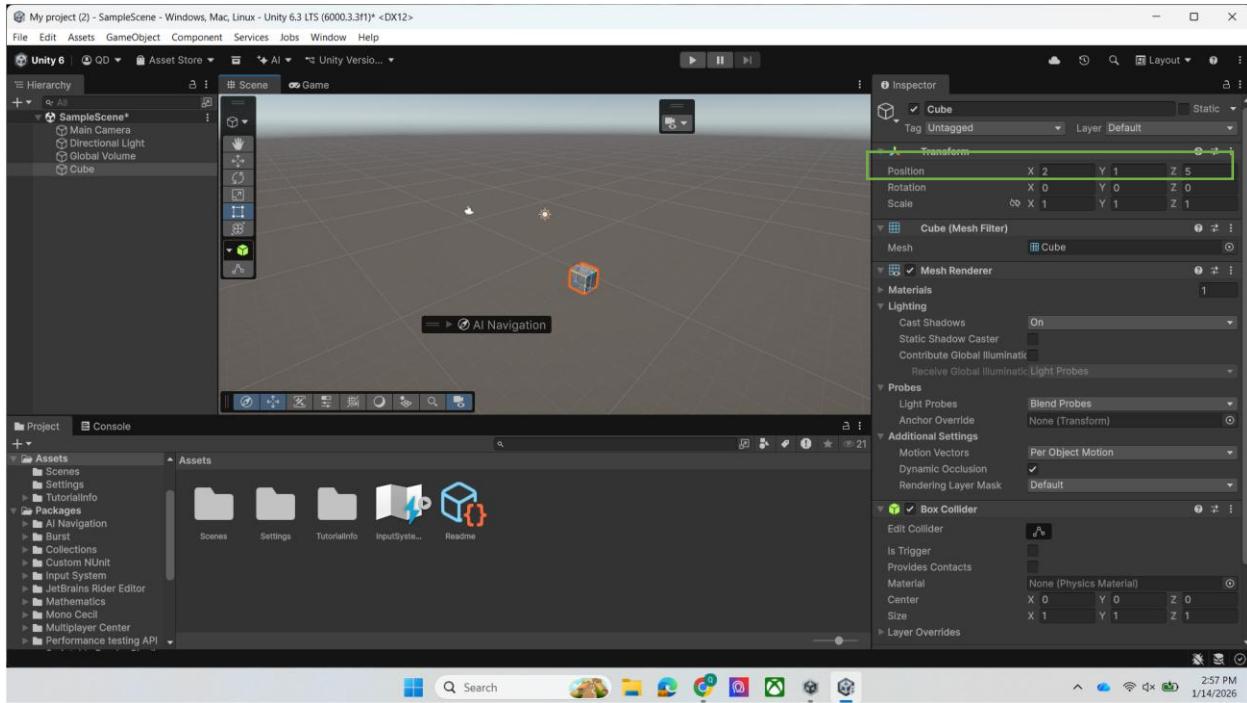


## PHẦN A – COORDINATE SYSTEM & WORLD SPACE (20%)

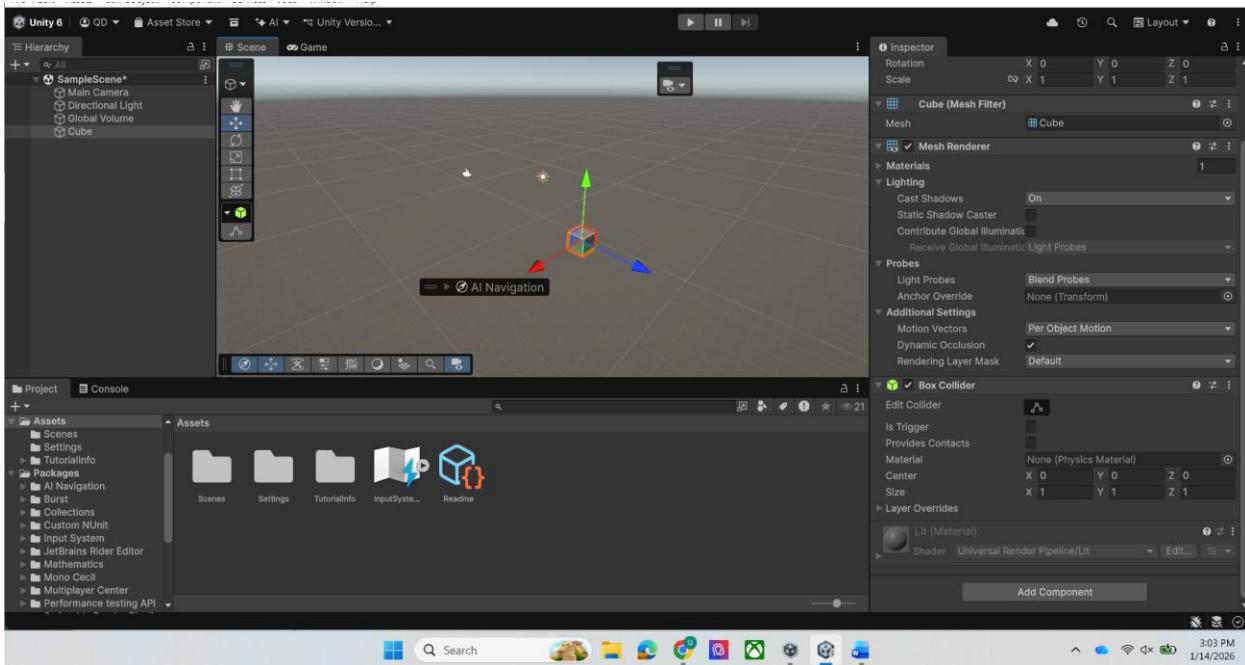
A1. Tạo một Cube tại vị trí:

X = 2, Y = 1, Z = 5



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

- Trục X
- Trục Y
- Trục Z



A3. Trả lời các câu hỏi:

- Trục nào hướng lên trên trong Unity?

Trục hướng lên trên là Y

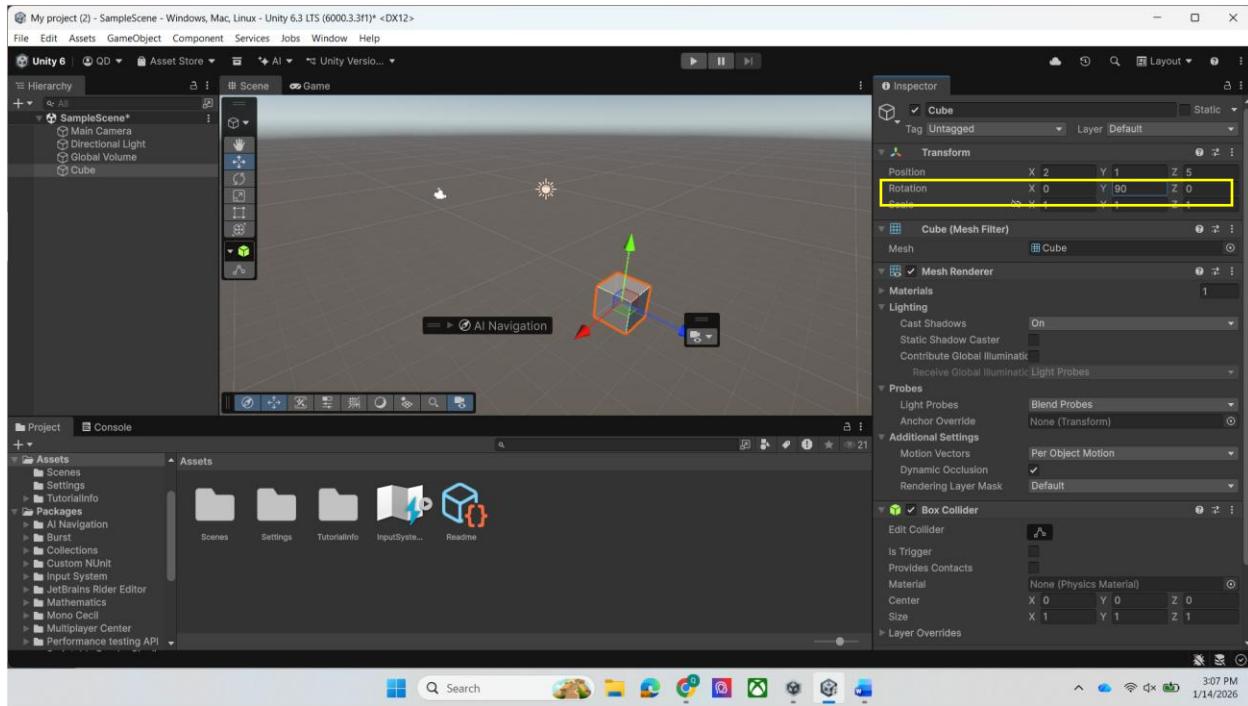
- Trục nào hướng về phía Camera?

Trục hướng về phía cam là Z

## **PHẦN B – LEFT-HANDED COORDINATE SYSTEM (15%)**

B1. Xoay Cube với Rotation:

$$Y = 90$$



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

- Cube quay theo chiều nào?

Cube quay theo bên phải

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

Khi xoay Cube quanh trục Y một góc dương ( $Y = 90$ ), object quay sang **bên phải**.

