CS4555 Introduction to 3D Computer Game Programming
Sample Questions

- **Introduction and Game Engine**
  - Write at least five names of game genres.
  - What does FPS stand for?
  - Write at least two distinct features of FPS game genre.
  - Write at least two benefits that you gain when using a game engine for your game development.
  - What do you mean by "Platform-independent" game engine?
  - Write three types of game engines. To which game type does Unity belong?
  - Write at least three names of Open Source Game engines.

- **Basic Math for Game Programming (see "Basic Math for Game" lecture note)**
  - Let vector v=(2,4,4). Normalize it.
  - Let two vectors v and w be v=(2,2,1) and w=(1,-2,0). Calculate their dot product and cross product.
  - Given two vectors v=(1,0,0) and w=(0,1,0), calculate their dot product. What is the smaller angle between these two vector?
  - Given two points A=(-4,5) and B=(2,-2), define a line passing through two points using the parametric line equation.
  - Write a plane equation defined by the three points (1,2,0), (2,0,-1), and (3,-2,1). What is the normal **n** of this plane?
  - Let S=(1,0,3) and V=(2,1,-1). Find the point at which the line P(t) = S+tV intersects the plane L=(a,b,c,d) = (1,1,0,5).

- **Unity Game Engine questions**
  - Write at least two distinct features of Unity game engine compared to other Game Engines.
  - Define "Assets". Give examples of art assets.
  - Define "Prefabs".
  - What does HUD stand for?
  - What does NPC stand for?
  - Define NavMesh Baking.
  - What is the main difference between `Start()` and `Update()` functions of `MonoBehavior` class?
  - What are three types of lights in Unity? Which light is like the Sun in real world?
  - In the following code segment, what does `[SerializeField]` attribute do?

```
[SerializeField] private float speed;
```

  - What does the code segment do?

```
using UnityEngine;
using System.Collections;
public class Unknown : MonoBehaviour {
  public float speed = 3.0f;
  void Update() {
    transform.Rotate(0, speed, 0);
  }
}
```

- What does the code segment do?

  ```
  transform.Rotate(0, speed, 0);
  transform.Rotate(0, speed, 0, Space.World);
  ```

- What is raycasting? Describe when (for what purpose) you use it for your homework.

- What is the main difference between a coroutine and a function? When do you need to use coroutines?
- What is the return type of a coroutine?
- When you invoke a function, you invoke it directly using its function name in your program. How do you invoke a coroutine?
- Which keyword causes a coroutine to temporarily pause, handing back the program flow and picking up again from that point in the next frame.

- Write the execution order of the following functions. Give number 1 through 5. Also write when each function will be executed.

  ```
  Start()
  Awake()
  OnGUI()
  Update()
  OnTriggerEnter()
  ```

- Assume that you have the following code segment in Sample.cs. What does it do? What happen if we use `Destroy(this);` instead?

  ```
  Destroy(this.gameObject);
  ```

- The following code segment instantiate a fileballPrefab object. Why do we need "`as GameObject`"?

  ```
  private GameObject fireballPrefab;
  _fireball = Instantiate(fireballPrefab) as GameObject;
  ```

- Describe the main difference between textures and materials.
- What is the size constraint for texture images in Unity?
- For texture mapping, each vertex/point of your 3D model is associated with (u,v) texture coordinate that indicates a pixel location in the texture map.
  - (u, v) coordinate range such that the u coordinate ranges from 0 to 1 from right to left, and the v coordinate ranges from 0 to 1 from bottom to top. True or False?
  - It is also legal to use texture coordinates (u,v) that go outside this range; you can have negative values, for instance, or numbers higher than 1. True or False?
  - (u,v) values associated with each vertex can be scaled and translated. True or False?
- Assume that you have a plane with a material whose Tiling is set as (4,6) and Offset is set as (1,1). How does the texture image appear on the plane?
- Write two texture wrap modes provided by Unity.
- Write two texture filter modes provided by Unity.
- Describe one scenario where texture minification or magnification is required.

