



Deadlift: A VR Calisthenics Experience

3D Games Programming Assignment

Kingston
University
London



GIORGOS KARAMBASIS RODRIGUEZ

K2376268

30/11/2024

Contents

1. Introduction	2
2. Techniques, Algorithms and Tools	3
2.1 Project Set-Up	3
2.2 Non-Player Characters	4
2.3 Hand Gestures & Spells.....	4
2.4 Body-Pose Detection.....	5
3. Evaluation	6
3.1 Methodology.....	6
3.2 Quantitative Results.....	6
3.3 Qualitative Results	7
3.4 Conclusions	7
4. Discussion of performance and optimisation techniques.....	8
5. Overview of LLMs and Gen-AI tools Used and How	8
6. References.....	9
7. Appendices	11

1. Introduction

Deadlift is a medieval fantasy VR arena fighter game that combines entertainment with fitness, specifically calisthenics. It is developed in Unity and built for Quest 3 due to its hand-tracking and body-tracking support.

Calisthenics is a term that describes generic bodyweight exercises. It has been proven to be an effective means of improving physical strength, and posture, without the need of any gymnastic equipment (Thomas, Bianco, Mancuso, Patti, Tabacchi, Paoli, Messina and Palma, 2017).



Figure 1: Player casting the Shockwave spell while surrounded by Enemy NPCs

The player takes the role of a Necromancer tasked with fighting waves of enemies assaulting their castle. They use their powers to dispatch them, powered by a source of magic called *Aether*. *Aether* can only be regenerated via calisthenics exercises, in the form of push-ups, squats and planks.

Deadlift comes with 2 game modes; Enemy Waves and Plank. Enemy Waves involve rounds of enemies attacking the player while they try to defend themselves using 4 spells cast with hand gestures; Teleport, Fireball, Shockwave and Raise Dead. Once the player defeats all enemies, they emerge victorious.

The Plank game-mode is a mini-game that involves prompting the player to do a plank exercise. The longer the player holds the plank, the higher the score they receive, which translates to their Max Health and *Aether*. They can then head into the Enemy Waves game mode with higher stats.

This game aims to entertain & improve players' physical health by promoting muscle-building exercises. While fitness games in other forms of media have been made before (Nintendo, 2008. Ubisoft, 2010. Nintendo, 2019) as well as in VR (*Beat Games*, 2018), no VR fitness game has fully gamified calisthenics, or muscle building exercises. This project aims to break new ground in this domain via utilising body-pose recognition.

The objectives are to achieve this aim through Meta Quest 3's hand tracking & full body tracking capabilities. Hand tracking was used for spell casting using Meta's All-in-One SDK, while body tracking was used for exercise recognition with Oculus Movement SDK (Meta, 2024).

2. Techniques, Algorithms and Tools

2.1 Project Set-Up

For a quick VR Project set-up, the Meta XR All-in-One SDK was used. This SDK was used because it provides straightforward and compatible solutions to body tracking & hand gesture recognition that is available with the Quest 3.

Through its building blocks, hand tracking, movement and full-body tracking were achieved.

A base script to control all the player-related variables and methods was created, named *PlayerController* (Table 1). This script contains all necessary methods and variables that affect the player's Health, Aether and Death State.

Consecutively, two Game-mode scripts were created. The *EnemySpawner* class contains an algorithm for spawning enemies while only allowing 3 enemies to be alive at any time (Figure 2).

As for the *PlankGameMode* script, when it is enabled, the longer a player holds the pose, the higher their *maxHealth* and *maxAether* are going to be, as seen in Figure 2. The conversion rate from pose held seconds to aether and health is 100 health per minute, that was derived from testing.