Điều này cho thấy Unity sử dụng **Left-Handed Coordinate System**, trong đó:

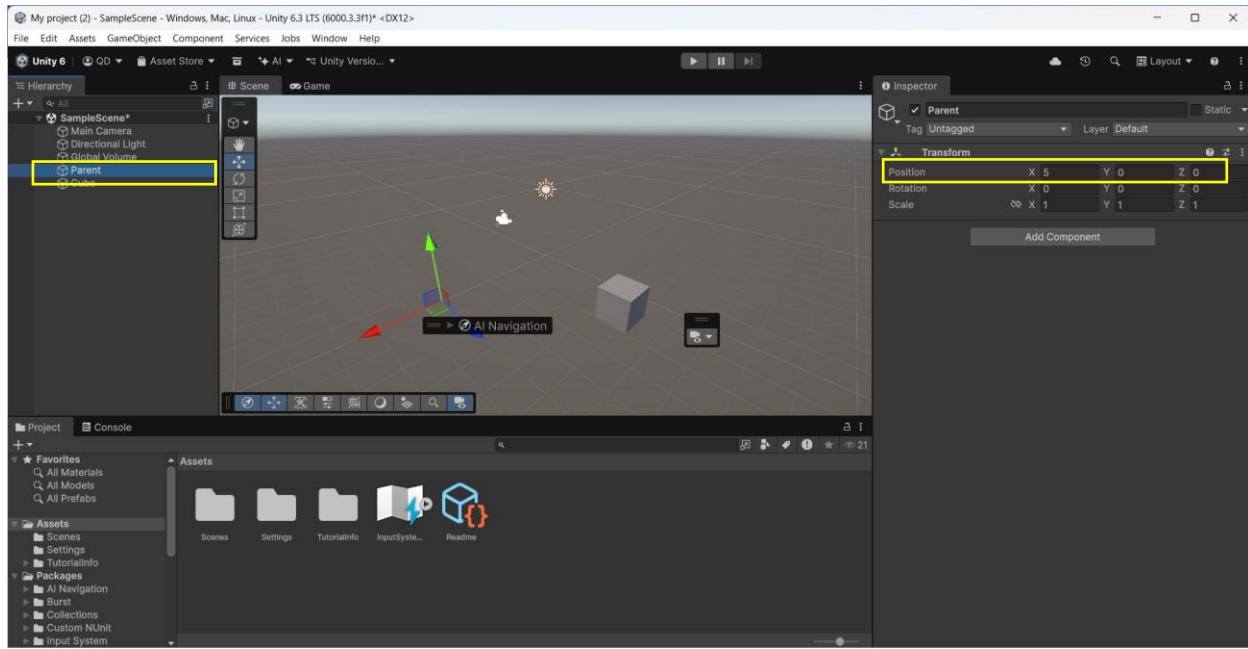
- Trục Y hướng lên trên
- Trục Z hướng ra phía trước
- Chiều quay dương tuân theo **quy tắc bàn tay trái**

Theo quy tắc này, khi ngón cái tay trái chỉ theo trục Y dương, thì các ngón còn lại chỉ chiều quay dương của object.

## PHẦN C – LOCAL SPACE VÀ WORLD SPACE (25%)

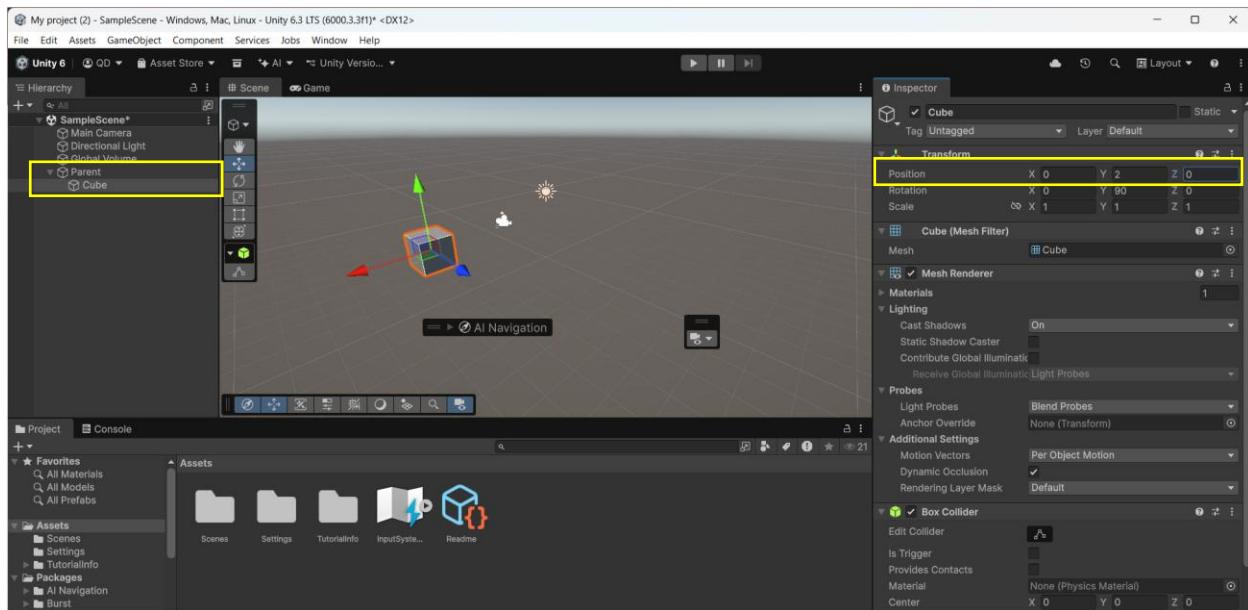
C1. Tạo một Empty GameObject tên là “Parent” tại vị trí:

$(5, 0, 0)$



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

Local Position của Cube = (0, 2, 0)

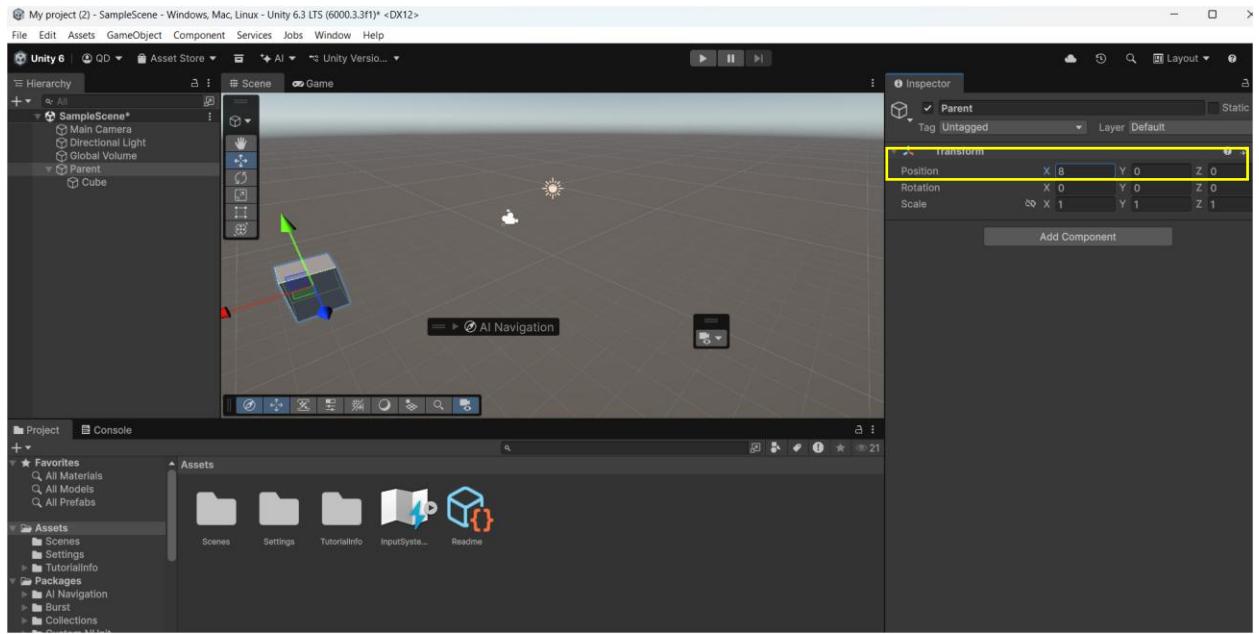


C3. Ghi lại:

- Local Position của Cube (0,2,0)
- World Position của Cube(5,2,0)

C4. Di chuyển Parent sang vị trí:

(8, 0, 0)



Trả lời:

- Local Position của Cube có thay đổi không?

