7th April 2019

3D Collision Simulation

Final Project Report

By
Rotte Priyanka Ajay - 160001051
Varakantham Vandana - 160001060
Computer Graphics and Visualizations Lab Project (CS352)
Course Instructor: Dr. Somnath Dey

Objective:

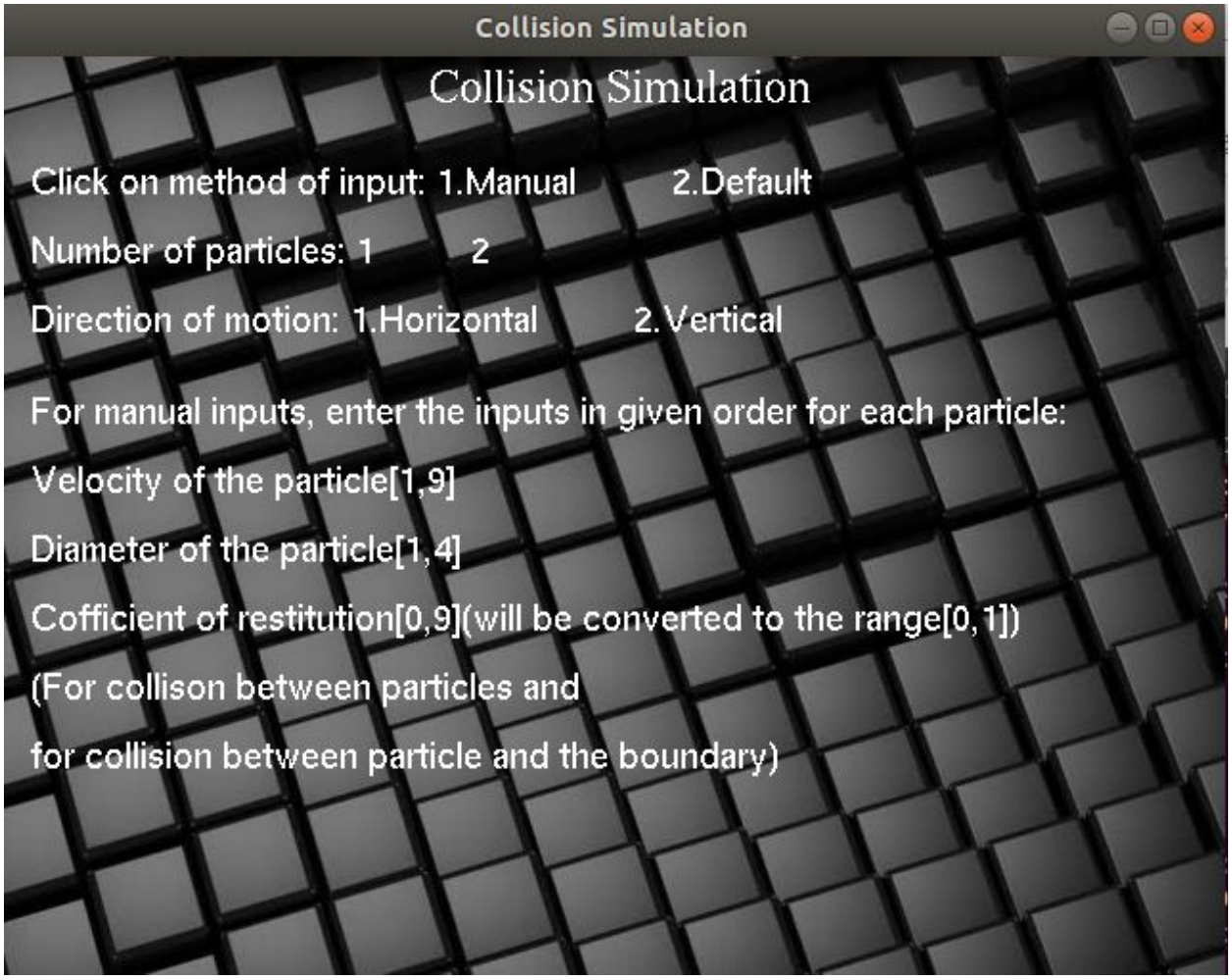
To develop a 3D model to simulate elastic and inelastic collisions between different objects

How to run the code:

To run the code, type in the terminal:

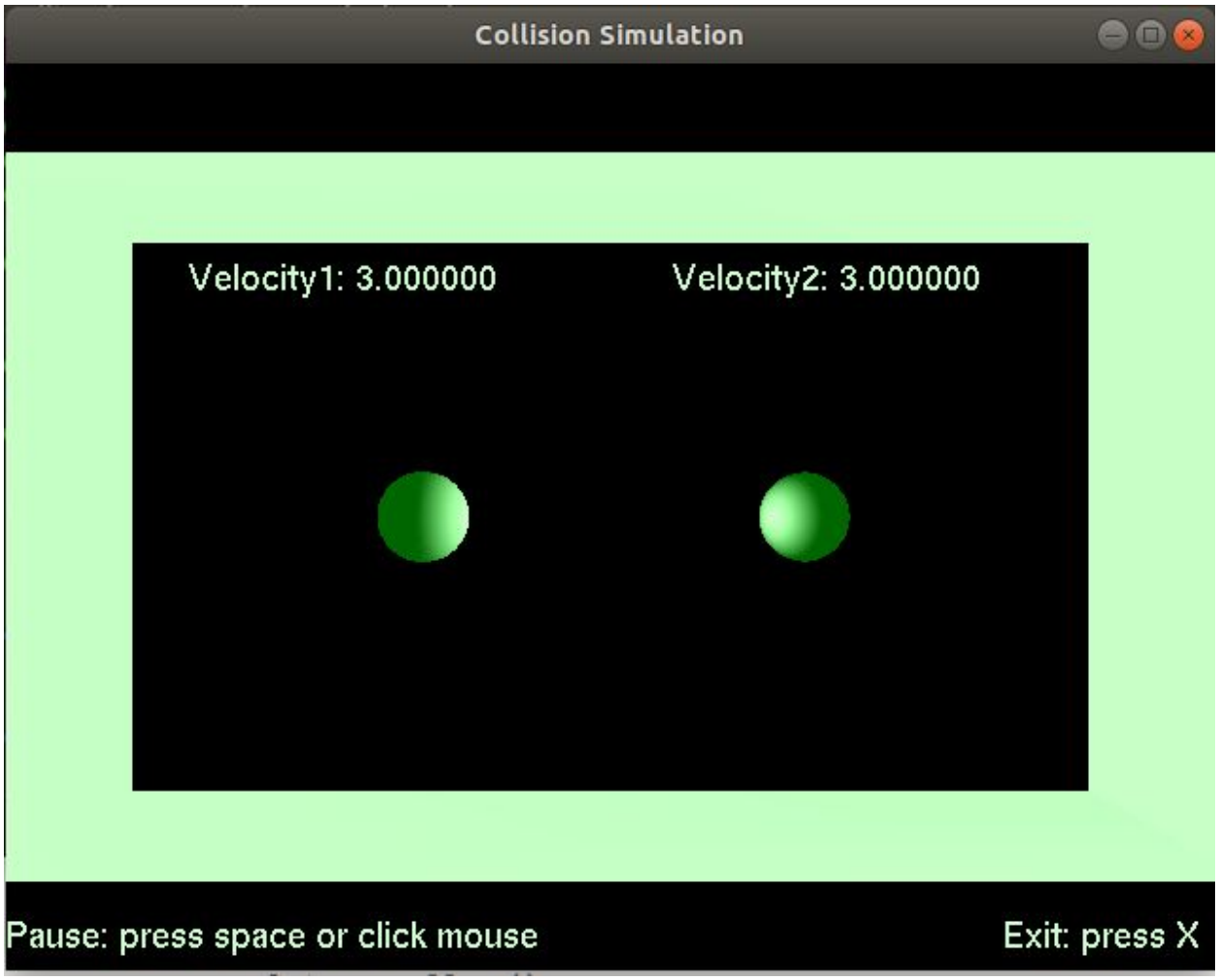
```
~$ g++ Collision.cpp -lGL -lGLU -lglut
~$ ./a.out
```

The console will ask the user to enter the space key. User inputs for the simulation will be asked in the new window opened.



There are two input methods and as the user clicks on the required option or types in the values of given parameters, values will appear in the console. After all the values have been inputted, the simulation will start.

```
ajay@ajay-Inspiron-7560:~/Desktop/cs352/collision$ g++ Collision.cpp -lGL -lGLU
-lglut -lSOIL -lIL
ajay@ajay-Inspiron-7560:~/Desktop/cs352/collision$ ./a.out
Enter space key
Manual
2 Particles
Horizontal
v2: 3
r2: 0.5
e1: 1
v1: 3
r1: 0.5
e: 1
Simulation started!
```



During the simulation, changing velocities of the particles can be seen on the screen.

Simulation can be paused/resumed by pressing the space key or left-clicking the mouse.
After pressing X (shift+x), the window will close.

Components of the model:
The code is written in C++ using OpenGL Utility ToolKit.
User input is taken using the keyboard as well as mouse.
The text is written in the window for input instructions as well as for showing the velocities.
Boundaries and the objects are in 3D and are colored.
There is a constant light source from where shading can be seen on the moving objects.
Also, texture is added in the background.

