

# Bài Tập Về Nhà Lập Trình Game

Họ và tên : Hà Khánh-BCS230045

Github : [Ngunhcho7654/unity-math](https://github.com/Ngunhcho7654/unity-math)

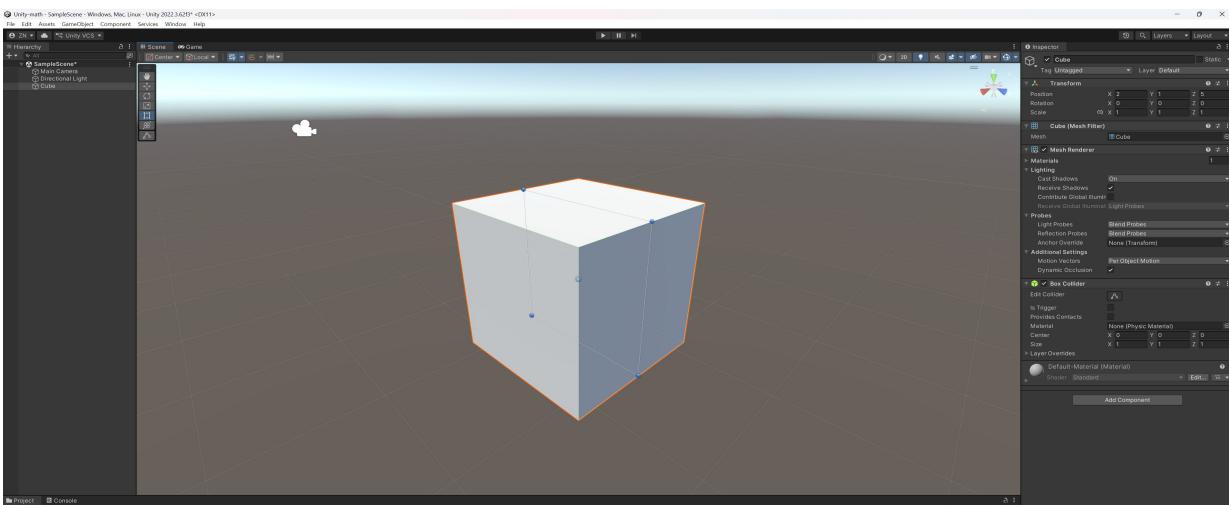
## PHẦN A – COORDINATE SYSTEM & WORLD SPACE

A1. Tạo một Cube tại vị trí: X = 2, Y = 1, Z = 5

A2. Bật Gizmos trong Scene View và chụp ảnh thể hiện: - Trục X (Đỏ) - Trục Y (Xanh lá) - Trục Z (Xanh Dương)

A3. Trả lời câu hỏi

1. Trục nào hướng lên trên trong Unity? Trục Y hướng lên trên
2. Trục nào hướng về phía Camera? Trục Z dương (+Z) hướng về phía trước (ra khỏi màn hình)



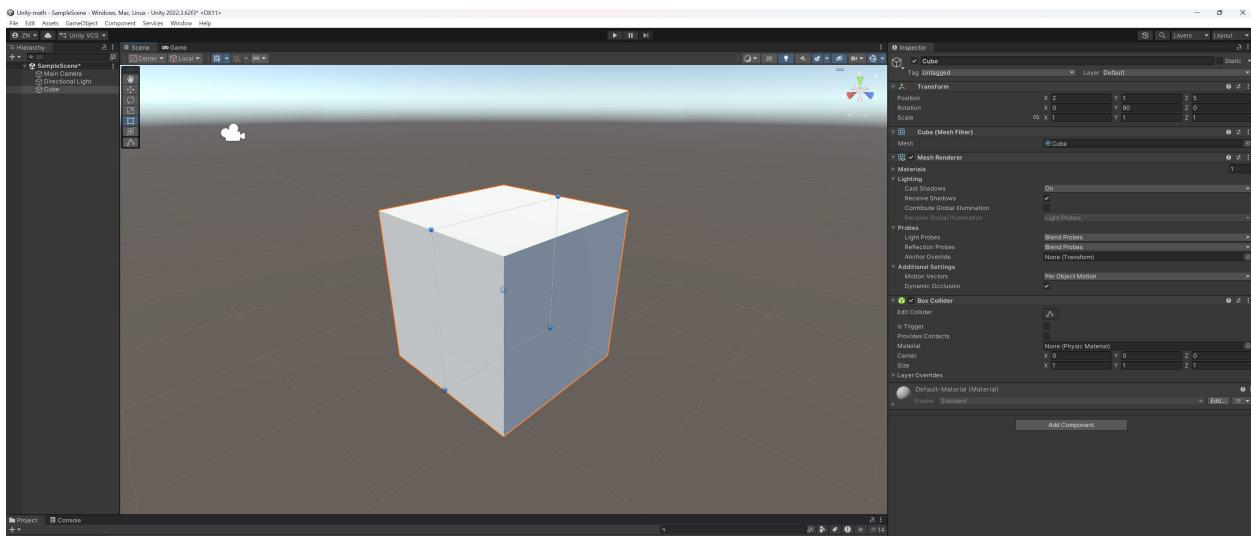
Hình 1: Scene View A

## PHẦN B – LEFT-HANDED COORDINATE SYSTEM

B1. Xoay Cube: Y = 90

B2. Trả lời câu hỏi

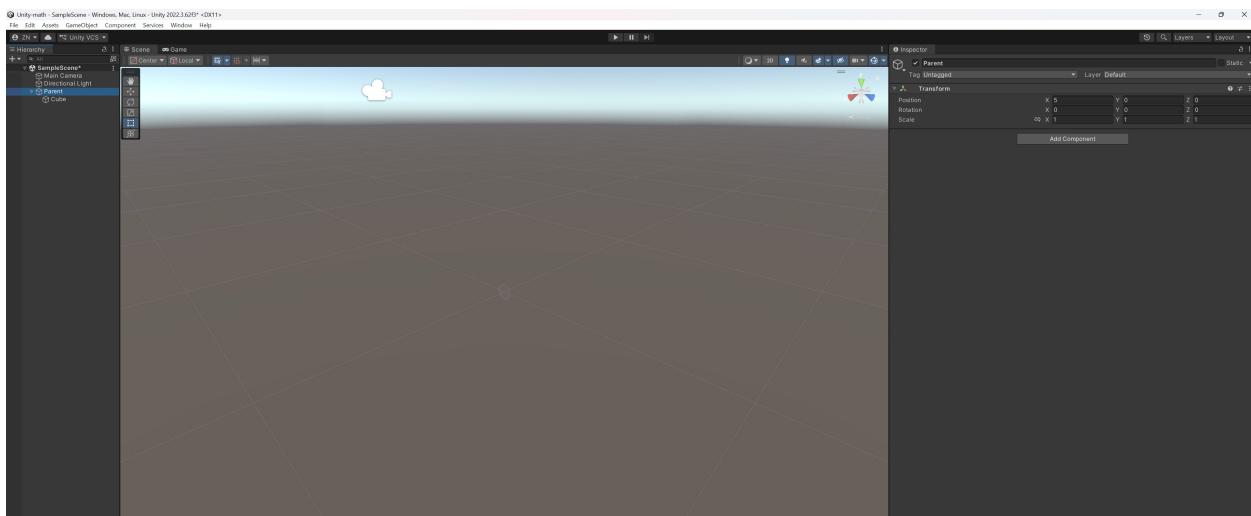
1. Cube quay theo chiều nào? Cube quay sang phải (theo chiều kim đồng hồ) khi nhìn từ trên xuống
  
2. Điều này thể hiện Left-Handed như thế nào?
  - Trục Z dương hướng về phía trước
  - Khi xoay theo trục Y dương, object quay theo chiều kim đồng hồ



Hình 2: Scene View B

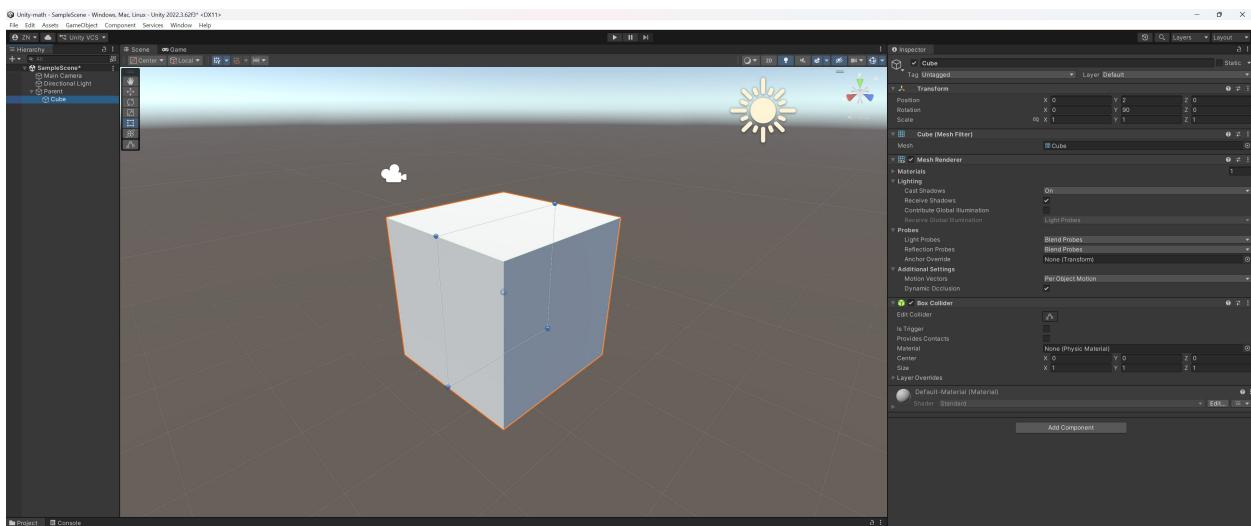
## PHẦN C – LOCAL SPACE & WORLD SPACE

C1. Tạo Empty Object “Parent” vị trí (5;0;0)



Hình 3: Empty Object Parent Position (5;0;0)

C2. Đặt Cube làm con: Local Position của Cube = (0, 2, 0)



Hình 4: Local Position của Cube Position(0;2;0)

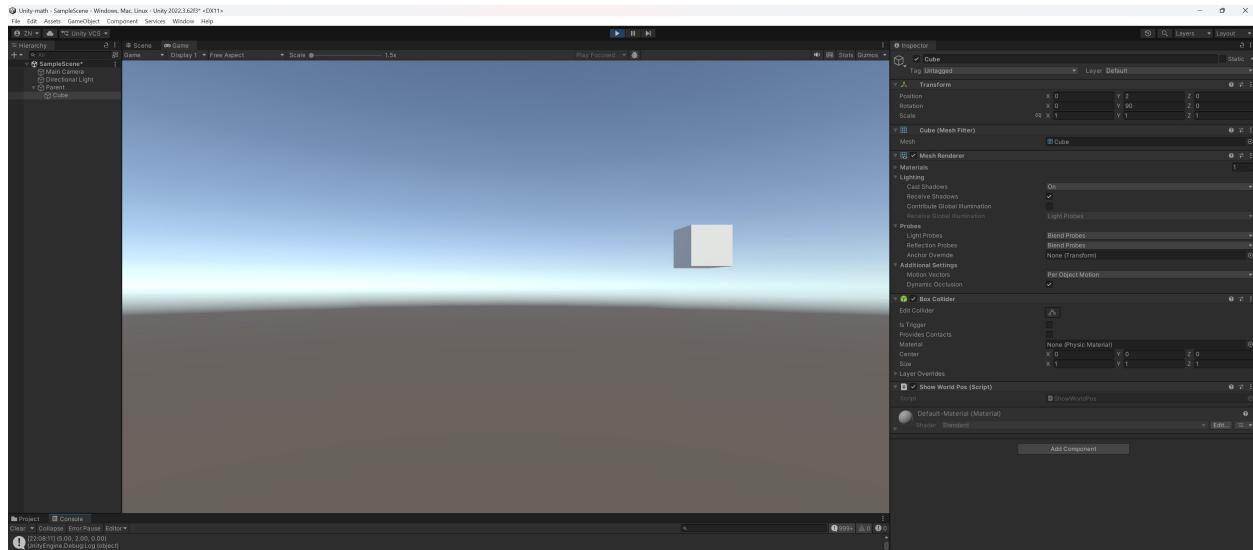
C3. Ghi lại vị trí:

1. Local Position của Cube:

(0, 2, 0)

2. World Position của Cube:

(5, 2, 0)



Hình 5: Toạ độ local position và world position của Cube

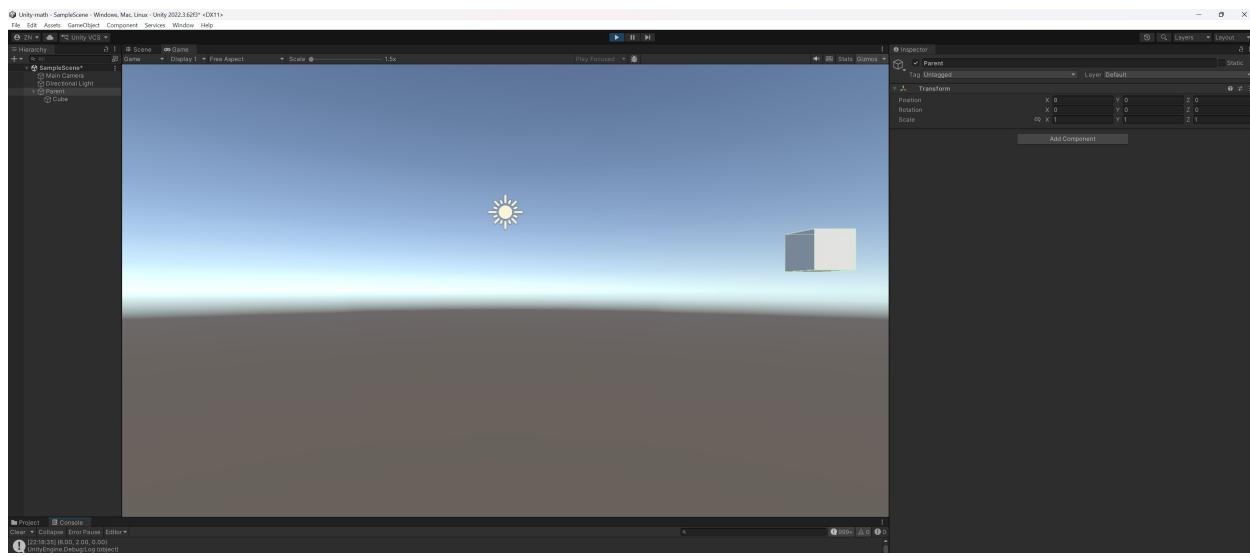
C4. Di chuyển parent từ vị trí (5;0;0) sang vị trí (8,0,0)

1. Local Position của Cube có đổi không?

Không đổi, local position của cube vẫn là (0, 2, 0)

2. World Position của Cube thay đổi thế nào?

World Position thay đổi thành (8, 2, 0), khi Parent được di chuyển, Cube di chuyển theo trong World Space, do đó World Position của Cube tăng thêm 3 đơn vị trên trục X.

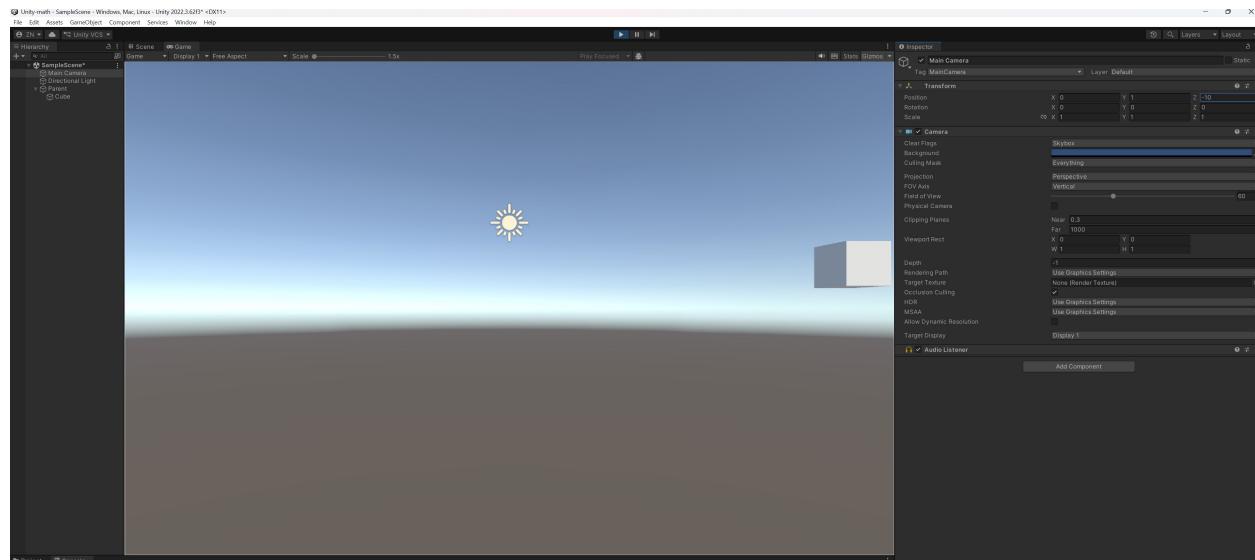


Hình 6: World Positon của Cube khi Parent thay đổi

## PHẦN D – GRAPHICS PIPELINE

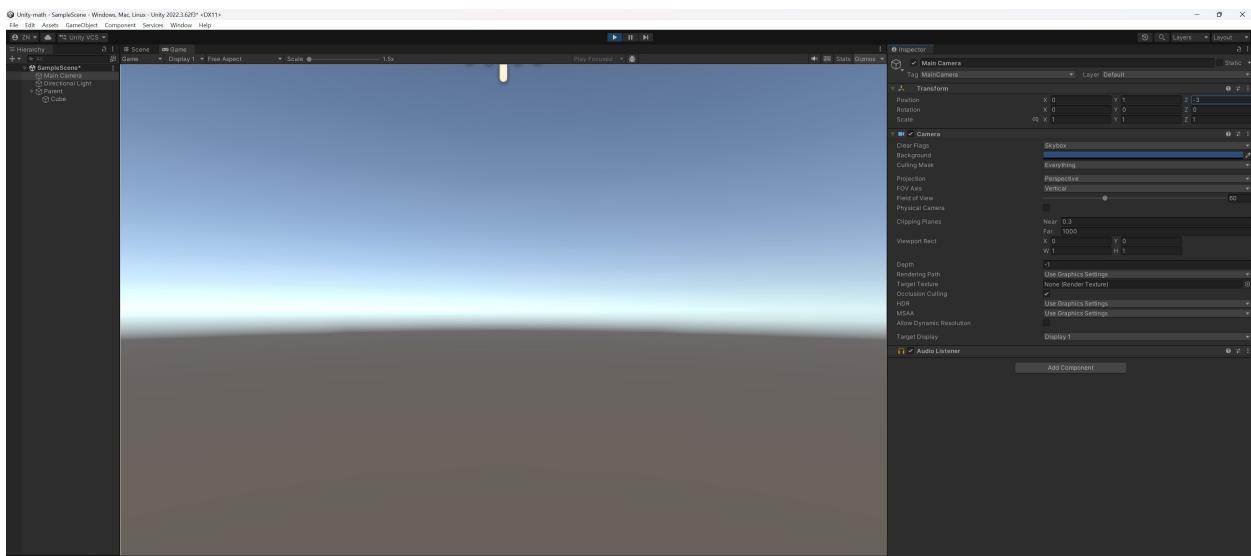
### D1. Di chuyển Camera

1. Position main camera  $z=-10$



Hình 7: Toạ độ main camera z=-10

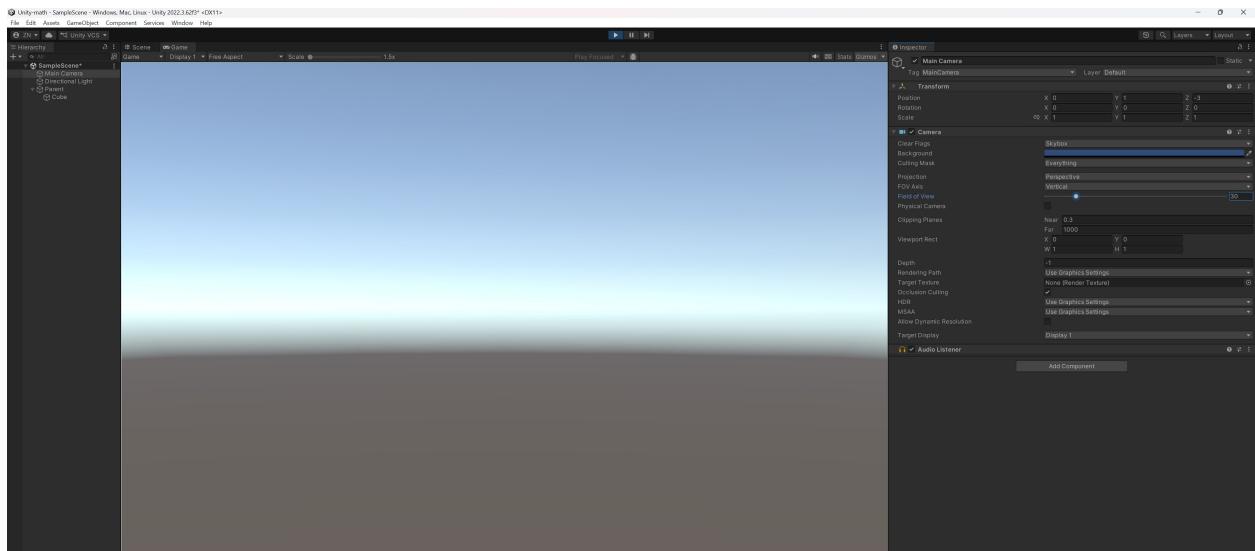