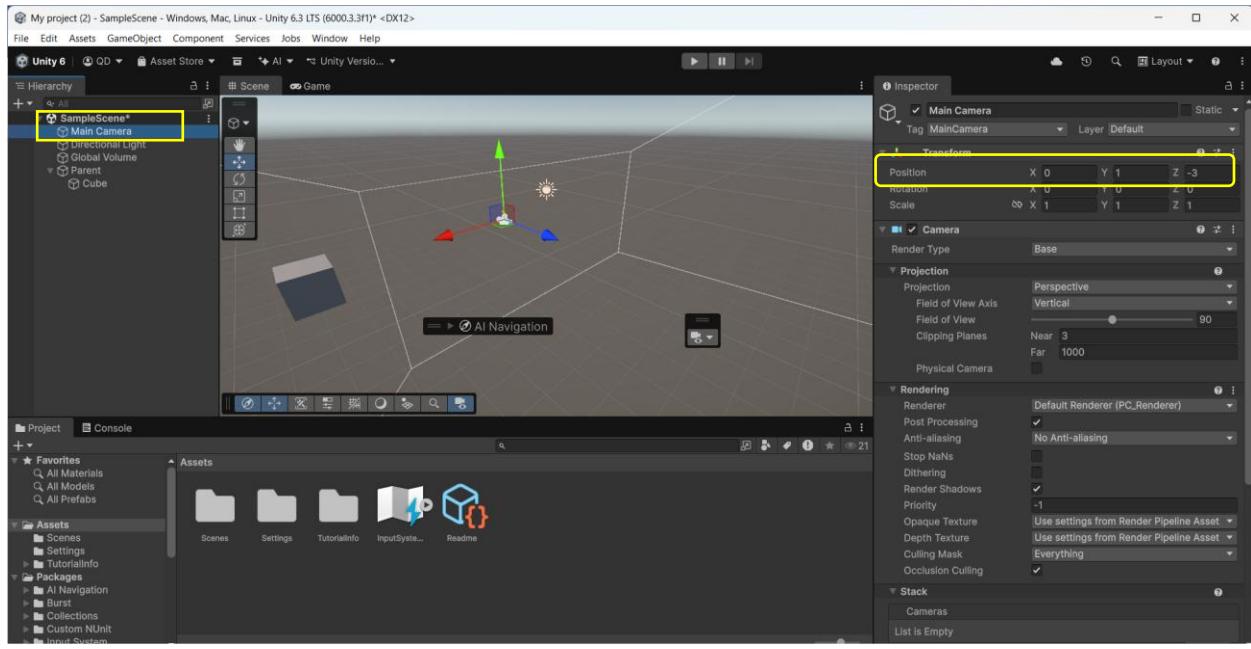
Local không đổi

- World Position của Cube thay đổi như thế nào?

World thành (8,2,0)

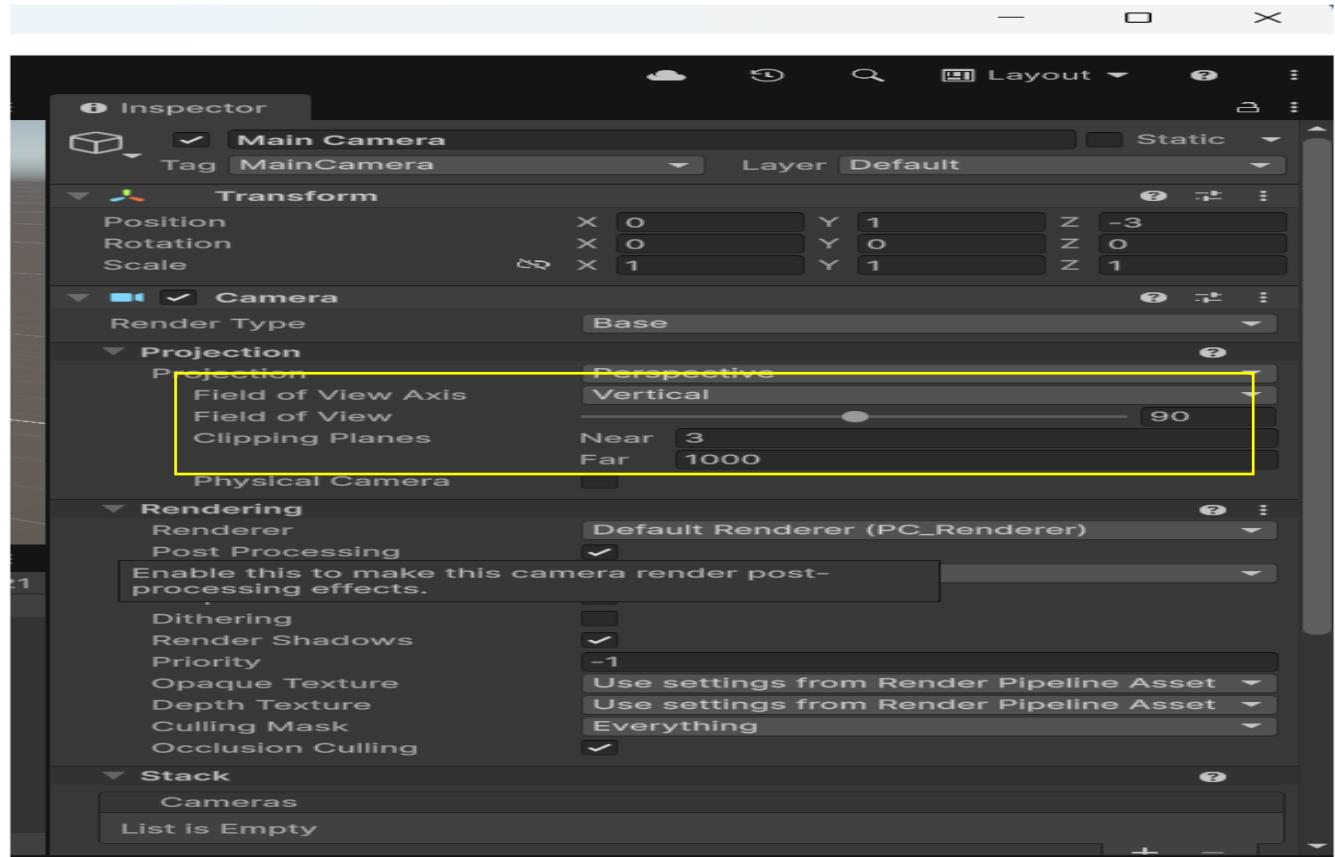
## PHẦN D – GRAPHICS PIPELINE (20%)

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



D2. Thay đổi các thông số của Camera:

- Field of View (30 → 90)
- Near Clip Plane(0.3 → 3)



- Vì sao object trông to hoặc nhỏ hơn dù không đổi vị trí?

