# CT3536 Marking Scheme, 2021-22

## Q.1.

(i)

| | |
|---|---|
| Written in an Update() method | [1] |
| Using Camera.main (or a public reference) to access the camera object | [1] |
| Applied to this.transform | [1] |
| Using transform.lookAt | [1] |
| Using camera.transform.position | [1] |

(ii)

| | |
|---|---|
| Update method. | [1] |
| transform.forwards as one argument to RotateTowards(). | [1] |
| (Player.activePlayer.transform.position – transform.position).normalized as the other argument. | [1] |
| Scaling rotation amount with Time.deltaTime. | [1] |
| Assignment of result to zombie object via transform.LookAt or similar. | [1] |

(iii)

| | |
|---|---|
| transform.forwards as one argument to Vector3.Dot(). | [1] |
| (Player.activePlayer.transform.position – transform.position).normalized as the other argument. | [1] |
| Testing for ‘almost straight’ via a dot product result of say >=0.8. | [1] |
| Updating transform.position by some value * transform.forward | [1] |
| Scaling movement amount with Time.deltaTime. | [1] |

(iv)

| | |
|---|---|
| Use of raycasting | [1] |
| Start raycast at zombie position | [1] |
| Raycast in direction to player (from zombie) | [1] |
| How is direction to player calculated | [1] |
| Use of layers to raycast only against boxes | [1] |

**Q.2.**

(i)

Definition of State Machine. [2]
Clear separation of logic at different times [2]
Example(s). [1]

(ii)

Screen space: on-screen pixels (2D). [1.5]
Viewport space: normalized on-screen position (2D). [1.5]
World space: position in the virtual world (3D). [1]
Camera class transforms between these spaces, according to the viewpoint of the camera. [1]

(iii)

Maintaining inactive objects in a data structure rather than destroying them [2]
How you set game objects inactive/active in Unity [1]
Importance of low-garbage code in games [2]

(iv)

Coroutines are Monobehaviour methods which can be paused for varying time [2]
Unity internally maintains a list of paused Coroutines and the Game Loop processes this each frame and resumes those whose pause time has elapsed. [1]
Situations might include: gathering timed logic together into one method; animating an object’s properties over times; waiting for other coroutines to end before continuing; carrying out CPU-intensive operations over multiple frames; and others! [2x1]

**Q.3.**

(i)

Collider is a component that defines a collision shape for use with physics. [1]
Typical use: to define physical bodies which will collide and respond, e.g. a ball and a floor. [1]
Trigger is a collider whose 'isTrigger' property is true (no physical collision response) [1]
Typical use: when you want your code to know when an object has moved into a region of space, e.g. a trap trigger in a dungeon. [1]
Interaction callbacks such as OnCollisionEnter() and OnTriggerEnter(). [2]

(ii)

OnCollisionEnter() happens when an object's collider touches another object's collider
Collision.contacts is an array of contact points. [2]
Iterate this array. [1]
Instantiate a 'spark' object at each contact's .point (which is a Vectror3 world position) [1]
Orient each 'spark' object to face towards the direction indicated by each contact's .normal (which is a Vector world direction). [2]

(iii)

transform.Rotate() [1]
Correct use of Space.Self [1]

transform.Translate() or direct setting of transform.position [1]
Correct use of Space.World [0.5]
Written in Update() [0.5]
Use of Time.deltaTime [1]

transform.Translate() or direct setting of transform.position [1]
Calculation of difference vector [1]
Normalization of vector and multiplication by 7 [1]

**Q.4.**

(i)

| | |
|---|---|
| Perform a transform.RotateAround() by a small amount each frame. | [3] |

(ii)

| | |
|---|---|
| Correct arguments (point, axis, angle). | [2] |
| Update() method. | [1] |
| No added components needed | [1] |

(iii)

| | |
|---|---|
| Reference to planet GameObject or its Transform. | [2] |
| Correct transform.RotateAround() arguments | [3] |

(iv)

| | |
|---|---|
| Inspector variable for rotation parent GameObject or its Transform. | [1] |
| Inspector variable for axis of rotation (Vector3). | [2] |
| Inspector variable for speed of rotation (float). | [1] |
| Update() method | [1] |
| Correct transform.RotateAround() arguments, including use of Time.deltaTime with speed of rotation | [3] |

**Q.5.**

(i)

| | |
|---|---|
| Raycast: source point and direction. | [2] |
| Testing for intersecting objects in the physics world | [2] |

(ii)

| | |
|---|---|
| Raycast downwards from a point inside the character, to a point just below their feet | [2] |
| True result means they're standing on something | [2] |
| Correctly written Physics.Raycast(). | [2] |

(iii)

| | |
|---|---|
| Iteration of list | [2] |
| Use of objects[i].transform.position | [3] |
| Calculation of vector difference magnitude | [3] |
| Returning of correct object | [2] |