- What are mipmaps? When is it useful?
- What is the anisotropic filtering? When do you need it?
- What does the code segment do?

```
 value = Input.GetAxis ("Horizontal");
```

- What are 4 pieces of the Unity Navigation System.
- For NavMesh Obstacle, we can turn on and off “Carve” option. When should we turn it off?
- If both NavMesh Agent and Rigidbody (non-kinematic) are active at the same time, you have race condition. True of False?
- You don’t need to add physics colliders to NavMesh Agents for them to avoid each other. True or False?
- NavMesh Agent and Animator without Root Motion can cause race condition. True or False.
- NavMesh Agent and Animator with Root Motion can cause race condition. True or False.
- NavMesh Agent and NavMesh Obstable do not work well. True or False.
- What does the following code segment do?

```
NavMeshAgent agent = GetComponent<NavMeshAgent>();
agent.destination = goal.position;
```

- Every GameObject has a Transform. Is this statement True or False?
- The position, rotation and scale values of a Transform are measured relative to the Transform’s parent. Is this statement True or False?
- If the Transform has no parent, the properties are measured in world space.
- In Unity, X, Y, and Z axes in 3D space are color-coded. What is the designated color for each axis? Is this statement True or False?
- What are global and local coordinates? Define them using the term “Parenting”.
- What does the following code segment do?

```
     void Update() {
          Vector3 targetDir = target.position -
          transform.position;
          angleBetween = Vector3.Angle(transform.forward,
          targetDir);
     }
```

- List at least three Unity primitive colliders.
- What is a mesh collider? When is it useful to use? Why does Unity discourage to use it?
- What is a trigger object?
- What are Static Collider, Rigidbody Collider, and Kinematic Rigidbody Collider?
- Write main differences between trigger object collisions and non-trigger object collisions.
- Complete the following collision action matrix. If collision messages are sent upon collision between two objects with corresponding components, mark “C”. If trigger messages are sent upon collision between two objects with corresponding components, mark “T”. Otherwise, leave it as blank.

Collision Action Matrix						
	Static Collider	Rigidbody Collider	Kinematic Rigidbody	Static Trigger	Rigidbody Trigger	Kinematic Rigidbody

				Collider	Collider	Trigger Collider
Static Collider						
Rigibody Collider						
Kinematic Rigidbody Collider						
Static Trigger Collider						
Rigidbody Trigger Collider						
Kinematic Rigidbody Trigger Collider						

  - If you want your object to react to physical collision with other objects and the game world, you must add a RigidBody component. Is this statement True or False?
  - The player's CharacterController will generate a trigger event when colliding with a trigger object. Is this statement True or False?
  - A character controller cannot walk through static colliders in a scene. Is this statement True or False?
  - Character Controller component gives the character a simple, cylinder collider that is always upright. Is this statement True or False?
  - If the player has a CharacterController component, the player character is be fully influenced by the Game Physics. Is this statement True or False?
  - Describe how Unity allows us to set the collision system to ignore (filter out) some collisions.

- **Unity Game Engine questions (for Extra Credits since there are not from topics for the midterm scope)**
  - What is the difference between Update() and LateUpdate() functions?
  - What does the following code segment do?

```
…
void Start() {
  _rotY = transform.eulerAngles.y;
  _offset = target.position - transform.position;
}
void LateUpdate() {
  float horInput = Input.GetAxis("Horizontal");
  if (horInput != 0) {
    _rotY += horInput * rotSpeed;
  } else {
    _rotY += Input.GetAxis("Mouse X") * rotSpeed * 3;
  }
  Quaternion rotation = Quaternion.Euler(0, _rotY, 0);
  transform.position = target.position - (rotation * _offset);
  transform.LookAt(target);
}
```

- Consider the following code segment.

