3D Game Programming
Lab Exercise

Compile and build programs in Release mode. Run a program in /bin/release.

Implement the programs in the folders p1, p2, etc.

p1. In this program, we are going to create a scene with multiple objects. The objects are arranged uniformly in a circle. The number of objects and the radius of the circle are input by the user.

Steps:
1. add createScene in TutorialApplication
2. in TutorialApplication::createScene
    - set ambient light colour
    - create entities
    - create scene nodes
    - attach an entity to a scene node

Problems:
- how can we generate unique names for the entities?
- how can we generate unique names for the scene nodes?

Pseudo code for createScene

```
1. set ambient light
2. ask to input numOfObjects and ask to input radius r.
3. for i = 1 to numOfObjects do
4.      sname = generateNameForSceneName( i )
5.      ename = generateNameForEntity( i )
6.      create scene node S
7.      create entity node E
8.      attach E to S
9.      a = i/( (float) numOfObjects ) *2*pi, (in radian)
10.     compute position p for object i, p = r*(sin a, cos a), where (a, r) are the polar
coordinates, a is the angle, and r is the radius of the circle.
11.     set position p for object i
12.     scale the object properly
13. end for loop
```

p2. Multiple scenes, multiple cameras, and multiple viewports.

Create four scene managers and four scenes. Then render the four scenes on the screen. The background colors of the viewports are yellow, green, blue and gray.

Thus, we need to create four scene managers, four cameras and four viewports. Furthermore, we can set the viewport to be focused by pressing key ‘1’, ‘2’, ‘3’ or ‘4’. Consider that a viewport is focused. Then the viewing direction of the camera can be changed by manipulating the mouse. There is only one viewport that can be focused at a time. The background colour of a focused viewport is light gray.

Thus we can finish this program in the following:

Step 1:

- create four scene managers, four cameras and four viewports
- create scenes

Step 2:handle keyboard and mouse events

- keyboard events: change the focus, get the pointer of the focused viewport, get the pointer of the corresponding camera
- assign the pointer of the camera to the pointer of the camera of the system.
- Assign background colour to the focused viewport

p3. Simple Animation.

Animate a set of free-falling spheres.

Steps:

1. Create a set of spheres.
    - create scene nodes
    - create entities
    - attach an entity to a scene node
    - assign initial velocities to the spheres

```
vx =random([0, 1]); // generate a random value within the interval
vy =random([0, 2]);
vz =random([0, 1]);
Vector3 v = Vector3 ( vx, vy, vz);
assign v to a sphere.
```

2. Free fall simulation based on Newton's second laws
    - update velocity

```
v = v + g*time_step, where g is the gravitational constant
```

    - update the position the spheres

```
position = position + v*time_step
If ( position.y < 0.0 ) {  // hit the floor
        // bounce up
        v = -v;
        position.y = 0.0 + epsilon (a small positive value);
}
```

Assumption: the y-coordinate of the floor is 0.