

Dynamics for Games and Animation

A3.1 Progress Report - Dynamics Project Checkpoint WIP

Student Name: Khang Trinh

Student ID: 102118468

Did anything change/Will anything change?

I went from using shuriken particle system to create the smoke plumes to using a mesh applied with a vertex displacement shader to create the plumes.

Reason: I wanted the FX to be 3D, which means the smoke plumes needed to be able to rotate in all 3 axis, but I was struggling to get it working. I've only been using animated billboards as particles in my past projects to make smokes (it was my go-to method for smokes), but those can only rotate in 1 axis without the viewer knowing that they're 2D (imagine having a 3D image on a piece of paper and trying to rotate it without breaking the illusion, you'd only be able to do that by rotating it flat).

Then I realized I forgot shuriken can accept 3D meshes as particles, but the idea of having a particle system spitting potentially thousands of rotating and animating smoke plumes per second wasn't very appealing, so I went back to the 2 Unity tutorial linked in the pitch, and decided to make the mushroom clouds via vertex displacement. I still wanted to stay away from the stylised aesthetics that both of them had, so I looked around some more til I came across [this tutorial](#).

What I have done so far:

- Mushroom cloud:
 - Created 3D models for Cap and Stem
 - Created the vertex displacement shader for the clouds
- Lighting:
 - Added HDRI sky
 - Played with lighting for the nuke
- Animated the values to create motion
- Post processing: glare + bloom

What I still need to do:

- House
 - Find furnitures and place them in the scene
 - Lighting

- Mushroom cloud:
 - Condensation rings:
 - 3D model
 - Material
 - Animation
 - Base dust cloud:
 - 3D model
 - Material
 - Animation
- The "Dynamic" part of the assignment:
 - Make 3D glass panel and their broken variation
 - Write knock back script for house stuffs
 - Create the shockwave fx
 - Camera distortion shader
 - Dust particles
- Juice:
 - Camera (motion blur, shaking, getting tumbled, etc.)

Thoughts & Opinions: I am less than 2 weeks from submitting. I AM SO FAR BEHIND

| How difficult it was for me to get to where I am. Rate tasks from 1 (being, "This is just a copy pasta from a past project") to 10 (being, "This is a new concept/technique to me and I've never seen/done this before and I have no idea where to begin")

