Sprint 1 – Scope of Work

Team 9

Dennis, Dhruv, Josh, Isaac, Matthew, Rodney

1. General Goals

The general goal for this project was to create a simplified molecular dynamics simulation in virtual reality, utilizing the Unity3D game engine. Conventionally, these simulations were usually done using supercomputers, with each simulations taking a large amount of time, and are used to help scientists figure out how a molecule would react to another molecule without having to expose themselves to the danger of conducting it live in a laboratory environment. This project, however, was done on the purpose of education, aiming to provide engaging and exciting chemistry simulation in virtual reality for high school, and first-year university students. After the project is done, it will be open sourced on GitHub, with the hopes of being used as a foundation for future development in this area.

2. Expected Results

Through the meeting with the client, it is made clear that the team are expected to meet several criteria. First, the team is not expected to handle the virtual reality aspect of the simulation and are instead expected to construct the building blocks for the simulation.

2.1 Functional Objectives:

From the meeting, the team along with the client have derived several objectives to be fulfilled.

Objective 1: Being able to run real time at 60-90 FPS

Frame rate greatly impacts the user experience. Unlike normal visual files or video games, VR programs are meant to “replace” the user’s surroundings, this means that the simulations need to run as close as possible to real life conditions. When the simulation isn’t acting in speeds as close as possible to real life, the movements will start to look choppy, and the items will look like they’re behaving in an unexpected fashion. The more these items jump around and jitter in an abnormal way, the more prone the user is to motion sickness, which is something we don’t want our users to experience when using our simulation.

Objective 2: Multithreaded code

Using Multithreading allows multiple processes to be run simultaneously, this means tasks like updating multiple particles as well as the effect of different forces on it would be done quicker, allowing the simulation to run smoother. This increased processing speed would also help accommodate the previous fps requirement.

Objective 3: Be able to house 10 Particles at a time

As explained before, simulations like the one our team is making are usually ran on supercomputers, with millions of particles interacting with each other. As this project is a smaller scaled one, the client only asked for the program to be able to house 10 particles on the simulation at the same time.

Objective 4: Isolated Environment

The program is expected to run several simulations in parallel. As stated in the previous objective, the program should be able to house 10 particles at the same time, the simulation should have the ability to run several different simulations, completely isolated from each other, giving the users the ability to observe different reaction outcomes from different configurations of particles.

Objective 5: The ability to create and destroy particles

The user should be able to add particle, as well as remove them from the simulation at any time. In the future, the simulation could be expanded, and more and more elements could be added, the user should be able to choose which element they want to work with and erase them from the “world” after they are done with it.

Objective 6: Scaled

The elements and particles of the simulations is expected to be identical in size and movement speed with the actual particles. This is important so that the simulation could be ran accurately, according to how the particles would react in real life, making the simulation more suitable for studying chemical reactions.

However, in addition to the expected objectives, the team have decided to set several extra goals for the project, which includes:

1. Setting up a “classroom environment” for the simulation sandbox.
   The team decided that the simulation would look better and be more user friendly if the simulation is running in a classroom background compared to a blank background. While it is not required by the client, the team feels like this is an objective that could be fulfilled, as one of the team members is experienced in 3D Modelling.

2.2 Technical Requirements

Language and Code Requirements

The main language that will be used in this project is C#, as the VR simulation will be made on Unity. Throughout the project, the team will keep modularity in mind. The project was created to be a foundation for further development in this area, thus, the team decided that each aspect of the code should be as independent as possible. The use of modularity in the design allows for future work to be done on only specific parts of the program, which is intended to be changed. Modularity also allows the team to divide the work between the members, as well as helping the team pinpoint any bugs and technical issues encountered during the project.

In addition to this, the client also emphasized that a well written code is crucial for this project, in order to accommodate and assist with further development of the project.

Hardware Requirements

The final simulation will be ran using a gaming laptop with the following specifications:

- CPU: Intel i7-8750H
- GPU: Nvidia GeForce GTX1060
- RAM: DDR4-2400 MHz 16GB

Subject Focused Requirements

- Interatomic Potential
  One of the team's main task is to emulate different reactions between particles, as well as the forces affecting the movement of each particles. The client have pointed out 3 vector fields the team would need to consider when attempting to emulate how particles would behave with each other:
    - Coulomb Potential
    - Lennard-Jones Potential
    - Morse Potential

- Collision Detection
  One of the main challenges of the project is to come up with a collision detection strategy for the particles, the collision detection system for the simulation would need to be foolproof, to tackle problems such as tunnelling, in which an object might move at a highspeed, that they just past through each other without the system detecting the collision.

- Data Structures
  The client emphasized that a good design architecture is crucial for this project, and one of the things we need to consider is how to store the data for each particle in the

simulation. The client suggested using an Entity Component System, a system which consists of the 3 terms as the name suggests.

- Entity: An entity in the ECS system is the tangible items in the simulation (the particles). The entity however, does not have any behavior or data, it is just a tangible representation in the sandbox.
- Component: a component is what gives the entity it's data, while it does not run processes and make the entity behave a certain way, it provides information of the entity's appearance, functionality, and behavior.
- Systems: is the functions that will be performed by the entity, such as graphics rendering, or how the particles will react with each other.

The ECS system would be beneficial for the project, as it would accommodate multi-threading, and the parallel processes used in the projects.