Do sự thay đổi của Field of View làm thay đổi góc nhìn nên nó trông to hoặc nhỏ hơn

- Vì sao object có thể biến mất khỏi màn hình?

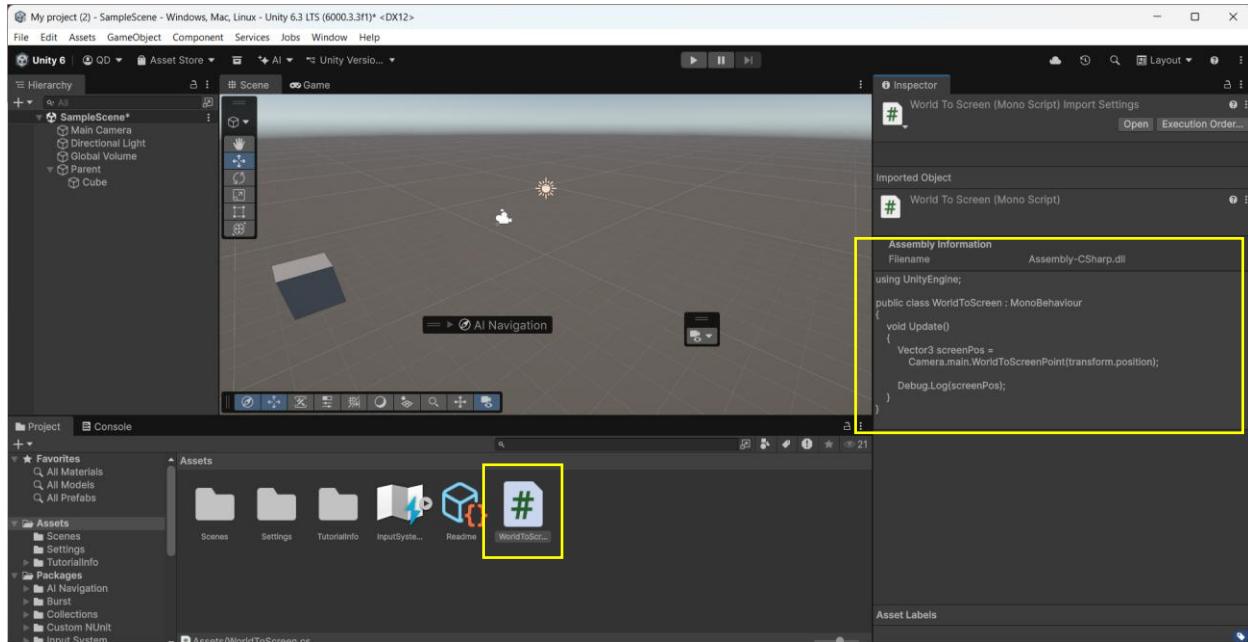
Do vật thể nằm ngoài vùng View Frustum cụ thể là bị loại bỏ bởi Near Clip Plane