PlayerController
+maxHealth: int +currentHealth: int +maxAether: int +currentAether: int +healthRegenVFX: GameObject +damagedSFX: AudioClip +aetherRegenVFX: GameObject +aetherRegenSFX: AudioClip +deathText: TextMeshProUGUI -isDead: bool -headPosition: Transform -groundPosition: Transform -minSquatDistance: float -maxSquatDistane: float -minPushupDistance: float -maxPushupDistance:float
Hit() CastSpell() RegenAether() RegenHealth() PerformSquat() PerformPushup() Death() DeathCoroutine()

Table 1: Player Controller Class

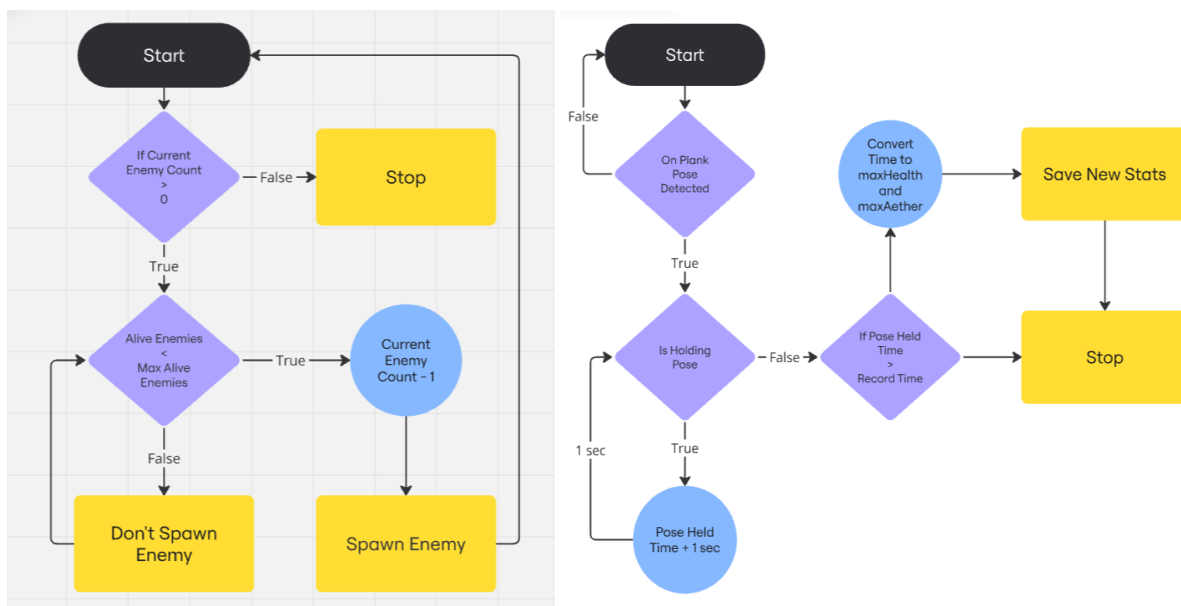


Figure 2: Flow Charts

2.2 Non-Player Characters

The NPCs in the game are all controlled by the NPC Controller class. There are two types of NPCs, friendly and hostile, assigned from a Team *enum* data type found in the class. They have three states: Search & Move, Attack and Death.

During the search and move state, the NPC looks for an enemy in the level (the player or an NPC of the opposing team) and will move towards them using Unity's Navigation mesh. Its animations are applied using Unity's Blend Trees, allowing for them to dynamically update based on the NPC's speed. Slower NPCs will be walking, whereas faster ones will be running.

When the NPC reaches an enemy, they will face towards their direction. One out of three attack animations will play at random intervals, and the collider on the NPC's sword will be enabled via the *WeaponScript* class that depends on *NPCController*.

Upon Death, the NPC's animator will be disabled, and the ragdoll physics will take hold. The more damage the NPC takes when it dies, the further it will be launched away via the overloaded method *Hit(int damage, Vector3 forceDirection)* which will apply a force multiplied by the damage in the direction that the NPC got hit from.

2.3 Hand Gestures & Spells

To cast spells, the player needs to perform one of the four hand gestures seen in Figure 3. The teleport spell is a modified version of the Teleport building block, which is part of the Meta All-In-One SDK. The rest of the spells (*Fireball*, *Summon Skeleton* and *Shockwave*) are fully custom and are integrated with Meta's Shape Recogniser Class through custom hand poses.

All methods required for the charging and casting of the spells reside in their respective script, inheriting from the Spell class. Once the hand gestures are formed, the Spell class instantiates a projectile with a respective Projectile Script that handles the spells' effects.