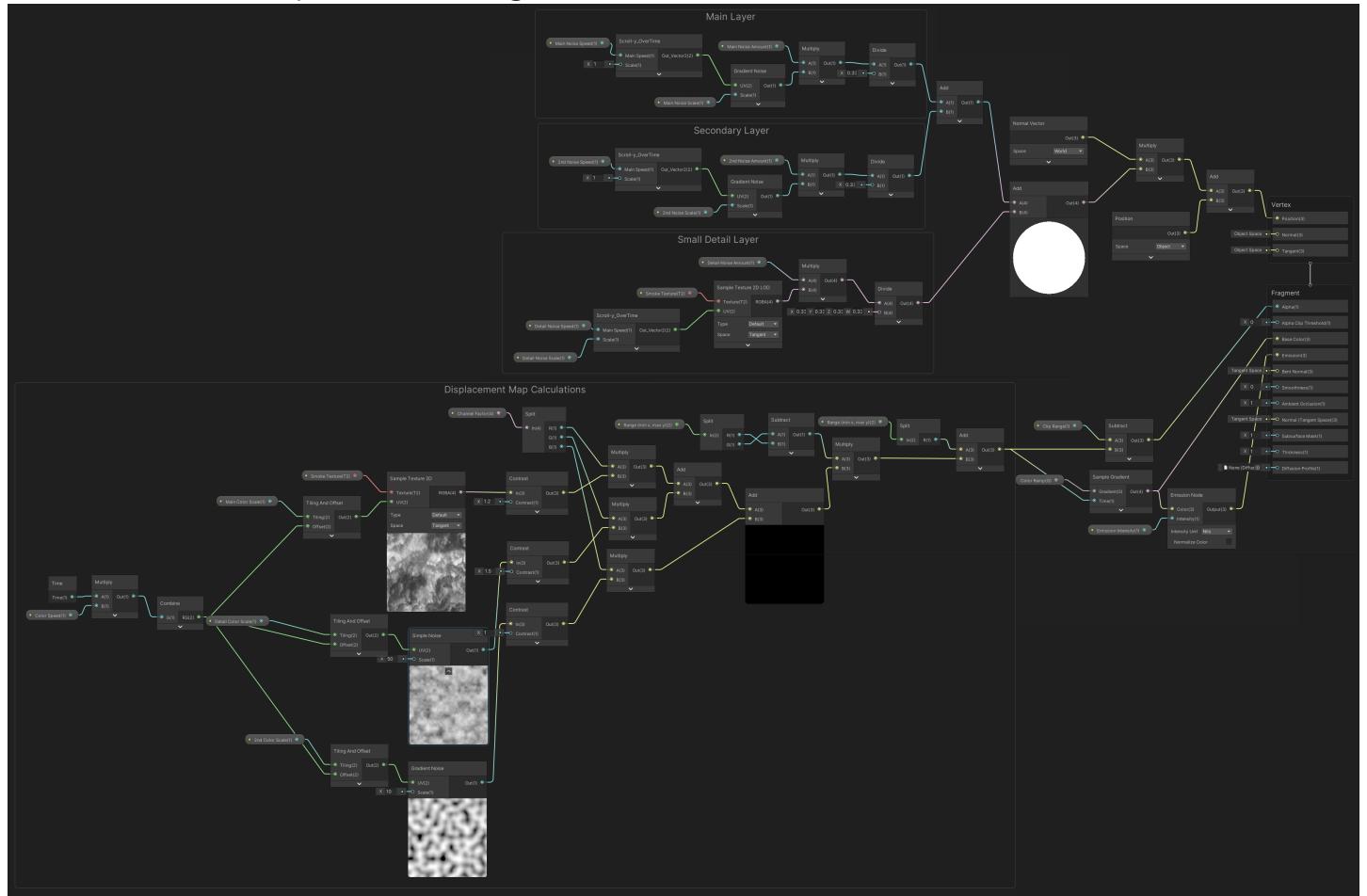
1/ Creating 3D models for Cap and Stem: 3 Rationale: Finding the right amount to use for the subdivision surface modifier for detail vs performance was something I don't think I did right cuz it took way longer than expected XD Too little vertices meant "edgy" displacement but fast performance, whereas too many vertices meant every pixel displaced on the texture can be represented by a vertex, *but at what cost*. I think I took way too long trying to optimize this part.

2/ Making the shader for the clouds: 9 Reason: This was an interesting challenge to take on, because the tutorial that I followed made the shader in hlsl, but I was using shader graph (the HDRP package in unity doesn't allow custom shaders to be used), so not only did I have to make my own interpreted version of the shader, I also had to translate bits and pieces of that code (especially how the logic uses the displacement map for both displacing vertices *and* distance coloring)