## PHẦN E – SCREEN SPACE (20%)

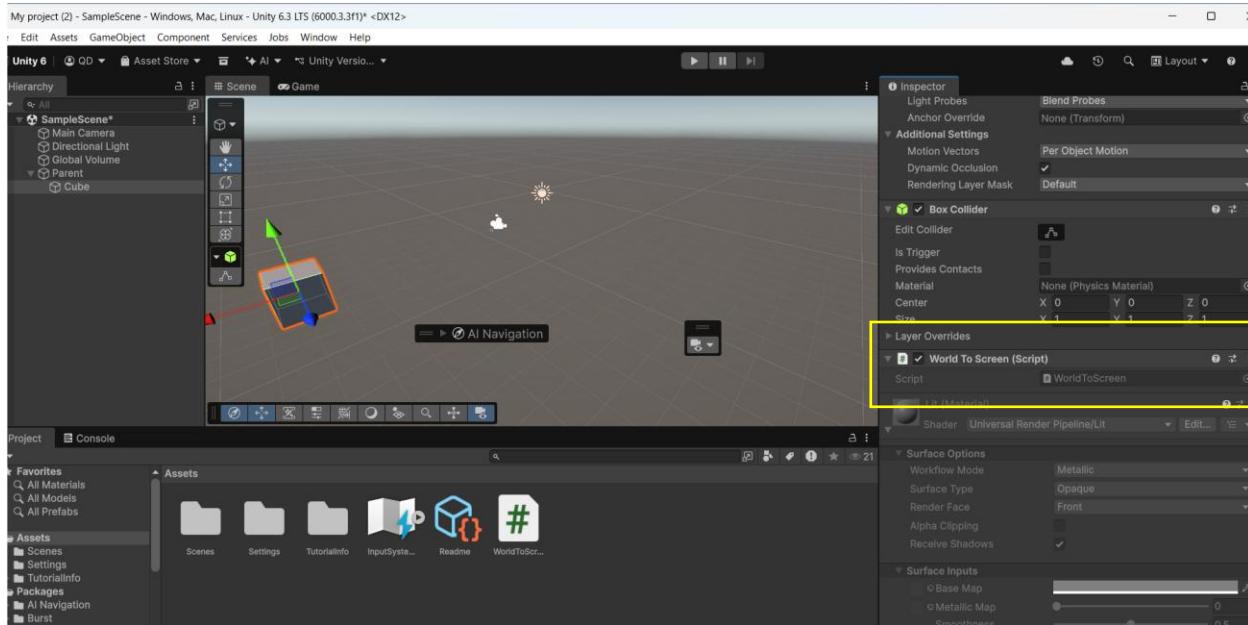
E1. Tạo script WorldToScreen.cs với nội dung:

using UnityEngine;

```
public class WorldToScreen : MonoBehaviour
{
    void Update()
    {
        Vector3 screenPos =
            Camera.main.WorldToScreenPoint(transform.position);
        Debug.Log(screenPos);
    }
}
```



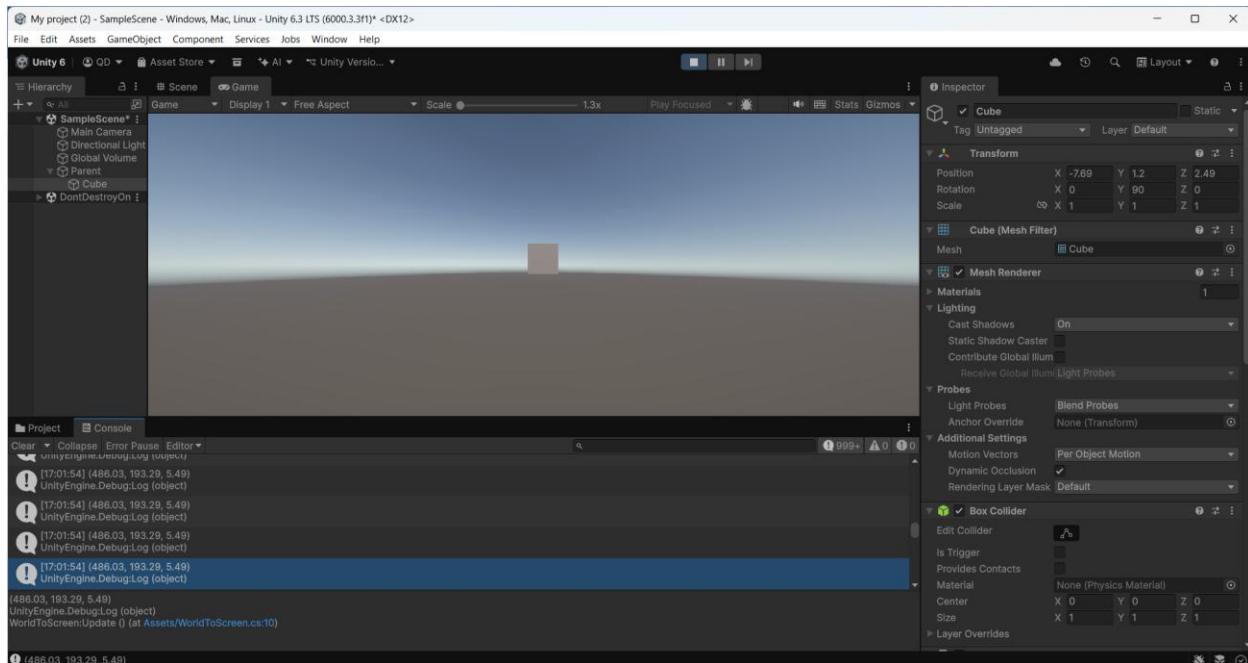
E2. Gắn script vào Cube và chạy game



E3.

