Dynamics for Games and Animation

Task 3 Report - Multiple Systems

Student Name: Khang Trinh Student ID: 102118468

===

Resources Used

> List (with names/titles) any websites/links that you found useful/helpful - don't include things like search engines

Campfire Making a campfire in Unity Campfire model

Water How real-time fluid simulation works A 2D way of achieving this simulation A custom made fluid system in Unity Asset Store (costs $$$$$)

Ray Marching Intro to Ray Marching 1 How Ray Marching works Everything about Ray Marching & Signed Distance Functions Signed distance function cheat sheet Smooth min for intersection

Compute Shaders Basics of writing shaders (HLSL) An overview of Compute shaders How to write basic compute shaders Writing HLSL for raymarching

> Which software did you use for this task?

Unity: Constructing/exporting the scene Photoshop: Making textures

===

Task Details

Scene Design/Vision

> Describe what scenario you were aiming to create, and what 'events/interactions' would be occurring, include reference images to help describe the look you were aiming for (include references/source links).

Event summary: A campfire starting and being extinguished by water. **Video References:**

- Overview of the entire FX

- Fire extinguishing in slowmo
- Fire burning wood on loop

Dynamic Systems and Forces

Referring to the physics phenomena behind the interactions of your scenario, are there any real-world interactions you were trying to mimic? What composition of forces did you put together to achieve more complex movement? Describe/explain, and use diagrams/reference images to support. **Specific interactions:**

Phase 1 - Starting the fire

- Fire burning wood:
  - Flame has more yellow colors than red
  - Very light grey smoke (amount dependent on firewood material)
  - Embers (amount/frequency dependent on firewood material)

Fig 1. Campfire reference image

Phase 2 - Extinguishing the fire

- Fire being extinguished by water
  - Flame shrinks overtime as more water is poured into the wood
  - Rate of embers emitting reduces as more water is poured
  - Just before the flame disappears, a lot more smokes will be emitted (they also seem to have a much longer lifetime, and they seem to be heavily affected by wind, probably because they're much denser now)
- Water splashes from being poured into the firewood
  - Has a certain viscosity level (friction with the surface and other water particles)
- There might have been water droplets being vaporized after touching the flame, but in the video this wasn't noticeable. So, while it might have happened, I won't be replicating this aspect to save performance.

