Zense Project Report

2D Collision of Balls

Idea: To make a program to visually simulate a 2D box with balls inside. The program is able to detect collisions between any two balls and resolve the collision accurately with regard to physics. It uses some basic vectors which was also written by me.

Technology used: Python 2.7 and pygame

Implementation: The program consists of three main parts: moving and drawing the balls, detecting collisions and resolving collisions.

Moving and drawing is simple. The ball's velocity vector is added to the ball's position vector at each iteration of the main loop(100 iterations per second). The ball is represented by a circle drawn in a pygame window.

The collision detection uses a grid system. The box is divided into a grid. The size of the cell is decided taking into account the balls radius. For each iteration, collision check happens for balls in the same cell and those in adjacent cells. Care has been taken so that redundancies don't take place(if ball[i] has been checked with ball[j], then ball[j] won't be checked with ball[i]). The grid system is more efficient than the brute force method(but significant differences will be seen only with large number of balls).

In resolving collisions, the major issue to address is the fact that the balls can be moving with speeds higher than 1 pixel per iteration. So, by the time collision detection detects collision between two balls, they might have already moved into each other. So, to preserve physical accuracy, the motion of the two colliding balls are reversed to the point where the just touch. Here all the calculations are done using vectors and basic physics to determine the new velocities. The reversed motion is now forwarded again, with the new velocities so the balls go to where the actually would have been in a real collision.

The program allows you to modify the radius of the balls, the number of balls and the coefficient of restitution between the balls and the wall(presently, only elastic collision can take place between the balls).

Future scope: the program itself can be improved. The balls can have individual radii and mass. With enough computational power, a 3D implementation can be used to simulate an ideal gas. This program can be useful in games like carom or billiards. The grid system of collision detection can be used in other games with a large number of particles.

Overall Experience: The most difficult part of this project was accurately translating the physics into computer code. The first attempt was only partially successful with the collisions sometimes showing abnormal behavior. The successful attempt was a fresh start after trying to fix the old algorithm for many hours. This project also helped me realize further how OOP is useful.

Video links:

https://youtu.be/59oFkynkX3k
https://youtu.be/lStanar51Jg