CO1301: Games Concepts
Lab 16 - Particles

1. Introduction

1. This lab introduces particles and particle systems.
2. A particle is an object with properties such as size, velocity and position. A particle system defines and controls the behavior of several particles to produce special effects such as sparks, smoke, fire, explosions etc.

2. Setting Up

1. Create a new TL-Engine project called "Lab16_ParticleSystem_Project".
2. Load and create a skybox and floor model using "stars.x" and "floor.x" respectively.
3. Create a 'kManual' camera and position it at (0, 60, -180).
4. You need to call the Timer() method in order to establish the first base time. The command needs to be placed outside the game loop, but make sure to place it immediately before the game loop. Call it using the following code:

```
myEngine->Timer();
```

5. You then need to call Timer() once per frame, at the beginning of the game loop, immediately after `myEngine->DrawScene();` . Declare a variable of type float called frameTime to store the time taken for the last scene to be rendered:

```
float frameTime = myEngine->Timer();
```

6. Use the draw command to output the frame time and FPS (frames per second can be calculated by 1/frameTime) to the screen.

3. One Particle

1. Load the model file: "quad.x" from the media folder on Blackboard, and implement particle movement as set out in the instructions below.
2. The particle should start at the world origin (0, 0, 0), and should move vertically with a starting velocity of 110 units. Remember to adjust the velocity using the frame time.
3. The particle will, of course, shoot off the screen when you run the program.
4. Now apply gravity to the particle. A reasonable value for gravity is -50 units. Gravity is acceleration. Use the equation given in the lecture to apply the acceleration (force of gravity).

$$v = v' + a \times \Delta$$

where new velocity = v
previous velocity = v'
gravity = a

5. The effect of gravity is downwards. Each frame you need to calculate the new velocity of the particle according to the effect of gravity. Note that gravity has a negative value. Use the frame time for delta-t (Δt).
6. You then move the particle using the new velocity (adjusted still by the frame time). The effect is the particle will rise up into the air and then fall downwards.

4. Rebirthing the Particle

1. When the particle hits the floor you need to reset its velocity. In this way the particle will be emitted, rise through the air; fall to the ground and then the process starts again
2. Orientate the particle every frame towards the camera by using the LookAt() method.

5.Movement to either side

1. It would be nice if the particle has a variance to its movement.
2. Define a structure to hold the movement vector of your particle. I would suggest the following:

```
struct Particle
{
    IModel *model;
    float moveVector[3];
};
```

3. You also need to define a function that returns a random value within a range of numbers:

```
// Return a random number in the range between rangeMin and rangeMax inclusive
// range_min <= random number <= range_max
float random( int rangeMin, int rangeMax)
{
        float result = (float)rand() / (float)(RAND_MAX + 1);
        result *= (float) (rangeMax - rangeMin);
        result += rangeMin;

        return result;
}
```

4. Use the random function to modify the direction of movement which the particle takes as it comes out of the emitter by assigning its result to the x-component of the particle as it leaves the emitter.
5. You can use right-angle triangles to calculate a new value of the y component of the particle if you want. This will ultimately produce a better particle system.

6. Advanced: Particle System

1. Create an array of particles. You can safely create an array of 1000 particles.
2. Start them moving. Emit one particle very frame.
3. Use a loop with the array to apply gravity and move the particles
4. When a particle hits the floor then reset its starting velocity as above.
5. If you have implemented a variance to the initial direction of movement of the particles then you will create a "fan" and something much like a fountain.