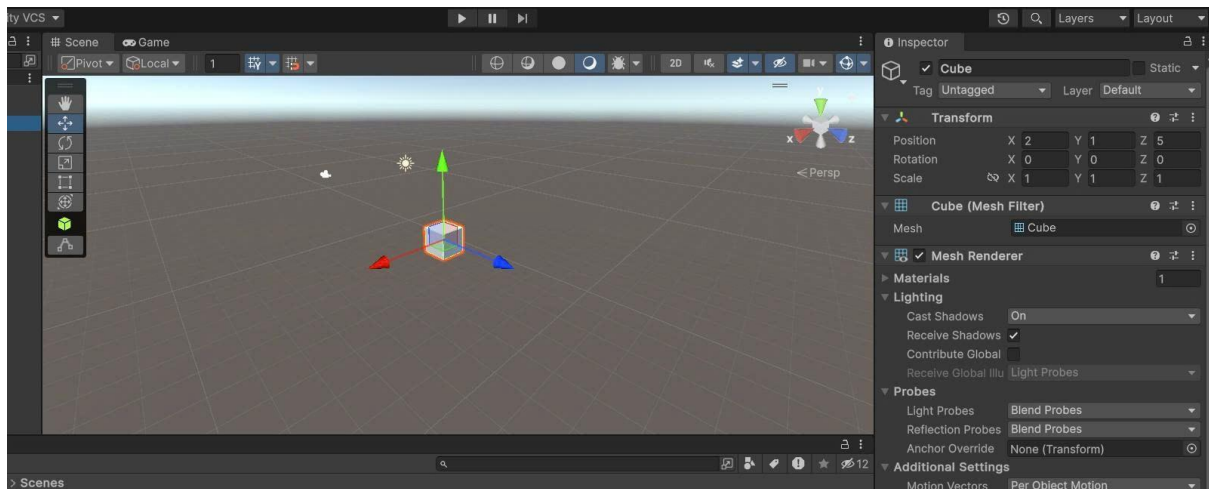


PHẦN A – COORDINATE SYSTEM & WORLD SPACE

A1+A2:

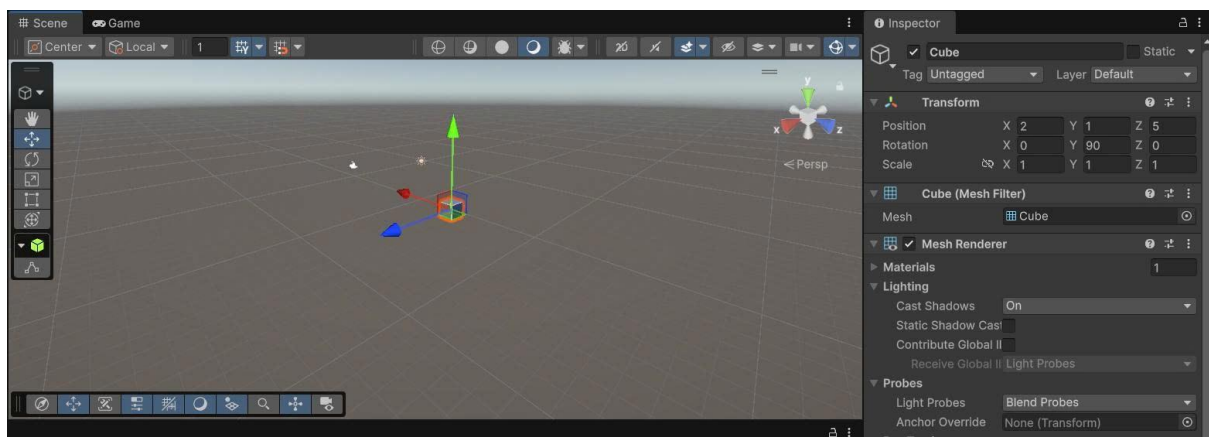


A3:

- ☐ Trục hướng lên trên: **Y**
- ☐ Trục hướng về phía Camera: **Z**

PHẦN B – LEFT-HANDED COORDINATE SYST

B1:

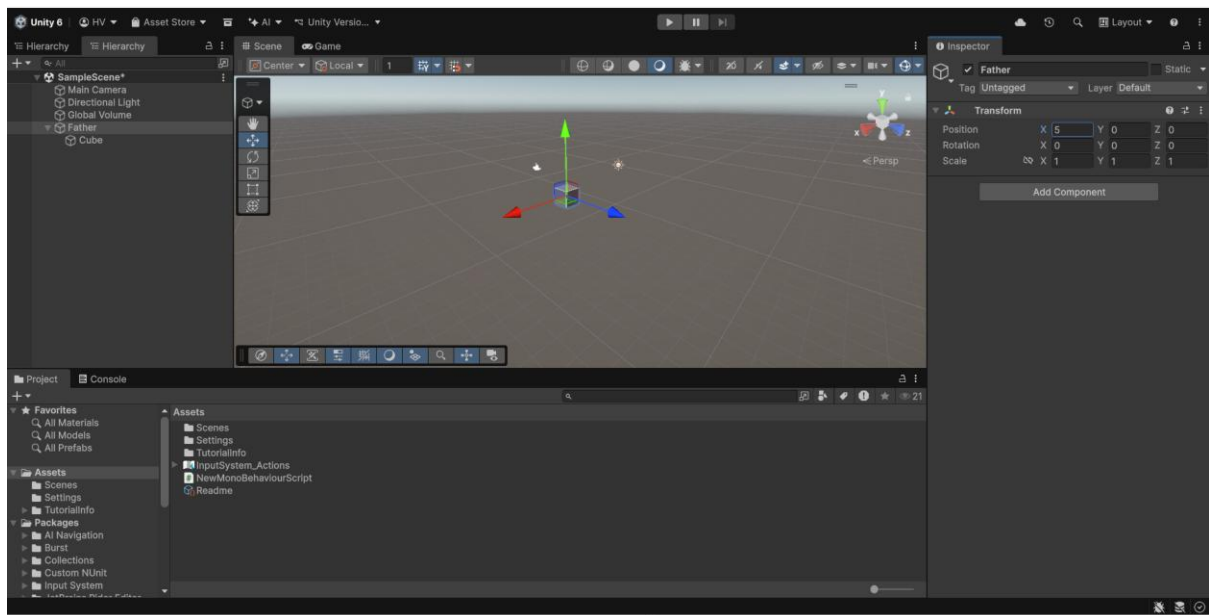


B2:

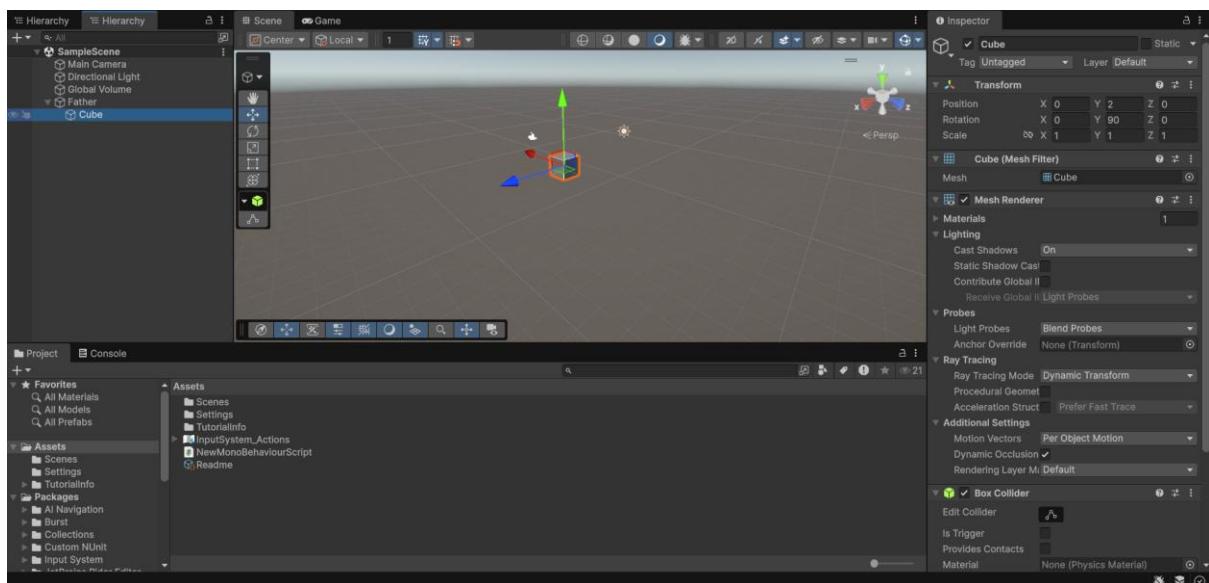
- ☐ Nhìn từ trên xuống:
 - Cube quay **theo chiều kim đồng hồ**
- ☐ Kết luận:
 - Unity dùng **Left-Handed Coordinate System**

