Implementation of deferred snow deformation technique used in the Rise of the Tomb Raider

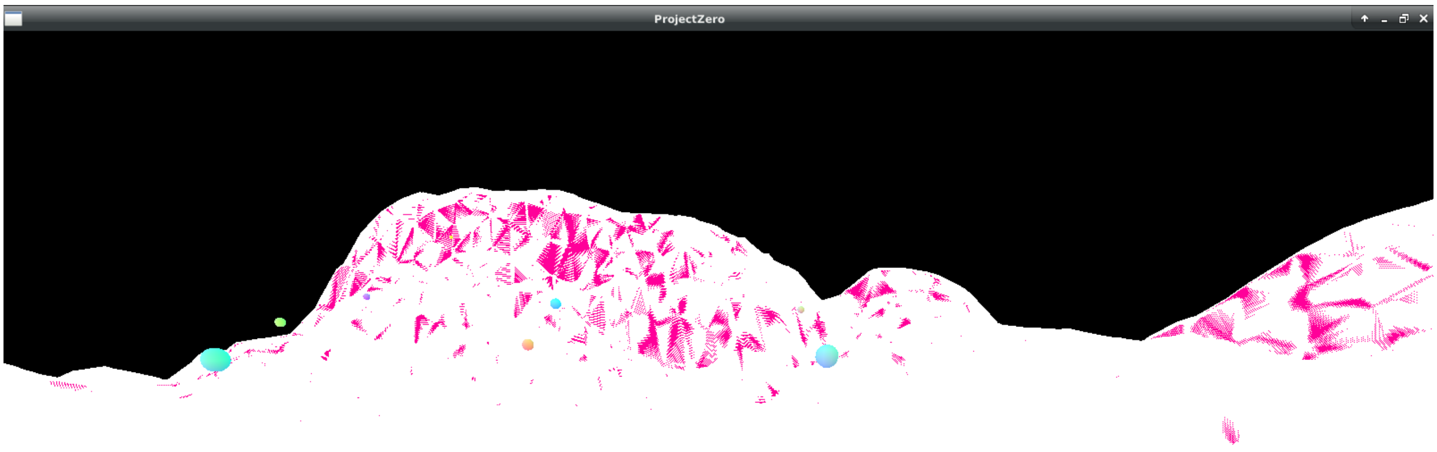
Improved physics of the spheres

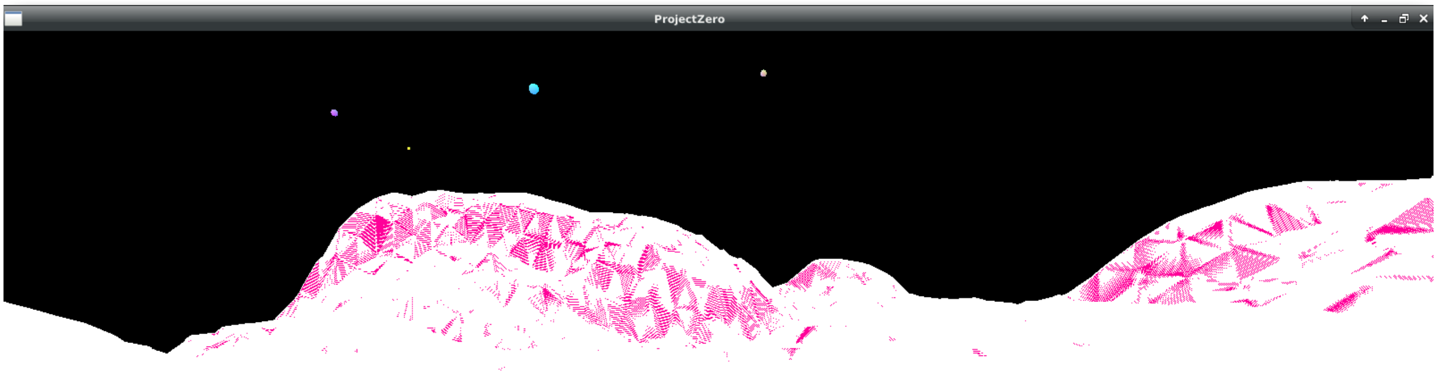
I have improved the behavior of the dynamic objects, they used to stutter and fall through the terrain a little. They are now moving mostly downhills and have some sense of gravity and velocity. It is not a perfect physical simulation, but the movement is smooth and good enough for the purpose of this project.

Snow rendering

The rendering of snow was actually pretty easy, I reused most of the terrain setup and shaders. I just reimplemented the tesselation evaluation shader and moved the snow a little bit higher than the ground, and gave it a white color. However, I hit a problem called z-buffer fighting. This happens when there are multiple planes/triangles, that hit the same number in the z-buffer after projecting into clip space. I solved the problem by pushing the near camera plane further away from the camera spot, which increased the buffer precision by lowering the range mapped into it.

z-buffer fighting





Writing into deformation texture

The method that I am implementing requires atomic writes into texture. Since the paper doesn't show much about the technical stuff and the few bits that are there are for DX11, I had to figure this out myself. The OpenGL documentation does mention atomic operation for images, but it is very sparse in terms of the necessary stuff that has to be done. For example, it didn't explicitly mention that for atomic operations support, I need to use 'image' uniform type instead of 'sampler'. But the most troublesome part was, that for 'image' uniform types, not only the texture has to be bound, but also a single layer of that texture has to be bound into so called image unit by special call 'glBindImageTexture'. I burnt several hours figuring out why the writing and reading to/from my deformation texture doesn't work, and finally discovered this command by luck. I wonder where are these preconditions written, because I really couldn't find them...