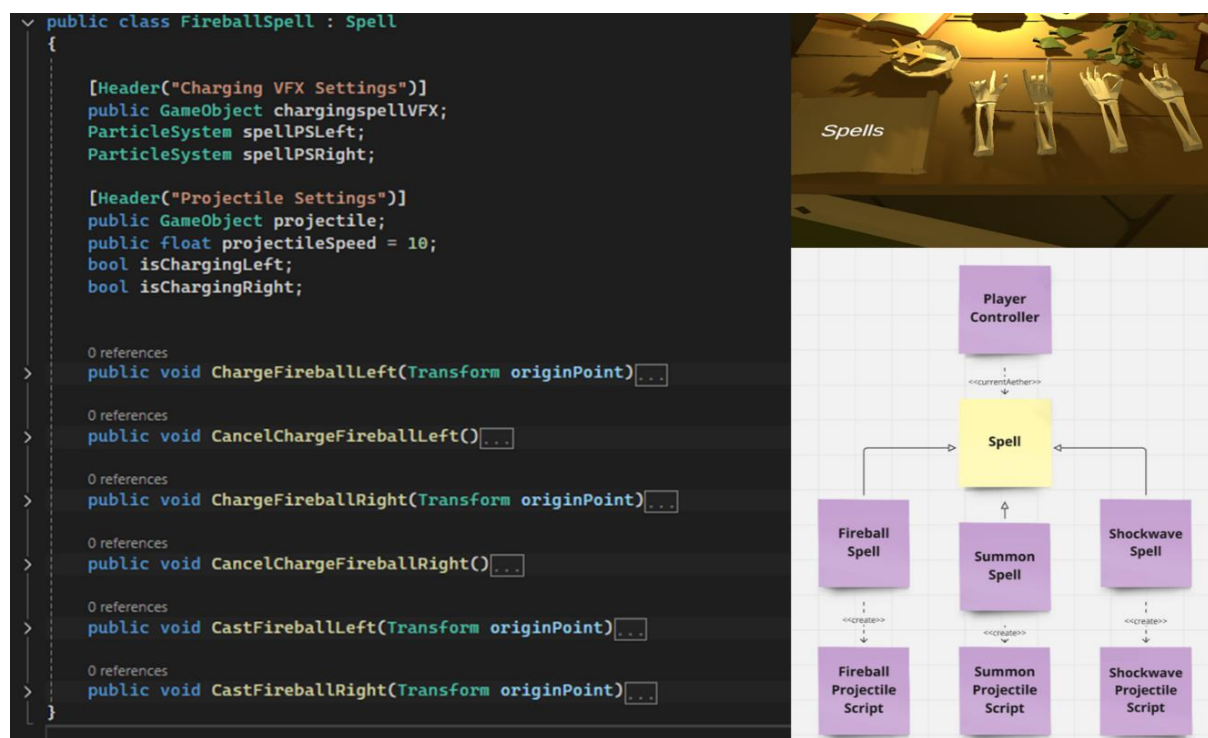


Figure 3: FireballSpell Class (Left), in-game hand pose tutorial (Top Right), Spell Class UML Diagram (Bottom Right)

2.4 Body-Pose Detection

While Body Tracking was achieved using Meta's All-In-One SDK OVR Body class, pose detection was done using another package from Meta, the Movement SDK. The package provided a Body Pose Alignment Detector which enabled recording and exporting poses as *Scriptable Objects* and comparing them to player poses during runtime. This allowed easy implementation of a variety of body pose detectors, one for each of the three exercises included in this project: Push-ups, Squats and Plank (Figure 4).

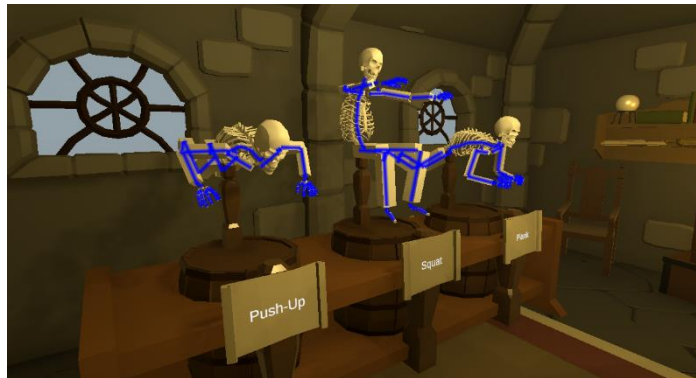


Figure 4: In-Engine screenshot of the tutorial stand for exercises.