Fig. 2 How much smoke can be emitted from extinguishing a campfire with water

Scene Composition Overview

As a summary/overview, list and describe the objects you have used/created in your software/file (i.e. "what is it, why is it there, what is it for?") Include a screenshot showing the composition in the software editor (showing force icons, object tree, names of objects, etc...)
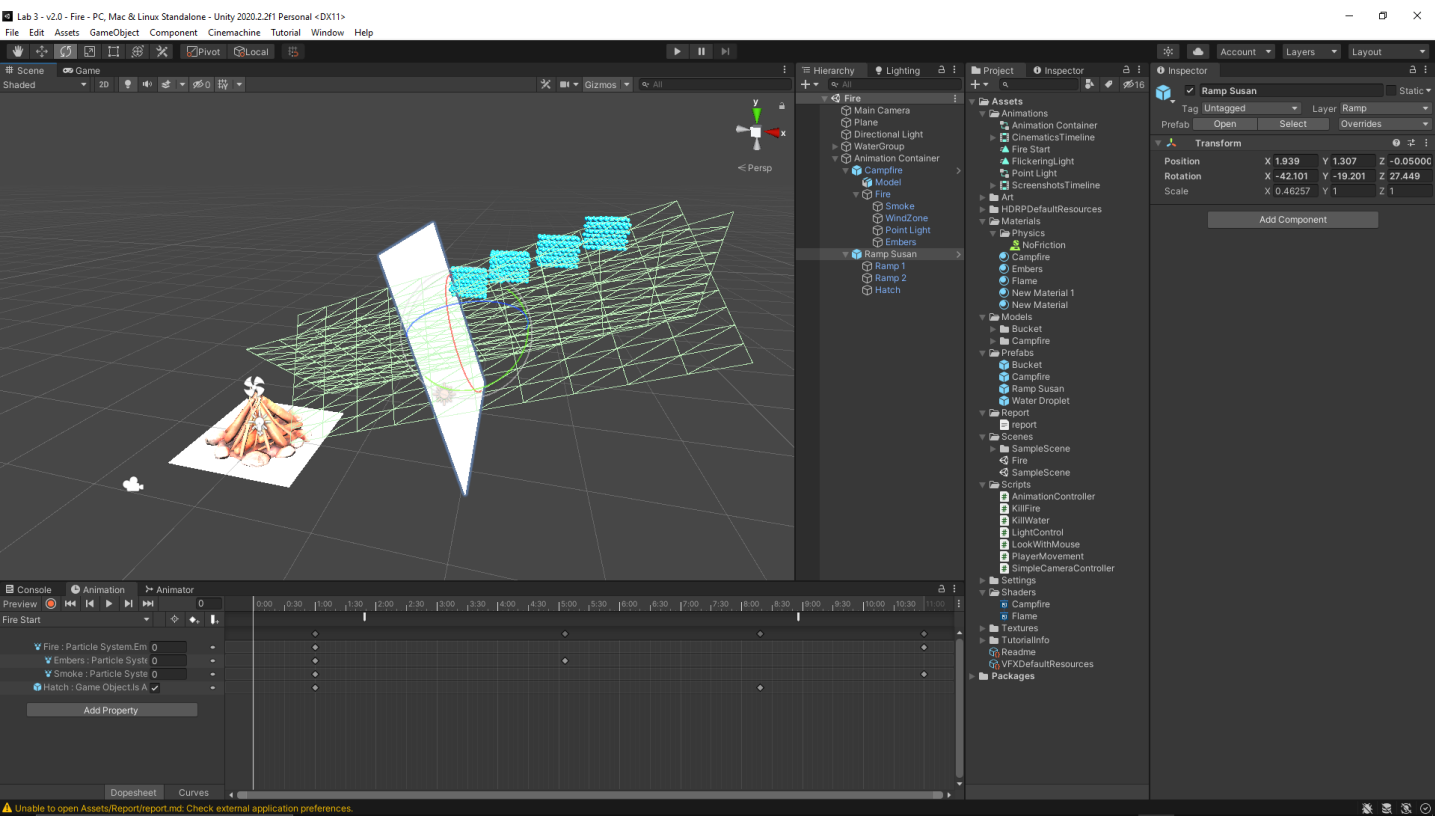

Fig. 3 Scene overview

- A container for ramp and campfire: This exists because I wanted to only have 1 animation component controlling everything that happens in the scene. To do that, the object that with the component has to be the parent for all child objects that I want to control, which in this case is the ramp's hatch and the many variables for the fire fx.
- Ramp: Consists of 2 planes intersecting each other and 1 in the middle as a hatch to hold the water particles.
- Campfire
  - Model
  - Particle systems
    - Windzone: This helps shaping the fire into the typical teardrop shape

    - Light (this is childed to the fire fx rather than the root object since it belongs to the fx, so moving the light with the fx makes more sense than moving the light with the campfire model).
- Water particles: This exists instead of a shuriken particle system for reasons noted below.

Systems Used

What systems did you use (rigid body, particles, cloth, hair/fur, etc...), and what are some key parameters/settings did you use/why? Describe/explain, and use diagrams/reference images to support.

