cost of hardware currently available for consumers in the PCVR space. This type of enthusiast user will often spend hundreds on systems such as haptic suits (OWO Game, 2026) or 8k headsets (Pimax, 2017), for a small immersion improvement. This means that they will naturally be interested in systems to improve realism in VR games and get the most out of their hardware.

What this project solves

The core problem which this artefact will address is the lack of realism in damage and ballistic systems in VR shooter games and how that can be improved in an efficient and visually appealing way.

Within VR shooter games there has been a persistent challenge with how players take damage and the ballistic properties of different materials. The popular title Pavlov VR (Vankrupt Games, 2017), only slows down the players movement temporarily when they take damage, along with having only hard and soft materials which either let all bullets through or none at all. This is extremely misleading in comparison to real life where different materials and different thicknesses of materials have unique ballistic properties.

Other games such as Ghost Of Tabor (Beyond Frames, 2023), make an attempt to implement a more realistic health system however fall short on the ballistics front. Their systems may allow for per limb damage affecting you in game, however, they have the same shortfalls as Pavlov (Vankrupt Games, 2017) with bullet penetration. Furthermore their armour implementation is purely just a damage reduction percentage and has no ballistic realism (Fandom, 2025) .

This project aims to solve that by implementing a sophisticated ballistic and health system into a VR shooter game that will be both performant and detailed, aiming to improve player immersion. The ballistic system referred to as KBallistic will be capable of calculating energy loss and energy delivery in a variety of mediums. The health system as a part of this will allow for per limb damage all driven by blood loss values.

Why is this needed

Shooter games in general have a tendency to lack realism when it comes to ballistics, with a wide range of modern titles such as the Call Of Duty games still opting for simpler raycast (hitscan) mechanics. This is the industry standard as it is computationally "cheap" and is how it has been done since the origin of 3D games (Arti Sergeev, 2020). However, in VR it causes a sensory disconnect for the player, as being able to hit enemies across an entire map instantly is not something that would be possible in real life.

While competitive shooters often prioritise speed and consistency over physics, such as Counter Strike 2 (Valve, 2023) or Pavlov VR (Vankrupt Games, 2017), there is a large demographic of PCVR enthusiasts who want more realism. This is evidenced by the long term success of games such as Hot Dogs, Horseshoes & Hand Grenades (H3VR) (RUST LTD, 2016), which is considered somewhat of a gold standard for VR weapon handling and physics.

Further adding to this issue, very few VR shooter games currently account for terminal ballistics. The popular title Pavlov VR (Vankrupt Games, 2017) highlights this problem entirely with materials being tagged as hard or soft either allowing all projectiles to penetrate or none at all. This makes tactical and more slow paced gameplay unrealistic and unintuitive as it does not reflect how weapons may perform in real life.

The next issue this project intends to address is the outdated nature of health systems. Many modern shooters still rely on simplistic single health bars which represent the player's entire health and go down consistently regardless of where you are hit. This is particularly problematic for VR titles where the player's physical movements are tracked. If a player takes damage on their arm but has 0 penalty to their accuracy the immersion is broken for them.

There is clearly a want for more detailed systems with the titles Ghosts Of Tabor (Beyond Frames, 2023) and Escape From Tarkov (Battlestate Games, 2019) being excellent examples of this. They have seen large communities form thanks to their health and looting mechanics and players agree that having to manage specific injuries and bleeding add a layer of excitement and risk which single health bar systems cannot replicate.

Issues that may be encountered

The main potential issues with this project will be of mathematical nature. The mathematics used in terminal ballistics can be extremely complex and hard to decipher making it challenging to implement correctly. Furthermore these calculations can be long and computationally expensive, potentially worsening cpu frametimes and gameplay performance. In a VR game this can have a detrimental effect on player experience, potentially leading to motion sickness and unplayability (Wang et al., 2023).

I aim to overcome these potential issues by creating a scale of ballistic detail from the standard raycast implementations up to fully calculated terminal ballistics. This will allow me to discover what elements affect the players experience to a greater degree so that cpu cycles can be saved where they are not needed.

Another potential issue is that this artefact will be on the cutting edge of what has been done with ballistic systems in video games. An issue resulting from this is a lack of documentation from other people to see where they went wrong along with a general lack of support when issues are encountered in development. Furthermore games with more sophisticated ballistics are often rather secretive on how it functions internally, and do not share this information anywhere that I would be able to access.

To work around this issue I will primarily be adapting mathematical models from mechanical engineering papers and studies. These calculations are often old and well known with several papers about them and various studies done on their performance and implementation.

Finally the project will likely encounter issues with working with VR hardware in general. This process can be complex and require the developer to frequently enter and exit VR which is generally inconvenient and can be fatiguing. Furthermore performance will also be harder to attain due to the nature of VR usually requiring 4k rendering at around 90fps.

To solve these issues the project will be cross compatible to work in flat screen mode for development. This will significantly reduce strain on development and make debugging substantially faster. As for performance the project will be made in a lightweight stripped back version of Unity HDRP (Unity Technologies, 2024). Unity is an extremely performant game engine with XR support and the high definition render pipeline is able to deliver the visual fidelity needed for immersion. By stripping back elements of the render pipeline, such as disabling screen space reflections, performance can be greatly enhanced to a point which is suitable for VR hardware even on lower spec personal computers.

Literature review

Ballistics in engineering

Currently the most advanced developments in ballistic technology are within engineering instead of interactive media. Fields such as mechanical engineering and terminal ballistics have spent years researching penetration mechanics and material responses for use in safety design and weapons systems.

One of the main computational approaches used in these fields is Finite Element Analysis (FEA). This approach uses the Finite Element Method to solve mechanical issues by breaking up structures and components into smaller elements, allowing for differential equations to be used to solve stress, strain, and energy transfer across a mesh. This is then recombined to estimate the behaviour of a full system, such as a spring breaking, or a bullet penetrating a sheet of metal. FEA is often used for high strain rate events such as ballistic impacts where a range of factors can simultaneously affect an object.

The advantages of FEA are in its ability to produce highly detailed and accurate simulations without the need of physical prototyping. It helps to enable iterative design processes and is capable of capturing complex non linear interactions between materials under extreme conditions. However, these do come with some significant limitations. FEA simulations are extremely computationally expensive, requiring substantial processing time and hardware especially when modelling high velocity impacts due to their chaotic and non linear nature. Furthermore FEA simulations depend heavily on the quality of their input data, requiring precise material properties and parametric models.

Due to these constraints, alternative methods have been made using analytical or empirical models. These are often used in situations where computational efficiency and speed is required compared to absolute accuracy. These models use simplified mathematical formulae to approximate ballistic behaviour focusing more on creating a threshold of what might happen such as a residual velocity and penetration depth, instead of exactly how two or more materials may deform. Whilst these are less detailed and accurate than FEA, they are substantially more efficient and can be solved quickly with consumer hardware. As a result these are widely used in military and engineering contexts for initial analysis and validation where repeatability and speed are more important than fully detailed models.

Ballistics and health in games

Ballistics in games usually implement simplified models focusing on performance and repeatability over physical accuracy like an FEA suite. Early implementations, such as in Counter Strike Source (Valve, 2004), use single raycasts where projectiles are instantaneous lines instead of simulated physics objects. Penetration behaviour in these systems is often done using pre determined material look up tables, where surfaces are assigned simple properties to determine whether or not projectiles will go through them.

Similar approaches exist in modern titles too such as in Valorant (Riot Games, 2020) released in 2020. Community driven analysis (quarkez, 2020) suggests that penetration is controlled by simplified classifications of materials and weapon types, similar to that of Counter Strike Source (Valve, 2004), with limited consideration of terminal ballistics such as projectile velocity or energy transfer. Whilst these systems are extremely efficient computationally, they fail to consider a lot of the factors that influence a real ballistic impact.

More recent developments show that the industry is gradually improving, such as Counter Strike 2 introducing penetration mechanics that also take into consideration material thickness and weapon penetration power (CS2 Official, 2025) . This allows for more predictable interactions between projectiles and surfaces, along with a better approximation of how they may have behaved physically. This system is particularly effective as it remains computationally simple and consistent enough to be used in competitive settings, whilst giving a more detailed indication of physical performance.

However, despite these improvements, current implementations still do not use and are not comparable to terminal ballistic models. In particular things such as projectile velocity, deformation, and energy dissipation are often simplified or disregarded entirely in both how a bullet interacts with its environment and in damage modelling. As a result of this there is still a significant gap between the models used in games and the solutions available in engineering and defence research.

For ballistics in a game to be good they need to meet two distinct criteria. Speed, and detail. These two parameters are often mutually exclusive so it is imperative to find a balance between them. An excellent example of this would be the systems used in the game Escape From Tarkov (Escape From Tarkov, 2026). One of the big improvements here over previously mentioned titles is the usage of cartridge specific statistics instead of a damage value per weapon. Furthermore bullets in flight are modelled in detail and are affected by things such as gravity and air friction along with being capable of ricochets when hitting objects. These environment driven parameters are used to affect the bullet's velocity which in turn affects its performance on impacts.

Mathematical models

Ballistic penetration

A range of analytical and empirical models have been developed to approximate ballistic penetration in materials with much better efficiency than full numerical simulations. One of the foundational approaches to this is the Alekseevskii-Tate model. This calculates long rod penetration by modelling projectile velocities as a function of time (Walker, 2021). While this approach is physically grounded, and widely used in defence research, it introduces computational complexity due to its dependency on time, making it unsuited for real time applications.

To address these issues a range of alternative models have been proposed which are based on indentation and hardness. Early work such as Bishop Et Al's theory of indentation and hardness (Bishop et al., 1945), show that penetration behaviour can be approximated using material strength and hardness parameters, greatly reducing computational expense. Continuing on this research, Forrestal and Warren present empirical penetration equations for ogive nose projectiles impacting aluminium targets (Forrestal & Warren, 2008). These models are experimentally validated and capable of producing accurate predictions for impacts up to around 1300 m/s whilst remaining computationally efficient. Compared to time dependent formulations, these are a good balance between accuracy and performance.

Injury and tissue modelling

Ballistic injury modelling is usually done using the transfer of kinetic energy into soft tissue with a common way of emulating this for measurement being calibrated ballistics gelatin. Two prominent approaches to this are the mTHOR model power-law equations (Schwartz, 2012), and the quantitative model proposed by Charles Schwartz (Schwartz, 2012). Both models estimate outcomes such as residual velocity and permanent wound cavity making them both suitable for assessing damage.

However, though they are both alike in their outputs they differ in flexibility. The Schwartz model allows for parameters such as density and tensile strengths allowing it to be adapted to different tissue types such as muscle and fat. It has also been shown to perform well in energy based formulations and in estimating probability of incapacitation (Schwartz, 2012). In contrast, the mTHOR models are more generalised and less adaptable to varying conditions. As a result of this, quantitative models with tunable parameters are more versatile for applications with multiple materials being represented (Schwartz, 2012).

While ballistic models describe projectile behaviour within a medium, additional modelling is required to represent a body's response following impact. One approach is to estimate total tissue damage using a scale applied to a wound volume, resulting in an estimate of overall disrupted tissue (Kaur et al., 2023). This provides a simple but computationally efficient representation of injury severity.

A more explicit approach involves modelling blood loss rate directly. Initial haemorrhage can be approximated using a Bernoulli-based derivation giving a baseline bleed rate for further time driven modelling (Kragh et al., 2011). This initial bleed rate can then be effected with clotting formulas from physiological literature, such as from the American Journal of Physiology Heart and Circulatory Physiology. These exponentially reduce blood flow rate with time as a product of clotting resulting in an exponential decrease in blood loss (Tsiklidis et al., 2019).

Finally, total blood loss can be linked to physical responses to estimate survivability. There are established classes to provide a basis for relating blood volume loss to physical effects such as increased heart rate, shock, reduced mental function, and loss of consciousness (Hooper & Armstrong, 2022). These thresholds can be used to make a value representing the survival state of a person, based on their total volume of blood lost.

Structural material penetration

Penetration of common materials such as wood or metal can be addressed with both generalised and material specific models. The DoD structural penetration equations (U.S. ARMY CORPS OF ENGINEERS, 2018) give a framework for estimating penetration across multiple material types such as wood and concrete using hardness based parameters. These models are computationally efficient, and widely used, making them well suited for large scale or mixed material scenarios.

However, limitations arise with more complex or unique materials. For example, the DoD approach to wood penetration (U.S. ARMY CORPS OF ENGINEERS, 2018) uses arbitrary hardness values for their range of materials which does not allow them to be extended to others. In contrast to this more specialised studies such as the Netherlands Defence Academy research on wooden targets (L. KOENE & F.R. BROEKHUIS, 2017) gives a more specific and controllable model which takes an input of Janka hardness allowing it to be used for any wood listed on the Janka scale. This model also employs the usage of real world testing to validate results which is more reliable and consistent than that of the DoD equations which show no testing or real world examples.

Methodology

Philosophy and worldview

This project employs a mixed methods worldview grounded in research through design, which assumes that insight comes from prototyping and testing of technical systems. The project balances the technical accuracy of systems with the subjective player perception of realism. This acknowledges that while the ballistics are based on objective mathematical frameworks, realism and immersion are subjective constructs that require a player centered approach to evaluate.

Planning and best practice

To ensure that the project is both technically sound and performant, an agile iterative development process will be used. This will involve a continuous cycle of design, implementation, testing, and reflection. This cyclical testing approach is the best practice for this project, as it allows for rectifying issues before they impact the end user experience.

Data collection and measurement

Data collection will focus on a comparative analysis of different ballistic methodologies. Rather than measuring mathematical ballistic accuracy of system performance, the research will gather qualitative and quantitative player feedback via surveyed users.

Participants will interact with different ballistic ranges representing existing ballistic solutions along with the one developed in this artefact. They will then provide data on their perceived realism and immersion with these systems, along with what elements have the biggest impact on these factors.

These data points will be gathered anonymously through the survey platform Google Forms, and analysed to determine which technical elements of ballistics in games are the most relevant to the players immersion and sense of realism.

Limitations and bias

The main limitation of my methodology is the presence of sampling bias. As the application is only available on PCVR, participants are limited to this platform. This reduces the diversity of the data gathered and may influence results as PCVR users may hold different expectations compared to standalone VR users.

Additionally self reflection bias (Karpen, 2017) is present as participation is voluntary. People who choose to take part likely have an interest in VR shooter realism which may result in more critical responses than those of a wider less specific audience. This project also relies heavily on subjective data which will vary between participants and is often influenced by prior experience rather than solid real world reference points. This may result in findings lacking consistency that are not fully generalisable.

A further limitation comes from order effects (Lucas, 1992), as ballistic systems are presented in a fixed progression from less realistic to more realistic implementations. This may influence testing results as later systems can be perceived more favourably in direct comparison with earlier ones. Furthermore, participants may view systems in a higher regard as they become more familiar with the games systems as they progress through the demo, this may affect results especially in the later, more detailed, ballistic systems

Ethics and risk assessment

Although this project does not involve sensitive data or high risk activities, several ethical concerns can still be acknowledged due to the human participants and nature of the project. All participants in data collection will be informed of the purpose of the study, the nature of the VR application, and their right to withdraw from testing at any time. No personal information will be collected and all data will be kept anonymous. Furthermore the VR environment will contain simulated firearms and combat scenarios. Whilst these are common in VR and flatscreen video games, participants will still be informed beforehand so that they may choose to not take part if they are uncomfortable with these elements.

The main risks throughout data gathering will be VR specific safety concerns along with damage to equipment. Due to how VR systems work the user will be visually impaired which can affect their sense of balance and location. This results in a higher risk of the player colliding with things within their playspace. Furthermore, if performance of the ballistic system is poor it could result in reduced framerate in the test application. On a flatscreen game this would not matter much, however, in VR this can result in motion sickness and loss of balance. Though performance is not being used as a data point in testing, maintaining consistent proper performance is still an important aspect to control to ensure participant safety and consistency of data. To rectify these issues participating members will be encouraged to take frequent breaks and the project will be extensively tested in flatscreen first to reduce the potential for performance related issues.

Artefact development

Planning

The first part of the development was creating a detailed and refined plan of the content that would be in the project and how it would be delivered. An initial list of wanted features for the MVP was created and then this was integrated into a Trello board to aid in project and time management as can be seen in Figure 1. Along with this an initial schedule was made in Figure 2 to plan the time at which main sections of the project would be completed. This was to try and keep the project in a reasonable state at all times and to make sure that it was delivered on time.

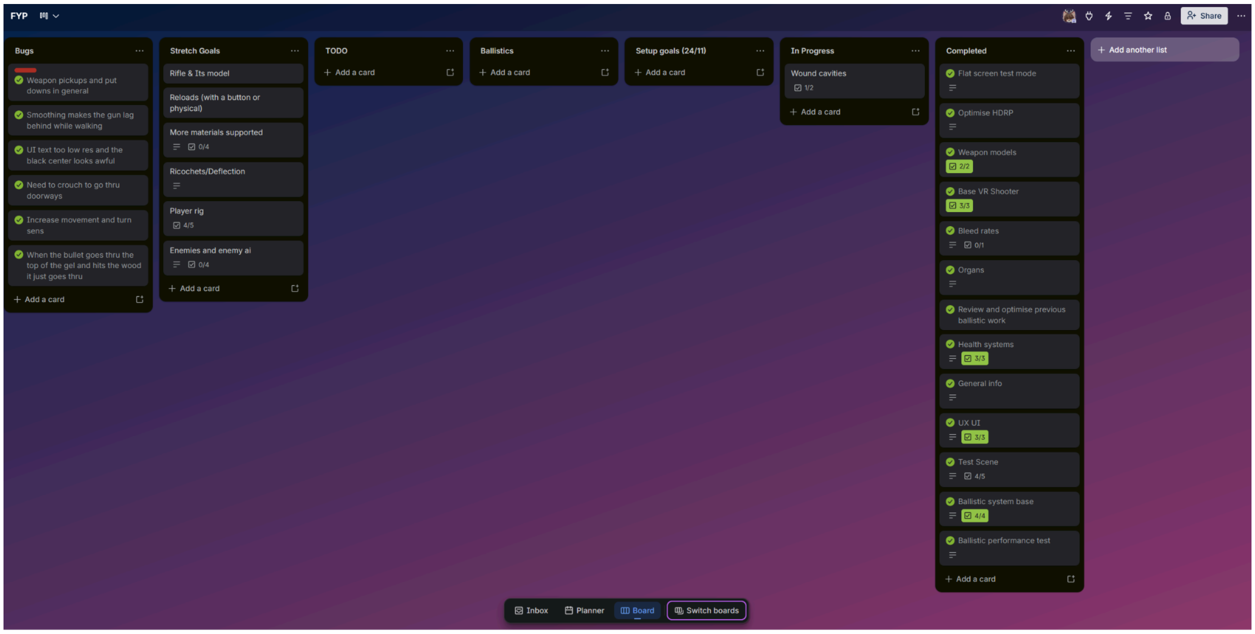


Figure 1 - An image of the projects now completed Trello board showing finished tasks and solved bugs.

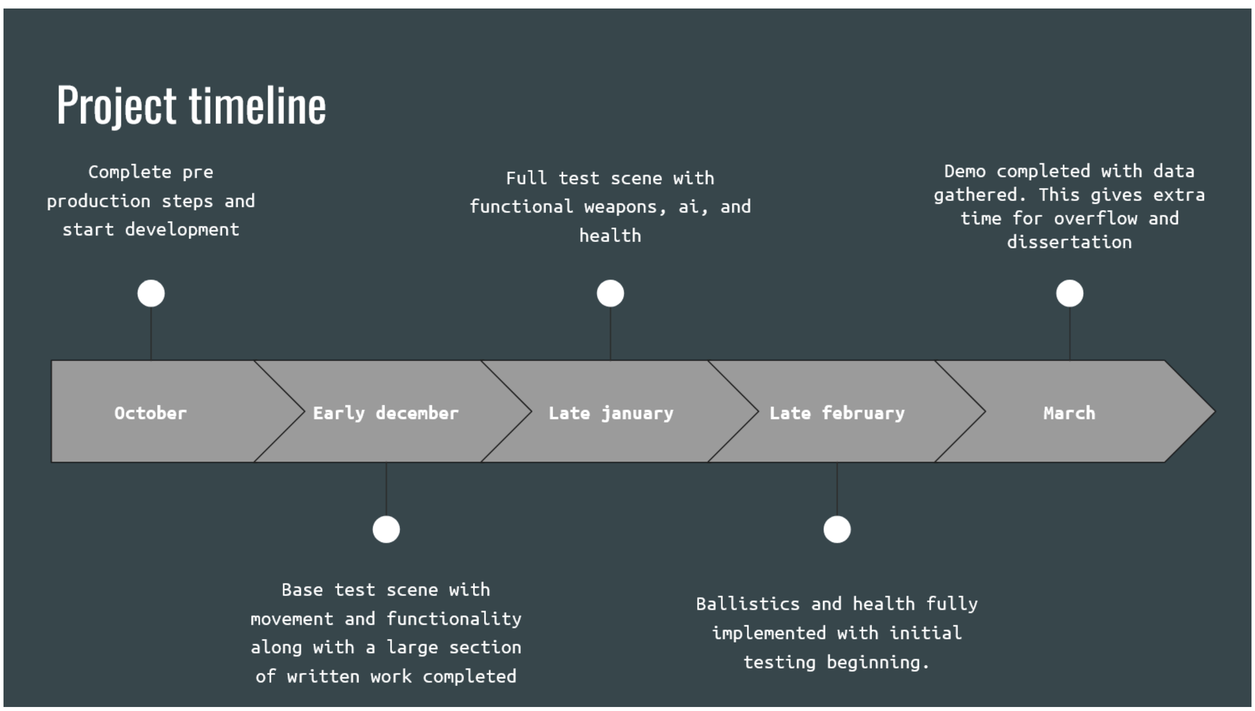


Figure 2 - An image of the project's development timeline showing estimated time periods for pre production, initial development, completed development, and testing.

Design and initial implementation

As planned the first part of the project was to make a VR shooter game designed around supporting a range of different ballistic and health methods in a convenient and efficient manner. One of the particularly interesting things here was trying to get over the limitations of development for VR platforms and trying to solve some of the issues with that. A good example of this would be the physical limitation of having to put a VR device on to test features. In practice this can be extremely inconvenient and time consuming during development and also physically fatiguing with spending extended periods using it.

To solve this issue I developed a system where the game could be controlled in both flatscreen and VR modes with the same player controller, this would allow for ease of testing and development of the non VR systems. This is something which Unity's built in XR toolkit does not support, and is something which I had to develop myself from scratch. Furthermore Unity's XR toolkit is lacking ease of support for other systems such as independent or full body trackers, it is also challenging to implement without using the rest of Unity's XR suite which is not necessary for this project and may reduce performance.

```
private void HandleHoldable(System.Action<HoldableObject> action)
{
   float _scaledRadius = _collider.radius * Mathf.Max(
       _collider.transform.lossyScale.x,
       _collider.transform.lossyScale.y,
       _collider.transform.lossyScale.z
   );

   Collider[] _hits = Physics.OverlapSphere(
       _collider.transform.position,
       _scaledRadius,
       ~0,
       QueryTriggerInteraction.Collide
   );

   // if we are colliding with an object that has component type HoldableObject
   foreach (Collider _hit in _hits)
   {
       // if collider overlapping with object tagged with Grabbable
       if (_hit.TryGetComponent<HoldableObject>(out HoldableObject _holdable))
       {
           action(_holdable);
           return; // first valid object
       }
   }
}
```

Figure 3 - The code used for interacting with the polymorphic grab functions on objects.

To solve this I developed my own XR and Flatscreen player controllers which can be switched between dynamically with a single button press during runtime. This system also allowed me to easily develop my own systems for grabbing and releasing objects in VR, such as in Figure 3, along with a range of systems around that which would be extremely useful when later developing the weapons systems. This multi platform player controller helped to streamline development and improve time spent on the project. It was particularly useful when testing the final integration of the project later in development and also allowed me to continue work on the project even without access to VR hardware.

One of the main issues I encountered with the XR controller was scaling the player to match their real world proportions. I initially intended to give the player a full rig with arms, legs, and a torso that would move with them so that they could be damaged by enemies to show how the systems would feel on a player. However, this turned out to be extremely complex to do and was a major time sink in this early development stage. The issue I encountered was that not everyone is uniformly sized, with different arm lengths and heights, and getting a character model to correctly deform to match this using inverse kinematics was extremely complex. I ended up falling back to just my minimum viable product for this part as it was taking too much development time to justify its creation. This choice to re-scope and abandon full body inverse kinematics greatly aided in ensuring that the project's development remained aligned with its time constraints.

Other than movement I also encountered several interesting issues with performance and optimisation. As the project is designed to run on VR hardware there is a large performance cost compared to flatscreen games, along with a requirement for a perfectly consistent framerate when playing. Performance was extremely important on this project as it could greatly affect a player's immersion if their experience was inconsistent. On the contrary to this visual fidelity is also important for a players immersion especially in the context of this project where realism and immersion are my targets.

To achieve my fidelity goals with this project I decided upon using Unity’s HDRP pipeline. This would give me all of the tooling and customisation required to make a visually appealing and computationally optimised project. To begin this I setup the scene with initial baked lighting and static objects, I then stripped the render pipeline of any unnecessary features such as screen space reflections and raytracing, as even when not in use these add substantial overhead to the cpu frametime of the project. Thanks to these optimisations, I got the project to a point where it could run at several hundred FPS at the reference resolution for the VR system I use, whilst retaining a level of visual fidelity appropriate for the project. This resulted in a VR frametime of only 5ms as can be seen in Figure 4. This was a substantial improvement over the initial performance where the project averaged around 75fps on a desktop 3090 in 1080p resolution. The project is now so performant that it is even able to run on mobile hardware and was able to reach 60fps in 4k on a laptop with no dedicated graphics processor.

Figure 4 - An image of the SteamVR frametime graph when testing the project. Anything below 11.1ms would indicate the player is receiving the full intended 90fps.

One of the other sections of optimisation which I spent time on was the weapon models which the player would be using. These also have to fit the same requirements of fidelity and performance so they were optimised to use atlased materials and have poly counts lower than fifteen thousand. This is more than reasonable for a hero asset in a VR game and results in very little performance loss whilst maintaining visual fidelity to improve the players experience as can be seen in Figure 5.

Figure 5 - A series of images of one of the weapon models created for this project rendered in Blender with materials produced in Substance Painter.

Ballistics and physics

The next major stage of the project was to implement the weapons which the player would be using and the systems that come with that. To meet the requirements of this project a series of systems will need to be made so that the player is able to pick up different weapons, use them, and switch between different ballistic methods on them. In practice this is actually quite a complex set of constraints as it requires entirely changing the weapons functionality dynamically in runtime. The player will be able to choose between three distinct ballistic and health combinations. KBallistic is the most detailed of these with the custom models developed in this paper. Multi Bar is the next step down and is a raycast system with multiple health bars per limb on a character. Finally the Single Bar system is the simplest using a single health bar to represent the full health of a character.

The first step of doing this was to make a simplistic generic weapons system that the player could pick up and put down. This was done by making a custom XR grab handler which allows the player to pick up objects and match their position and rotation to their hand. This works well for single handed weapons however does not work at all with two handed weapons as both hands would be conflicting.

```
// if we have both hands on the weapon
if (_isTwoHand && _hasVstock)
{
   // our position is that of the primary hand
   Vector3 _targetPosition = _primaryHandTransform.position;
   _primaryGrabTransform.position = Vector3.Lerp(_primaryGrabTransform.position, _targetPosition, _smoothingFactor *
Time.deltaTime);
   // our rotation is pointing at our secondary hand
   Quaternion _targetRotation = Quaternion.LookRotation(_secondaryHandTransform.position -
_primaryHandTransform.position);
   _primaryGrabTransform.rotation = Quaternion.Slerp(_primaryGrabTransform.rotation, _targetRotation, _smoothingFactor
* Time.deltaTime);
}
else
{
   // if its only 1 hand on the weapon it needs to handle position and rotation
   // its possible it could just be the secondary hand thats on the weapon if for example they let go of the primary
for some reason
   if (_primaryGrab)
   {
       Vector3 _targetPosition = _primaryHandTransform.position;
       _primaryGrabTransform.position = Vector3.Lerp(_primaryGrabTransform.position, _targetPosition, _smoothingFactor
* Time.deltaTime);

       Quaternion _targetRotation = Quaternion.LookRotation(_primaryHandTransform.forward, _primaryHandTransform.up) *
Quaternion.Euler(_oneHandPrimaryOffset);
       _primaryGrabTransform.rotation = Quaternion.Slerp(_primaryGrabTransform.rotation, _targetRotation,
_smoothingFactor * Time.deltaTime);
   }
   else    // this is holding it 1 handed with the only hand being the secondary one
   {
       // get the offset from secondary grip to primary grip
       Vector3 _secondaryToPrimaryOffset = _primaryGrabTransform.position - _secondaryGrabTransform.position;
       Vector3 _targetPosition = _secondaryHandTransform.position + _secondaryToPrimaryOffset;

       _primaryGrabTransform.position = Vector3.Lerp(
           _primaryGrabTransform.position,
           _targetPosition,
           _smoothingFactor * Time.deltaTime
```

Figure 6 - The update loop used for the “virtual stock” behaviour where the rear hand controls a weapons position and the front controls the rotation.

To solve this issue I took inspiration from Pavlov VR’s “virtual stock” functionality. This is a system where the player's rear hand controls the position of the gun, and the front hand controls its rotation. This system shown in Figure 6 is good as it makes steadying the weapon easier for the player and it is reasonably intuitive to use especially for someone who is familiar with VR shooter games. This was reasonably simple to implement with enough trial and error, however did lead to a discovery of several issues with how the Unity input system handles XR controllers. The first of which was that the generic grip function simply does not appear to work with any XR controllers, I ended up having to use the Oculus Touch grip function which strangely works fine for both my personal Pico controllers and my testers Meta and even Valve Index controllers. Furthermore there is no such thing as a generic grip function, you cannot just have a grip input and get the hand from it later, you need to have two separate grip functions with one being for each hand. This seems flawed in several ways from an efficiency perspective but is the way that it worked and even though I am remaking a lot of Unity's XR components, remaking the entire input system would be too large a time sink to be reasonable for this project.

One of the major time sinks that came from weapon handling was an ambition for fully functional weapon operation with things such as reloading and operating the weapons bolt. As can be seen in Figure 5 this was something the models I made were designed around doing, however I did not end up implementing due to the time required. This was mostly because it was never entirely clear how to actually make a constrained movement path for something like a weapon bolt. I intended to use blendshapes initially, however getting this to work well was far too time consuming for what would be a small improvement for some users and a minor inconvenience for others. Because of these reasons I decided to entirely disregard the need for reloading to further simply the project and keep this section within its allotted timeframe.

The next step was actually implementing the different ballistic physics methods and a way of changing them. This was done by using a generic ballistic method interface that let a weapon dynamically switch between calling entirely separate shoot functions. This meant that a weapon could shoot in several entirely separate ways without any changes to its base weapons system.