The problem with body tracking with the Quest 3 is that it only provides true tracking information for the hands and head. The rest of the limbs of the player's avatar are inferred using a combination of Inverse Kinematics (IK) and contextual cues. This provides a challenge for full body pose detection, since most of the limbs are not reliable and prone to error, especially when the player forms a pose that is outside of the norm (e.g. lying on the ground).

To combat this, two solutions were developed and implemented. First, there was extensive tuning on the tracked body joints, to ensure that the necessary bones are being tracked (Figure 5). Joints like feet and lower legs were ignored since they were deemed very unreliable. On the contrary, limbs like the hands and head, which bore more reliable results, had their "wobble room" reduced.

Another issue is that the body pose detector does not take into account the player's rotation and distance from the floor. For example, if the player holds their hands up close to their face and looks up, the detector might falsely detect a push-up, even though the player is standing upright. To combat this, before triggering a detection, a method in the PlayerController Class (Table 1) checks for the distance of the player's head from the floor. If the head is within the expected margins (close to the ground, in the case of a push-up), then it triggers and regenerates aether.

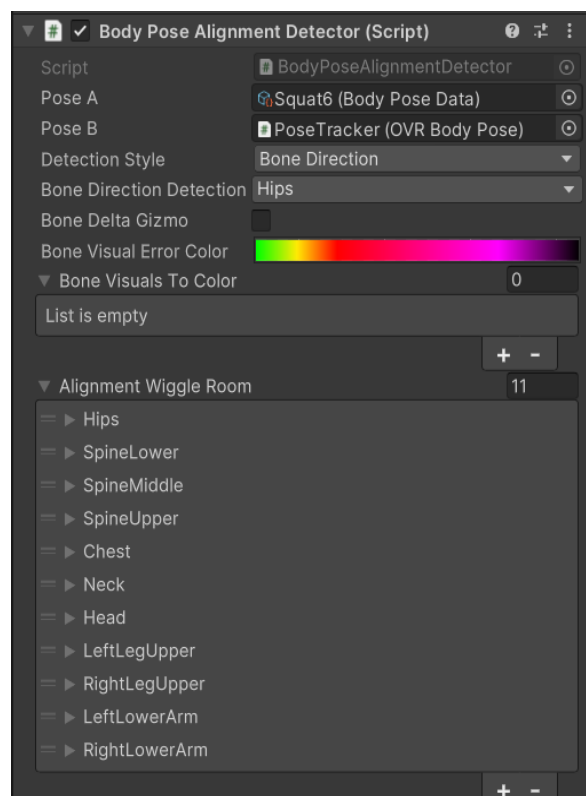


Figure 5: Editor View of the Body Pose Alignment Detector Class

3. Evaluation

3.1 Methodology

A convenience sample of 5 adults of varying physical shapes, between the ages of 18 – 35 was surveyed and interviewed after a play session.

The play session was recorded for each participant and was then manually reviewed to label the ground truth for the body-pose detector’s accuracy per exercise. Only the push-ups and squats were recorded since the plank is an alternative game mode and is not as prominent throughout the game. The data was then used to calculate the Precision, Recall and F1 Score per participant and consecutively accumulated score.

After the gameplay section, participants were given a short survey to fill out with 1-5 Likert-style questions on the responsiveness of both the body-pose detector and the hand. Then they were promptly interviewed, asking open-ended questions about the technologies they experienced during play.

3.2 Quantitative Results

As mentioned before, the footage was recorded and manually reviewed to extract information per participant. The full table can be found in the Appendices, and the total score can be seen in Table 2.

It is important to note, that not all participants were able to perform all the exercises. Two of the participants were not able to perform push-ups, hence only 3 participants’ push-up data were recorded.

To begin with, the squats pose-detector yielded a low precision score of 66.7%, meaning that there were a significant of instances where the detector failed to detect a squat. This, however, was not true for every participant. Some participants got an individual precision score of up to 86% but others got as low as 14%.

Another interesting note on the squat pose-detector was the recall score. While it is standing at 83.9%, the only time that False Positives occurred, was when the participant was in the process of lying down to form a plank or push-up.

The push-ups, on the other hand, proved to be more responsive with an overall F1 score of 82.5%. The few events where false positives occurred, were during the plank exercise, since the two poses are similar to one another. As for false negatives, they mostly occurred between push-ups, which may be because of the detector's time to detect a pose.