Once the deformation texture become usable, it was time to write the deformation according to the descriptions from the paper. The hardest part of this step weren't actually the bit shifts to store the two numbers in single uint. I spent more time mapping the texture correctly onto the ground around the camera, with configurable resolution, so that I can tune this parameter later. This step is almost done, but the texture is now moving with the camera, so the deformation traces in the snow are moving as well, which is obviously wrong. Other than this, the writing and reading from the texture works fine. There is however a second part that I haven't implemented yet, and that is the snow elevation along the trails.

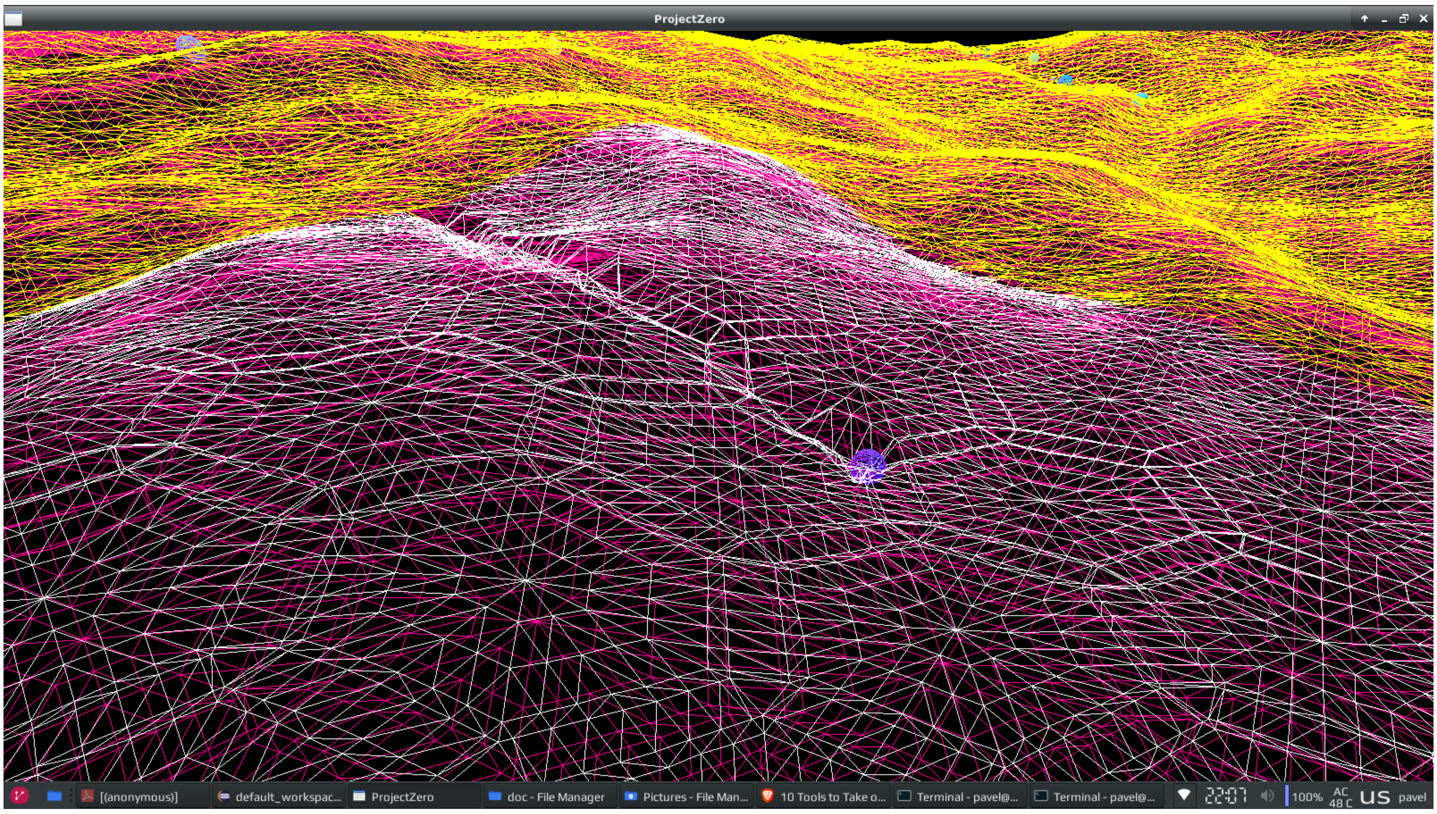
Distance units refactoring

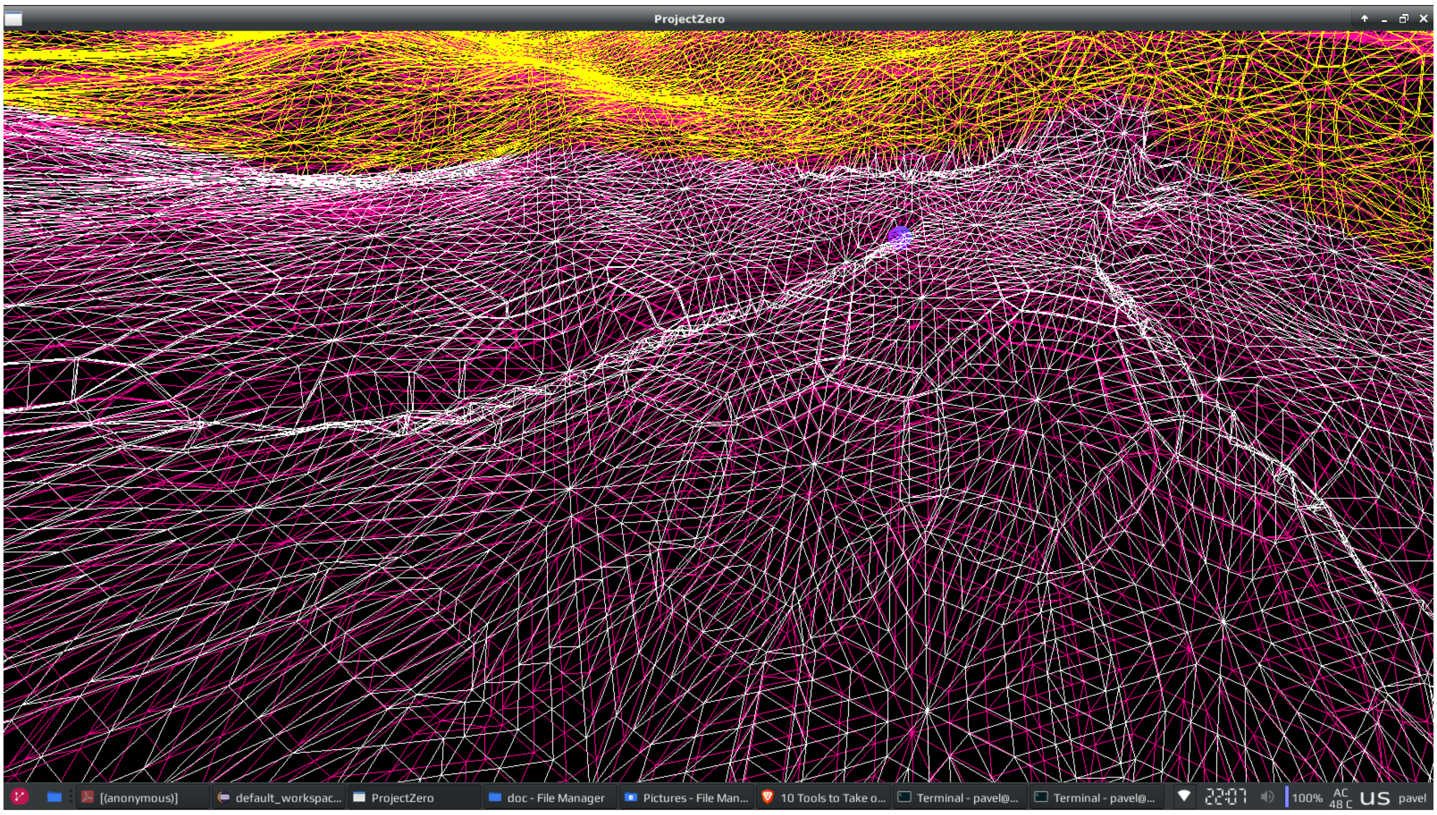
The OpenGL supports only 32 bit integer type for atomic operations, so I had to store the two deformation values into 16 bit values. This however meant that the deformation was rounded to the nearest integer, and that didn't work well with my mapping: 1 numeric unit = 1 meter. So I refactored the measure units to achieve configurable units per meter ratio.

Following steps

The first thing to do now is to make the mapping of deformation texture to terrain work properly, so that the snow trails don't move together with the texture. Next step is probably going to be shading implementation, because I am currently hitting a performance drops with OpenGL polygon mode and the visibility of the stuff is also not enough for the details like snow elevation.

deformation - white represents the deformation texture mapping, the yellow is outside the deformation texture so no deformation is happening


ProjectZero
[(anonymous)]
default_workspac...
ProjectZero
doc - File Manager
Pictures - File Man...
10 Tools to Take o...
Terminal - pavel@...
Terminal - pavel@...
22:07
100%
US
pavel


ProjectZero
[(anonymous)]
default_workspac...
ProjectZero
doc - File Manager
Pictures - File Man...
10 Tools to Take o...
Terminal - pavel@...
Terminal - pavel@...
22:07
100%
US
pavel