Games programming Lab02: Point Masses and Particle Systems

Games heavily use particle in the gameplay. These include explosions, weather effects, fire, destruction particles, etc. Your challenge in this lab is to implement and explore them.

Lab 2: Tasks

1. Practice vectors by representing and adding simple physical animation.

2. Create a class for the object you are going to apply the physics to. This should be different from the class that displays the object. For example, you can be displaying a more graphically complex object than the one you are running the physics for. This is usually is the case in games and it makes the physics simulations more straightforward.

3. Use the Newtonian mechanics approach covered in the lecture to produce a 3D object to which you can drop from the top of your volume of space to the bottom under gravity. When it gets to the bottom just make it vanish.

4. Create a class for a particle in a particle system. This can be similar to the last class, but it needs a lifespan property: an integer that can be counted down to 0 would do. It will also need an accessor to check whether the particle is dead or not.

5. You'll also need an emitter class that creates new particles and initialises them. You could use a fixed number and an array to hold them, or a variable number and a vector, which

actually behaves rather like a list. If you have 360 particles you can rotate 1 degree for each to get a circular effect. Note there is no standard degree to radian conversion function in C++, use your own or the maths library.

6. To visualize the particles, you need a class that represents a particle either as a point or a single quad. This needs a field for transparency, so the particle can fade away, and also for RGB colour: you might want different particles to be different colours.

7. Create a 3D explosion with a large number of particles.

8. Enhance your particle to create more effects,

Simple Impacts

Games heavily use physics to calculate and process all parts of gameplay (movement, logic, AI, etc). Your challenge in this lab is to implement and explore the simplest impact and collision physics.

In the previous code, you dropped a particle down the screen. Turn this into a bouncing ball where the bounce uses the physics of impact. You will need to calculate the forces, bounding boxes and physics.

Note that unless you take some energy out of the system at each bounce, the ball will bounce forever, or until you exit.