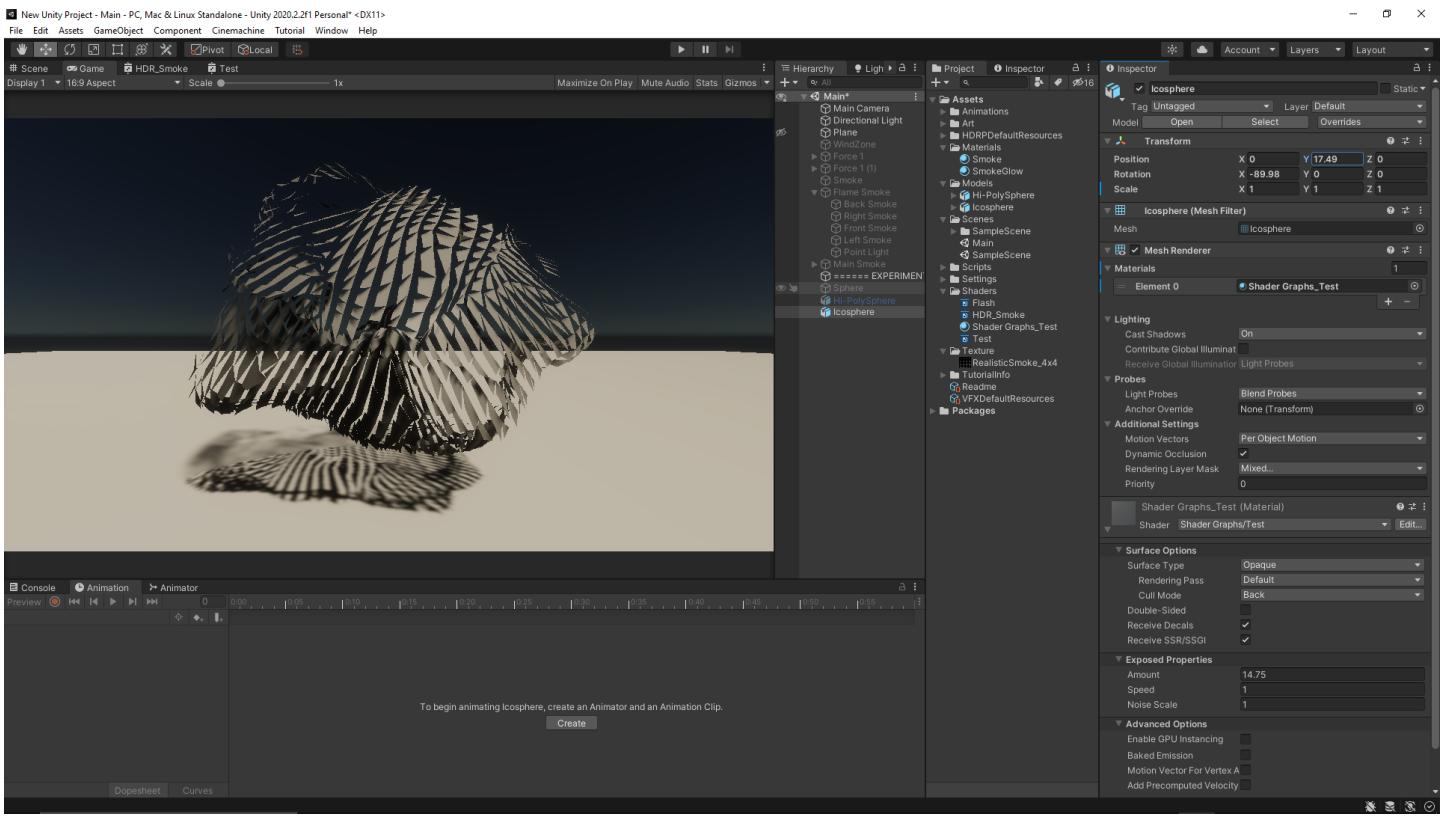
I tried to make a smoke texture in photoshop for the albedo, but that failed miserably (Refer to Day 2 video, I also learned I don't have enough skills to make a smoke texture look realistic on a mesh), not to mention the mesh's scale was constantly changing due to the resolution optimization, so I needed to make the texture procedural using a mix of simple noise + perlin noise + a different smoke texture found online and photoshoped to make it tileable. Each noise texture can be a layer of detail, and each can be individually controlled to change the look of the smoke.

I also had a bit of fun playing around with subsurface scattering as well. Clouds have that, so the FX needs that. *I definitely did not die trying to get the values right.*

This is what I ended up with (click image for full-sized version)



And then this happened. The mesh's normals apparently were not merged (it took roughly 1-2hrs of hair pulling to diagnose this), so it was back to Blender and merging the normals, but for some reason the normals merge didn't work on an existing mesh, so I needed to remake the meshes for both (which now bumped the difficulty of task 1 from 3 to 4).



3/ HDRI Sky: 8 (*it's only 8 because of the criteria, I'd wanna give this a 50 if every fail search counts as a point*) This was again, harder than I expected. The reason's because I didn't expect that finding an HDRI sky with specific criteria make my life so much more miserable . . .

The first criteria was to find an HDRI with flatlands. This is to make it easier for me to place the nuke closer or further away from the camera without background props breaking the depth perception.

Case in point: This nuke looks very close to the camera (it's closer than the mountain in the distance and seems like it's only ~100m away), and I didn't want the camera man to immediately die from the explosion. Not only that, it also looks quite small because it's closer than the mountain, yet it's not even as tall as the mountain from afar.



