I decided to implement a particle system in my Vulkan framework.

First, a ParticleEmitter script is placed on a game object. This handles all of the data for particle spawning and parameters, while the particles themselves contain very simple simulation calculations.

I pass the particle system a parameter when it is created that determines how many particles there are. I did not have enough time to implement a resizing function that changes the max number of particles in the system, so if you wish to change the number of particles, you will have to do so manually. It starts out with 1000 particles max and a spawn rate of 0.1 seconds.

All of the particles are kept within an array in the particle emitter. I sort the active particles from the inactive ones using a list, in order to avoid unneeded computations.

The active particles are determined to be active based on their life. If they are, they are restarted from the spawn point. They are given a somewhat random starting velocity move over time based on that.  
If the spawn timer hits zero, a particle is added to the list of active particles.   
If the list has more than the maximum number of particles, the first particle in the list, which would be the oldest, is removed and a new one inserted at the end.

When iterating over the particles, I set the current active particle’s instance data to the current loop index position in a buffer containing the data needed for instance rendering. This will allow me to set the current number of instances to be rendered dynamically, without needing to perform many memory operations.

The relevant data is in ParticleEmitter.scr and Particle.scr. ParticleRenderer.scr is a specialized renderer for particles, as I needed a way to get the number of active particles when rendering out the system.

Build for x86 in debug.

You can still fly around with WASD, shift and space.

Yes, I know about the two not-deleted memory and buffer handles. I know where they are I think. They SHOULD be being deleted, but I don’t know why the destructor seems to just not be called.