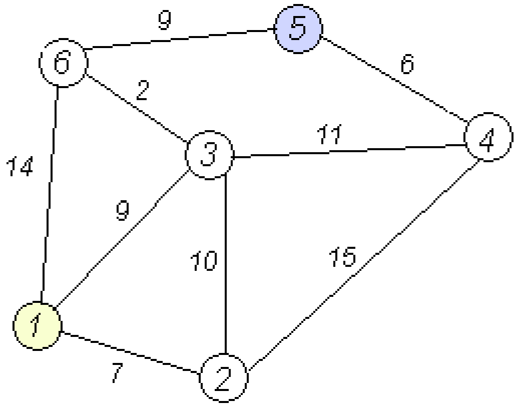
  ```
  void Update() {
    …
    Vector3 movement = Vector3.zero;
    float horInput = Input.GetAxis("Horizontal");
    if (horInput != 0) {
      movement.x = horInput;
      Quaternion tmp = target.rotation; // save the target’s angle
      target.eulerAngles = new Vector3(0, target.eulerAngles.y, 0);
      movement = target.TransformDirection(movement);
      target.rotation = tmp; // recover the target’s angle
      transform.rotation = Quaternion.LookRotation(movement);
     …
   }
  ```

  - What does the following line do?
    ```
    target.eulerAngles = new Vector3(0, target.eulerAngles.y, 0);
    ```
  - What does the following line do?
    ```
    transform.rotation = Quaternion.LookRotation(movement);
    ```
  - What does the program do?

- **Collision Detection**
  - Overlapping test and intersection test are two basic techniques for collision detection. Describe what they are. Why do we need intersection test?
  - There are two general issues in collision detection.
    - If collision detection process is performed for the original model geometry, it is highly inefficient and expensive. Write at least two ways that solves this issue.
    - If a collision detection process yields $O(n^2)$ time complexity, it is highly inefficient and expensive. Write at least two ways to improve time complexity.
    - What does Unity provide to solve one or both of these issues?
  - Compute the time of collision for two spheres, each of radius 0.25, that starts at t=0 at the position (0,0) and (1,1), respectively, and both end up at (1,0) at t=1.
  - Draw the Minkowski sum of a circle and a triangle.
  - Let’s assume that the player character is moved on a Terrain map (triangular mesh) whose y-axis is for height. Given the identified triangle with vertices at (1,0,0), (0,0,1) and (0,1,0) for the collision, a character’s feet are currently located at (0.2,0,0.2). Update y-value of the feet such that the character’s feet are on the plane to avoid collision.
  - Given a triangle with vertices at (0,0,0), (1,0,0), and (0,0,1), calculate the Barycentric coordinates of the points (0.5, 0, 0) and (1,0,1).
  - In games, collisions between characters are not usually very precise, as often times the arm or leg of one character will penetrate another character. If precise collision detection is employed, what problems or issues might arise? What advantage does modeling character-character collisions with spheres or cylinders offer?

- **AI and Pathfinding Algorithms**
  - Write three ways to represent a search space for pathfinding algorithm.

- Write at least three reasons that Navigation mesh representation is superior to Waypoint graph representation.
- For every algorithms covered in the class (Breadth-First, Best-First/Greedy, A*, and Dijkstra), answer the following questions:
  - Is the algorithm an exhaustive or a heuristic algorithm?
  - Does the algorithm always find the optimal path?
  - Is the algorithm a complete algorithm?
- Why is A* preferred over Breadth-First and Greedy algorithm?
- Consider the following map.



Heuristic function h(n):

h(1)=15
h(2)=15
h(3)=10
h(4)=5
h(5)= 0
h(6)= 7

Node 1 is the initial (departure) node and Node 5 is the goal (destination) nodes. The step cost between two nodes is given as an edge cost in the figure. The heuristics **h(n)** (cost from each node **n** to the goal) are given next to the graph. The task is to find a path to the goal node from the initial node. Write a path found by each algorithm (BFS, Greedy, A*, Dijkstra algorithm). Is the path optimal?