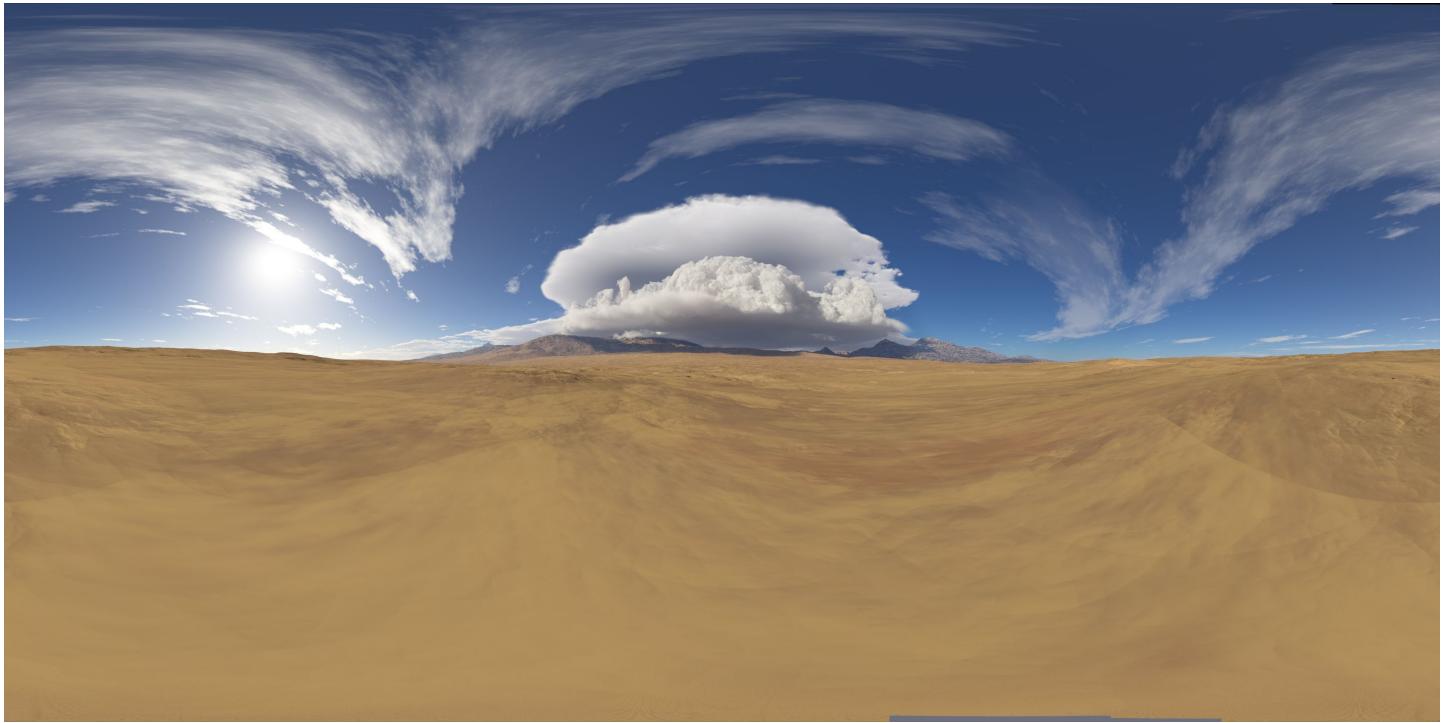
The second criteria was to find an HDRI without clouds, and every, *single, one of 'em* that I found had to have some form of cloud (It's almost like it's normal for the sky to have clouds at any given time, even on a nice, sunny day, ~~the perfect day to launch a nuke~~).

The problem with having clouds in the HDRI is, when a nuke is dropped, should it form condensation rings (which I am planning to have), the rings would be made from the clouds surrounding it (in other words the sky would only have those rings as "clouds", no normal clouds would exist in its immediate vicinity, [refer to this video](#)). But an HDRI is a static image, and the rings that I'll be making will be an object on its own, and we can't have a ring and normal clouds existing right next to each other.

