

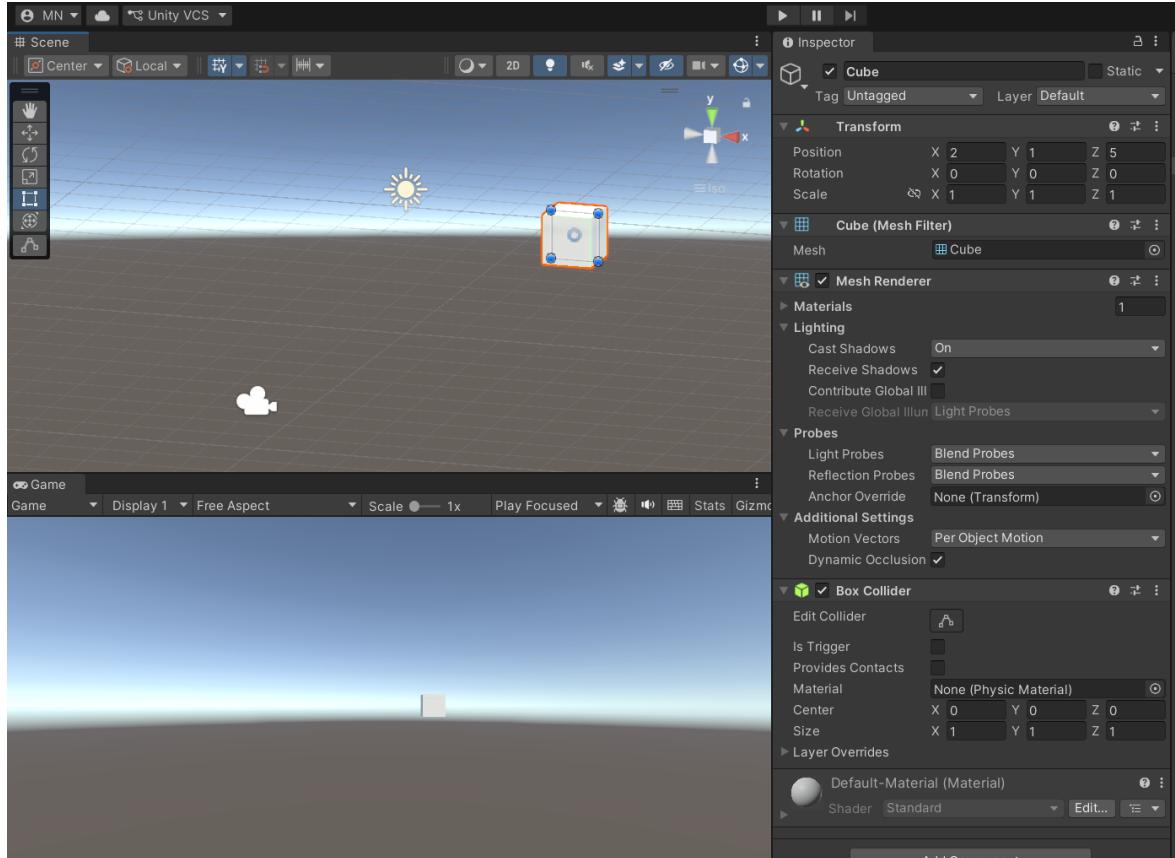
# Báo cáo bài tập:

## Mathematics in Unity

### PHẦN A – COORDINATE SYSTEM & WORLD SPACE

#### A1. Tạo một Cube tại vị trí:

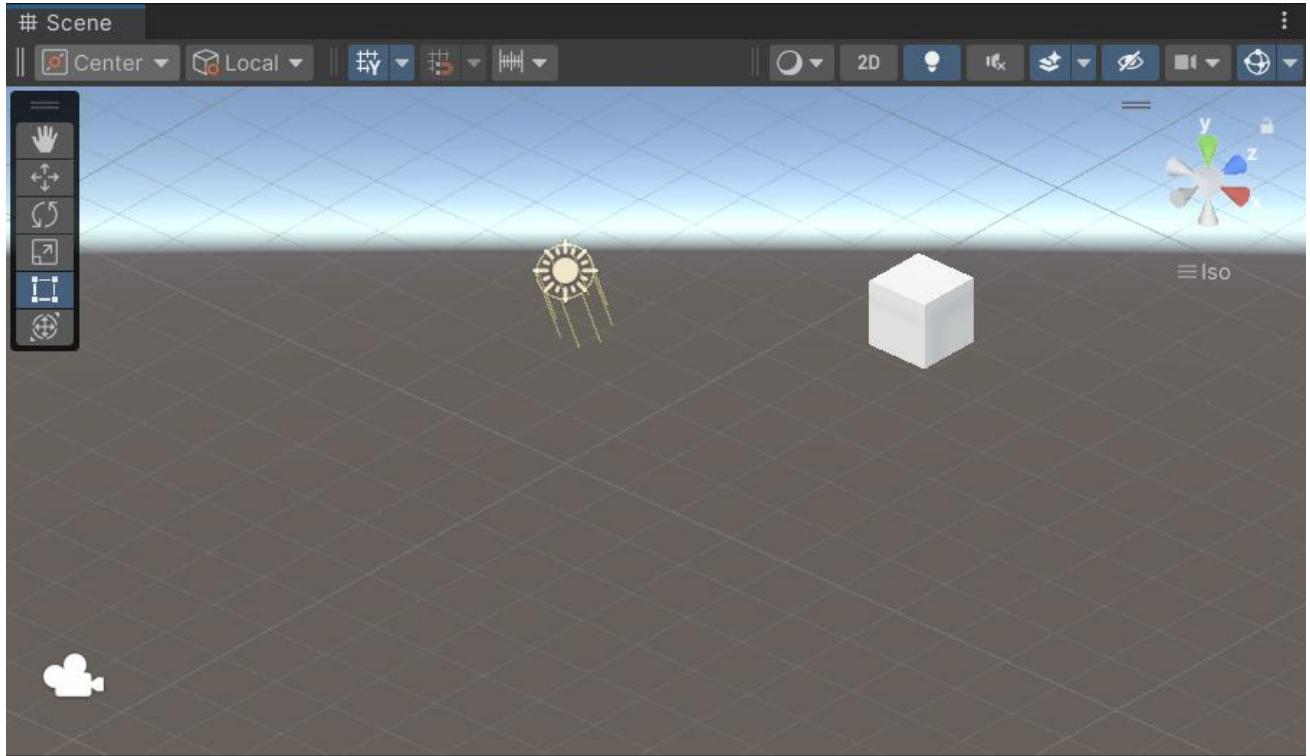
X = 2, Y = 1, Z = 5



Hình a1: Ảnh sau cube được tạo ở vị trí (2, 1, 5)

## A2. Bật Gizmos trong Scene View và chụp ảnh thể hiện:

- Trục X
- Trục Y
- Trục Z



Hình a2: Ảnh ở màn hình Scence View cùng với Gizmos ở góc trên bên phải

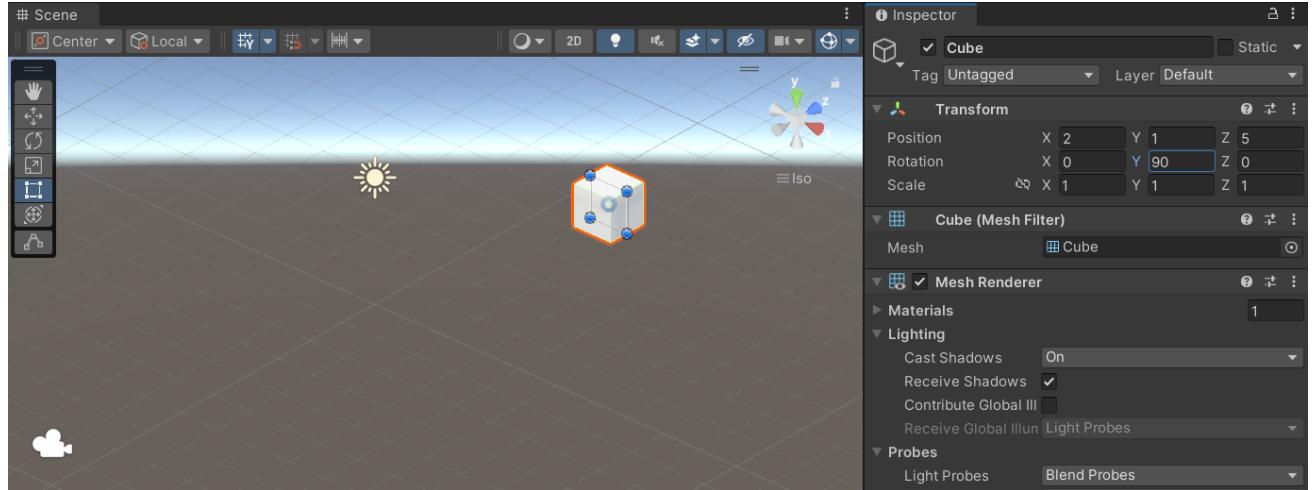
## A3. Trả lời các câu hỏi:

- Trục nào hướng lên trên trong Unity?
  - Trục Y (màu xanh lá)
- Trục nào hướng về phía Camera?
  - Trục Z (màu xanh dương)

## PHẦN B – LEFT-HANDED COORDINATE SYSTEM

### B1. Xoay Cube với Rotation:

$Y = 90$



Hình b1: Ảnh cube sau khi xoay trục Y 90 độ

### B2. Quan sát hướng quay của Cube và trả lời:

- Cube quay theo chiều nào?

→ Cube quay sang phải (theo chiều kim đồng hồ khi nhìn từ trên xuống)

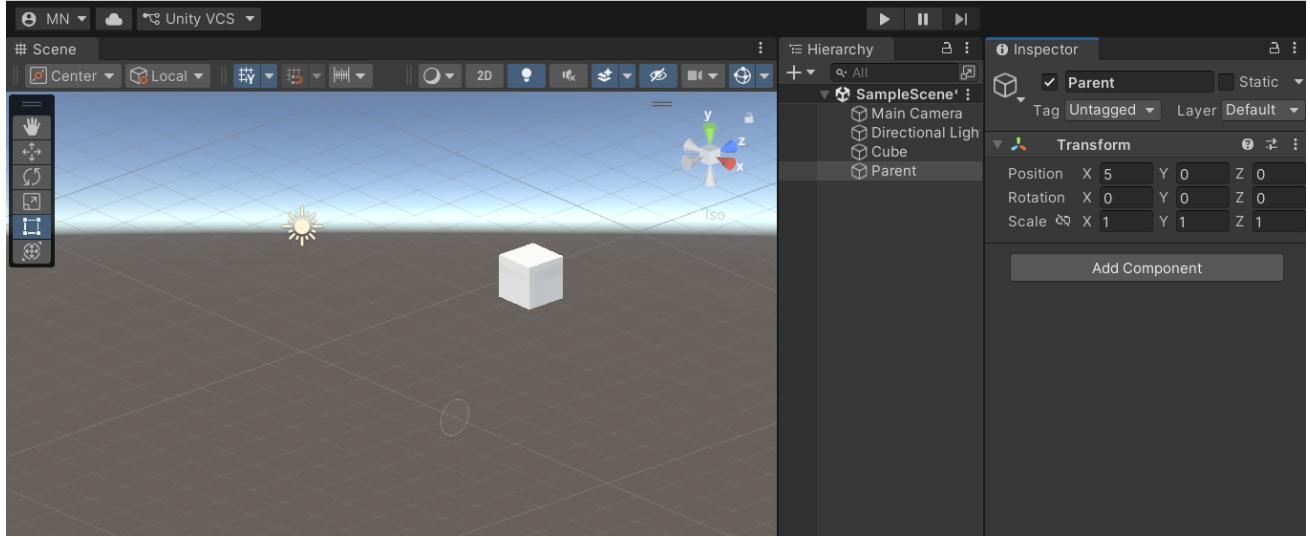
- Điều này thể hiện Unity sử dụng Left-Handed Coordinate System như thế nào?

→ Khi trục Y quay theo chiều dương thì đối tượng quay theo chiều kim đồng hồ

## PHẦN C – LOCAL SPACE VÀ WORLD SPACE

C1. Tạo một Empty GameObject tên là “Parent” tại vị trí:

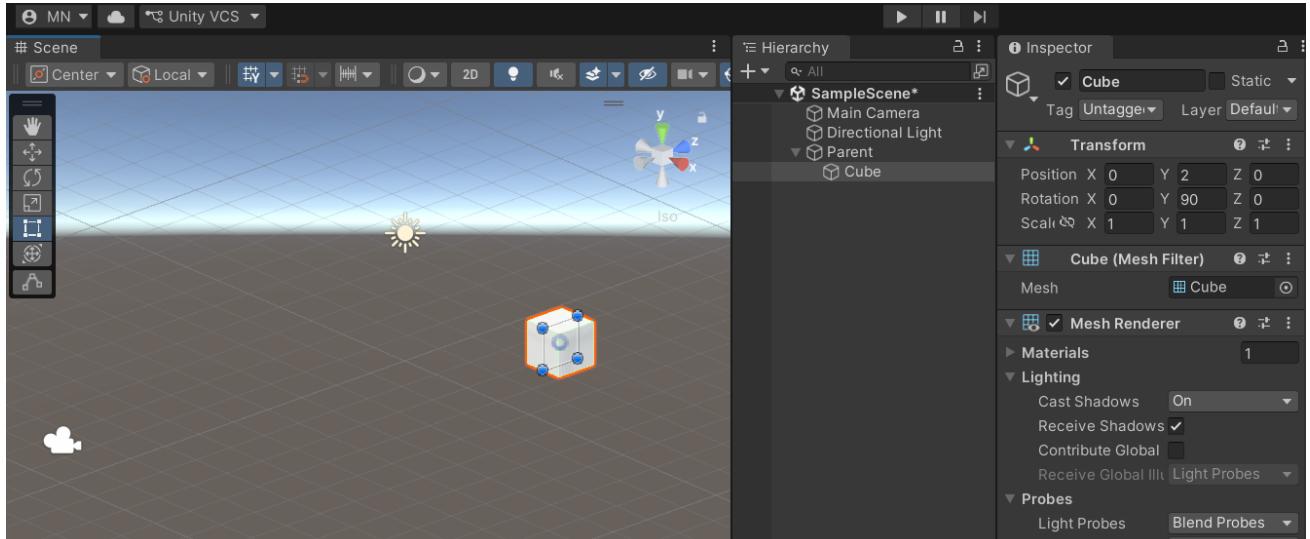
(5, 0, 0)



Hình c1: Ảnh sau khi tạo Parent (được thể hiện bằng đường mờ) ở vị trí (5, 0, 0)

C2. Đặt Cube làm con của Parent và thiết lập:

Local Position của Cube = (0, 2, 0)



Hình c2: Ảnh cube sau khi được đặt vào Parent và thiết lập Local Position (0, 2, 0)

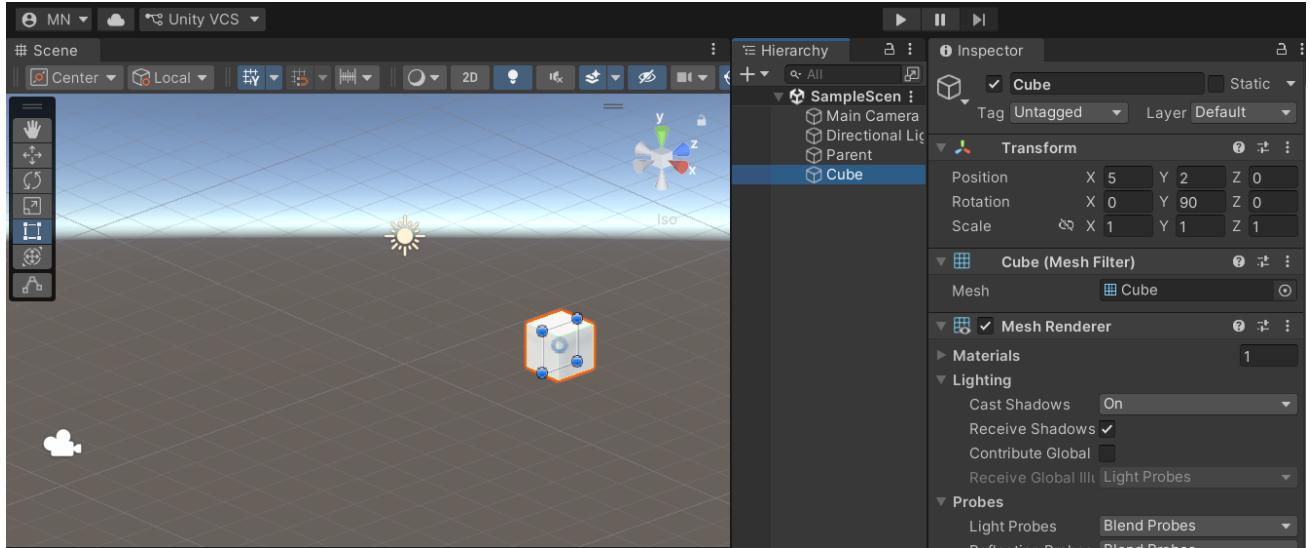
### C3. Ghi lại:

- Local Position của Cube:

$$\rightarrow (0, 2, 0)$$

- World Position của Cube

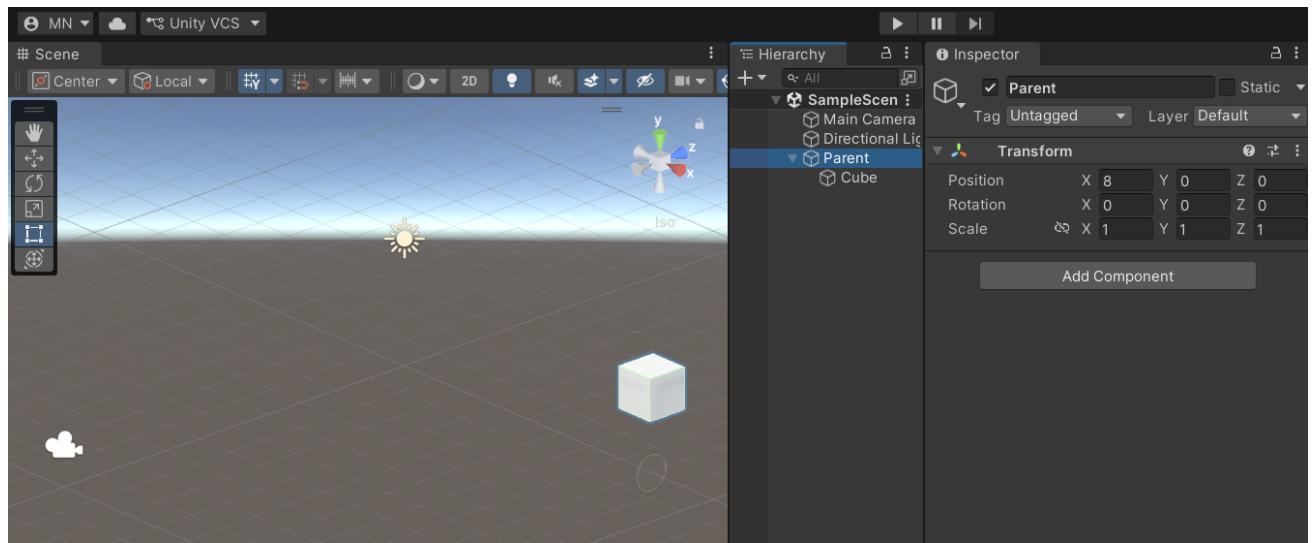
$$\rightarrow (5, 2, 0)$$



Hình c3: Ảnh cube với World Position của Cube (5, 2, 0)

#### C4. Di chuyển Parent sang vị trí:

(8, 0, 0)



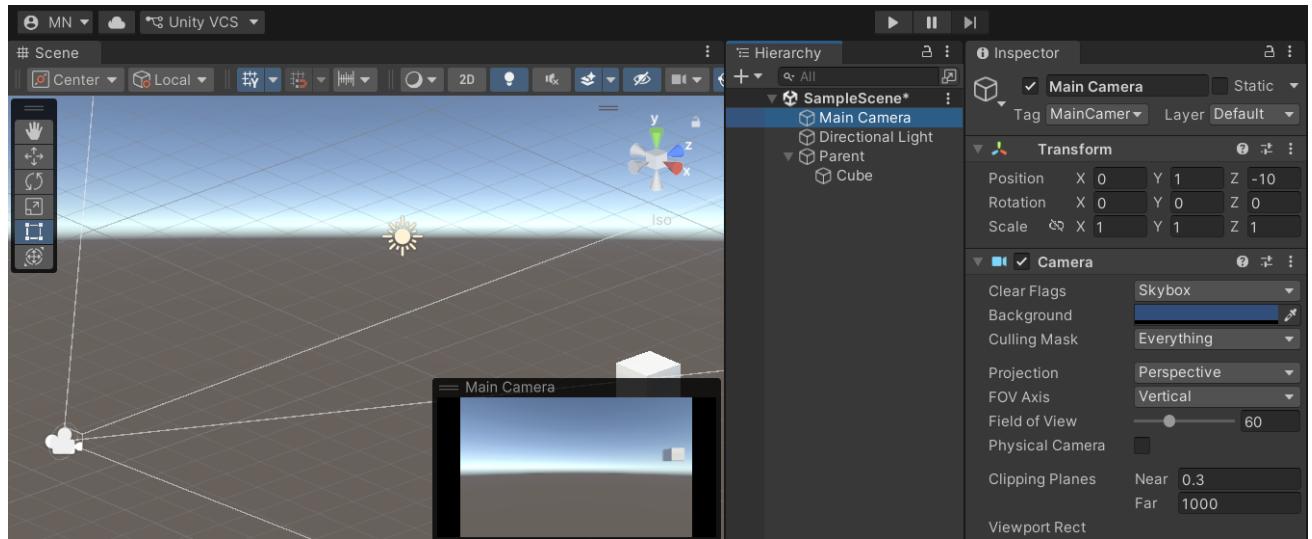
Hình c4: Ảnh sau khi di chuyển Parent sang vị trí (8, 0, 0)

Trả lời:

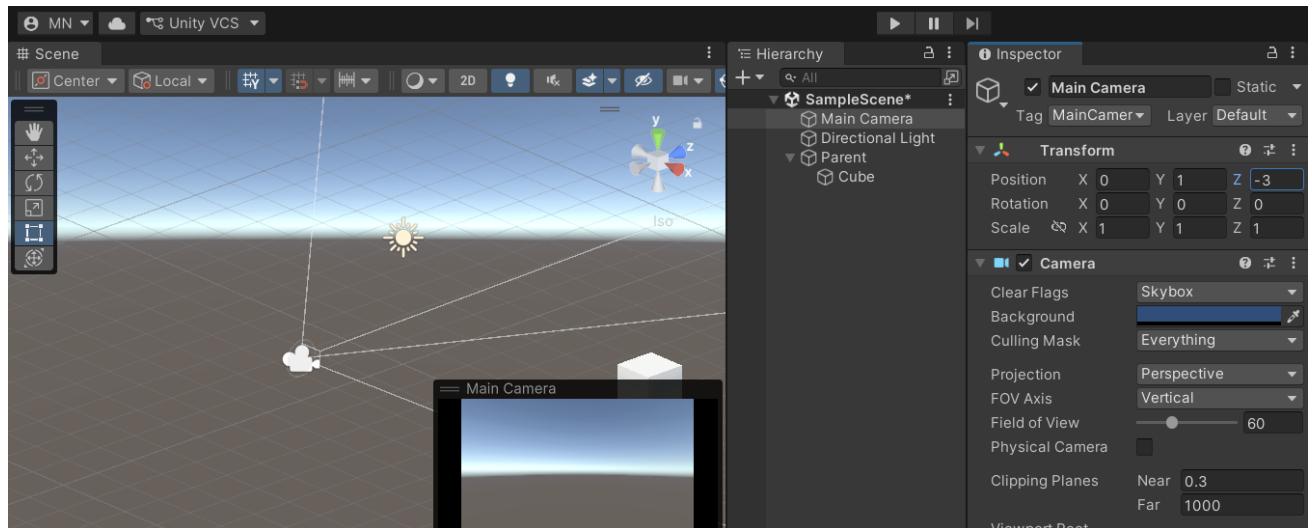
- Local Position của Cube có thay đổi không?
  - Không đổi. Local Position của Cube vẫn ở (0, 2, 0)
- World Position của Cube thay đổi như thế nào?
  - Chuyển thành (8, 2, 0)

## PHẦN D – GRAPHICS PIPELINE

### D1. Di chuyển Camera dọc trục Z từ -10 đến -3



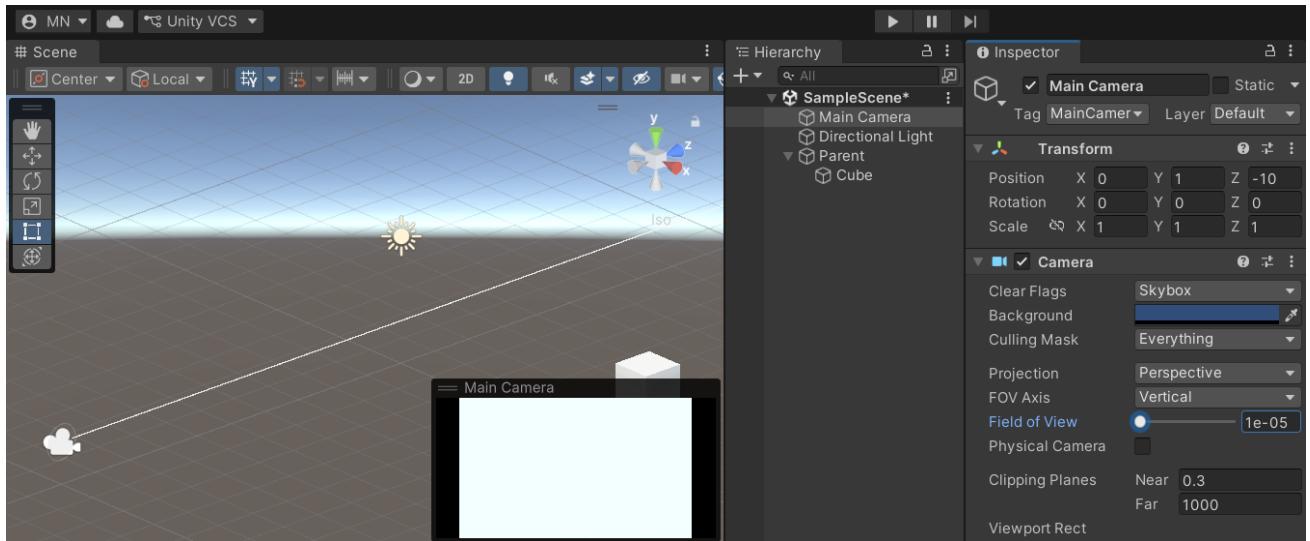
Hình d1.1: Ảnh trước khi di chuyển camera ở vị trí (0, 1, -10)



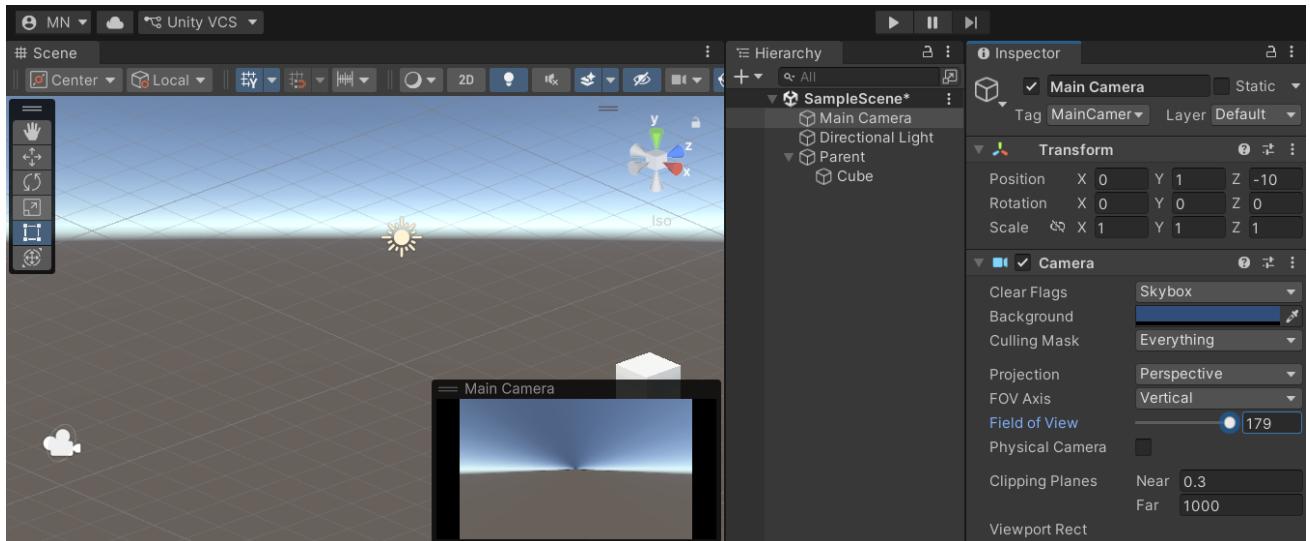
Hình d1.2: Ảnh sau khi di chuyển camera sang vị trí (0, 1, -3)

## D2. Thay đổi các thông số của Camera:

### - Field of View

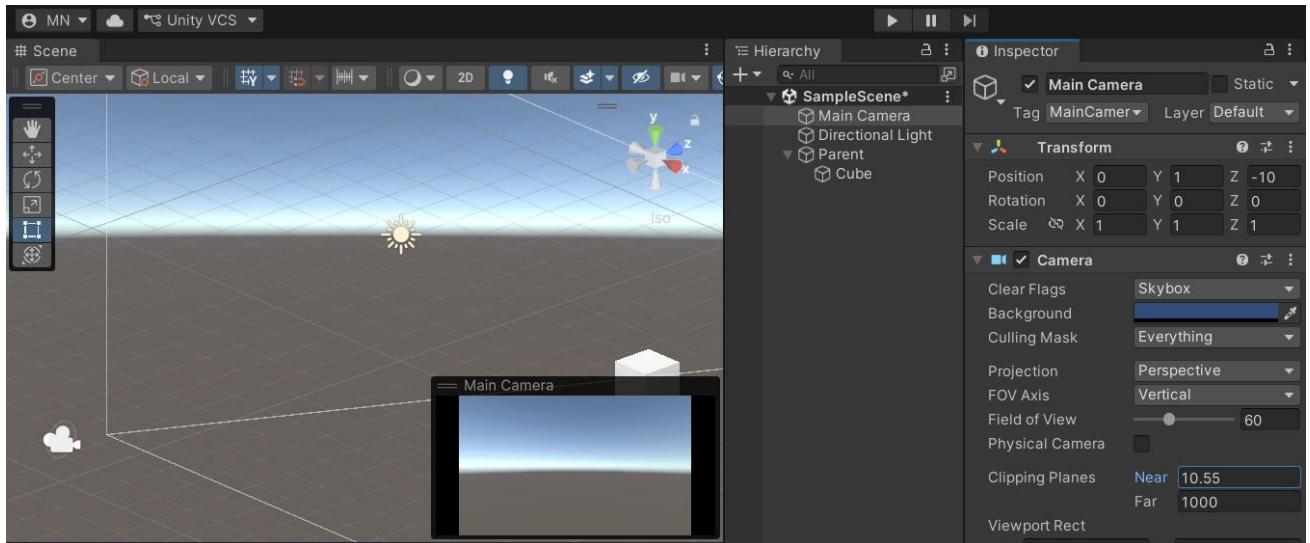


Hình d2.1: Ảnh sau khi chỉnh Field of View xuống thấp nhất ( $1e-05$ )