For the KBallistic physics method I ended up adapting a ballistic system I had previously worked on. This system uses a range of different mathematical models for ballistics including the Q Model (Schwartz, 2012) for flesh mediums along with a range of other models for metal and wood (U.S. ARMY CORPS OF ENGINEERS, 2018). Although the base mathematics here remain mostly unchanged from my previous work apart from some unit corrections that were resulting in incorrect values, the system for calculating what the bullet is actually hitting along with its projectile movement had to be entirely remade from the ground up. With the previous system it had several fundamental issues. These were that the bullet could not handle cavities in objects with things such as air gaps, and it could also not handle composite materials where one was inside of another or two were closely stacked as can be seen in Figure 7. This required an entire rewrite of projectile movement to solve and took around a week of development time to complete.

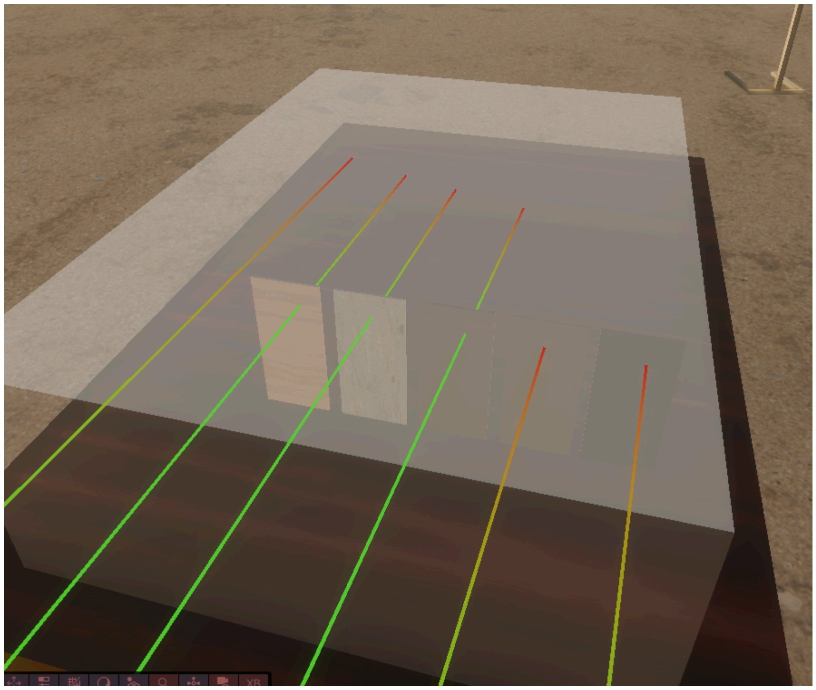


Figure 7 - An example of composite materials where there is a plate inside of a piece of ballistics gelatin. My prior system was not capable of accounting for this and the current newly developed one can handle it without issue.

These issues are particularly interesting as they primarily come from constraints of the physics engine built into Unity which I had to solve here. The problem is that collision geometry in Unity is single sided, aligning with the normals of a mesh. This is problematic when you need to get the thickness of an object that you have hit relative to your current direction. To solve this I used a system where upon hitting an object a ray would be fired from an opposite vector with the point of contact being used to calculate the thickness of the object which was being hit. In concept this works well, however, in the case of a room with 1cm thick wooden walls it would treat it as potentially a several meters thick piece of wood, as it could not account for the cavity in the middle. This is also why composite materials did not work as it could not account for anything inside of that initial collider.

To solve this I kept that initial system except once it hit that outer wall it fires an additional ray inwards from it. For this to hit a forward facing normal it means there must be a cavity in the object or another object nested inside of its collider as can be seen in the code in Figure 8.

```
// no inner objects hit this frame, progress thru material
transform.position += _step;
_trajectoryPoints.Add(transform.position);
UpdateTrajectory();

// check if we reached the end of the penetration
if (Vector3.Distance(transform.position, _penetrationEndPosition) <= 0.001f)
{
   _isPenetrating = false;
   Velocity = Velocity.normalized * _penetrationExitVelocity;

   if (_penetrationExitVelocity <= 0)
   {
       StopProjectile(transform.position);
   }
   else
   {
       // successfully exited this object
       _activeContainers.Remove(_currentPenetratingCollider);
       transform.position += Velocity.normalized * ExitOffset;
       _trajectoryPoints.Add(transform.position);
       UpdateTrajectory();
   }
}
```

Figure 8 - A section of the code used for making a projectile penetrate in steps instead of instantaneously when inside of an object.

This process is repeated iteratively with it shooting a ray in the opposite direction of this impact to find the thickness of the void or of the internal object. This system now allows for the bullet to hit nested composite objects and handle them appropriately along with properly dealing with voids in objects. I also had to change the way that penetration worked previously as it originally teleported the bullet to its exit point or the point where it finished penetrating instantly. This would not work with composite materials as it would mean the bullet could technically travel through meters of loose or soft materials instantly. To solve this I used a system of keeping the projectile penetrating until it reaches its determined end position. This means that the projectile now moves smoothly through objects using the code shown in Figure 9.

```
// sweep from backcast offset toward entry point
Vector3 _rayOrigin = hit.point + Velocity.normalized * RayBackcastOffset;
Vector3 _rayDirection = -Velocity.normalized;

// only look for distance up to the entry point plus a tiny bit of padding
float _maxSearchDist = RayBackcastOffset + 0.1f;
float _traversedDist = 0f;
int _safety = 0;
Ray _ray = new Ray(_rayOrigin, _rayDirection);
RaycastHit _closestExit = new RaycastHit();
bool _foundExit = false;

while (hitCollider.Raycast(_ray, out RaycastHit _stepHit, _maxSearchDist - _traversedDist))
{
   // check if the face is actually in front of our entry point
   if (Vector3.Dot(_stepHit.point - hit.point, Velocity.normalized) > 0)
   {
       _closestExit = _stepHit;
       _foundExit = true;
   }

   float _stepSize = _stepHit.distance + 0.001f;
   _traversedDist += _stepSize;
   _ray.origin = _rayOrigin + _rayDirection * _traversedDist;

   if (++_safety > 10 || _traversedDist >= _maxSearchDist) break;
}

if (_foundExit)
{
   var _thickness = Vector3.Distance(hit.point, _closestExit.point);
   double _residualVelocity = 0;
   double _penetrationDistance = 0;

   if (IsInLayerMask(hitCollider.gameObject, BallisticsMedium))
   {
       CalculateMaterialDamage(hitCollider, _thickness, _thickness * MToInches, ref _residualVelocity, ref
_penetrationDistance);
   }

   StartPenetrating(hitCollider, _thickness, (float)_penetrationDistance, (float)_residualVelocity,
_closestExit.point);
   return true;
}
```

Figure 9 - A section of code from the ballistic penetration functions illustrating how the projectile determines what it has hit and the thickness of it.

These various improvements greatly improved the realism of this system and were worth their development time however they did bring performance issues with them that took several days to improve and optimise. The issue with these nested raycasts is that they can be quite computationally expensive. In particular I was using debug draw rays so that I could see what was happening in the editor and found that all of these additional render calls resulted in frame times of several hundred ms if a bullet was selected in the editor while it was moving. To solve this issue I optimised the calls to the projectile's trail renderer, removed debug draw rays once the system was complete, and made sure not to select the projectile while it was moving. This issue was never present in builds however was inconvenient and time consuming to solve in development.

The next thing to do was building a simple raycast based weapons system to be used as a point of comparison as to how most games do ballistics. This will be used for the Single and Multi bar sections. This system was designed to be similar to one of my main points of comparison on this project which is Pavlov VR. In that game they use a system where materials can be tagged as either soft, hard, or impenetrable. Soft materials can be penetrated by any bullets, hard materials can only be penetrated by rifles, and impenetrable materials cannot be penetrated. This system is of course flawed, as it does not take into account material thickness or layered materials. To emulate this in my project I devised a system shown in Figure 10 which fired a raycast from the gun of an infinite length and returned all the colliders which it hit. It then progressed through these colliders in order of what was hit first with soft objects being ignored and impenetrable objects stopping the bullet and calling its damage functions on them. This results in a simple but fast and efficient raycast ballistics solution with some penetrable materials.

```
RaycastHit[] _hits = Physics.RaycastAll(transform.position, transform.forward, _maxDistance, _hitMask);
System.Array.Sort(_hits, (a, b) => a.distance.CompareTo(b.distance));

Vector3 _endPoint = transform.position + (transform.forward * _maxDistance);

foreach (var _hit in _hits)
{
   if (_hit.collider.TryGetComponent(out MaterialType _mat))
   {
       CalculateMaterialDamage(_hit.collider, projectileType);
   }

   if (_hit.collider.CompareTag("SoftPen"))
   {
       continue;
   }
   else
   {
       _endPoint = _hit.point;
       break;
   }
}
```

Figure 10 - A section of code for the simplistic raycast ballistics. This code is what allows it to penetrate some materials and not others.

Health and user experience

Finally I had to develop the health systems for use in this project. This all had to be done from scratch using the research conducted prior.

First I developed a generic framework of abstract classes for the health system to run on top of as can be seen in Figure 11. This means that a generic object with the correct colliders can serve as a standard single health bar, a more sophisticated multi weighted health bar system, or as my unique KBallistic health system. To do this an object has a range of colliders placed on each area of it including areas such as upper and lower arms, legs, and torso. This system has potential for greatly increased detail with elements such as nested bones or organs being possible in future renditions thanks to my ballistics now supporting composite materials.

```
public class UpperTorsoHealthCollider : HealthColliderBase
{
   protected override void Awake()
   {
       base.Awake();

       SignificanceMultiplier = 1f;

       VascularityBase = 0.0004f;
       VitalChance = 0.1f;
       VascularityVital = 0.02f;
   }
}
```

```
public class HealthSingleBar : IHealthMethod
{
   // consts
   private const float StartingTotalHealth = 100f;

   // public params
   public float CurrentOverallHealth { get; set; }
   public HealthSystem HealthSystem { get; set; }

   // functions
   public void Init()
   {
       CurrentOverallHealth = StartingTotalHealth;

       // hide all health uis
       HealthSystem.ShowIndividualHealths(false);
   }

   // the dict here is a reference to the one on HealthSystem
   public void TakeHit(ProjectileTypeSO projectile, HealthColliderBase healthCollider, Dictionary<HealthColliderBase, float> healthByCollider,  float velocity = 0f, double penetrationDistance = 0f, double woundVolumeCmCubed = 0f)
   {
       CurrentOverallHealth -= projectile.Damage;

       if(CurrentOverallHealth < 0f)
           CurrentOverallHealth = 0f;

       Debug.Log($"Took damage, current health overall: {CurrentOverallHealth}");
   }
   public void Reset()
   {
       CurrentOverallHealth = StartingTotalHealth;

       // hide all health uis
       HealthSystem.ShowIndividualHealths(false);
   }
}
```

Figure 11 - These two classes are examples of how new health colliders or systems can be implemented. This process is very simple because of the use of interfaces and abstract base classes.