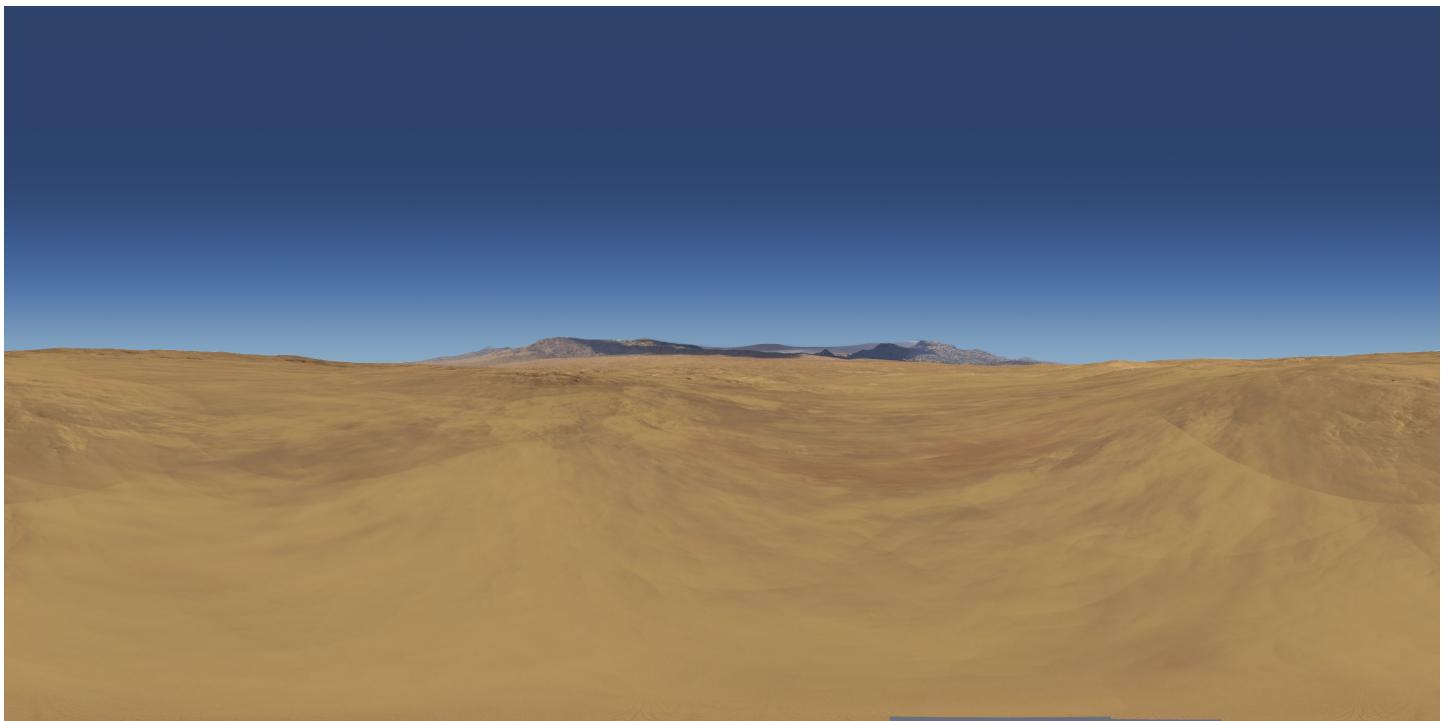
The direct solutions to this was to either

- Find an *animated* HDRI *with* clouds going from normal to ring, which would've been impossible because
 - i. I'd 100% have to pay for that because it's a lot of cost and work for the creator for the reasons below, and I'm still just a uni student with no income whatsoever
 - ii. You'd need to have a nuke to achieve that FX, and world peace has kinda prevented that from existing (for now), not to mention the nuke in the HDRI will be obstructing *my* FX, in which case I might as well use the entire HDRI as my FX and only make the shockwave :D, or
- Find an HDRI *without* clouds to start with