PHẦN C – LOCAL SPACE & WORLD SPACE

C1:



C2:

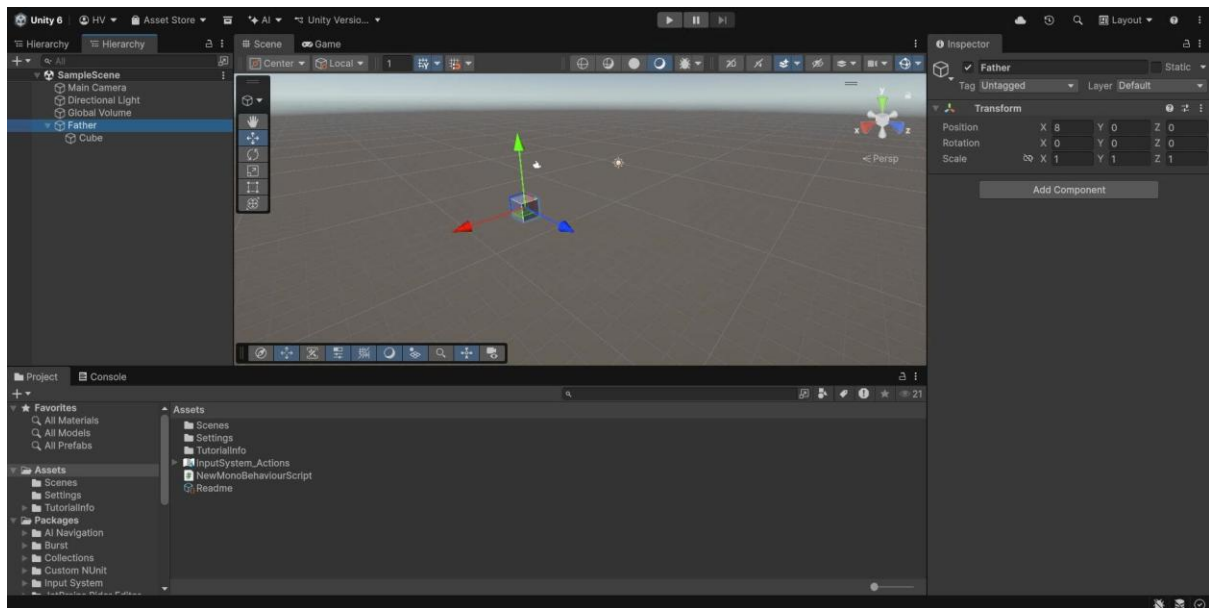


C3:

Vị trí: Local Position của Cube (0, 2, 0)

World Position của Cube (5, 2, 0)

C4:

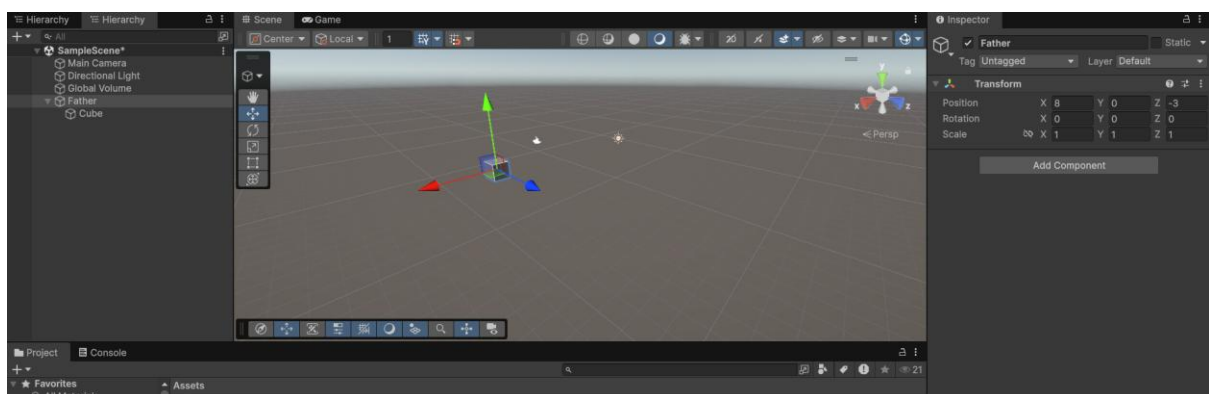
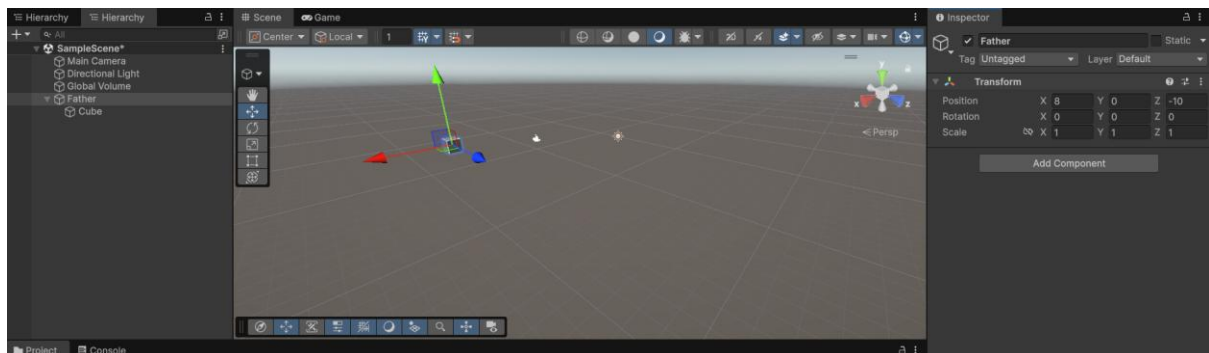


Local Position của Cube không thay đổi vẫn là (0, 2, 0)

World Position của Cube thay đổi thành (8, 2, 0)

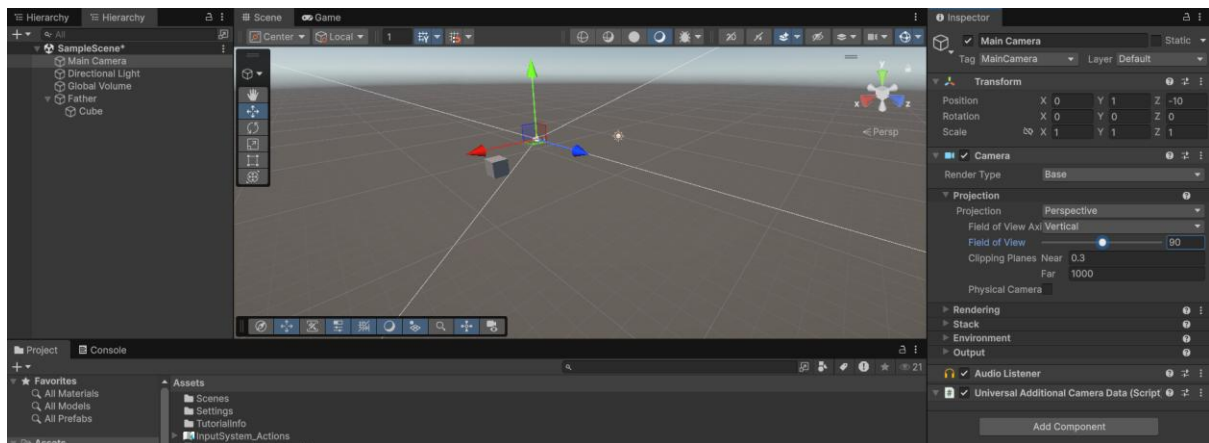
PHẦN D – GRAPHICS PIPELINE (20%)