## 2. Position main camera z=-3



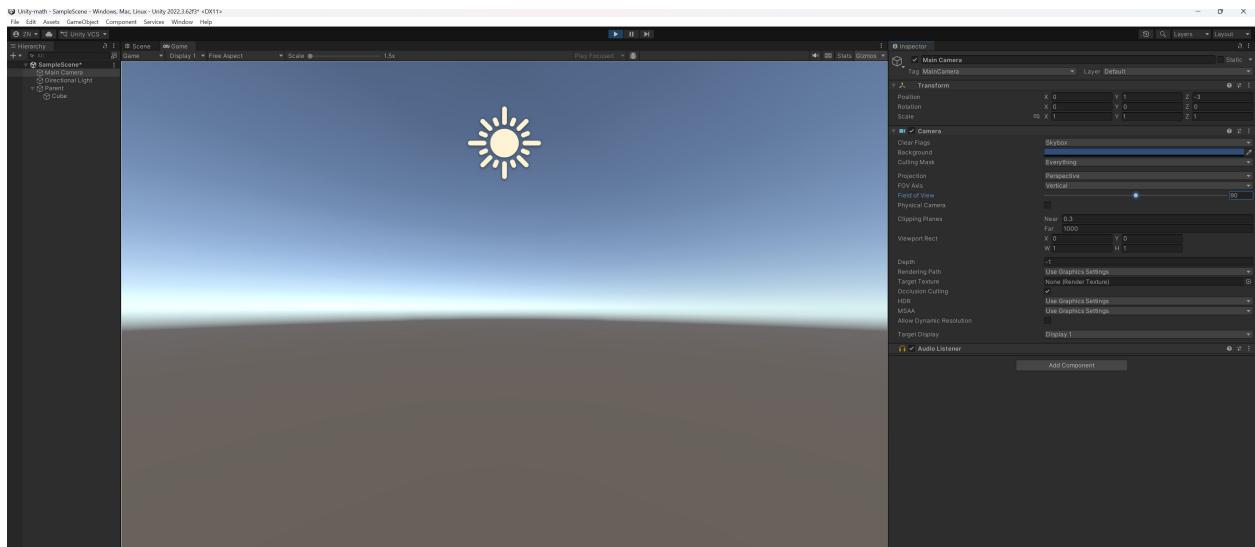
Hình 8: Toạ độ main camera z=-3

## D2. Thay đổi Camera Settings

### 1. Thay đổi Field of View (FOV):



Hình 9: Giảm FOV (từ 60 xuống 30)

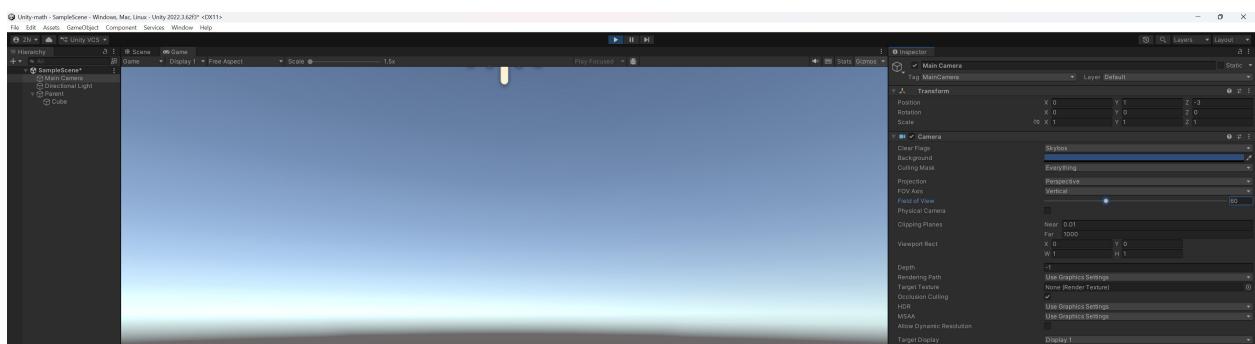


Hình 10: Tăng FOV (từ 60 lên 90)

Quan sát Game View:

- FOV lớn thì sẽ góc rộng và object nhỏ
- FOV nhỏ thì sẽ có góc hẹp và object to

## 2. Near Clip Plane:



Hình 11 : Near Clip Plan 0.01

## 3. Trả lời câu hỏi :

Vì sao object trông to/nhỏ dù không đổi vị trí?

- Object trông to hoặc nhỏ hơn do Camera sử dụng phép chiếu phối cảnh (Perspective Projection), trong đó kích thước hiển thị phụ thuộc vào khoảng cách giữa Camera và object cũng như Field of View, mặc dù vị trí World Space của object không thay đổi.

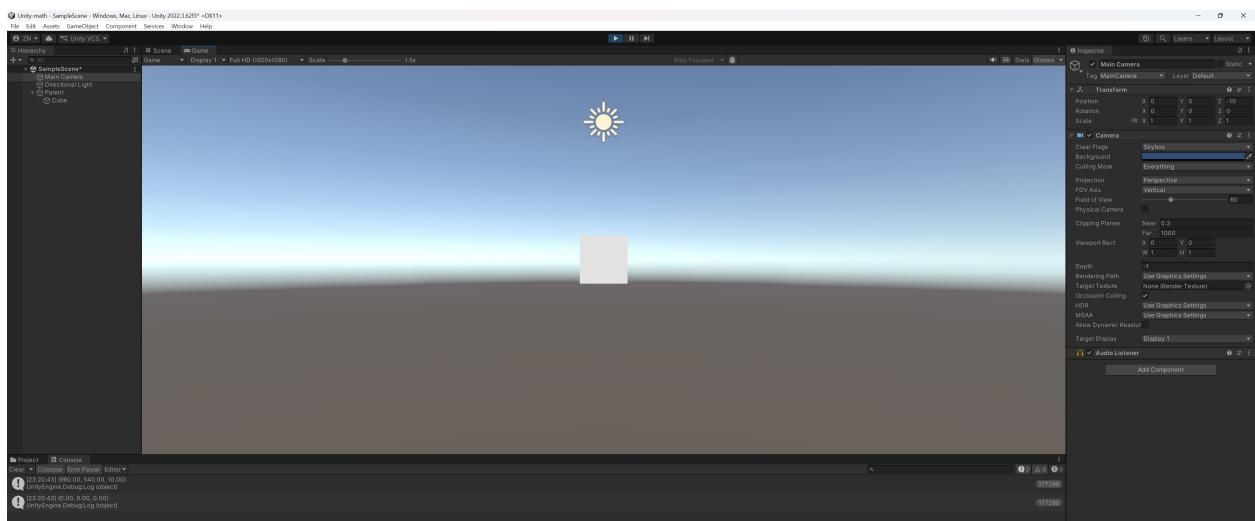
Vì sao object có thể biến mất khỏi màn hình?

- Object có thể biến mất khỏi màn hình khi nằm ngoài vùng render của Camera, cụ thể là khi nằm trước Near Clip Plane hoặc sau Far Clip Plane. Trong trường hợp Camera tiến quá gần, object bị cắt bởi Near Clip Plane nên không được hiển thị.

## PHẦN E – SCREEN SPACE

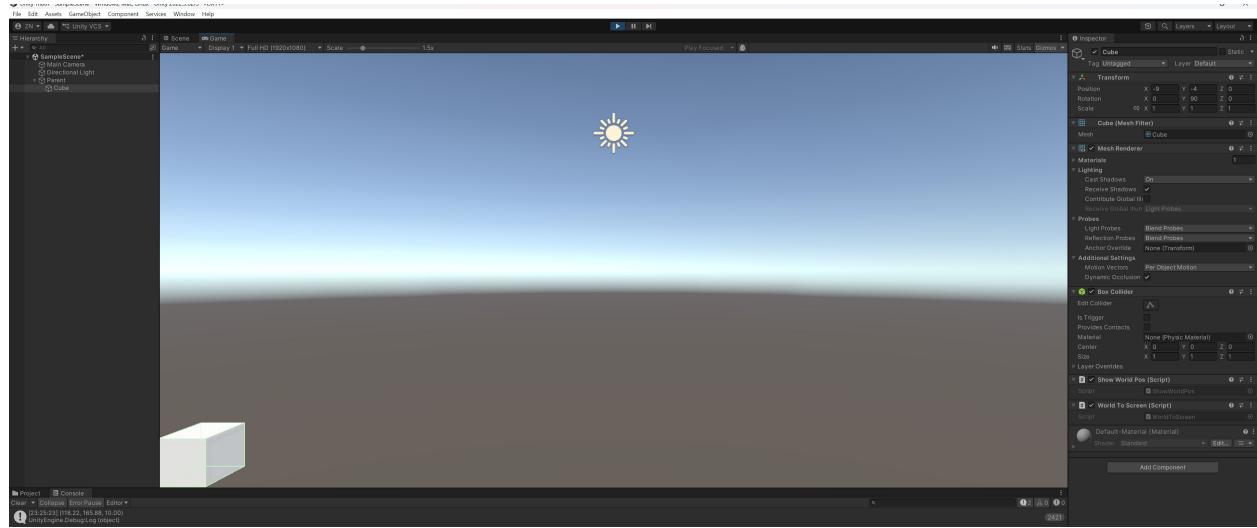
### E1. Ghi lại

#### 1. Screen Position khi Cube ở giữa màn hình (960;540;10)



Hình 12 : Screen Position Cube ở giữa màn hình

## 2. Screen Position khi Cube ở góc dưới bên trái (118.22;165.88;10.00)



Hình 13: Screen Position Cube ở góc dưới bên trái

### E2. Trả lời câu hỏi

#### 1. Góc tọa độ Screen Space ở đâu?

Góc tọa độ của Screen Space trong Unity nằm tại góc dưới bên trái màn hình, với trục X tăng từ trái sang phải và trục Y tăng từ dưới lên trên (Góc dưới bên trái màn hình (0,0)).

#### 2. Screen Space khác World Space thế nào ?

Screen Space là hệ tọa độ hai chiều của màn hình hiển thị, trong đó góc tọa độ nằm ở góc dưới bên trái màn hình.

World Space là hệ tọa độ ba chiều của toàn bộ Scene, được sử dụng để xác định vị trí các object trong môi trường 3D.

Screen Space sử dụng đơn vị pixel và phụ thuộc vào độ phân giải màn hình, trong khi World Space sử dụng đơn vị 3D và không phụ thuộc vào kích thước màn hình. Camera chịu trách nhiệm chuyển đổi tọa độ từ World Space sang Screen Space trong Graphics Pipeline.

## **PHẦN F – NHẬN XÉT CÁ NHÂN**

Thông qua bài tập này, em đã hiểu rõ hơn về cách Unity sử dụng hệ tọa độ trong không gian ba chiều và cách các đối tượng trong Scene được hiển thị lên màn hình. Bài tập cũng giúp em hiểu rõ hơn về hệ tọa độ Left-Handed mà Unity sử dụng, đặc biệt thông qua việc quan sát hướng xoay của object khi thay đổi Rotation. Ngoài ra, phần thực hành với Camera và Graphics Pipeline giúp em nhận thức được vai trò của Field of View và Near Clip Plane trong việc quyết định kích thước hiển thị cũng như việc object có được render hay không. Việc object có thể biến mất khi Camera tiến quá gần là một ví dụ trực quan giúp em hiểu rõ hơn về khái niệm clipping trong đồ họa máy tính. Phần chuyển đổi từ World Space sang Screen Space thông qua script giúp em kết nối được lý thuyết với thực hành, đồng thời hiểu rằng Camera đóng vai trò trung gian trong quá trình hiển thị hình ảnh từ không gian 3D sang màn hình 2D. Nhìn chung, bài tập không chỉ giúp em làm quen với các thao tác cơ bản trong Unity mà còn tạo nền tảng quan trọng để học các nội dung nâng cao hơn về đồ họa 3D và phát triển game sau này.