Total Results	
Total Squats	78
True Positives	52
False Positives	10
False Negatives	26
Precision Score	0.667
Recall Score	0.839
F1 Score	0.743
Total Push-ups	32
True Positives	26
False Positives	5
False Negatives	6
Precision Score	0.813
Recall Score	0.839
F1 Score	0.825

Table 2: Accuracy Analysis on Squat Pose-Detector and Push-up Pose-Detector

3.3 Qualitative Results

The interviews were codified to provide a better understanding of the emotions that players felt when playing the game and interacting with the hand and body-pose tracking technologies.

Participants were mostly satisfied with the game. One participant said that “it was fun and imaginative”, while others expressed feeling immersed and challenged. They were able to perform physical exercises with no hindrance, despite them wearing a VR headset.

However, most players mentioned some issues with hand tracking, which were broken down into three areas:

- **Frustration:** Occasionally, the hand tracking would not respond to the players' hand signs. Sometimes it would not register, while other times it would be because they were out of Aether. This created a feeling of frustration among participants.
- **Feeling Overwhelmed:** Due to restricted time and a short tutorial, players expressed feeling overwhelmed when deciding what spell to cast, and remembering the correct hand sign for the spell.
- **Unreliable Aiming:** Players expressed difficulty in aiming using the fireball spell. They described the fireball as inaccurate, not launching towards the direction they were pointing.

When it comes to body tracking, most participants found it very responsive and they achieved expertise quickly. A participant expressed that “*The resource management by doing pushups did not hinder the flow of the game at all.*”. This shows that even though players are fully immersed in the world, they are still confident to enter a push-up position without too much disorientation, or physical obstruction from the headset.

However, most participants expressed uncertainty about whether their Aether was regenerating when they were performing an exercise. While the game does include a VFX that spawns at the feet of the player, participants were focused on the aether bar, located on the right wrist of the player avatar. There was no animation to inform the player that it was regenerating, thus they were left unsure if they were performing the exercises correctly.

3.4 Conclusions

To conclude, the project was able to achieve its goals to create an entertaining VR muscle-building game. Participants experienced feelings of challenge, and body detection for exercises like the plank and the push-ups was very effective.

Future implementations will involve improvements in hand tracking and body-pose recognition. Hand tracking needs to be streamlined, by coming up with simpler hand signs to cast spells, while also signposting when the player is out of aether more clearly to reduce frustration. While body tracking proved to be sufficiently functional in the case of the push-ups, the squats occasionally did not function for some participants. More testing is required to figure out the cause, but the ground-to-head detection in the PlayerController script is suspected to be the issue.

4. Discussion of performance and optimisation techniques

It was important to maintain a high framerate for this project to provide a smooth experience to the player. To achieve this, a variety of optimisation techniques were employed.

First, occlusion culling was applied, using Unity's native system, that bakes occlusion data and hides static objects hidden behind other objects. This allows the engine to render less meshes, lowering the frame by frame tri-count, increasing frames per second (FPS).

The project was made in Unity's Universal Rendering Pipeline, since it prioritizes performance, compared to the Standard Render Pipeline. To further improve performance, the lighting settings were tweaked, lowering shadow & light resolution in the project settings.

5. Overview of LLMs and Gen-AI tools Used and How

ChatGPT was used for searching up different ways of tackling a task, at a high level. For example, when a NPC is killed, they are pushed away, based on the damage of the projectile that hit them. To figure out the optimal function to apply this force within Unity, ChatGPT was prompted to provide the variety of options available to tackle the specific task, as seen in Figure 6.

This way, even if ChatGPT hallucinates and gives misinformation, it will be spotted upon looking up the documentation for the specific function that is needed to be used.

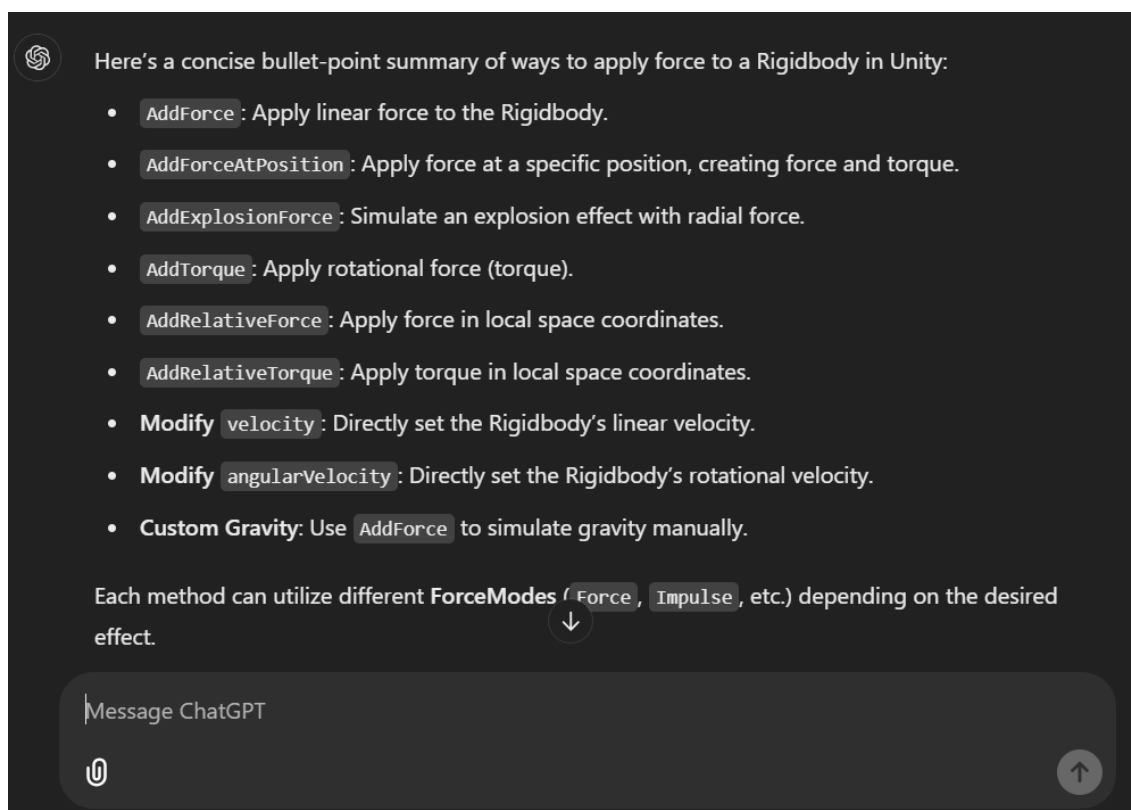


Figure 6: ChatGPT's response to the prompt "Using unity, what are the different ways I can apply a force to a body? Please keep it short."

6. References

Academic

1. Thomas, E., Bianco, A., Mancuso, E.P., Patti, A., Tabacchi, G., Paoli, A., Messina, G. and Palma, A. (2017). The effects of a calisthenics training intervention on posture, strength and body composition. *Isokinetics and Exercise Science*, 25(3), pp.215–222. doi:<https://doi.org/10.3233/ies-170001>.

Games

2. Beat Games (2018). *Beat Saber*. [video game] (Microsoft Windows, PlayStation 4, Oculus Quest). Beat Games.
3. Nintendo (2008). *Wii Fit*. [video game] (Nintendo Wii). Nintendo.
4. Nintendo (2019). *Ring Fit Adventure*. [video game] (Nintendo Switch). Nintendo.
5. Ubisoft (2010). *Your Shape: Fitness Evolved*. [video game] (Xbox 360). Ubisoft.

Software Development Kits

1. Meta (2024). *Meta All-in-One SDK*. Available at: <https://developers.meta.com/horizon/downloads/package/meta-xr-sdk-all-in-one-upm/> (Accessed: 1 December 2024).
2. Meta (2024). *Movement SDK*. Available at: <https://developers.meta.com/horizon/documentation/unity/move-overview/> (Accessed: 1 December 2024).

3D Assets

3. Broken Vector (2019). *Ultimate Low Poly Dungeon*. Available at: <https://assetstore.unity.com/packages/3d/environments/dungeons/ultimate-low-poly-dungeon-143535#version-original> (Accessed: 1 December 2024).
4. Moreno, J. (2018). *Cartoon FX Remaster Free*. Available at: <https://assetstore.unity.com/packages/vfx/particles/cartoon-fx-remaster-free-109565> (Accessed: 1 December 2024).
5. Polytope Studio (2020). *Lowpoly Medieval Skeleton - Free - MEDIEVAL FANTASY SERIES*. Available at: <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/lowpoly-medieval-skeleton-free-medieval-fantasy-series-181883#releases> (Accessed: 1 December 2024).
6. Polytope Studio (2021). *Lowpoly Modular Armors - Free - MEDIEVAL FANTASY SERIES*. Available at: <https://assetstore.unity.com/packages/3d/characters/lowpoly-modular-armors-free-medieval-fantasy-series-199890#releases> (Accessed: 1 December 2024).

Sound Effects

7. BBC Sound Library (2023). *Heavy Explosions – Heavy Explosion*. Available at: <https://sound-effects.bbcrewind.co.uk/search?q=07037325>
8. Zumer, J. (2014). charging power. Available at: <https://freesound.org/people/JavierZumer/sounds/257229/>
9. EminYILDIRIM (2020). *Magic Fire Spell Impact & Punch*. Available at: <https://freesound.org/people/EminYILDIRIM/sounds/541478/>
10. The-Sacha-Rush (2022). Electric Shock 2 Hit. Available at: <https://freesound.org/people/The-Sacha-Rush/sounds/657803/>
11. Kanyonwyvern (2024). Skeleton Gasp / Final Breath / Death Breath. Available at: <https://freesound.org/people/kanyonwyvern/sounds/736663/>

7. Appendices

Ground Truth & F1 Score Table per Participant

Participants	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
Total Squats	6	7	22	37	6
True Positives	6	1	19	25	1
False Positives	0	0	4	0	6
False Negatives	0	6	3	12	5
Precision Score	1	0.14	0.86	0.68	0.17
Recall Score	1	1	0.83	1	0.14
F1 Score	1	0.25	0.84	0.81	0.15
Total Push-ups	6	N/A	8	N/A	18
True Positives	1	N/A	8	N/A	17
False Positives	0	N/A	4	N/A	1
False Negatives	5	N/A	0	N/A	1
Precision Score	0.17	N/A	1	N/A	0.94
Recall Score	1	N/A	0.67	N/A	0.94
F1 Score	0.29	N/A	0.8	N/A	0.94

User Study Survey & Interview Form



Deadlift

For this user study, you will be partaking in a short playtest of the VR game Deadlift. You will play two of the game modes after receiving a quick introductory tutorial by the interviewer. After the playtest you will respond to a few survey and interview questions.

Your gameplay will be recorded for further research, but your participation is 100% anonymised. No personal data or identifying features will be collected.

If at any point during this survey you want to back out, let the interviewer know, and the study and any data collected up to that point will be scrapped. If you experience any VR related illnesses like dizziness, nausea or disorientation, please let the interviewer know.

Should you accept the terms and conditions of this study please

jorgekaramba@gmail.com [Switch accounts](#)



Not shared

* Indicates required question

I've read the terms and conditions of this study and agree to take part in it. *

- Agree
- Disagree

Next

Clear form

Quantitative Feedback

Please answer the questions below based on your gameplay experience.

On a scale from 1-5, where 1 is Very Easy and 5 is Very Difficult, how would you rate the difficulty of the game? *

	1	2	3	4	5	
Very Easy	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Difficult

On a scale from 1-5, where 1 is Very Incompetent and 5 is Very Competent how would you rate your Skill level during play? *

	1	2	3	4	5	
Very Incompetent	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Competent

On a scale from 1-5, where 1 is Very Unreliable and 5 is Very Reliable, how would you rate the reliability of the Body Pose Detector during the game. (When you did Squats, Push-ups and Plank) *

	1	2	3	4	5	
Very Unreliable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Reliable

On a scale from 1-5, where 1 is Very Unreliable and 5 is Very Reliable, how would you rate the reliability of the Hand Pose Detector during the game. (When you casted spells) *

	1	2	3	4	5	
Very Unreliable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very Reliable

Qualitative Feedback

Pass the laptop to the interviewer

How did you find the game *

Your answer

What did you think of the Hand Tracking for Casting Spells? *

Your answer

What did you think about the Body Tracking for Exercise *

Your answer

Back

Next

Clear form