- Flame, smoke and embers: shuriken particle system was used.
- The water could've been simulated using the shuriken system, but the built-in collision module was hard to use, and they didn't support the use of physics material.
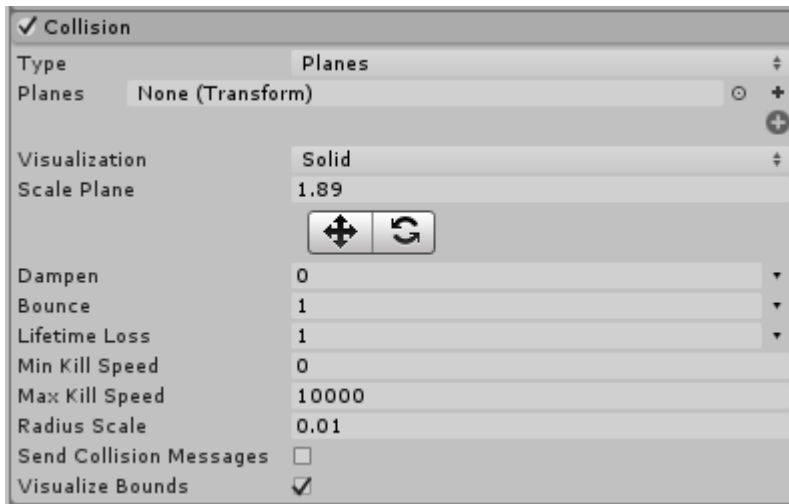
Fig. 4 Collision module for the shuriken system (there are no slots to insert the physics material, unlike a normal collider does)
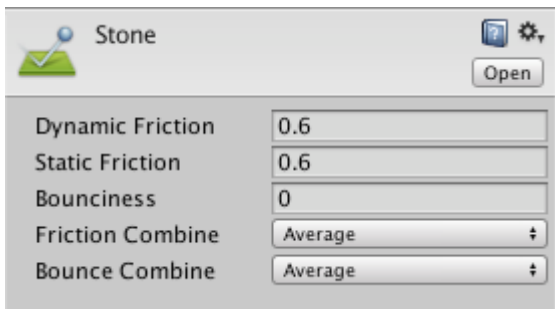

Fig. 5 A physics material

Physics materials allow control over an object's friction and bounciness against a surface, and not being able include this meant it would be difficult to make the system behave like water. So, I decided to make the fluid system from scratch by creating/duplicating multiple spheres in the scene, giving them a rigid body and physics material to make each water particle behaving like a blob of water, while hoping I could learn ray marching in time to complete the simulation.

Other Objects and Concepts

> List/explain any extra things you have added to your scene that aren't addressed in other parts of your report that you want to highlight.

- Campfire model as prop for the fire
- Invisible ramp to transfer water particles (having it visible was unnecessary since it blocks the view of the water particles)
- Flickering light - realism

==

Insights/Questions/Comments

Q: Did you attempt any new dynamic systems not covered in the unit so far (i.e. anything other than rigid bodies and particles)? What motivated you to try/use it (or not try them)?

A: Fluid system. It was the next step in my list of dynamic systems to learn.

Q: Did you run into any issues with any processes you did in this task? List/explain what you tried/how/if/ you solved it or not.

A: **1. Limited options for basic water simulation** Due to the system needing to be made from scratch instead of being built-in like the shuriken system, variables like density, viscosity & surface adhesion had to be re-interpreted differently (number of particles and friction against surface as well as adjacent particles respectively).

2. Creating Metaballs A key part of a fluid system is to be able to render particles into metaballs to make the entire group of particle look like one whole mesh. This required the understanding and implementation of ray marching in compute shader code, which turned out to be too much in the given timeframe (I only understand basic shaders language, but I've only ever made shaders in shader graph). The rendering of the particles was also out of scope of the unit (the technique focusses on rendering, while the unit's goal focusses on physically accurate simulation).

3. Colliders checking Due to bad optimization, collision checking with 1024 particles against a mesh collider was too much for a mid-range pc. A number of particles were passing through the campfire and the ground without colliding. Changing to a higher end pc temporarily fixed the issue, enough for exporting.

4. Importing materials for campfire model Textures were provided with the model, but the materials were broken as expected due to them originating from a different program. Custom shaders had to be created to accommodate the new textures.

Q: What was easier in this task than you expected? Why?

A: Making the fire. It was a reused asset from the meteor's trail, smoke and embers FX. I didn't expect a trail FX to be able to be converted into a static FX so easily.

Q: If you had extra time what would you want to improve on?

A:

- Implement ray marching to make water particles actually look like water
- Optimize to allow more water particles to exist in the scene (the more *and* smaller the particles the clearer the simulation can be shown)
- Camera FX: Heat distortion

Q: Any other comments/feedback/ideas?

A: