CMPU4030

****

**DUBLIN INSTITUTE OF TECHNOLOGY**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DT228 BSc. (Honours) Degree in Computer Science**

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

**DT211 BSc. (Honours) Degree in Computer Science (Infrastructure)**

**Year 4**

**DT508 BA. (Honours) in Game Design**

**Year 3/4**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SUPPLEMENTAL EXAMINATIONS 2018/2019**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Games Engines 1**

Dr. Bryan Duggan

Dr. Deirdre Lillis

Mr. Patrick Clarke

Mr. Alan Fahey

Mr Sergio Calabria

TBC

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 bot manager system in a Unity real time strategy (RTS) game has the following rules:

1. There are three possible bot types and each has an equal probability of being spawned.
2. There should always be 5 bots in the scene.
3. Bots are spawned at a rate of 2 bots per second.
4. Bots are spawned in front of the player at random positions, a minimum of 500 units and a maximum of 2000 units from the player.
5. When a bot is spawned, it will travel from its spawn point to a random position within a 200 unit radius of the player.
6. If the player comes within 200 units the bot will shoot at it
7. When the bot is hit with a player bullet it explodes and dies
8. When a bot dies, it can respawn at a new location,

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

(8 x 5 marks)

**Question 2**

1. How can you efficiently determine if one game object is in front of or behind another game object?

(5 marks)

1. Explain in detail how you would implement an FPS controller component that implements walk, strafe, pitch and roll using keyboard and mouse. What changes would you need for this to be usable in VR?

(20 marks)

1. How would you adapt your solution to part (b) to implement third person camera following?

(5 marks)

**Question 3**

1. 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:
2. Units of measurement and representations for these quantities. (5 marks)
3. Equations that describe the relationships. (5 marks)
4. A description of how to update the state of a Newtonian physics particle with respect to time in a 3D computer game. (5 marks)
5. 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 4**

1. Compare *jobs* with *threads.*

(10 marks)

1. Figure 1 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.
2. What fields should SwayJob and SwayManager have?

(10 marks)

1. 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 1