# Project Overview

This project is a chance to practice my optimization skills. I am given a Virtual Reality (VR) game that is extremely un-performant, and I am to use my knowledge to optimise it to a consistent 90fps.

# Why this project?

Performance optimization is one of the most important skillsets for a VR developer, since apps running below the target framerate are essentially unplayable, causing nausea and discomfort. This project simulates a real-world scenario where a VR game has been hastily built with a focus on functionality rather than performance (a common approach while prototyping). And by optimisation I can make it a performant app ready for release.

# Learning Points

This project is about combining various ideas and skills I have been practicing throughout the course. They include:

* Using static and dynamic batching
* Optimizing physics
* Creating object pools
* Caching variables and optimising code
* Setting up lighting with Multi Sampling Anti-Aliasing (MSAA)
* Baking lighting
* Using light probes
* Using the profiler for performance optimisation

# Build Process:

For this project, I will be optimising the performance of a VR game in which the User will bounce balls on as many trampolines as possible. The integrity of the gameplay has to be preserved. Therefore:

1. Trampolines cannot be removed, and must retain their ability to score a point and show particles when a ball hits them.
2. The lights in the scene cannot be removed and must show up in either baked or realtime.
3. A ball must spawn every 0.5 seconds.
4. There must be a score visible to the player from their starting position.
5. There must be a countdown timer visible to the player.
6. The player must be able to pick up and throw balls at the trampolines.

Steps

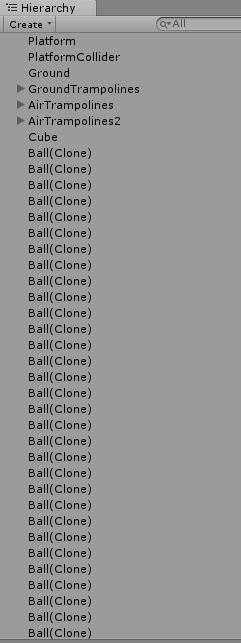
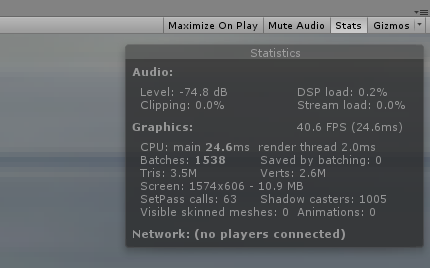
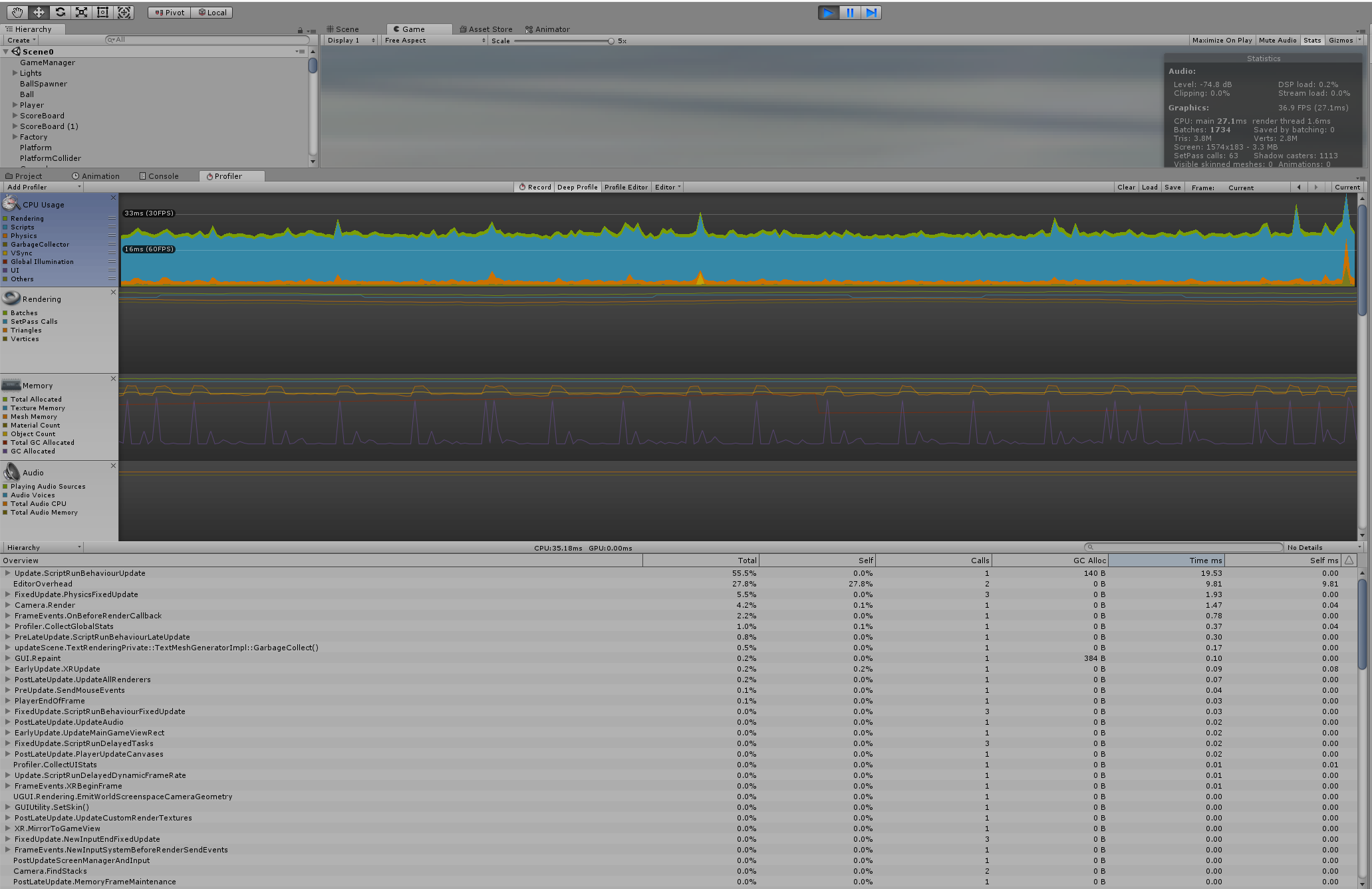


Figure 1. Instantiated

Before:





<Include video of pre-optimisation>

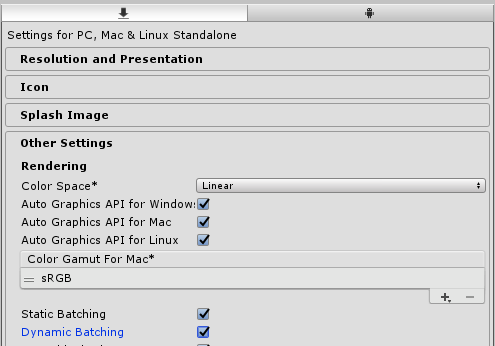


Figure 2. Enabling Static and Dynamic Batching

