

## **Fundamentals of Game Development Course - Final Exam**

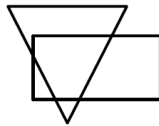
***(Lab + Main Course Mixed Exam)***

### **A. PRESENTATION (5%)**

- a. Summarize your presentation in one or two paragraphs.
- b. Summarize one of your classmates' presentations in one or two paragraphs.  
(Your own words.)

## B. GENERAL THEORY (12%)

- Give an example of when you would incorporate LiveOps in your game.
- What are the points of having a game design document for a new project?
- How would you turn a perspective camera into orthographic?
- What happens in the world update phase of a game engine?
- State the pros and cons for Navigation Mesh and Navigation Graph?
- How would the Sutherland-Hodgman algorithm clip the triangle for the rectangle screen? Draw each step.

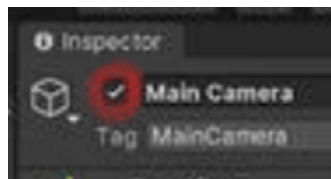


- What's back-face culling? What are some cons of it?
- Run the A\* algorithm on the following figure. Show each step as well as the final route.
  - Cost of horizontal and vertical move = 2.3
  - Cost of diagonal move = 3.5
  - Heuristic: Manhattan Distance
  - S: Start, D: Destination



### C. UNITY (25%)

- a. In what order Unity's built-in methods are called? Name at least 5 of them in order and explain each of them.
- b. Why would we want to use MonoBehaviour instead of a normal C# class in Unity?
- c. Why would we want to use Coroutines in Unity? What's an alternative for coroutines?
- d. Why would we want to use prefabs? What about prefab variants?
- e. What are some useful built-in components of Unity? Name and explain 4 of them.
- f. A MonoBehaviour child class only has the Awake method implemented and nothing more. A game object uses this component. Why can't we disable/enable this component from its respective game object through the inspector?



- g. What's one component that can never be removed from a Unity game object?
- h. We have a button in our scene, we have assigned a function for its "OnClick" property through inspector, but it's NOT detecting any clicks. What would be some of the possible reasons for this problem?
- i. How are normal game objects (like a normal 3D cube) rendered differently from the UI elements (like a text object) in Unity?
- j. What's the difference between world position and local position in Unity?

#### D. PROBLEM SOLVING LEVEL 1 (8%)

Considering Unity and/or software engineering and/or game AI concepts, present possible solutions for the following problems in a **few words**:

- a. We want to keep a game object consistent (exist) between different Unity scenes.
- b. We want to create a system that can generate unlimited number of levels by entering only a few settings (like heart count, number of NPCs spawned randomly)
- c. We want to save user's stats (such as highscores) [not a database.]
- d. We want to only have one instance of a class's object in the whole project during runtime.
- e. We want to implement gun aiming and shooting a target for a game.
- f. We want to work as a team simultaneously on a single project and be updated on the latest changes.
- g. We want to implement a group of fish NPCs for an underwater scene.

## E. PROBLEM SOLVING LEVEL 2 (25%)

Present solutions in the asked format for each problem:

- a. **(description and code)** We need to implement a simple NPC boat in our game that only goes from point A to point B and comes back. (present **2 different** ways; one must include purely code, one must use Unity's features)
- b. **(state machine)** We need to implement a NPC for a shooting game that:
  - 1) Senses when a player is in its vision
  - 2) Approaches the player
  - 3) Shoots the player
  - 4) Takes defense if the player starts shooting back
  - 5) Retreats if its health is below 20%
- c. **(code)** We need to implement a jetpack for a 2D platformer game which has a limited capacity; it ends after more than 10 seconds of accumulative use. It can be kept in inventory as long as it's not fully used. Everytime you press and keep pressing a certain button, it keeps adding to the character's y coordinate and when you stop pressing the button, it decreases the value until the character collides with a surface. The value change is affected by the character's weight and a set amount of gravity.