To make the actual health systems I first made the single bar 0 - 100 range health system. This one is extremely simple and just subtracts a damage value from the projectile from the overall health value of the system. This could be extended to include separate weighted health for things such as the head and feet as many games often do, however for this example it made the most sense to do the most simplistic form of this type of health system. Next I worked on making a system for the multi bar health system.

For this system each one of the colliders has its own 0 - 100 health range and affects the main overall health by a weighted quantity depending on the significance value of this collider. This is most similar to tactical, slower paced games, such as Escape From Tarkov, however can also be used in a more simplistic form for the aforementioned single bar with weighted damage for head and feet. This system references back to a dictionary on the overall health system with each collider being the key for a health value of 0 - 100. Damage is done to these individually and then a significance multiplier is taken from each collider to determine how much damage it should do to the overall health. This system is not only extremely simple but also extremely effective and easy to add onto or remove elements from, it's a very good example of how a modular multi bar system can be done and also the benefits of these simpler health systems.

Finally I had to develop my own extremely detailed health system using a computation sheet based on my research shown in Figure 12. Though this would be based on the core of the multi bar system, it features its own unique blood loss based model which greatly changes how it works in game. The first problem with this was actually calculating how much blood being shot in different locations would lose. To do this I first took the wound cavity generation from my initial ballistic penetration of flesh which is based on the Q model (Schwartz, 2012).

$$MPC = \pi \times \left(\frac{1}{2} D_{cm}\right)^2 \times S \times \rho \times \Phi$$

I then used this to calculate the surface area of the wound and the estimated initial bleed rate using a derivation of the Bernoulli equations (Kragh et al., 2011).

$$Q = A\sqrt{\left(\frac{2\Delta P}{\rho} + v_2\right)},$$

This allowed me to calculate an initial bleed rate of a wound in milliliters per second. Next I calculate how the wound will clot over time using an exponential clotting function with a clotting constant and the time since impact (Tsiklidis et al., 2019).

$$y = Ae^{-kt}$$

This is important as even large wounds will slow their bleeding over time, with smaller wounds stopping much faster. Finally I calculate the overall blood loss so far in this limb, this value is then used to calculate whether or not this limb would be considered fully functional or not using a simple calculation of if a limb has lost more than 4% of the total initial blood volume of the person. Finally I use all of this to calculate the state of the person overall which allows me to classify their injuries into four classes (Hooper & Armstrong, 2022). Class one is a loss of 15 percent of total blood volume resulting in minimal change to the person. Class two at 15 to 30 percent blood loss results in a state of shock with increased heart rate, at this point a person would be fully conscious still but likely becoming anxious. Class three is a 30 to 40 percent loss of total blood volume. This results in significant changes in blood pressure and mental state and can cause loss of consciousness. Finally a class 4 injury at over 40% loss of blood can result in highly increased heart rates, greatly reduced mental function and likely death if the injury is not treated soon (Hooper & Armstrong, 2022).

Figure 12 - An image of the computation spreadsheet created to test my chosen formulae and calculate performance of them.

In the case of the system I use here for now it is just deemed that 40% blood loss is equivalent to death and the players overall health is represented on a scale from 0 - 40% blood loss with 0 being 100 health and 40% being 0.

Once all of these formulas had been found and decided upon the main difficulty was just in converting the spreadsheet maths into C# shown in Figure 13, and then putting this into a health system inspired by that of the multi bar one. This was actually a surprisingly simple process as most of the mathematics being done here is extremely simple with the only real issues encountered being related to unit conversions for the various formulas in use.

```
(double volumeMCubed, double penetrationM, double radiusM, double surfaceAreaMCubed) Calculate3DProperties(double 
penetrationDistance, double woundVolumeCmCubed)
{
   double _volumeMCubed = woundVolumeCmCubed / 1000000;
   double _penetrationM = penetrationDistance;
   double _radiusM = Math.Sqrt(_volumeMCubed / (Math.PI * _penetrationM));
   double _surfaceAreaMCubed = 2 * Math.PI * _radiusM * _penetrationM;

   return(_volumeMCubed, _penetrationM, _radiusM, _surfaceAreaMCubed);
}

double CalculateInitialBleedRate(double surfaceAreaMCubed, double vascularityCoefficient)
{
   const double _meanArterialPressure = 13332;
   const double _bloodDensity = 1040;

   double _initialBleedRateM3S = (surfaceAreaMCubed * vascularityCoefficient) * math.sqrt((2 * _meanArterialPressure) 
/ _bloodDensity);
   double _initialBleedRateMls = _initialBleedRateM3S * 1000000;

   return _initialBleedRateMls;
}

double CalculateClotting(float timeSinceImpact, double initialBleedRateMls)
{
   double _currentBleedRatemls = initialBleedRateMls * math.exp(-ClottingConstant * timeSinceImpact);

   if(_currentBleedRatemls < 0.1)
       _currentBleedRatemls = 0;

   return _currentBleedRatemls;
}

double CalculateBloodLoss(WoundProperties wound)
{
   double _bloodLossMlsInTime = (wound.InitialBleedRate / ClottingConstant) * (1 - math.exp(-ClottingConstant * 
wound.TimeSinceWound));

   return _bloodLossMlsInTime;
}
```

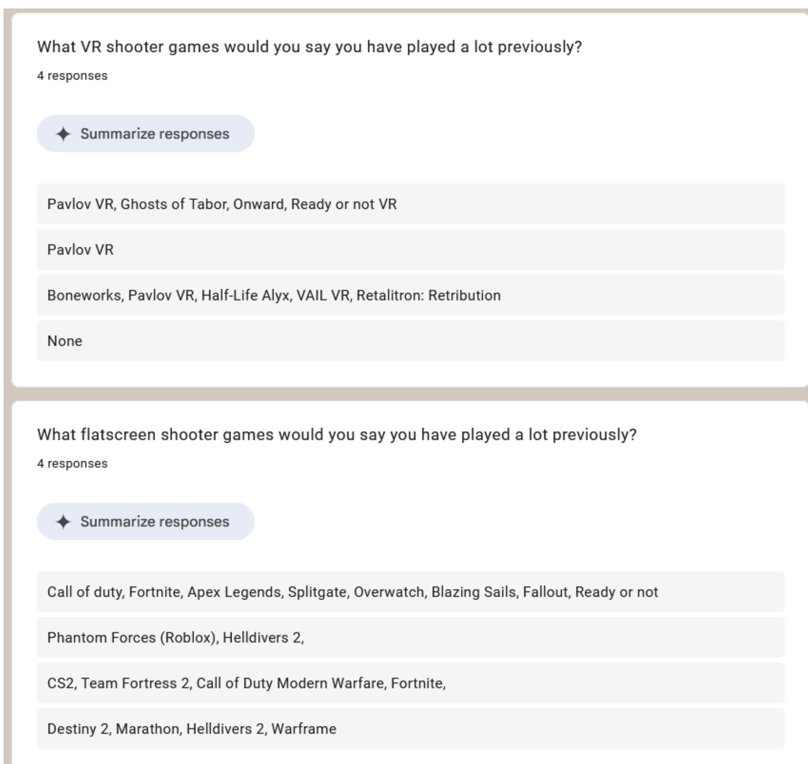
Figure 13 - The sections of code used for most of the calculations in KBallistic health converted over from the computation sheet into C# code.

At this point all of the game's health systems were complete and tied into ballistics, and the only remaining elements were related to the user experience. These include modelling of the structures for use in the game, the addition of various UI elements such as text explaining the use of areas, and a settings system for each range allowing the player to choose the cartridge used by their weapon. All of this was implemented with very little difficulty or issues resulting in the artefact being completed on time.

Results

Surveys were conducted on four participants within the target audience of age range 16-49 using PCVR systems. Users were presented with a VR scene with three identical shooting ranges in a line each with written text explaining their functionality and differences. The first one is referred to as the single bar range and uses simple raycast ballistic physics along with a single overall health bar for enemies. The second range is referred to as the multi bar range and uses the same simple raycast ballistic physics with the addition of multiple health bars on enemies. These health bars are on each vital area of the enemy and affect their overall health in different ways. Finally the last range is referred to as the KBallistic range and uses the physics and health system which was the primary focus of this paper. This range uses complex ballistic physics allowing for material penetration along with an adaptation of the multi bar health system which allows for dynamic bleeding and blood clotting based on injury locations. Users were instructed to play through the demonstration and then complete the online form using Google Forms.

Previous games played


What VR shooter games would you say you have played a lot previously?

4 responses

Summarize responses

Pavlov VR, Ghosts of Tabor, Onward, Ready or not VR

Pavlov VR

Boneworks, Pavlov VR, Half-Life Alyx, VAIL VR, Retalitron: Retribution

None

What flatscreen shooter games would you say you have played a lot previously?

4 responses

Summarize responses

Call of duty, Fortnite, Apex Legends, Splitgate, Overwatch, Blazing Sails, Fallout, Ready or not

Phantom Forces (Roblox), Helldivers 2,

CS2, Team Fortress 2, Call of Duty Modern Warfare, Fortnite,

Destiny 2, Marathon, Helldivers 2, Warframe

Figure 14 - Three out of four participants reported prior experience with VR shooter titles. All participants have prior experience with flat screen shooter games.

