Proposal for COS412 Class Project
Elliot Chandler, 27 Jan. 2019

For this semester, I propose the implementation of more dynamic landscape elements in my existing game.

The first priority is implementing terrain deformation based on collision by masses or forces e.g. vehicle wheels. This should be relatively simple, as the behavior has been studied and a published algorithm is available and looks like something I could implement with some work. This should be implemented in a way that allows storing and deterministically reproducing the results of deformation, so changes to the landscape are persistent between sessions. (Relatedly, it would be useful to be able to reset the terrain to its default state, and interesting to have some method of allowing terrain editing within the game, although creating a good user interface for the latter would likely be a challenge.)

Separately, to get better behavior of wheels (e.g. correct handling of sharp dropoffs without the rendered wheel glitching through the land) it will likely be necessary to find a replacement for (or more likely, implement a replacement for) the current WheelCollider component, since it only seems able to collide at a single point.

These first three changes should substantially increase the enjoyment of the game, as it would allow the environment to feel more variable and dynamic.

If I can implement those this semester, which itself is probably no small task, there are other things that would also be nice to have. First, the water should be able to have better behavior. Specifically, it should correctly move when things enter it, and it should be able to have flow vectors within it that affect the player characters when they are submerged. These are a bit more complex. The first involves modelling the wave behavior (presumably only computing the effects on the volume in the immediate vicinity of the object moving in the water, and blending it into unchanged volumes further away), and the latter I don't even know how I'd start going about that. (Note that I'm not proposing full volumetric flow modelling, in that e.g. water would not be affected by gravity and the flow forces would be constant and predefined rather than equalizing as it gets itself organized like it does in reality; I think that would be even more ridiculously difficult.) Second, it would be nice to have destructible environment objects (trees, etc.), which could also have effects upon them be persistent.

If I accomplish all of those goals, I could probably think of more, but I suspect even just the first couple are hugely ambitious, given that there's a lot of mathematics in how these things move and interact, but I find mathematics and similar manners of thinking very difficult.

I have collected a few resources that I think will be valuable in the course of developing these changes. Here is a list of what I have selected for consideration so far:

- Sumner, Robert W., O'Brien, James F., and Hodgins, Jessica K. 1999. "Animating Sand, Mud, and Snow". In *Computer Graphics Forum*, vol. 18 no. 1 pp. 17–26. http://web.archive.org/web/20190125213550/http://graphics.berkeley.edu/papers/Sumner-ASM-1999-03/Sumner-ASM-1999-03.pdf
- Knutsson, Viktor. 9 March 2016. "Terramechanics based wheel-soil model in a computer game environment". Master's thesis, Umeå Universitet. http://web.archive.org/web/20190125222644/http://www.diva-portal.org/smash/get/diva2:914164/FULLTEXT01.pdf
- Zagrebelnyy, Pavel. 16 Nov. 2017. "Mud and Water of Spintires:MudRunner". Web. https://web.archive.org/web/20190127213144/https://www.gamasutra.com/blogs/PavelZagrebelnyy/20171116/309626/Mud_and_Water_of_SpintiresMudRunner.php

As well as some selections from Unity's documentation and Web site, the links from class, *etc.*:

- Unity Technologies. "Manual: Colliders". 2018. In *Unity: Documentation*, 2018.3 ed. http://web.archive.org/web/20190127214804/https://docs.unity3d.com/Manual/CollidersOverview.html
- Unity Technologies. "Scripting API: Collider.bounds". 2018. In *Unity: Documentation*, 2018.3 ed. http://web.archive.org/web/20190127214757/https://docs.unity3d.com/ScriptReference/Collider-bounds.html
- Tomer-Barkan *et al.* "Difference between bounding volumes of a collider and actual collider". First publ. 15 Nov. 2013. In *Unity Answers*. http://web.archive.org/web/20190127214732/https://answers.unity.com/questions/576028/difference-between-bounding-volumes-of-a-collider.html
- IgorAherne *et al.* "Am I inside a volume? (without colliders)". First publ. 8 Jan. 2014. In *Unity Answers*. http://web.archive.org/web/20190127214740/https://answers.unity.com/questions/611947/am-i-inside-a-volume-without-colliders.html
- Cold999 *et al.* "Getting the Volume of an Uneven Mesh". First publ. 26 Aug. 2015. In *Game Development*. http://web.archive.org/web/20190127214748/https://gamedev.stackexchange.com/questions/106318/getting-the-volume-of-an-uneven-mesh
- TriangleKing *et al.* "How does SetDensity find volume, and how can I get it?". First publ. 26 May 2013. In *Unity Answers*. http://web.archive.org/web/20190127214816/https://answers.unity.com/questions/463659/how-does-setdensity-find-volume-and-how-can-i-get.html
- Catlike Coding. "Mesh Deformation, a Unity C# Tutorial". *n.d.* Web. http://web.archive.org/web/20190127214829/https://catlikecoding.com/unity/tutorials/mesh-deformation/