## F. PROBLEM SOLVING LEVEL 3 (25%)

In this part, you must design the architecture for a game project and show the **components** used for each object (like Rigidbody or a custom script component) as well as the non-MonoBehavior classes and the relationship between the entities.

If you want to add a custom MonoBehaviour or a normal C# class, you don't need to write the code for it, just write the headers for the important methods. No need to include unnecessary details. The goal of this question is to see how you would **design** a game's code base as a software engineer.

### Your Game:

Calculate **the sum of the sum of** the last two digits of your student number (e.g. 9852xx88 => 88 => 16 => 7) Based on this value check the following table and answer the question for that game.

Value	Game
1 / 3 / 8	1
2 / 5 / 7	2
4 / 6 / 9	3

## 1: Minecraft (Adventure-Survival)

- Walk, run, jump, swim
- Collect things by hitting them using ax/pickaxe
  - Wood Piece (ax can collect this)
  - Stone (pickaxe can collect this)
- Using a sword:
  - Attack a zombie NPC
- Zombie NPC:
  - Walks
  - Senses the player in the field of vision with adjustable radius, and attacks.
  - Has limited hearts and can die.
- Placing objects:
  - You can place the objects that you collect on the map.
- Crafting:
  - At any time, if you press E, you can use a **popup menu** to create stuff using the stuff that you collect. In order to make new things, you need to spend the objects that you collect.
  - 2 wood + 3 stone = pickaxe
  - 2 wood + 2 stone = ax
  - 1 wood + 4 stone = sword
- General features:
  - At any moment, you can have multiple objects on your inventory. But you can only have one object in your hand at any moment. In the same menu that appears by pressing E and is used for crafting, you can choose what item to hold in your hand.

## 2: Counter strike (Shooter Game)

- Walk, run
- Aim and shoot the gun
- Take damage and die
- Buy weapons:
  - It has a menu
  - You can buy 2 weapons each round (with different price, name, damage, speed, weight)
  - Buy a vest (if you have a vest, you'll have twice the heart size)
- Enemy and friendly NPCs (no need to make their AI advanced, keep them as dumb as possible)
- Changeable amount of team and opponents count (but they must be equal)
- The weight of the weapons affect your running speed.
- You can change weapons by pressing a button.
- Each successful gunshot decreases the opponent's health as much as the damage of the gun is set to.
- You gain money by eliminating opponents.
- The team whose members all die first, lose the round
- The winner team gains more money than the loser team at the end of the round (adjustable amounts for winner and loser)



### 3: Age of Empires (Real-Time Strategy Game)

- Different types of human units: soldier, villager (worker)
  - Each of them has adjustable heart count
- Worker:
  - Can build:
    - Can build a farm and military building.
    - You should choose a place for your target building on the map.
  - Can collect resources:
    - Can collect food from animals (1 food per second)
    - Can collect wood from trees (1 wood per second)
    - Can collect gold from gold mine (1 gold per second)
    - If you assign a worker to collect resources, he/she cannot do anything while you haven't canceled the collecting task for them.
- Soldier:
  - Can attack
    - Does not do this automatically (the player has to order the soldier to attack)
  - Can die
- Buildings:
  - Farm:
    - Generates 1 food every 2 seconds automatically.
    - It takes 25 seconds to get built. And it costs 50 wood.
  - Military Building:
    - Can make a soldier unit (costs 50 wood + 50 food + 50 gold)
    - It takes 15 seconds to create each unit (soldier creation task).
    - It takes 50 seconds to get built and it costs 80 wood + 30 gold.
  - Town Center:
    - Placed automatically when the map is loaded for the player. You can build workers from there.
- General features:
  - You can choose a single or multiple human units at the same time and order them to move somewhere on the map (so you also have a path-finding component)
  - Assume there are trees, animals, and gold mines spawned on the map.
  - Workers can have multiple tasks in a sequence. You must be able to see the task queue for each worker and cancel the tasks that you want.