Hardware used

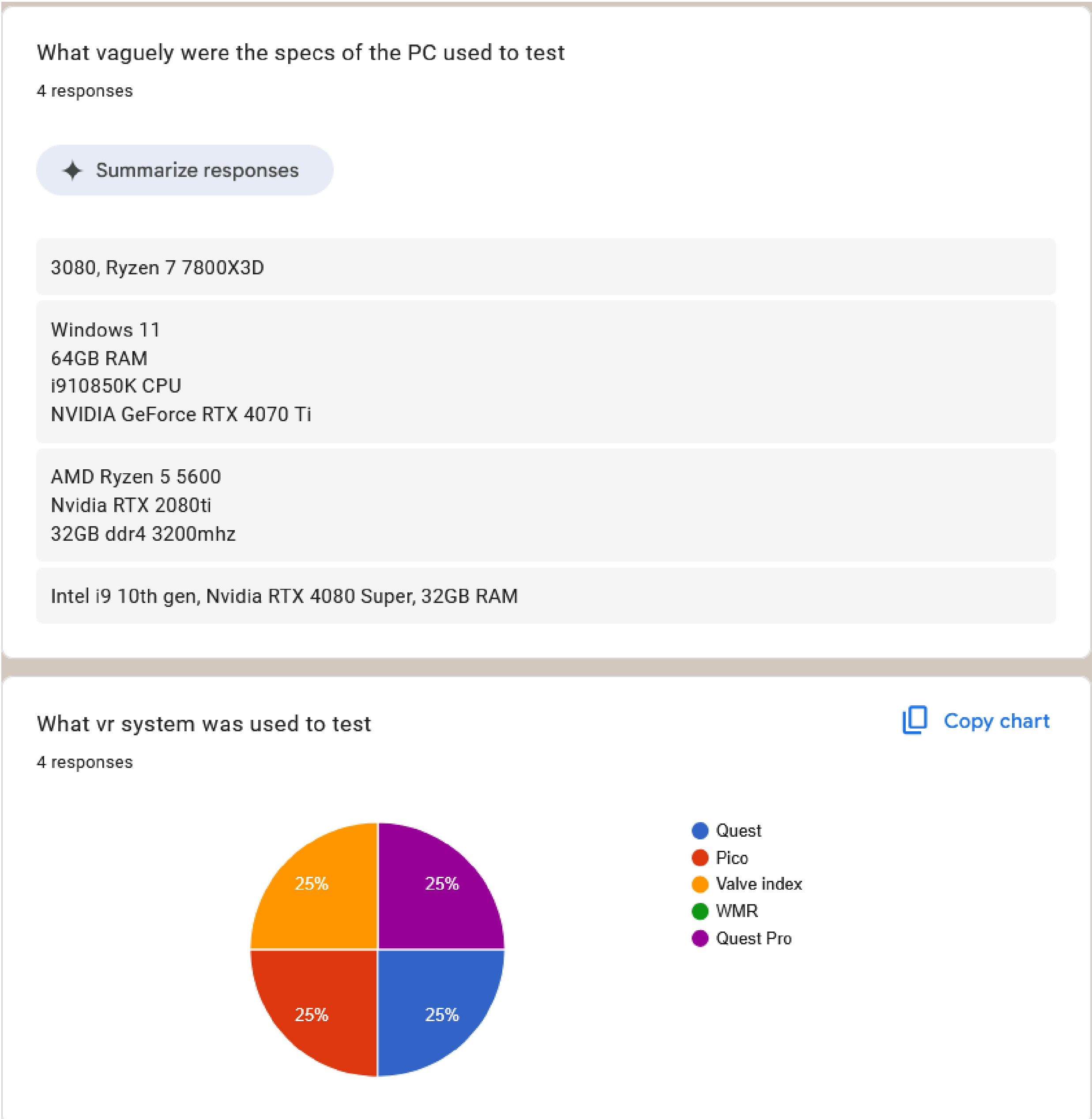


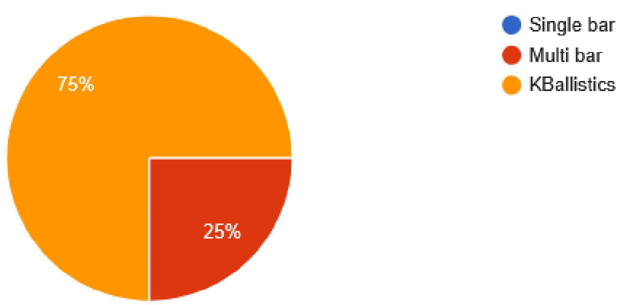
Figure 15 - Participants used a range of VR systems. All of which were powered by high end consumer PC hardware.

Overall preference

Range preferences

Which shooting range did you prefer overall?

4 responses



What specific aspects influenced your preference (e.g. weapon feel, feedback, realism, clarity)?

4 responses

Summarize responses

The realism of K ballistics and the logical consistency of the physics was extremely cool to see

I felt that the Multi bar range seemed to work the best to me, as I noticed that the bullet tracing and hitbox stayed for the KBallistics range and the Single bar range was very simplistic.

The bleeding and clotting system in the Kballistics section was much more interesting than the other two sections as it was more akin to real life. the system provided, in my opinion, a fairer approach to health systems in games. The ballistic physics was also a welcome touch as one thing I don't enjoy about games is when I'm unable to shoot through surfaces such as thin wooden boards or fences, and this system provides a solution to this issue. the weapon handling in all three sections was about the same as the dummies were placed in the same areas in each.

I particularly enjoyed how the different materials in the range (Types of walls, and enemies) affected the bullet trajectories. I feel as though this could be used to great effect when creating maps, as players can learn where walls can be shot through and can use this knowledge to conduct more engaging and unique gameplay. Combined with the advanced healthbar system of this range, these systems work in conjunction to create a sense of heightened realism which, especially in a VR game, will likely result in increased player engagement as they learn to employ new tactics and playstyles indicative of a high level of emergent gameplay and deeply enjoyable experience

Were there any elements from the other ranges that you would have liked to see included in your preferred range?

4 responses

✦ Summarize responses

I liked the individual weighted healthbar system a lot - I feel like the basic all in one healthbar is frustrating when you lose a lot of HP even when hit in the foot, however I feel like the bleed system would only feel health in gameplay in a realistic hardcore tactical survival shooter

I think the ballistics gel was quite cool, and it would be interesting to see how it could be expanded on.

I would like to see the ballistic gel in each of the sections rather than just the kballistics one as it provided an insight into the calculations behind each shot, especially when shooting through materials inside the ballistic gel.

One factor that slightly concerned me was how a single shot to the head can instantly eliminate a player, unlike in the Single Bar range where it takes two or three. Whilst this may be more realistic, my concern is that if headshots are easy to perform, being eliminated so quickly may detract from the gameplay experience. However, this may be an opportunity for further realism - if players are given helmets which, utilising the KBallistics system of calculating bullet range and speed drop off, could slow or stop a bullet before it enters a player's head, resulting in a slower bleed-out time and giving players a chance to react

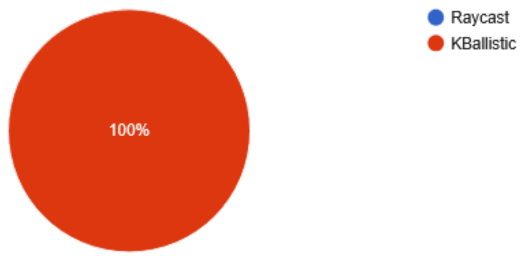
Figure 16 and 17 - Three participants indicate a preference for the KBallistic range whilst one participant prefers the multi bar range. Participants highlight a range of features from the KBallistic section including the ballistic gel demonstration. Some participants also report confusion regarding some elements of the test ranges.

Ballistic systems

Which ballistic system did you prefer overall?

4 responses

Copy chart



What made this system feel better than the other?

4 responses

Summarize responses

- More consistent to logic and common sense, what id imagine a bullet could do was consistent with the actual gameplay
- I felt that overall KBallistic felt more consistent, though the tracing staying around and dealing damage wasn't expected.
- The raycast system felt too simplistic for a VR game, especially at long distances as there was no bullet dropoff which felt unrealistic. On the other hand, the Kballistic system provided a significantly more realistic simulation of bullet dropoff which helped boost the overall experience of the game.
- The realism of how shots are affected by the type of material and how much of it they travel through allowing shots from close range or better angles to penetrate materials they would not be able to in the raycast system.

Did either system feel more realistic or responsive? Please explain why.

4 responses

Summarize responses

- KBallistics was far more realistic and immersive, for example I see a bit of concrete and go "A bullet would never go through that" and what I believe to be the case always is true, sometimes for other ballistic systems easily penatratable objects are not penetrated, or unrealistically large objects are penetrated with ease
- Neither system felt majorly more responsive nor realistic, but I do believe that KBallistic seemed to almost always act as I would have expected while using the pistol.
- Both systems felt about the same when it came to responsiveness due to the lack of long-range dummies.
- The KBallistic system felt more real, as the bullets would react more dynamically with the world around them.

Figure 18 - All participants preferred the ballistic behaviour in the KBallistic range over the raycast based systems. Reported reasons include perceived realism and expected bullet behaviour. One participant reports that bullet tracers in the KBallistic range appeared to persist and deal damage.

Ballistic improvements

For your preferred ballistic system, what changes or additions would improve the experience?

4 responses

✦ Summarize responses

N/A

I believe that removing or fixing the tracers staying behind and dealing damage would improve the KBallistics system. Overall, it would also be nice if there was some kind of feedback in the future as generally it was hard to tell how the bullet was affected by materials at times. I felt no difference in any of the ammo options personally. Simple sound and markings along the bullet's path would go a long way.

adding more targets further away would help a lot as it would give an opportunity for the game to demonstrate the differences between the raycast and Kballistics systems at long ranges which is where both systems differ the most. A scope for the large weapon would also be a welcome improvement as it would help me see targets that are further away.

For the KBallistics system, it could be improved by adding effects to the game world where bullets interact with it, in order to give visual (and potentially audible) feedback to players when shots either do or do not penetrate a material. This could be in the form of bullet holes and/or marks when something gets hit. On top of this, an on-screen or audible cue when a player is hit through a wall could help players take a greater advantage from the wall-penetration system, although the balance of such a feature would need to be considered.

For the other ballistic system, what changes could make it more enjoyable or usable?

4 responses

✦ Summarize responses

N/A

I believe that overall raytracing would also benefit from feedback from the bullet, visual and audible. I could not tell any difference in ammo, which would be fixed by the prior suggestion as well.

nothing really, the system is fine as it is.

I feel as though my suggested improvements to the KBallistics system could also be applied to the raycast system. On top of which, the system could be improved with some kind of variable bullet fall-off, potentially a middle ground between the current implementation and the full KBallistics system.

Did you notice any inconsistencies, unexpected behavior, or learning curve issues with either system?

4 responses

Summarize responses

The differing bullet velocities upon exit of objects will require some time to adjust to

The KBallistics system seemed very reliable with the pistol, though I did have some difficulties with the rifle.

For Raytracing, I felt that it did not act how I expected with materials at times. I do not know if this was due to me taking guns between the first and second areas or if it was due to other factors.

In addition, I felt that the rifle was difficult to use both for trying to find where to grip with my secondary arm and for not being comfortable to use overall. The pistol was very nice, though I would have liked to be able to grip with my secondary arm.

nothing of note for either system, they worked well.

The KBallistics system certainly presented a greater learning curve than the raycast system, as I had to learn not only which materials a gun could penetrate, but also how different bullet types would interact with each material at different angles. Overall I believe this to be a good thing however, as it raises the skill ceiling for players without particularly affecting the skill floor, as at a baseline players can still effectively play within the system without needing to shoot through any walls.

Figure 19 and 20 - Participants suggested the need for increased visual and auditory feedback when indicating bullet penetration and damage. Some participants note that the KBallistic system may be harder to learn initially. One participant reported issues with the bullet tracer behaviour and rifle functionality whilst others indicated general satisfaction with how the systems worked.

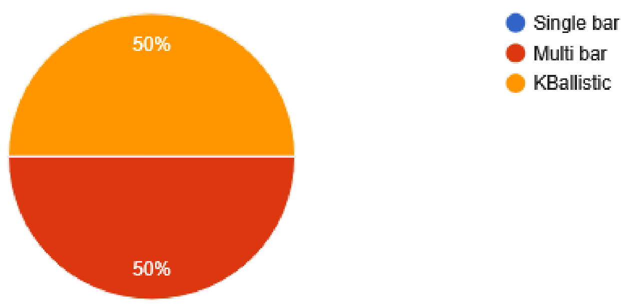
Health systems

Health Systems

Which health system did you prefer overall?

Copy chart

4 responses



What made this system better than the others?

4 responses

Summarize responses

In my opinion, for the vast majority of games multi bar is the goldilocks zone so to speak. The all in one health bar is frustrating to play with as getting shot in the foot does a lot of damage, the bleed system feels overly punishing and one single bullet is pretty much garunteed to kill you whereas the multi bar system seems like a good middle ground

I would not say I significantly preferred any health system over another, but while KBallistic felt the most responsive, it did not always act how I expected with the health.

Introducing bleed rates for the dummies made the game feel more rewarding as my shots were dealing lasting effects to the dummies which made the game more interesting and set it apart from other shooter games where you get shot once and stop bleeding instantly.

I preferred the KBallistics system as it feels far more realistic than the other systems. The idea of health-loss over time determined by what area of the body gets hit is a novel and interesting one which I believe could be used effectively to create engaging and interesting gameplay

Did any system feel confusing or less intuitive? Why?

4 responses

Summarize responses

The bleed was difficult to understand exactly what changes the bleed rates

I felt that none of the systems were confusing or less intuitive. They all mostly acted as I would expect of a health system.

all of the systems felt intuative as they are similar to ones in other games I have played.

The multi-bar system was the least intuitive system, as it was often unclear how much damage would be dealt by a hit to any individual hurtbox, and even more so how combinations of hits in different locations might add up to effect the overall health of the enemy.

One might think that the KBallistics system would suffer from a similar issue - however, I found the added element of the bleed-out time to mitigate this issue. This is because you can always be sure that any hit will cause some amount of bleeding out, and a hit to a limb is quite intuitively going to cause slower bleed-out than a hit to the chest or neck, for example.

Which system felt more realistic or immersive and why?

4 responses

Summarize responses

The bleed system is more realistic, however the multi bar systems feels more intuitive and immersive

The second and third systems felt the most immersive. The first system was very simplistic and overall less realistic.

The Kballistic system felt more realistic due to the bleeding and clotting mechanics and per-limb health zones. one gripe I have with most games is that when you get shot in the toe 3 times you die, whereas with this system, you slowly bleed out of your foot but don't die instantly which made me more conscious about where I was landing my shots.

The accuracy of the KBallistics system certainly made it feel much more realistic, as it is closer to how players could expect people to react when shot. This is particularly prevalent in a VR experience, where this added realism has a positive impact on immersion.

Figure 21 and 22 - Three participants preferred the KBallistic health system with one preferring the multi bar system. Participants describe the KBallistic system as feeling more realistic and immersive. One participant reported difficulty understanding factors affecting bleed rates. Another participant reports confusion on how damage affects overall health in the multi bar system.

Health improvements

For your preferred health system, what changes or additions would improve the experience?

4 responses

✦ Summarize responses

Slower bleed rates on bleed health and only a couple seconds of main bleed then slows down to a near stop

Mainly better visual indication of damage to particular parts or where you exactly hit, as it was hard to see at times. I think it would be nice if overall damage was increased more on shooting the head.

I would like to see a visual or auditory indicator when the dummy has died as it was hard to read the text from a distance.

I think that a method of blood-loss prevention could be valuable to allow players to recover from a bleed that would otherwise be fatal. This could force players to be more careful in their playstyle, having to decide between retreating to patch wounds or remaining in an engagement, with the consequences of both decisions amplified more so than in the other systems.

For the other health systems, what changes could make it more enjoyable or usable?

4 responses

✦ Summarize responses

N/a both do as intended but just some less intuitive

The first system was too simplistic to be realistic or as immersive, the third system interacted oddly with the tracing lines but otherwise seemed to work well. I believe that as with the second system, the third system would also benefit from more indication.

same again, having an indicator for when the enemy has died would make the experience better overall.

Other than balancing changes, a factor potentially more relevant in the less realistic scenarios utilising the Single or Multi bar systems as opposed to the KBallistics system, the main improvement I would suggest for them is to improve realism, as such pushing any other system closer to how the KBallistics system functions

Did you notice any inconsistencies, unexpected behavior, or learning curve issues with any system?

4 responses

✦ Summarize responses

N/A but bleed is a bit challenging

I did not notice any major issues with the health system.

nothing to note.

Both the multi-bar system and the KBallistics system presented some learning curve required to fully understand their respective interactions with the ballistics system utilised in each scenario - but I did not feel as though I had any issues with either. However I can see how players in a live gameplay scenario might struggle to grasp the intricacies of these systems, especially the KBallistics system. However, with a shooting range concept similar to that in the demonstration I believe that either of these systems can be effectively understood.

Figure 23 and 24 - All participants suggested improvements to the visualisation of bleeding and damage with two participants suggesting reducing or allowing for mitigation of blood loss. Two participants reported no issues with the health systems whilst two expressed concerns regarding the learning curve of the KBallistic health system.

Final thoughts

Final thoughts

What stood out most during your experience?

4 responses

✦ Summarize responses

KBaillistics

I think that the ballistics gel was one of the coolest parts to me, the absence of the actual bullet in games has always been a bit immersion-breaking to me and the gel gives a great demonstration of realistic objects that can be used in a game instead of simple metal or basic targets for weapons.

the ammo type selector is very confusing to me as I don't know a lot about ammo types and don't know what the numbers mean. I couldn't tell if they were arranged from smallest to largest or what their properties were. having some sort of indicator for this would help the experience greatly for players like me.

Something that particularly stood out to me was that in the KBallistics scenario was how bullets shot into a material they would normally pass through could be stopped completely when entering the material at a small acute angle. I also found how bullets can over-penetrate enemies to be interesting - at one point a bullet shot into one enemy passed through two walls behind it and grazed another enemy, causing them to start to slowly bleed out. I believe this kind of interaction is precisely the kind of new and interesting gameplay players can hope to enjoy from a system striving for such realism as KBallistics.

Any additional comments or suggestions?

1 response

the similarity of the sections was confusing and I got a bit lost while playing through them. changing the layout of each section might help. The text was also quite hard to read. the peroformance of the game was very good overall though and was locked to 90fps the entire time.

Figure 25 - Participants overall reported interest in the KBallistic system, particularly its ability to simulate bullet penetration through multiple targets and materials. One participant reported confusion regarding ammunition types. One participant commented positively on overall system performance whilst also noting issues with the layout of the demonstration overall.

Reflection

A broad overview

The goal of this artefact and project was to create a health and ballistic system for use in VR to improve realism whilst maintaining performance and improving immersion. The project overall has absolutely succeeded at this. An extremely unique and innovative set of health and ballistic systems were made which function as intended in VR. The artefacts performance was more than adequate for testers. And overall testers preferred the systems that had been developed compared to the current industry standards.

At a high level the KBallistic set of systems increased immersion for players by giving them a greater sense of realism when using weapons and shooting enemies without a large performance cost or too much of an increase in perceived complexity.

Reflection on project management and methodology

One of the major issues over the course of this project was absolutely scope and time management. Several early intended features had to be cut to deliver the artefact on time with elements such as full body inverse kinematics and interactive weapons proving too complex and time consuming to complete in a reasonable manner. These features consumed significant development time without directly contributing to the main focus of the project's research.

Initially the project had been planned out on a Trello board and split into different sections with the intention of taking an Agile approach to development and iteratively developing and refining systems within small development cycles. However, as a result of inconsistent time management and insufficient scope control, the project switched to more of a Waterfall methodology where systems were worked on sequentially with the focus shifting to the delivery of a functional minimum viable product. This shift in methodology ultimately improved the project's progress substantially and resulted in it being completed on time.

Finally the project was rescoped with several secondary systems being cut down such as additional weapons, advanced animation systems, and enemy ai agents being removed entirely. This shrinking in scope allowed development to be focused again on the core mvp and on the project's research. In hindsight a defined MVP earlier in the development alongside clearer prioritisation of the project would have prevented substantial time loss and reduced work on non essential systems.

Reflection on literature review and research

My research into how other games handle ballistics was extremely useful. The insight into how they handle material penetration and calculating material thickness was particularly useful in my development as I ended up employing a somewhat similar system to some of the games researched.

To improve upon my literature review I should have conducted more research into material penetration to find more suitable approaches for different materials such as concrete and plastic. Using the ones I currently have gives a very limited set of general materials to use and it would've improved the systems detail by supporting more. Another thing which I should have researched more into was projectile deflection and deformation. I did not end up implementing projectile deflection and if more research had been done it may have been possible that I found a simpler solution which was more reasonable to implement and use.

Evaluation and reflection of artefact development

Overall the artefact development was a success and resulted in a functional set of systems that effectively meet the criteria of this project. Despite this, several elements of the project were substantially more challenging to develop than originally anticipated. One of the best examples of this is the attempt at implementing full body inverse kinematics for the player's character. Initially this system was only expected to take a few days to complete however integrating it with the existing animation and gameplay systems was substantially more complex than expected. Around a week of development time was spent trying to develop this feature before it was ultimately discarded to focus on the systems more aligned with the project's research goal.