Working of the model:

The formula for the velocities after a collision is:

$$v_a = \frac{C_R m_b (u_b - u_a) + m_a u_a + m_b u_b}{m_a + m_b}$$
$$v_b = \frac{C_R m_a (u_a - u_b) + m_a u_a + m_b u_b}{m_a + m_b}$$

where

v_a is the final velocity of the first object after impact

v_b is the final velocity of the second object after impact

u_a is the initial velocity of the first object before impact

u_b is the initial velocity of the second object before impact

m_a is the mass of the first object

m_b is the mass of the second object

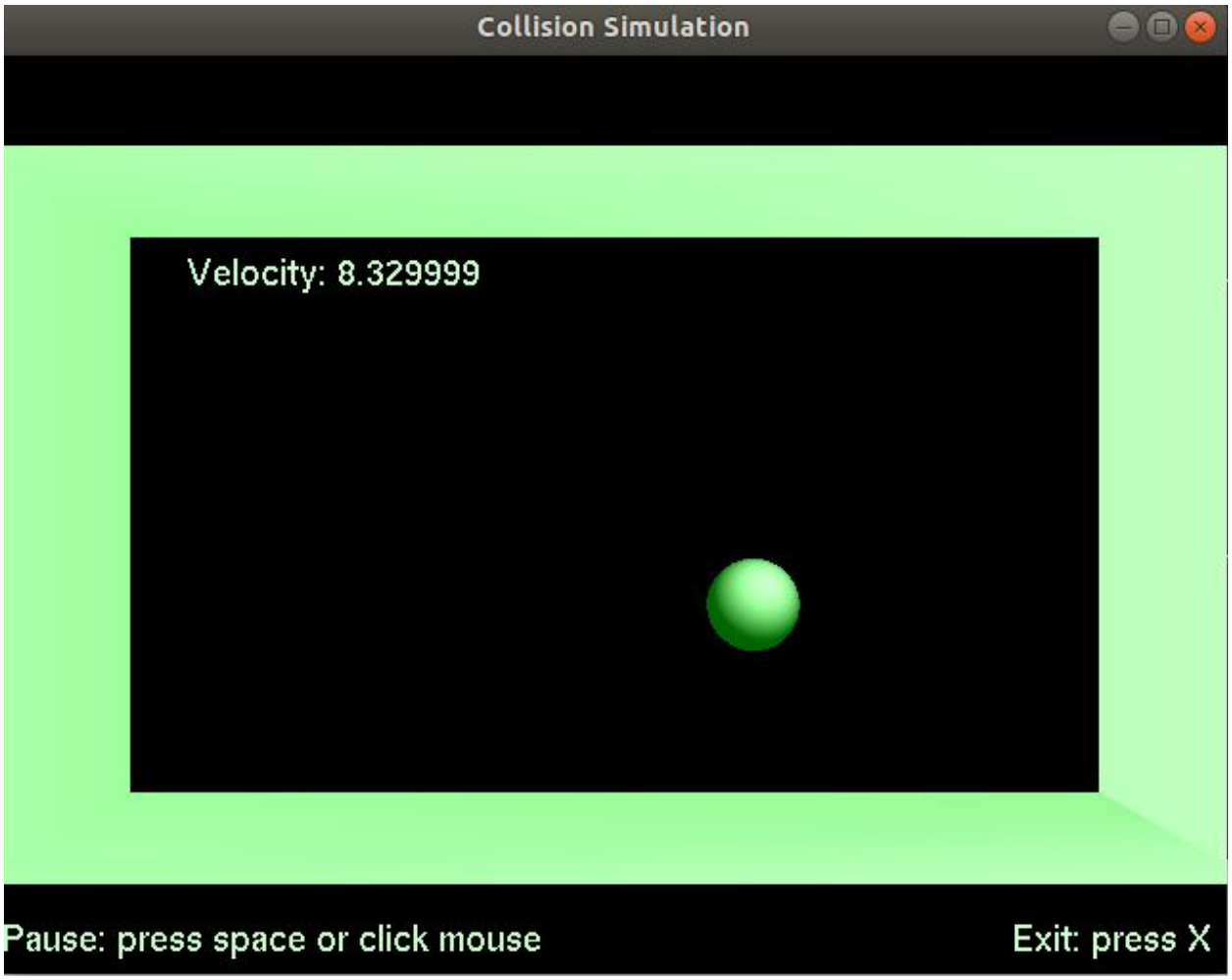
C_R is the coefficient of restitution; if it is 1 we have an elastic collision; if it is 0 we have a perfectly inelastic collision.

Mass of the object is directly proportional to (radius)^3(Assuming constant density).
For boundaries, it is assumed that the mass >> mass of the objects and velocity is zero.

Using these formulas, we developed the code for collision.
Case 1:
Number of particles: 1
Motion: Horizontal
In this case, the object will move between the two boundaries and its velocity will change according to the formula. There is no acceleration involved.

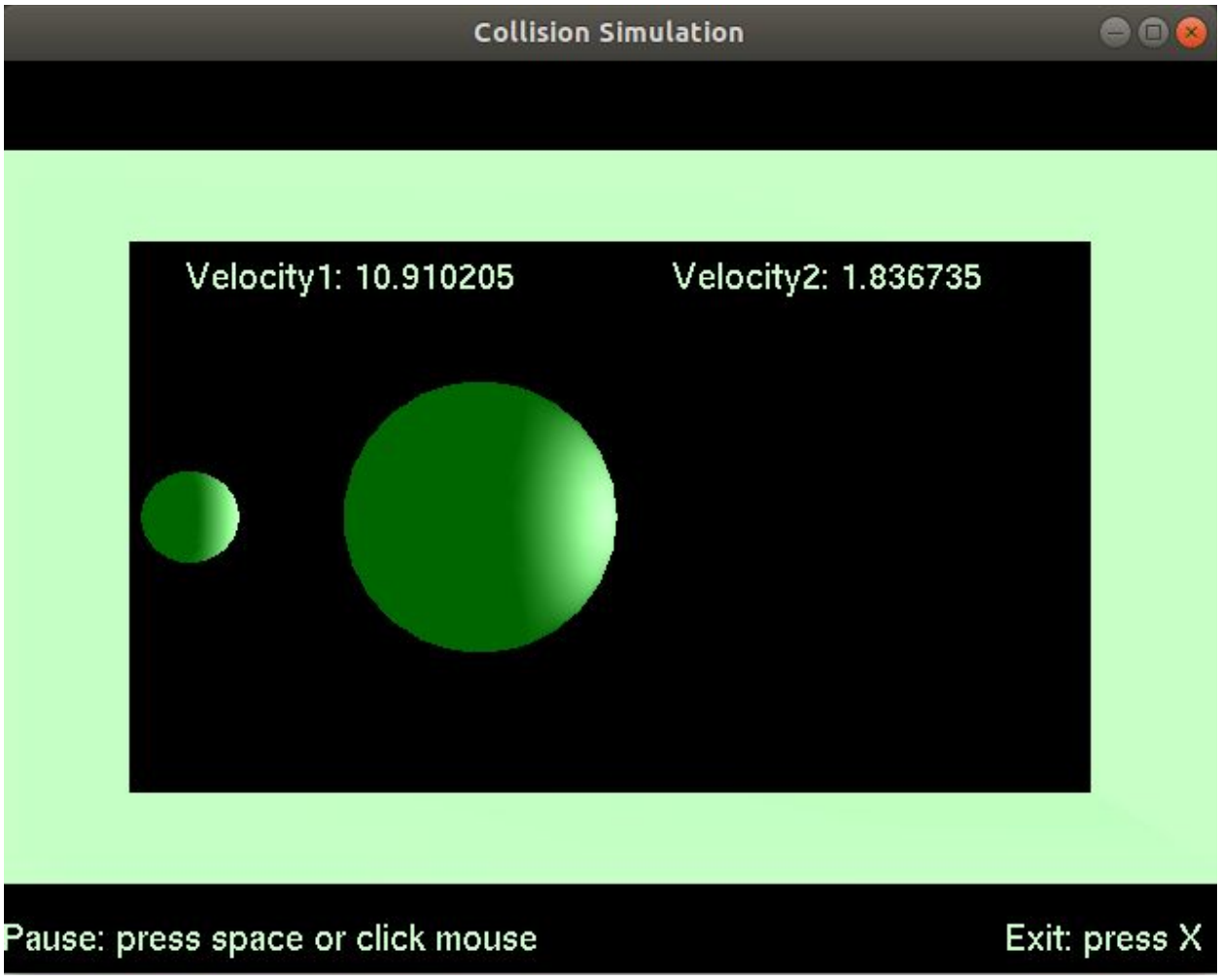
Case 2:
Number of particles: 1
Motion: Vertical
In this case, the object will fall down from the top with some initial velocity and accelerate downwards due to

gravity. After the collision, it will go upwards with deceleration and will eventually stop.



Case 3:
Number of particles: 2
Motion: Horizontal
Two objects will come in the opposite direction and after the collision with each other, depending on their mass and initial velocities, final velocities and directions will be decided. As the objects go back to the boundaries, there

will be a collision between the boundary and the object and it will continue.



Complete Code can be found here.