# Emma! V1.1 Optimization Plan

This report evaluates the project’s performance before optimization and generates possible optimization plans based on what needs to be optimized.

## Scene Loading

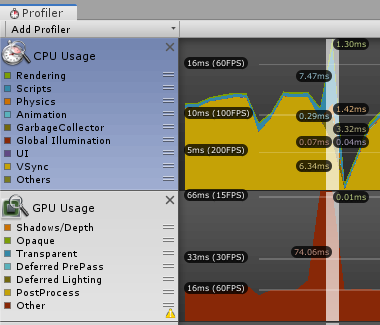


Figure 1 Spike on scene changing

CPU Usage: ~ 20ms

GPU Usage: ~ 74ms

As shown in the figure, every time a new scene loads, there is a peak usage rate. It is considered normal for Unity to use relatively more resources when loading a new scene. Because the program does not appear to be stuck, and the CPU and GPU usage are within a reasonable range, no optimization is needed for this part.

## Menu Scenes

For the start menu scene and the end menu scene, one optional optimization is lowering the frame rate limit (ex: maximum 30).

## Debug.Log

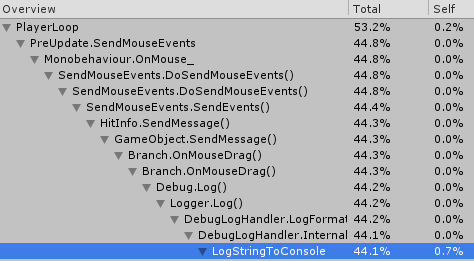


Figure 2 Logging to console cpu usage

There are several Debug.Log calls under Update for temporary test use. These calls are resource-consuming since they are being called every single frame. They will be removed.

## Memory Usage



Figure 3 Acorner level memory usage data

In the game start scene, there are two interactive UI buttons only. The total memory usage is about 380MB, of which Unity Profiler accounts for nearly half, and the memory actually occupied by the game program is below 200MB. The memory usage increased by 50MB when entering the first game scene because the scene contains more objects, which is considered normal and does not need to be optimized.

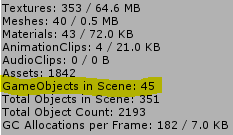


Figure 4 Gameobjects count increase abnormally

According to the memory usage analysis, the number of gameobjects in the scene gradually increases. Every 2 times the scene switches (every 2 usage peak), the gameobject increases by 1. According to this, there is one gameobject that was not destroyed with the scene on scene switching.

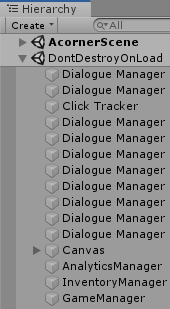


Figure 5 Multiple non-singleton managers

Multiple instances of dialogue manager are created, which is a memory leak caused by logic problems. This gradually slows down the running speed. It must be optimized. The optimization method will be to make the dialogue manager into a singleton.

## Polygon Colliders

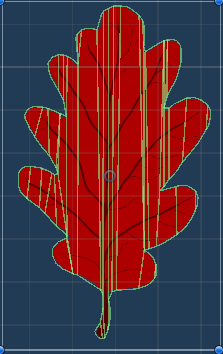


Figure 6 Unoptimized collider on prefab

The polygon collider auto-generated by Unity is detailed but unoptimized, this can cause significant performance drop during physics simulation. In future updates, colliders on gameobjects(especially non-static ones) will be optimized by reducing the complexity.