Hình d2.2: Ảnh sau khi chỉnh Field of View lên cao nhất (179)

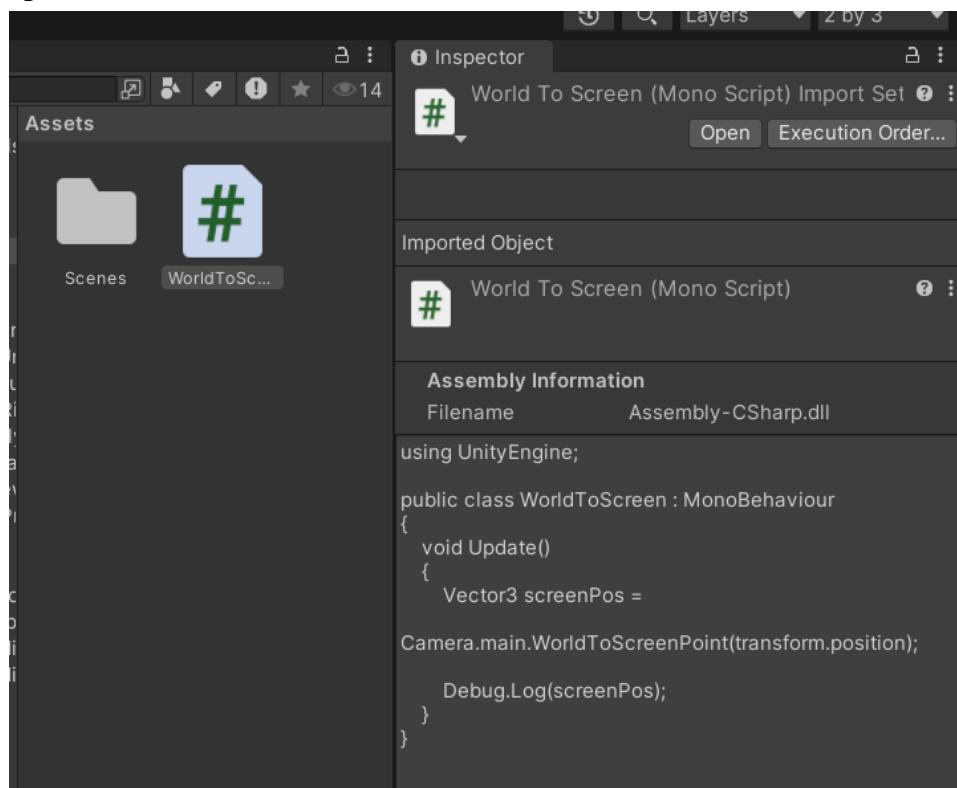
- Near Clip Plane



Hình d2.2: Ảnh sau khi chỉnh Near Clipping Planes thành 10.55 (Cube biến mất khỏi tầm nhìn của camera)

## PHẦN E – SCREEN SPACE

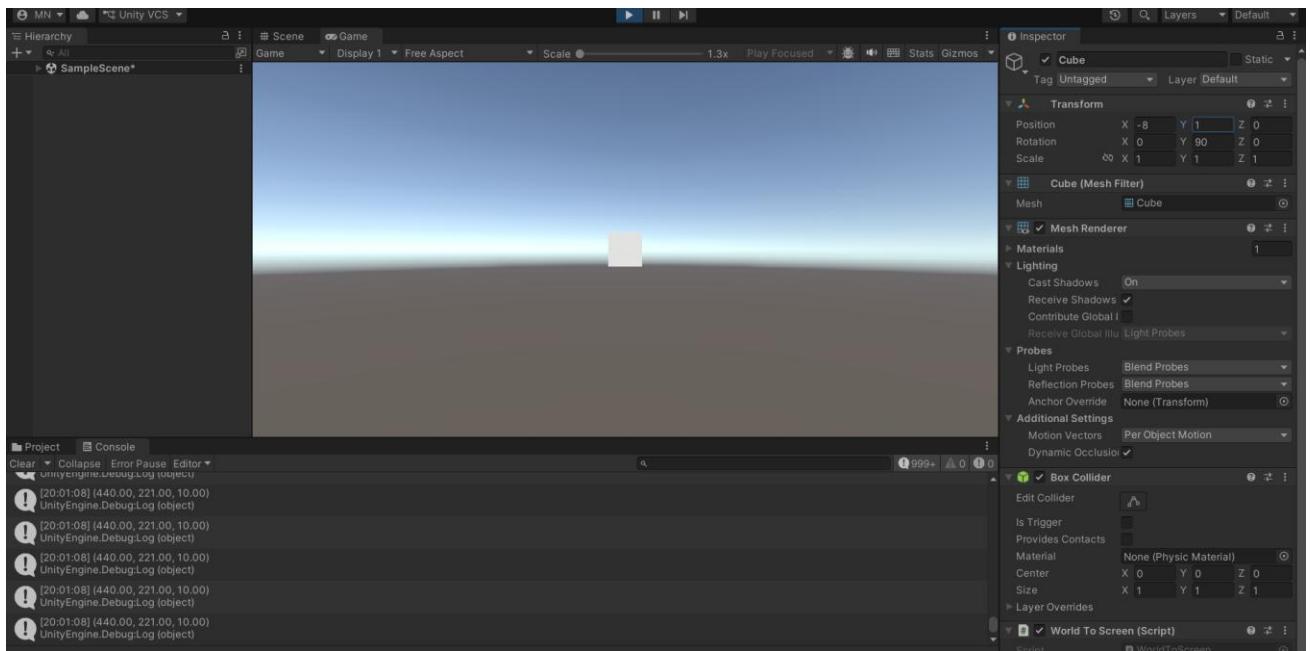
### E1. Tạo script WorldToScreen.cs



Hình e1: Ảnh sau khi tạo script WorldToScreen.cs

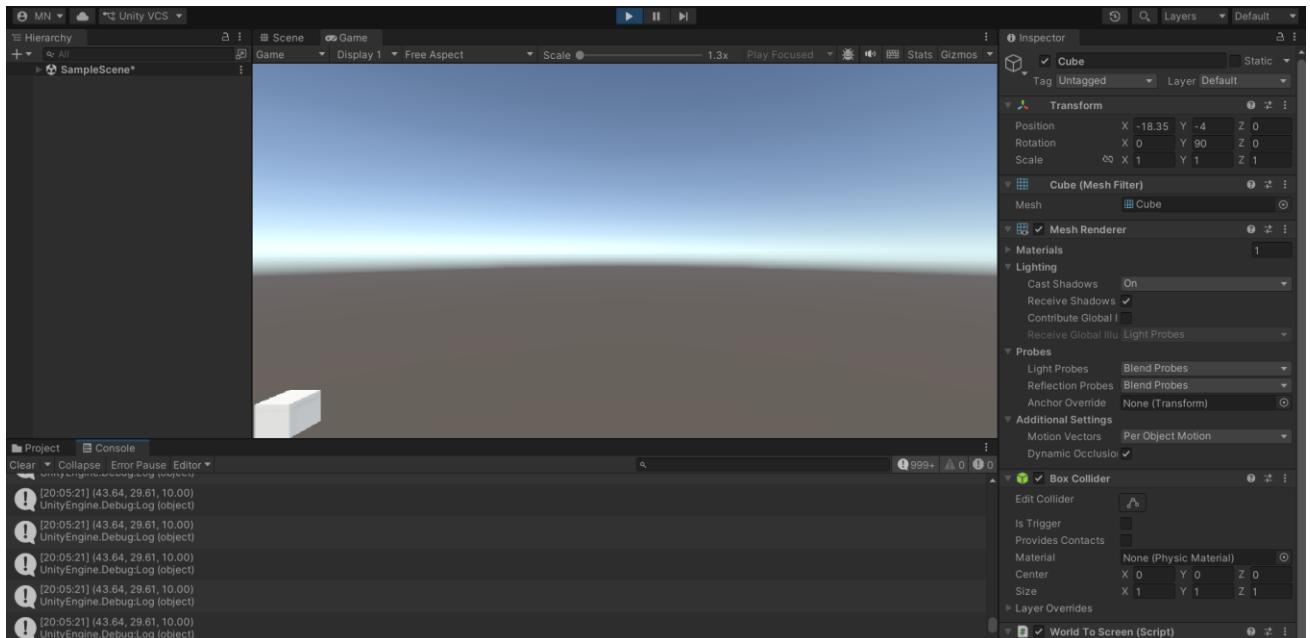
### E2. Gắn script vào Cube và chạy game

### E3. Ghi lại:



Hình e3.1: Ảnh Cube ở vị trí (-8, 1, 0) khi game đang chạy

- Screen Position khi Cube ở giữa màn hình: (440.00, 221.00, 10.00)



Hình e3.2: Ảnh Cube ở vị trí (-8, 1, 0) khi game đang chạy

- Screen Position khi Cube ở góc dưới bên trái: (43.64, 29.61, 10.00)

#### E4. Trả lời:

- Gốc tọa độ của Screen Space nằm ở đâu?
  - Góc dưới bên trái màn hình (0,0)
- Screen Space khác World Space như thế nào?
  - World Space sử dụng tọa độ 3D trong không gian game, trong khi Screen Space là hệ tọa độ 2D dựa trên màn hình hiển thị của người dùng.

#### Nhận xét cá nhân:

Unity sử dụng hệ tọa độ Cartesian ba chiều (X, Y, Z) để xác định vị trí của các đối tượng trong không gian. Trong đó, trục Y đóng vai trò là trục hướng lên trên, còn trục Z biểu diễn chiều sâu của cảnh và hướng nhìn mặc định của Camera. Các đối tượng được đặt trong World Space và được Camera thu nhận để xử lý và hiển thị lên màn hình thông qua Graphics Pipeline.

Unity sử dụng hệ tọa độ Left-Handed, điều này được thể hiện rõ qua cách đối tượng quay khi xoay theo trục Y. Cách hoạt động này ảnh hưởng trực tiếp đến việc định hướng, xoay và di chuyển các đối tượng trong quá trình phát triển game. Bên cạnh đó, mỗi GameObject có thể tồn tại trong Local Space hoặc World Space; Local Space phụ thuộc vào đối tượng cha, trong khi World Space thể hiện vị trí tuyệt đối của đối tượng trong Scene.

Khi Camera di chuyển hoặc thay đổi các thông số như Field of View và Clip Plane, hình ảnh của đối tượng trên màn hình sẽ thay đổi về kích thước hoặc có thể không còn được hiển thị, mặc dù vị trí của đối tượng trong World Space không thay đổi. Cuối cùng, việc chuyển đổi từ World Space sang Screen Space giúp Unity xác định tọa độ hiển thị 2D của đối tượng trên màn hình, với gốc tọa độ nằm ở góc dưới bên trái, từ đó phục vụ cho việc hiển thị giao diện và xử lý tương tác trong game.