COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

Department of Computer Science
Terminal Examination Spring 2024
Subject: Game Development
Class: BSE 7A/B
Max Marks: 30
Instructor: M. Ibtisam Gul
Date: 24-July-2024
Time Allowed: 30 Minutes

MULTIPLE CHOICE QUESTIONS

Instructions: Choose the correct option. Each question carries 1 mark.

---

1. Which of the following is not a common role in a game development team?
   A) Game Designer
   B) 3D Modeler
   C) Sound Engineer
   D) Network Administrator

2. What is the primary difference between Unity 2D and 3D?
   A) Graphics quality
   B) Camera perspective
   C) Audio quality
   D) Network capabilities

3. What is the main advantage of using the Universal Render Pipeline (URP)?
   A) Enhanced graphics quality
   B) Compatibility with a wide range of devices
   C) Faster rendering times
   D) Improved sound quality

4. What is the purpose of a material in Unity?
   A) To control the physics of an object
   B) To manage the animations of an object
   C) To define how a surface should be rendered, including textures, colors, and shader properties
   D) To handle user input

5. What is the purpose of baking lightmaps in Unity?
   A) To create dynamic lighting effects
   B) To enhance sound quality
   C) To manage network traffic
   D) To improve performance by pre-calculating lighting

6. Which of the following is an example of a game design element?
   A) Game Mechanics
   B) Programming Languages
   C) Development Tools
   D) Network Protocols

7. Which Unity method is used to update physics calculations?
   A) Update
   B) FixedUpdate
   C) LateUpdate
   D) OnGUI

8. Why is frame rate independence important in game development?
   A) To ensure consistent gameplay across different hardware
   B) To improve graphical quality
   C) To enhance sound effects
   D) To reduce development time

9. What is Cinemachine in Unity?
   A) A tool for creating animations
   B) A camera system for advanced control
   C) A physics engine
   D) A networking library

10. What is the main difference between a Collider and a Trigger?
    A) Colliders can detect collisions, Triggers cannot
    B) Triggers can detect collisions, Colliders cannot
    C) Colliders affect the physics of objects, Triggers do not
    D) There is no difference

11. Which component must an object have to detect collisions in Unity?
    A) Rigidbody
    B) Light
    C) AudioSource
    D) Animator

12. How can you check if a GameObject has a specific Tag in a script?
    A) Using the CheckTag method
    B) Using the CompareTag method
    C) Using the ValidateTag method
    D) Using the TagMatch method

13. How can you interact with UI elements using a script in Unity?
    A) Using the UIManager class
    B) Accessing the component through the Canvas
    C) Using Event Listeners
    D) Using Physics components

14. Which method would you use to apply force to a Rigidbody in Unity?
    A) AddForce
    B) ApplyForce
    C) SetForce
    D) EnableForce

15. How do you access the parent of a Transform in Unity?
    A) transform.root
    B) transform.parent
    C) transform.child
    D) transform.sibling

16. How can you create a Prefab in Unity?
    A) By right-clicking in the Scene view and selecting Create Prefab
    B) By dragging a GameObject from the Hierarchy into the Project window
    C) By using the Create menu in the Inspector
    D) By selecting an option in the Rigidbody component

17. What happens to the children of a GameObject when it is destroyed?
    A) They are also destroyed
    B) They remain in the scene
    C) They are moved to the root of the scene
    D) They are duplicated

18. What parameters are required to instantiate a GameObject at a specific position and rotation?
    A) Prefab, position, rotation
    B) Prefab, scale, position
    C) Prefab, rotation, scale
    D) Prefab, position, parent

19. Which keyword is used to pause a Coroutine in Unity?
    A) yield return
    B) pause
    C) wait
    D) delay

20. What is the primary purpose of Terrain in Unity?
    A) To create animations
    B) To render complex 3D models
    C) To generate large outdoor environments
    D) To handle UI elements

21. How do you apply acceleration to a vehicle in Unity?
    A) WheelCollider.motorTorque
    B) WheelCollider.acceleration

C) WheelCollider.move
D) WheelCollider.speed

22. Which component is typically included in a First-Person Character Controller?
A) Camera
B) AudioSource
C) Light
D) Animator

23. How do you determine what ray hits in Unity?
A) By checking the RaycastHit object
B) By using the RayDetect method
C) By using the Collision component
D) By analyzing the Transform component

24. Which component is essential for using the Timeline in Unity?
A) Animator
B) Rigidbody
C) AudioSource
D) Playable Director

25. Which component is used to control the playback of Animation Clips in Unity?
A) Rigidbody
B) Animator Controller
C) AudioSource
D) Collider

26. How can you create a Ragdoll in Unity?
A) Using the Ragdoll Wizard
B) By scripting it from scratch
C) By using the Audio Mixer
D) By importing from the Asset Store

27. How do you apply an impulse force to a Ragdoll in Unity?
A) Using AddForce with ForceMode.Impulse
B) Using ApplyImpulse
C) By increasing the Ragdoll's mass
D) By adjusting the Ragdoll's Animator component

28. Which process is used to combine two parents to produce offspring(s) in genetic algorithms?
A) Mutation
B) Crossover
C) Selection
D) Evaluation