In hindsight this issue really highlights the weaknesses with both planning and feature prioritisation on this project. If the project timeline had sufficiently accounted for failure and technical difficulty it would have let more time be allocated to critical features such as adding more depth to material penetration, and implementing systems such as projectile deflection and ricochets.

Despite these issues the remaining systems were delivered well with the ballistic and health systems being a particular highlight. The modular systems which they rely on made development and integration extremely simple and helped allow for rapid iteration and tuning. One of the strongest aspects of the project was the implementation of the modular ballistic system which was capable of dynamically switching between entirely different ballistic methodologies at runtime. This greatly improved development efficiency and also allowed test participants to directly compare traditional ballistic approaches with the more advanced KBallistic systems during evaluation.

The health systems also benefited substantially from this modular structure. A single set of base systems was able to support multiple different approaches including single bar, multi bar, and the physiologically detailed KBallistic health system. This flexibility made balancing and testing substantially easier and demonstrated that more detailed health systems can be implemented without making the overall experience excessively complicated for players.

User feedback during testing further reinforced the success of the artefact. Participants consistently preferred the KBallistic projectile and penetration systems over the more traditional approaches presented during testing. In particular the ballistic gel demonstrations and penetration mechanics were highlighted as improving immersion and realism. Feedback on the health systems was more divided, however participants still generally found the systems more engaging and interesting than existing methods as a result of the increased level of detail and interaction.

Performance was also a major success of the project. None of the participants reported noticeable performance problems during testing despite the increased complexity of the ballistic systems, with one tester specifically noting stable framerates throughout use. This demonstrates that the systems developed were not only functional but also sufficiently optimised for use within VR environments which was one of the primary goals of the artefact.

Current limitations and future work

Whilst the overall project was a success, there are still several limitations which remain, with the primary set of issues being the ballistic mechanics in use. Though the current ballistics have accurate and realistic penetration, there is only a small range of materials supported. Furthermore, there is currently no support for projectile deflection or ricochets, which are normally core elements of how projectiles react with materials. Currently, projectiles either penetrate or stop without accounting for fragmentation or nose deformation. These features would have substantially improved realism and detail if implemented, however the technical complexity made them impractical to complete within the scope of this project.

Another area in which the project fell short was the user experience. The addition of AI enemy agents and full body inverse kinematics would have greatly improved player immersion and allowed users to further test and experience the ballistic systems available to them. The inclusion of these features would have allowed more testing, better gathered data, and an overall more full and conclusive artefact in comparison to what was ultimately produced. In the future, this could also be improved with particle effects and auditory stimuli to make it far more apparent to the user what was happening, along with a more refined way of showing health such as a radial display on limbs or clothing progressively saturating with blood. Finally, the text which explained the tests to the participants was challenging to read in VR and was generally a flawed way of delivering information to them. In the future, this artefact would greatly benefit from integrated short video guides showing how the systems function, alongside some form of more refined tutorial.

Finally, there are several areas which this artefact has not touched on, primarily multiplayer support and further detail within the health system, most importantly healing. The artefact did not touch on multiplayer support at all or how these systems would function when scaled to the level of a full game and, even though they would work conceptually, this is an area which requires further research. Health is another area which has been explored but not fully developed. There is potential to add a substantially higher level of detail such as bone breakages, internal organs, and a range of ways to heal yourself and others through mechanics such as tourniquets and bandages. All of these systems would greatly improve the level of detail shown in the health system and would make it far more interesting to use.

Conclusion

The aim of this project was to design and implement a detailed health and ballistic system for use in VR experiences to improve perceived realism. This aim was achieved through the development of the KBallistic system and its accompanying health systems.

The results from testing show that participants generally preferred the KBallistic approach over traditional raycast based systems primarily due to the perceived realism of penetration mechanics and the detail within the health systems. Performance remained stable and consistent across all tests showing that the increased ballistic detail can be achieved without negatively impacting usability in a VR title.

While the system is effective and meets the set criteria in its current form, it still has limitations pertaining to material variety, projectile behaviour, and the overall user experience. Whilst the surveyed group is small, valuable insight was still gained about ballistics in games and there is a clear set of solutions posed in this artefact to be expanded upon in the future.

Overall the project effectively demonstrates that more physically detailed ballistic and health systems can improve immersion in VR without an unacceptable loss of performance.

References

Arti Sergeev. (2020, July 17). *How Does Shooting Work in Games: Hitscan and Projectile Ballistics*. 80.Lv; 80lv. https://80.lv/articles/how-does-shooting-work-in-games-hitscan-and-projectile-ballistics

Battlestate Games. (2019). *Save 15% on Escape from Tarkov on Steam*. Steampowered.com. https://store.steampowered.com/app/3932890/Escape_from_Tarkov/

Beyond Frames. (2023). *Ghosts of Tabor on Steam*. Steampowered.com. https://store.steampowered.com/app/1957780/Ghosts_of_Tabor/

Bishop, R. F., Hill, R. E., & Mott, N. (1945). The theory of indentation and hardness tests. *Proceedings of the Physical Society*, *57*(3), 147–159. https://doi.org/10.1088/0959-5309/57/3/301

CS2 Official. (2025). *Steam Community :: Guide :: How does bullet penetration work?* Steamcommunity.com. https://steamcommunity.com/sharedfiles/filedetails/?id=275573090

Escape From Tarkov. (2026, April 23). *Ballistics*. Escape from Tarkov Wiki. https://escapefromtarkov.fandom.com/wiki/Ballistics

Fandom. (2025). *Ballistics*. Ghosts of Tabor Wiki; Fandom, Inc. https://ghosts-of-tabor.fandom.com/wiki/Ballistics

Forrestal, M. J., & Warren, T. L. (2008). Penetration equations for ogive-nose rods into aluminum targets. *International Journal of Impact Engineering*, *35*(8), 727–730. https://doi.org/10.1016/j.ijimpeng.2007.11.002

Hooper, N., & Armstrong, T. (2022). *Hemorrhagic Shock*. National Library of Medicine; StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK470382/

Karpen, S. C. (2017). Biased self-assessment: Causes, consequences, and countermeasures. *American Journal of Pharmaceutical Education*, *82*(5), ajpe6299. https://doi.org/10.5688/ajpe6299

Kaur, G., Mukherjee, D., & Moza, B. (2023). A Comprehensive Review of Wound Ballistics: Mechanisms, Effects, and Advancements. *International Journal of Medical Toxicology & Legal Medicine*, *26*(3and4), 189–196. https://doi.org/10.5958/0974-4614.2023.00071.2

Kragh, J. F., O'Neill, M. L., Beebe, D. F., Fox, C. J., Beekley, A. C., Cain, J. S., Parsons, D. L., Mabry, R. L., & Blackbourne, L. H. (2011). Survey of the indications for use of emergency tourniquets. *Journal of Special Operations Medicine: A Peer Reviewed Journal for SOF Medical Professionals*, *11*(1), 30–34. https://doi.org/10.55460/36SQ-1UQU

L. KOENE, & F.R. BROEKHUIS. (2017). Bullet Penetration into Wooden Targets. *29th International Symposium on Ballistics*. https://doi.org/10.12783/ballistics2017/16976

Lucas, C. P. (1992). The order effect: reflections on the validity of multiple test presentations. *Psychological Medicine*, *22*(1), 197–202. https://doi.org/10.1017/s0033291700032852

OWO Game. (2026, May 8). *OWO Haptic Vest – Feel Every Impact In-Game | OWO Game*. OWO Game. https://owogame.com/product/original-edition-kit/

Pimax. (2017, September). *Pimax | VR Headsets, HMDs & Equipment*. Pimax.com. https://pimax.com/

quarkez. (2020, June 15). *Reddit - The heart of the internet*. Reddit.com. https://www.reddit.com/r/VALORANT/comments/h9npsl/valorant_wall_penetration_explained_and_why_it_is/

Ramirez, S. (2026, May 7). *Virtual Reality Statistics 2026: Market Size, Users and Data*. SQ Magazine. https://sqmagazine.co.uk/virtual-reality-statistics/

Riot Games. (2020, June 2). *VALORANT: Riot Games' competitive 5v5 character-based tactical shooter*. Playvalorant.com. https://playvalorant.com/en-gb/

RUST LTD. (2016, April 5). *Hot Dogs, Horseshoes & Hand Grenades on Steam*. Store.steampowered.com. https://store.steampowered.com/app/450540/Hot_Dogs_Horseshoes__Hand_Grenades/

Schwartz, C. (2012). *Quantitative Ammunition Selection*. iUniverse.

Sony. (2023, February 22). *PlayStation®VR2 | The next generation of VR gaming on PS5*. PlayStation. Retrieved May 8, 2026, from https://www.playstation.com/en-gb/ps-vr2/

Tsiklidis, E. J., Sinno, T., & Diamond, S. L. (2019). Coagulopathy implications using a multiscale model of traumatic bleeding matching macro- and microcirculation. *AJP Heart and Circulatory Physiology*, *317*(1), H73–H86. https://doi.org/10.1152/ajpheart.00774.2018

U.S. ARMY CORPS OF ENGINEERS. (2018). *UFC 4-023-07 - DESIGN TO RESIST DIRECT FIRE WEAPONS EFFECTS | GlobalSpec*. Globalspec.com. https://standards.globalspec.com/std/14491770/ufc-4-023-07

Unity Technologies. (2024). *Create high-quality graphics and stunning visuals | Unity HDRP*. Unity. https://unity.com/features/srp/high-definition-render-pipeline

Valve. (2004, November 1). *Counter-Strike: Source on Steam*. Store.steampowered.com. https://store.steampowered.com/app/240/CounterStrike_Source/

Valve. (2023, September 27). *Counter-Strike 2 on Steam*. Store.steampowered.com. https://store.steampowered.com/app/730/CounterStrike_2/

Valve. (2025, November 12). *Steam Frame*. Steampowered.com. https://store.steampowered.com/hardware/steamframe

Vankrupt Games. (2017, February 17). *Pavlov on Steam*. Store.steampowered.com. https://store.steampowered.com/app/555160/Pavlov/

Walker, J. D. (2021). The Tate—Alekseevskii Model. *Cambridge Core*, 333–370. https://doi.org/10.1017/9781108684026.009

Wang, J., Shi, R., Zheng, W., Xie, W., Kao, D., & Liang, H.-N. (2023). Effect of Frame Rate on User Experience, Performance, and Simulator Sickness in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, *29*(5), 2478–2488. https://doi.org/10.1109/tvcg.2023.3247057

Wikipedia Contributors. (2019, September 23). *Janka hardness test*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Janka_hardness_test