16.01.20

09.30 - 11.30am

CMPU 4030 Games Engines 1

KE G43, Kevin Street

Programme Code: DT211C, DT228, DT282, DT508
Module Code: CMPU 4030
CRN: 22528, 22418, 31083, 27945

TECHNOLOGICAL UNIVERSITY DUBLIN

KEVIN STREET CAMPUS

BSc. (Honours) Degree in Computer Science (Infrastructure)

BSc. (Honours) Degree in Computer Science

BSc. (Honours) Degree in Computer Science (International)

Year 4

DT508 BA. (Honours) in Game Design

Year 3

SEMESTER 1 EXAMINATIONS 2018/19

Games Engines 1

Dr. Bryan Duggan
Dr. Deirdre Lillis
Ms. Pauline Martin – DT211C
Mr. Patrick Clarke – DT228/282

2 HOURS

INSTRUCTIONS TO CANDIDATES

ANSWER QUESTION 1 (COMPULSORY) AND ANY 2 FROM THE REMAINING QUESTIONS
QUESTION 1 IS WORTH 40 MARKS, THE REMAINING QUESTIONS ARE WORTH 30 MARKS EACH

Question 1

A spawner system for creatures in a virtual world programmed in C# for Unity has the following features:

1. There is a list of possible creature types and these are spawned in order.
2. Creatures are spawned inside the players field of view, a random distance from the player between a minimum and maximum range.
3. Creatures spawns on top of the environment.
4. Creatures that are a certain distance behind the player get suspended.
5. When a new creature is needed, a previously suspended creature can be used instead of instantiating a new one.
6. Creatures fade in when they are created.
7. Creatures are spawned and suspended at a programmer specified rate.
8. The spawner is a singleton.

Taking each of the features above, how would you program them in Unity?

(8 x 5 marks)

Question 2

(a) Discuss the relationship between the quantities of *force*, *velocity*, *acceleration*, *position*, *distance*, *time* and *mass* in relation to 3D computer games. In your answer include:

i. Units of measurement and representations for these quantities. (5 marks)
ii. Equations that describe the relationships. (5 marks)
iii. A description of how to update the state of a Newtonian physics particle with respect to time in a 3D computer game. (5 marks)

(b) A *gravity gun* in 3D games allows the player to grab an object and hold it at a point in front of the camera. Explain in detail how you would implement a gravity gun effect in a Unity project.

(15 marks)

Question 3

(a) In relation to digital audio, explain the following terms: *sample rate*, *resolution*, *frame size*, *spectrum*, *bin width*.

(10 marks)

(b) Figure 1 shows an extract from a Unity C# script that visualises the frequency spectrum of an AudioSource.

```
void CreateVisualisers()
{
    float theta = (Mathf.PI * 2.0f) / (float)AudioAnalyzer.frameSize;
    for (int i = 0; i < AudioAnalyzer.frameSize; i++)
    {
```

```
            Vector3 p = new Vector3(
                Mathf.Sin(theta * i) * radius
                , 0
                , Mathf.Cos(theta * i) * radius
                );
            p = transform.TransformPoint(p);
            Quaternion q = Quaternion.AngleAxis(theta * i * Mathf.Rad2Deg, Vector3.up);
            q = transform.rotation * q;

            GameObject cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
            cube.transform.SetPositionAndRotation(p, q);
            cube.transform.parent = this.transform;
            cube.GetComponent<Renderer>().material.color = Color.HSVToRGB(
                i / (float)AudioAnalyzer.frameSize
                , 1
                , 1
                );
            elements.Add(cube);
        }
    }

    // Update is called once per frame
    void Update () {
        for (int i = 0; i < elements.Count; i++) {
            elements[i].transform.localScale = new Vector3(1, 1 + AudioAnalyzer.spectrum[i] *
scale, 1);
        }
     }
```

Figure 1

(c) In relation to the code answer the following questions:

(i) What shape will the generative visual have? How is the position of each segment in the visual calculated?

(ii) How is the orientation of each segment calculated?

(iii) How is the colour of each segment determined? What will the colour look like?

(iv) What aspect of the visual will be affected by audio? How would you improve the visual so that it was more responsive to the audio characteristics of music?

(4 x 5 marks)

Question 4

(a) Compare *jobs* with *threads*

(10 marks)

(b) **Figure 2** shows an extract from a procedural animation system that implements a harmonic motion. In porting this code to the C# job system, a `SwayJob` struct is created that extends `IJobParallelForTransform` a new class `SwayManager` is created to manage and schedule the job.

i. What fields should `SwayJob` and `SwayManager` have?

(10 marks)

ii. What methods will `SwayJob` and `SwayManager` have in order to process, manage and schedule the job?

(10 marks)

```
public class Sway : MonoBehaviour {
    public float angle = 20.0f;
    public float frequency;
    public float theta;
    public Vector3 axis = Vector3.zero;
      // Use this for initialization
      void Start () {
        if (axis == Vector3.zero)
        {
            axis = Random.insideUnitSphere;
            axis.y = 0;
            axis.Normalize();
        }
    }
      void Update () {
        transform.localRotation = Quaternion.AngleAxis(
            Mathf.Sin(theta) * angle, axis);
        theta += frequency * Time.deltaTime * Mathf.PI * 2.0f;
      }
}
```

Figure 2