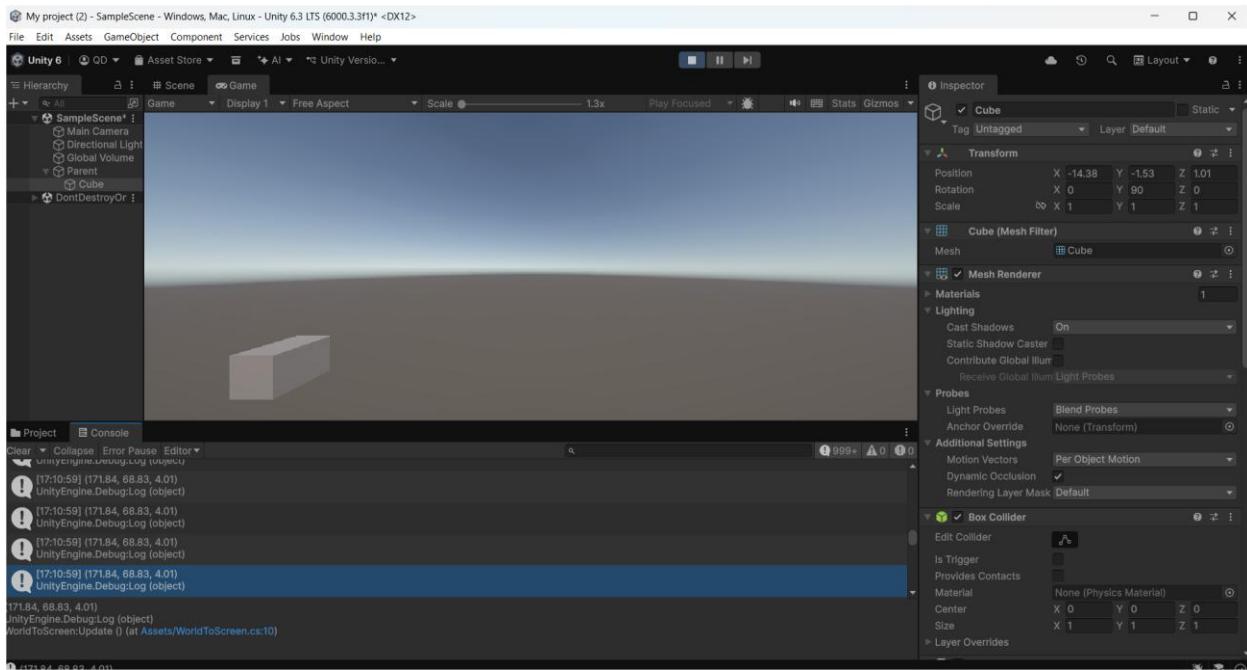
- Screen Position khi Cube ở giữa màn hình

Tọa độ(454.18, 191.83, 7.00)



- Screen Position khi Cube ở góc dưới bên trái

Tọa độ(171.84, 68.83, 4.01)



## Góc tọa độ của Screen Space nằm ở đâu?

Góc tọa độ của Screen Space nằm tại **góc dưới bên trái màn hình**, có tọa độ (0, 0), và các giá trị tọa độ được tính theo đơn vị pixel.

## Screen Space khác World Space như thế nào?

Screen Space là hệ tọa độ **2D dùng để hiển thị trên màn hình**, phụ thuộc vào độ phân giải và Camera, trong khi World Space là hệ tọa độ **3D biểu diễn vị trí thực của object trong không gian scene**.

## Nhận xét

Thông qua bài thực hành về Coordinate System và các không gian trong Unity, em đã hiểu rõ hơn cách Unity tổ chức và xử lý không gian 3D. Việc làm quen với hệ trục tọa độ X, Y, Z giúp em hình dung chính xác vị trí và hướng của các object trong Scene, từ đó thao tác di chuyển, xoay và sắp xếp object một cách hiệu quả hơn. Đặc biệt, phần thực hành xoay object đã giúp em hiểu rõ Unity sử dụng hệ tọa độ Left-Handed Coordinate System và cách chiều quay dương được áp dụng trong thực tế.

Bên cạnh đó, bài tập về Local Space và World Space giúp em phân biệt rõ mối quan hệ giữa object cha và object con. Em nhận ra rằng Local Position của object không thay đổi khi di chuyển Parent, trong khi World Position thay đổi theo, điều này rất quan trọng khi xây dựng các mô hình phân cấp trong game hoặc ứng dụng 3D. Phần Graphics Pipeline và Camera cũng giúp em hiểu được vai trò của Camera trong việc hiển thị hình ảnh, đặc biệt là ảnh hưởng của Field of View và Clip Plane đến kích thước cũng như khả năng hiển thị của object.

Cuối cùng, thông qua phần Screen Space và việc chuyển đổi từ World Space sang Screen Space bằng script, em hiểu rõ hơn cách Unity hiển thị object lên màn hình và cách xác định vị trí hiển thị theo tọa độ pixel. Bài thực hành này giúp em củng cố kiến thức nền tảng quan trọng trong Unity, tạo tiền đề để tiếp tục học các nội dung nâng cao như UI, điều khiển camera và lập trình gameplay sau này.