D1:

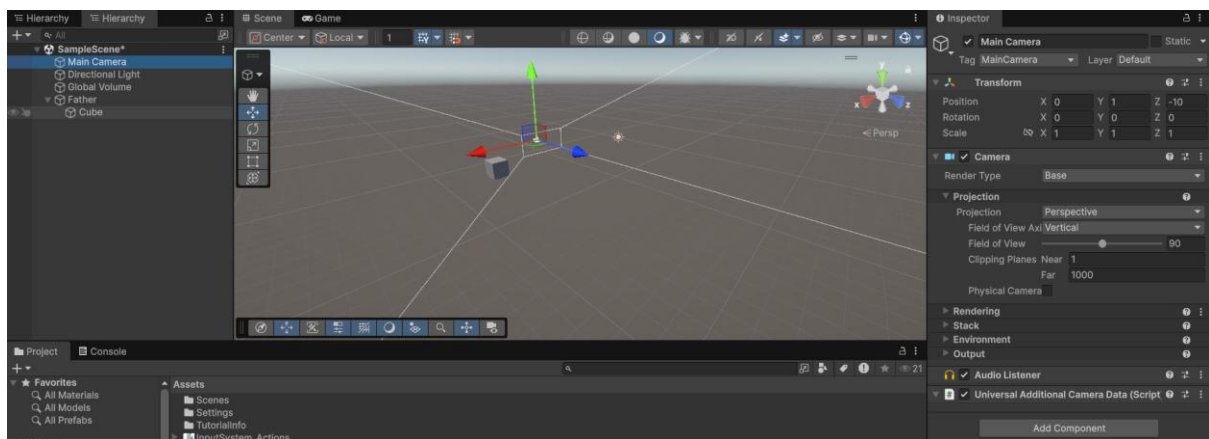


D2:

Field of View



Near Clip Plane



Do FOV:

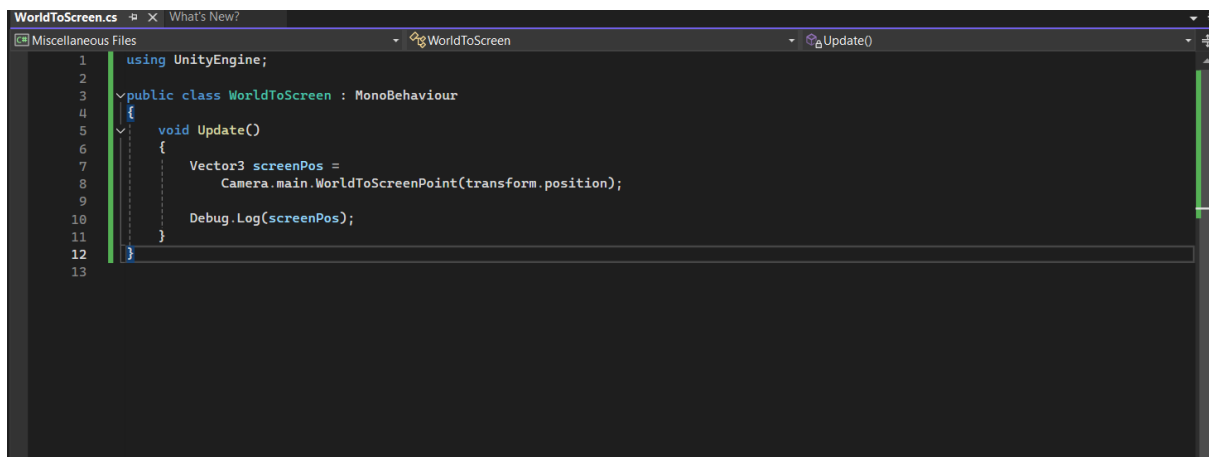
- FOV lớn → góc nhìn rộng → object trông nhỏ hơn
- FOV nhỏ → góc nhìn hẹp → object trông to hơn

