**Performance Testing**

Though gameplay testing is vital to any game’s creation, the performance of said game is equally crucial. Even the best games can flop if they don’t run with a comfortable frame rate, and as such the team will be testing the game thoroughly to ensure that it runs at peak performance on multiple systems with both high- and low-end specs.

**Profiling:**

Unreal Engine 4 has built in CPU and GPU profiling tools, this allows us to dictate what part of the system is acting as the games bottleneck and from there dictate what part of the game is causing the issues. The use of these tools requires us to open the Unreal Engine console while our game is running and using the commands:

**Commands:**

***stat fps:***

Unreal loads a basic ‘frames per second’ and ‘length of frame’ monitor. This will allow us to see the exact frame rate we are getting in specific areas of the game.



Figure - FPS counter

***stat unitgraph:***

Unreal opens a visual graphing tool, this acts as a graphical representation of the data being transferred during runtime.

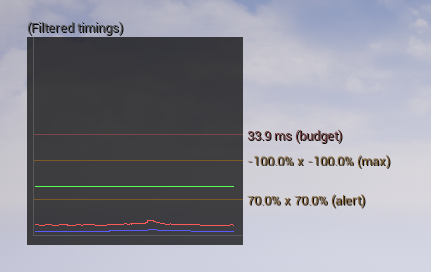
****

Figure - Data transfer graph

***stat SceneRendering***

the stat SceneRendering command, opens up a table showing general rending statistics for the scene. This will allow us to pin-point specific areas of the rendering pipeline as the problem and fix them accordingly.

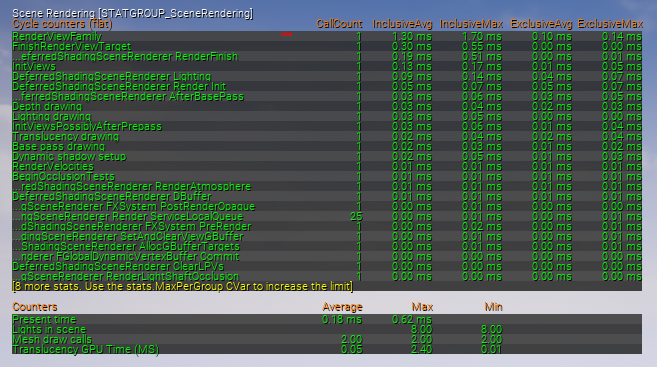
****

Figure - Scene render data

***stat gpu***

This command gives us direct information as for what the graphics card is currently used for. This will also be a quick and easy way of finding specific problems in areas of the level.

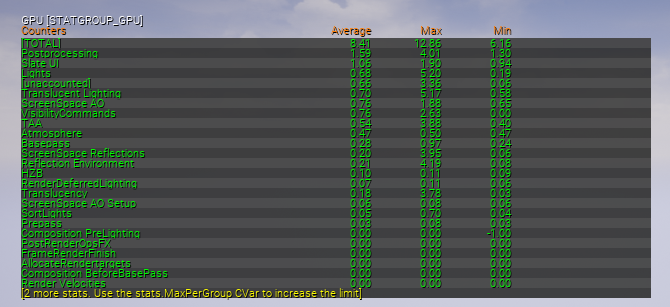
****

Figure 4 - GPU render data

There are around 100 other commands used for performance testing built into Unreal’s console. These include tests for audio and sound clips, rendered text and even for specific game actors like the player controller. With these we will be able to do more then enough extensive tests so that the final game is running fast and consistent.