



### **Autumn Examinations 2019**

**Course Instance Code(s)** 3BCT  
**Exam(s)** BSc (CS&IT)  
**Module Code(s)** CT3111  
**Module(s)** Next Generation Technologies  
**Paper No.** 1  
**External Examiner(s)** Dr. Jacob Howe  
**Internal Examiner(s)** Prof. Michael Madden  
\*Dr. Sam Redfern

**Instructions:** Answer any three questions.  
All questions carry equal marks.  
Note that the final page of this exam paper lists useful classes from the Unity3D SDK.

**Duration** 2 hours  
**No. of Pages** 4  
**Discipline(s)** Information Technology  
**Course Co-ordinator(s)** Dr. Des Chambers

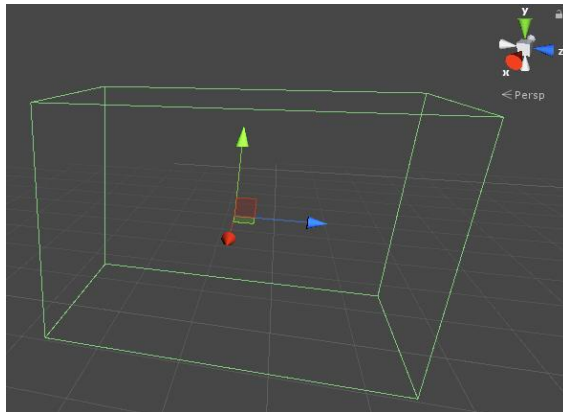
**Requirements:**

Release in Exam Venue	Yes	<input type="checkbox"/>	No	<input type="checkbox"/>
MCQ Answersheet	Yes	<input type="checkbox"/>	No	<input type="checkbox"/>
Handout	None			
Statistical/ Log Tables	None			
Cambridge Tables	None			
Graph Paper	None			
Log Graph Paper	None			
Other Materials	None			
Graphic material in colour	Yes	<input checked="" type="checkbox"/>	No	<input type="checkbox"/>

## Q.1.

(i) Explain how Unity's MonoBehaviour class provides tight integration with the Game Loop. Refer to appropriate methods of the MonoBehaviour class in your answer. [6]

(ii) What is a Coroutine in Unity, and how do Coroutines integrate with the Game Loop? [4]



(iii) The Game Object depicted has a Box Collider component, whose 'isTrigger' property is true. A script on the game object contains a reference to the Box Collider and to a prefab of a ball.

```
public BoxCollider bc;  
public GameObject ball;  
  
public IEnumerator SpawnBallsInBox(){  
}
```

Write code for the SpawnBallsInBox() coroutine, so that it continually instantiates balls, at a rate of one ball every two seconds. The balls should be initialised to a random position somewhere inside the Box Collider. (Hint: use the 'bounds' property of the Box Collider, which has 'min' and 'max' properties, each of which are of type Vector3). [10]

## Q.2.

Making appropriate use of local and global co-ordinates, write Unity3D/C# code to perform the following transformations. You may assume that references to the runtime gameobjects are provided:

- rotate a gameobject 5 degrees around its own x axis [2]
- move a gameobject 6 units downwards in the world's co-ordinate system [2]
- move a gameobject 7 units directly towards another gameobject [3]
- move a gameobject 10 units forward in whatever direction it is facing [3]

(ii) Write code for the following method, which considers the supplied list of objects and returns the one which is furthest away from the specified 3D point: [10]

```
public static GameObject GetFurthestObject(List<GameObject> objects, Vector3 pos) {  
  
}
```

### Q.3.

(i) In 3D games development, what does the term '**raycast**' mean, as supported by various static methods of the Unity3D SDK's Physics class? Explain, with illustrative C# code, how you could use a raycast to allow the user to click with the mouse and select a gameobject from the scene [10]

(ii) In a shooting game, assume you are using raycasts to determine what the player has hit when they fire their gun. You may assume that you are given a reference to the gun object in the 3D scene.

- Write appropriate Unity3D/C# code to perform a raycast when the gun is fired, to determine what is hit by the bullet. The gun should have a maximum range of 500 metres. [6]
- Write appropriate Unity3D/C# code to instantiate an 'explosion' object at the position that the bullet hits. You may assume that a prefab exists for this explosion object. [4]

### Q.4.

(i) Bearing in mind that, in Unity's physics engine, gravity only operates along a fixed world vector, how could you simulate a moon orbiting a planet? Write Unity3D/C# code to achieve this, identifying the appropriate methods in which it should be written, as well as identifying the appropriate component(s) which have been added to the game objects. [10]

(ii) Write Unity3D/C# code to accomplish the following:

- instantiate a gameobject at runtime, from a prefab [2]
- obtain a reference to the Rigidbody component which is assumed to be attached to it [2]
- attach a new Rigidbody to the gameobject, if it did not have one already [3]
- set the gameobject moving in a straight line using the physics engine [3]

### Q.5.

Write technical notes on each of the following [5 x 4]

- (i) How you would display (and update) a score on the screen while a game is being played, using the Unity GUI system.
- (ii) Garbage collection in Unity, including how to write low-garbage code.
- (iii) Triggers and Colliders in Unity – how to use them and why they're useful for games development.
- (iv) Screen space, viewport space and world space in Unity.

## Some Useful Unity3D SDK Classes

### **GameObject: static methods**

Instantiate()	Destroy()	DestroyImmediate()	Find()
---------------	-----------	--------------------	--------

### **GameObject: methods**

AddComponent()	SendMessage()	GetComponent()	SetActive()
----------------	---------------	----------------	-------------

### **GameObject: data members**

activeInHierarchy	transform	tag	
-------------------	-----------	-----	--

### **MonoBehaviour: methods**

Start()	OnDestroy()	Awake()	Update()
FixedUpdate()	LateUpdate()	OnDisable()	OnEnabled()
OnBecameInvisible()	OnBecameVisible()	OnCollisionEnter()	OnCollisionExit()
OnCollisionStay()	OnTriggerEnter()	OnTriggerExit()	OnTriggerStay()
SendMessage()	BroadcastMessage()	SendMessageUpwards()	GetComponent()
GetComponentInChildren()	GetComponentInParent()	GetComponents()	GetComponentsInChildren()
GetComponentsInParent()	GetInstanceID()	Invoke()	StartCoroutine()

### **MonoBehaviour: data members**

enabled	gameObject	transform	name
---------	------------	-----------	------

### **Transform: methods**

Rotate()	Translate()	TransformPoint()	InverseTransformPoint()
LookAt()	RotateAround()	SetParent()	TransformVector()
InverseTransformVector()	TransformDirection()	InverseTransformDirection()	

### **Transform: data members**

position	localPosition	rotation	localRotation
lossyScale	localScale	parent	right
up	forward	gameObject	

### **Rigidbody: methods**

AddForce()	AddForceRelative()	AddForceAtPosition()	AddTorque()
AddRelativeTorque()	MovePosition()	MoveRotation()	

### **Rigidbody: data members**

drag	angularDrag	mass	velocity
angularVelocity	centerOfMass		

### **Camera: methods**

ScreenToWorldPoint()	WorldToScreenPoint()	ScreenToViewportPoint()	
ViewportToScreenPoint()	WorldToViewportPoint()	ViewportToWorldPoint()	
ViewportPointToRay()	ScreenPointToRay()		

### **Physics: static methods**

Raycast()	SphereCast()	OverlapBox()	BoxCast()
-----------	--------------	--------------	-----------