

Question 1: For each shader model, how many vertex instructions and fragment instructions does your simple shader need? How many does your complex shader need?

Answer

Type	Target	Vertex Instructions	Fragment Instructions
Simple	2.0	6	1
Simple	3.0	6	1
Complex	2.0	238 (pretty close to max value of 256 instructions)	6
Complex	3.0	838 (Max value of 1024 instructions possible for shader model 3.0)	6

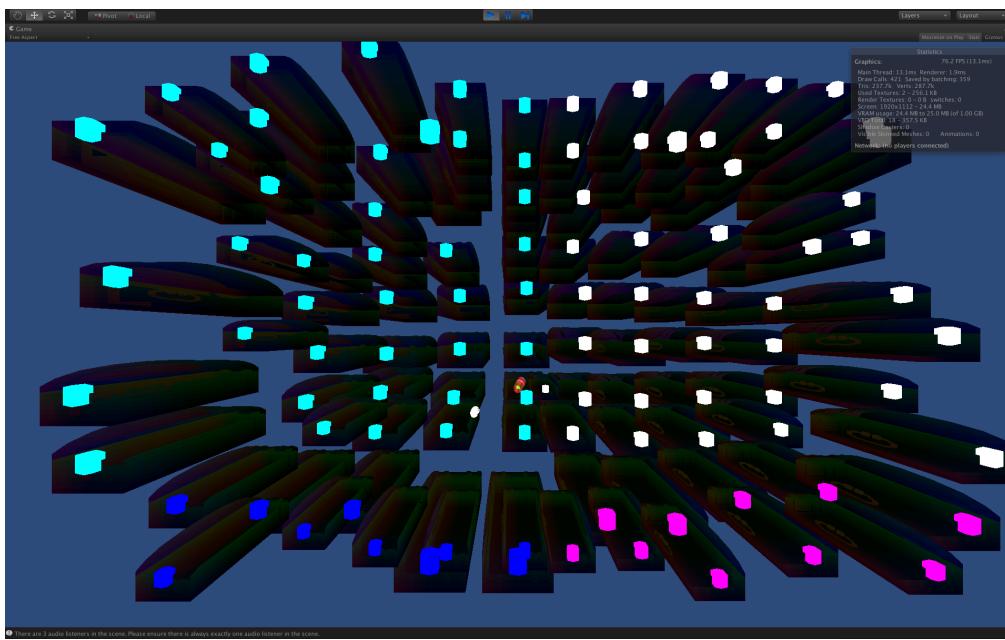
Question 2: What are the performance metrics, "Renderer time" and "milliseconds per frame," for the four cases? (SM 2.0 vs 3.0, simple vs. complex code)

Answer

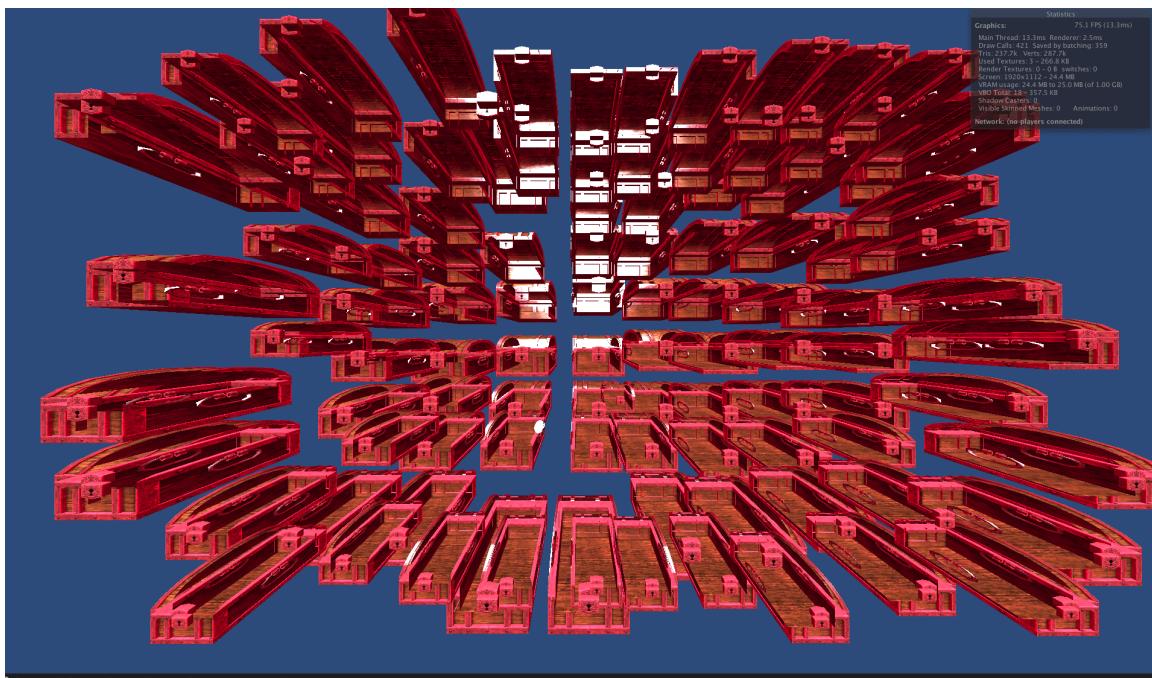
Type	Target	Renderer Time	Milliseconds per frame
Simple	2.0	1.9 to 2.0 ms	12.8-13.6
Simple	3.0	1.7 to 1.9 ms	13.1-14.4
Complex	2.0	2.3 to 2.6 ms	12.8-13.4
Complex	3.0	2.3 to 2.6 ms	12.8-13.4

Question 3: Comment on how the performance metrics of "Renderer time" and "milliseconds per frame" relate to the instruction count. Do you conclude that there is a straightforward mapping from instruction count to milliseconds per frame or renderer time?

Answer :According to my experiments where I maxed out the number of instructions for each shader model (2.0 and 3.0) for each shader (complex and simple) and the results I got from them, I can only conclude that there is no straightforward or linear mapping from instruction count to milliseconds per frame or renderer time. A slight increase is observed when the number of instructions is increased in renderer time,(while FPS and milliseconds per frame nearly remained the same) and thus it is safe to say that if number of instructions are increased there can only be an increase and definitely not a decrease in renderer time and milliseconds per frame... but by how much and what is the relation between these parameters and the instruction count remains debatable... (as it is seen in this case where it was only when the number of objects was increased significantly a change in renderer time was observed... but with a small number of objects such an observation was not seen...Hence this shows that there are various parameters apart from just the instruction count that play a role in determining the renderer time and milliseconds per frame and hence finding a direct relation between the instruction count and milliseconds per frame is not possible)
What I also found was that shader model 3.0 supported 1024 instructions whereas shader model 2.0 supported 256 instructions... i.e. shader model 3.0 supported roughly 4 times the number of instructions shader model 2.0 supported. Also the limit on number of constant registers which unity can use is 96



Simple Shader test case



Complex Shader test case