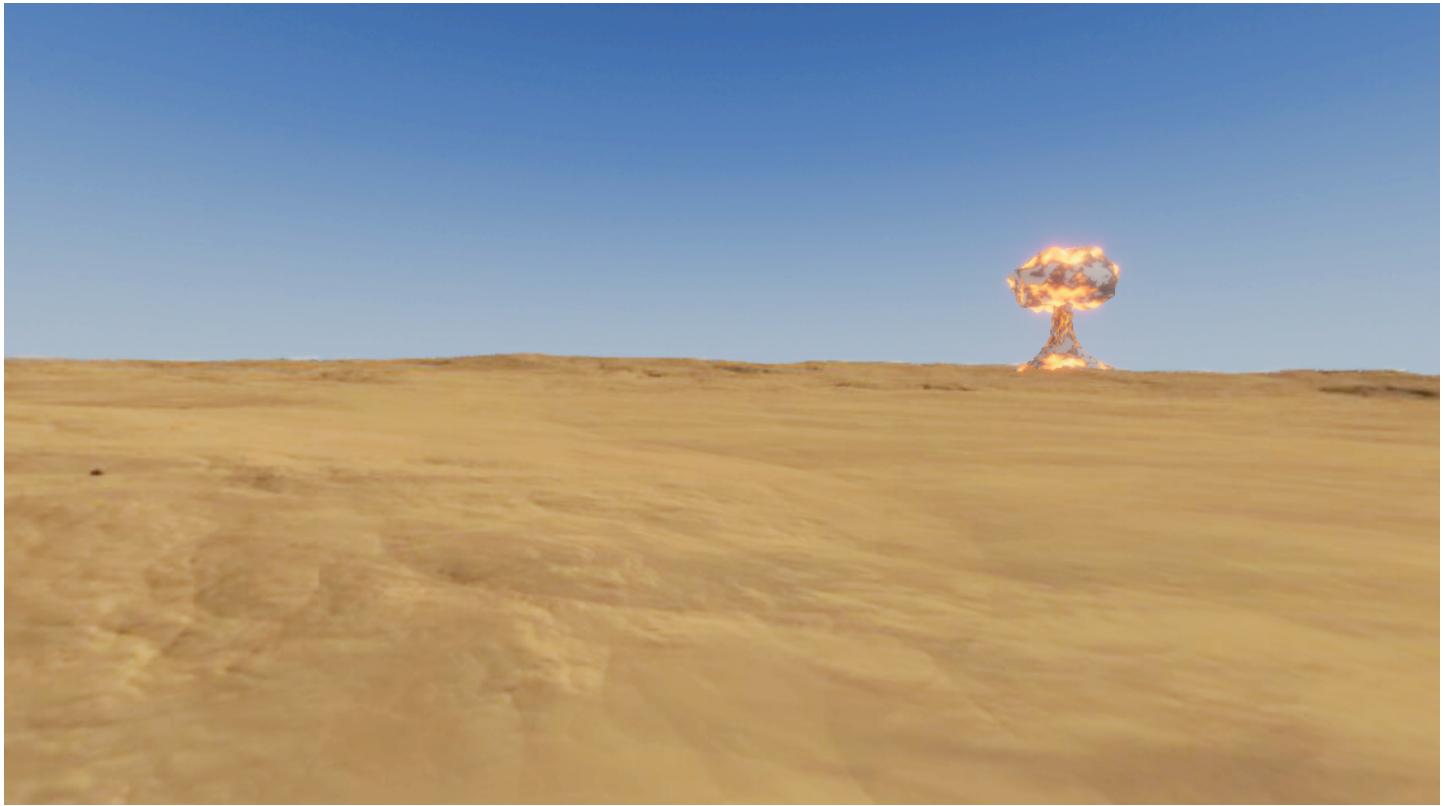
As I got desperate from the search, I decided to download one with clouds *and* flatlands and crop the clouds away from the image, this is the flattest land I could find, and it just had to throw a thicc, massive cloud in the middle of it . . .



That was the "Before", this is the "After". I also took the sun away because having the sun in the same shot as the nuke would mean 2 bright light sources in the same shot, which would only make it distracting for the viewer.

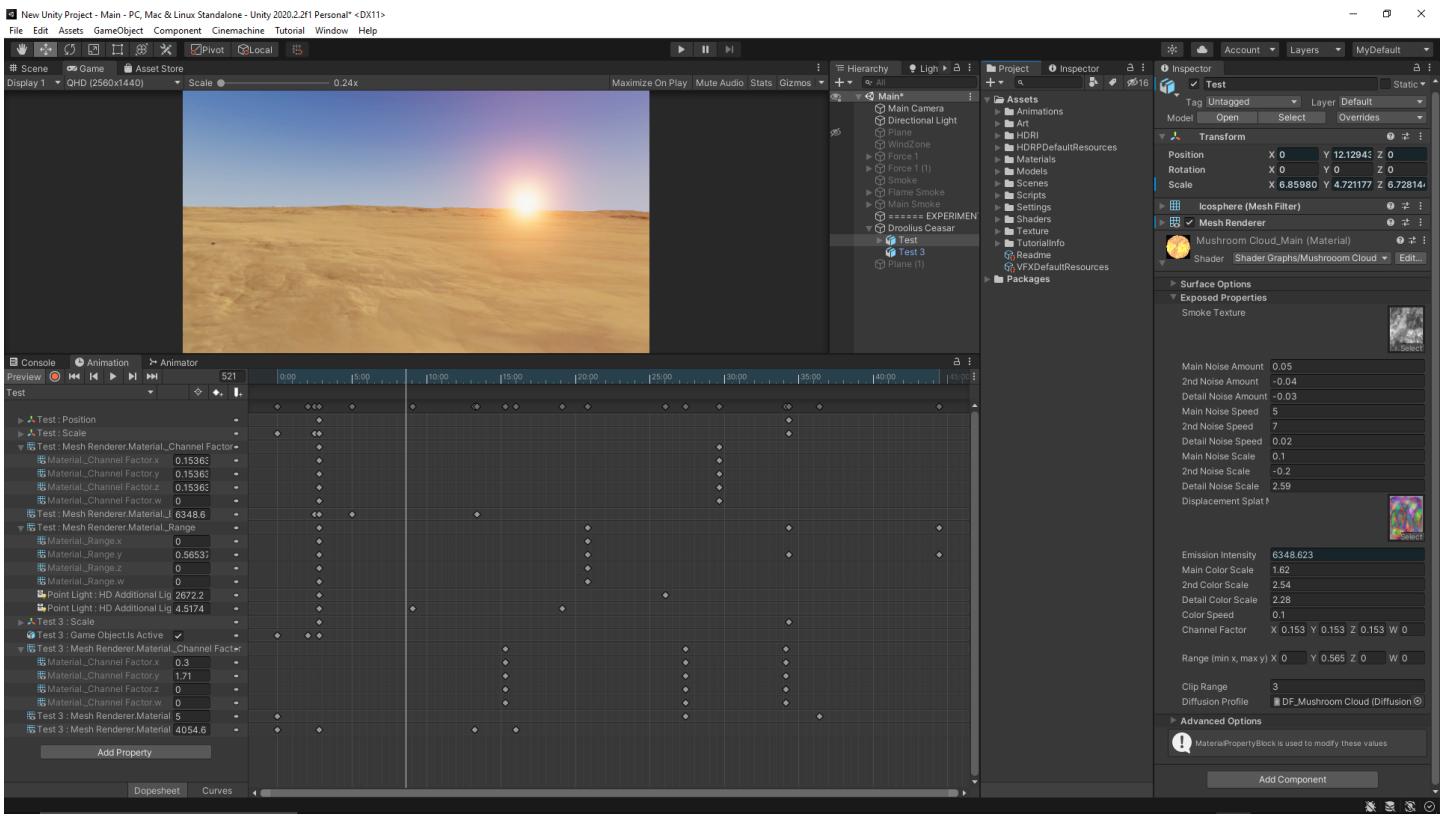


This is the result:



4/ Post Processing: 5 This was a 5 solely because I never wanted to keep the various HDRP scene settings as is, but now I do, and I never had to handle things that were so bright (ie. nuke at 100,000 lux + emission intensity) that it makes the camera go white. *It should've been a 2.*

5/ Animating the nuke: 9 This was the hardest task I never expected to find so challenging (I was expecting this to be a 4 like other submissions for this unit). Trying to make the nuke look like irl references was like trying to make a movie with flashy explosions like how Michael Bay would. The amount of vertex displacement from each noise layer were not independent off of each other, so changing how bumpy 1 layer is would mean also changing how bumpy the other layer is, there were 5 individual values that could control how the FX would fade away, etc. TL;DR: The shader's math made it difficult to predict how the FX would behave, which resulted in many, *many*, **many** keyframes in between to effectively change the rate at which the value is interpolated over time.



6/ Final result: $(4+9+8^*+5+9)/5 = 7$: This is more difficult than I expected. If the rest of the project is going to be like this, I'll have to drop my expectation of the end result in order to not spend too long tweaking things instead of making them (I've only done 5 tasks and they took 2 weeks, and I still have more than 5 left, big or small).

Note: The rest of the report structure is up to you. See task sheet for a guide of what to address.