17/1/2019

09.30 - 11.30am

Courtyard, DIT Aungier Street

DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science
DT282 BSc. (Honours) Degree in Computer Science (International)

Year 4

DT508 BA. (Honours) in Game Design

Year 3

SEMESTER 1 EXAMINATIONS 2018/2019

GAMES ENGINES 1 [CMPU4030]

DR. BRYAN DUGGAN
DR. DEIRDRE LILLIS
MR. PATRICK CLARKE

THURSDAY 17TH JANUARY 9.30 A.M. – 11.30 A.M.

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 3D tower defence game made in Unity has the following rules:

1. Players place towers by using the mouse to select a location on the map.
2. Towers become active when a creep comes in range.
3. When a tower becomes active it will turn to face the creep and continue targeting the creep so long as it stays in range.
4. Towers can fire 5 bullets per second.
5. Bullets disappear after 5 seconds if they don't hit anything
6. There are three possible types of creeps and each has an equal probability of being spawned.
7. Creeps follow a path to get to the players base.
8. When a creep is hit with a bullet, it explodes and after a few seconds, sinks into the ground and gets removed from the scene.

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

(8 x 5 marks)

Question 2

(a) When generating a mesh procedurally, a programmer may generate arrays of *vertices*, *normals*, *colors*, *triangles*, and *uv*. Identify the purpose and data type of each of the italicised terms.

(15 marks)

(b) A terrain mesh contains a grid of quads of size 50 x 50. How many vertices are in this mesh assuming no vertices are shared between triangles?

(5 marks)

(c) Figure 1 is an extract from a game component that generates a terrain mesh. Explain the main features of this code.

(10 marks)

```
Vector3 bottomLeft = new Vector3(-samples.x / 2, 0, -samples.y / 2);
coll = AddComponent<MeshCollider>();
  int vertex = 0;
   for (int y = 0; y < samples.y; y++)
   {
       for (int x = 0; x < samples.x; x++)
       {
           Vector3 sliceBottomLeft = bottomLeft + new Vector3(x, 0, y);
           Vector3 sliceTopLeft = bottomLeft + new Vector3(x, 0, y + 1);
           Vector3 sliceTopRight = bottomLeft + new Vector3(x + 1, 0, y + 1);
           Vector3 sliceBottomRight = bottomLeft + new Vector3(x + 1, 0, y);

           sliceBottomLeft.y += SampleCell(x, y) * amplitude;
           sliceTopLeft.y += SampleCell(x, y + 1) * amplitude;
```

```
                sliceTopRight.y += SampleCell(x + 1, y + 1) * amplitude;
                sliceBottomRight.y += SampleCell(x + 1, y) * amplitude;

                int startVertex = vertex;
                gm.initialVertices[vertex++] = sliceBottomLeft;
                gm.initialVertices[vertex++] = sliceTopLeft;
                gm.initialVertices[vertex++] = sliceTopRight;
                gm.initialVertices[vertex++] = sliceTopRight;
                gm.initialVertices[vertex++] = sliceBottomRight;
                gm.initialVertices[vertex++] = sliceBottomLeft;

                for (int i = 0; i < 6; i++)
                {
                    gm.meshUv[startVertex + i] = new Vector2(x / samples.x, y / 
samples.y);
                    gm.meshTriangles[startVertex + i] = startVertex + i;
                }
            }
        }

        mesh.vertices = gm.initialVertices;
        mesh.uv = gm.meshUv;
        mesh.triangles = gm.meshTriangles;
        mesh.RecalculateNormals();

        coll.sharedMesh = null;
        coll.sharedMesh = mesh;
```

Figure 1

Question 3

(a) Figure 2 shows an extract from a generative physics system that creates the caterpillar animat given in Figure 3.

```
     void Awake()
    {
        float depth = size * 0.05f;
        Vector3 start = - Vector3.forward * bodySegments * depth * 2;
        GameObject previous = null;
        for (int i = 0; i < bodySegments; i++)
        {
            float mass = 1.0f;
            GameObject segment = 
GameObject.CreatePrimitive(PrimitiveType.Cube);
            Rigidbody rb = segment.AddComponent<Rigidbody>();
            rb.useGravity = gravity;
            rb.mass = mass;
            segment.name = "segment " + i;
            Vector3 pos = start + (Vector3.forward * depth * 4 * i);
            segment.transform.position = transform.TransformPoint(pos);
            segment.transform.rotation = transform.rotation;
            segment.transform.parent = this.transform;
            segment.transform.localScale = new Vector3(size, size, depth);
            segment.GetComponent<Renderer>().shadowCastingMode =
UnityEngine.Rendering.ShadowCastingMode.Off;
            segment.GetComponent<Renderer>().receiveShadows = false;

            segment.GetComponent<Renderer>().material.color =
Color.HSVToRGB(i / (float)bodySegments, 1, 1);

            if (previous != null)
            {
                j.autoConfigureConnectedAnchor = false;
                j.anchor = new Vector3(0, 0, -2f);
                j.connectedAnchor = new Vector3(0, 0, 2f);
                j.axis = Vector3.right;
                j.useSpring = true;
                JointSpring js = j.spring;
                js.spring = spring;
                js.damper = damper;
                j.spring = js;            }
            previous = segment;
        }
    }
```

Figure 2

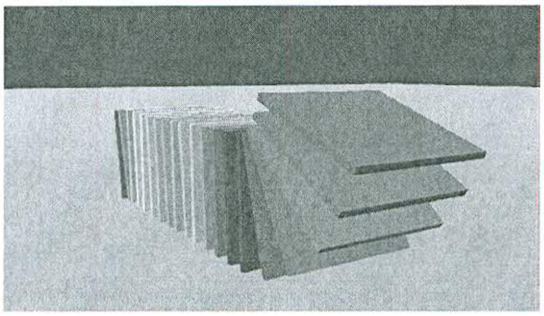

Figure 3

Answer these questions about the code:

(a) How are the position, rotation, anchor points and scale of each segment calculated? Include a diagram in your answer.

m

(10 marks)

(b) How are the segments constrained to move relative to one another?

(10 marks)

(c) How is the colour of each segment determined?

(3 marks)

(d) What should the hierarchy look like after the Awake method has been called?

(2 marks)

(e) How would you procedurally animate the caterpillar so that torque is applied to to each segment in sequence?

(5 marks)

Question 4

(a) What are the main features of the C# Job System?

(5 marks)

(b) Figure 4 shows an extract from a procedural animation system that implements a harmonic motion. How would you convert this code to use the C# Job System? In your solution include a description of:

i. What new classes/structs that you would need to create.

(5 marks)

ii. What fields and their types would need to be on these new classes/structs.

(10 marks)

iii. What methods you would need to create on the new classes/structs

(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 4