29. What is Augmented Reality (AR)?
A) A technology that fully immerses users in a virtual environment

B) A tool for creating 3D models
C) A method for generating audio effects
D) A technology that overlays digital information on the real world

30. What is a common challenge when developing VR applications?
A) Creating realistic audio effects
B) Achieving high frame rates to avoid motion sickness
C) Designing 2D user interfaces
D) Implementing basic physics

COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

Department of Computer Science
Terminal Examination Spring 2024

Instructor: M. Ibtisam Gul
Subject & Class: Game Development-BSE-7A/7B
Date: 24-July-2024
Time Allowed: 150 Minutes
Maximum Marks: 45

Question 1 - CLO-2

Design a comprehensive game development strategy for a first-person shooter game. Your strategy should include the following elements:
a. Creating a first-person character controller, including basic locomotion and attack mechanics.
b. Creating an enemy controller with detection, follow, and attack capabilities.
c. Creating particle effects & sound effects to simulate realistic environments and combat scenarios.

Question 2 - CLO-3

Design an advanced character animation system for a Unity game that includes both animated sequences and ragdoll physics. Your design should include:
a. Creating an Animator Controller and integrating it with Animation Clips and an Avatar to manage complex character animations.
b. Creating Timeline Sequences to synchronize game events.
c. Assembling Ragdoll Physics to enable realistic character responses to physics interactions and damage.

Question 3 - CLO-4

Describe how the Start, Update, InitializePopulation, SelectAndBreed, Crossover, and Mutate functions work within the genetic algorithm. Explain the role of each function and how they collectively drive the algorithm toward optimal solutions.

Below is the provided code snippet for reference:

```
public class GeneticAlgorithm : MonoBehaviour
{
  [SerializeField] int numberOfCarsToSpawn = 50;
  [SerializeField] GameObject carPrefab;
  List<GameObject> cars;
  float startTime = 0f;
  float generationTime = 30f;
  float elapsedTime = 0f;
  int generationNumber = 1;
  void Start()
  {
    startTime = Time.realtimeSinceStartup;
    InitializePopulation();
  }
  void InitializePopulation()
  {
    cars = new List<GameObject>();
    for (int i = 0; i < numberOfCarsToSpawn; i++)
    {
      GameObject car = Instantiate(carPrefab, transform.position, transform.rotation);
      AIController aiController = car.GetComponent<AIController>();
      aiController.steeringSensitivity = Random.Range(0.1f, 0.3f);
      aiController.maxTorque = Random.Range(180f, 220f);
      aiController.lookAhead = Random.Range(18f, 22f);
      aiController.maxSteerAngle = Random.Range(50f, 70f);
      aiController.maxBrakeTorque = Random.Range(4500f, 5500f);
      aiController.accelCornerMax = Random.Range(18f, 22f);
```

```
            aiController.brakeCornerMax = Random.Range(3f, 7f);
            aiController.accelVelocityThreshold = Random.Range(18f, 22f);
            aiController.brakeVelocityThreshold = Random.Range(8f, 12f);
            aiController.antiroll = Random.Range(4500f, 5500f);
            cars.Add(car);
        }   }
    void Update()
    {
        elapsedTime += Time.deltaTime;
        if (Time.realtimeSinceStartup > startTime + generationTime)
        {
            elapsedTime = 0;
            SelectAndBreed();
        }
    }
    void SelectAndBreed()
    {
        startTime = Time.realtimeSinceStartup;
        List<GameObject> sortedCars = cars.OrderByDescending(o =>
o.GetComponent<AIController>().fitness).ToList();
        int halfList = (int)(sortedCars.Count / 2);
        cars.Clear();
        for (int i = 0; i < halfList; i++)
        {
            AIController parent1 = sortedCars[i].GetComponent<AIController>();
            AIController parent2 = sortedCars[i + 1].GetComponent<AIController>();
            Crossover(parent1, parent2);
        }
        foreach (GameObject car in sortedCars)
        {
```

```
            Destroy(car);
        }

        generationNumber++;
    }

    void Crossover(AIController parent1, AIController parent2)
    {
        GameObject offspring1 = Mutate(parent1, parent2);
        GameObject offspring2 = Mutate(parent2, parent1);
        cars.Add(offspring1);
        cars.Add(offspring2);
    }
    GameObject Mutate(AIController parent1, AIController parent2)
    {
        GameObject offspring = Instantiate(carPrefab, transform.position, transform.rotation);
        AIController offspringAIController = offspring.GetComponent<AIController>();
        offspringAIController.steeringSensitivity = parent1.steeringSensitivity;
        offspringAIController.lookAhead = parent1.lookAhead;
        offspringAIController.maxTorque = parent1.maxTorque;
        offspringAIController.maxSteerAngle = parent1.maxSteerAngle;
        offspringAIController.maxBrakeTorque = parent1.maxBrakeTorque;
        offspringAIController.accelCornerMax = parent2.accelCornerMax;
        offspringAIController.brakeCornerMax = parent2.brakeCornerMax;
        offspringAIController.accelVelocityThreshold = parent2.accelVelocityThreshold;
        offspringAIController.brakeVelocityThreshold = parent2.brakeVelocityThreshold;
        offspringAIController.antiroll = parent2.antiroll;
        return offspring;
    }
}
```