Do Near / Far Clip Plane

- Object nằm ngoài vùng cắt sẽ **không được render**

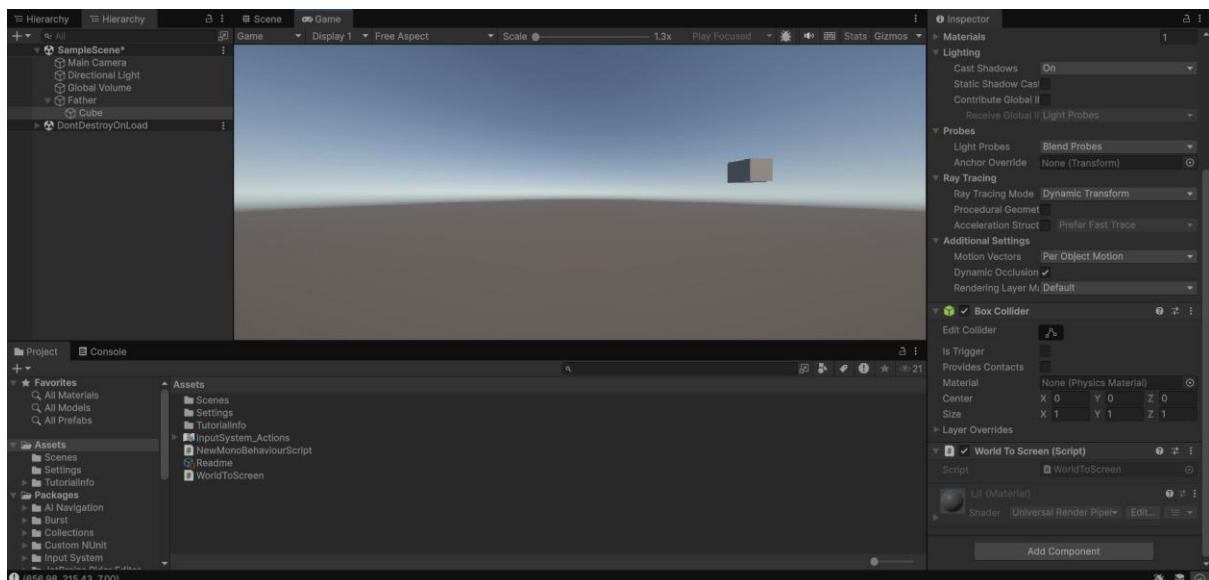
PHẦN E – SCREEN SPACE (20%)

E1:



```
1 using UnityEngine;
2
3 public class WorldToScreen : MonoBehaviour
4 {
5     void Update()
6     {
7         Vector3 screenPos =
8             Camera.main.WorldToScreenPoint(transform.position);
9
10        Debug.Log(screenPos);
11    }
12 }
13
```

E2:



E3:

Cube ở giữa màn hình:

$$x \approx \text{Screen.width} / 2$$

$$y \approx \text{Screen.height} / 2$$

Cube ở góc dưới bên trái:

$$x \approx 0$$

$$y \approx 0$$

E4:

Gốc tọa độ Screen Space ở đâu?

Gốc dưới bên trái màn hình (0, 0)

Khác nhau giữa Screen Space và World Space:

World Space:

- + Không gian 3D
- + Đơn vị mét
- + Phụ thuộc vị trí object
- + Dùng cho logic game

Screen Space:

- + Không gian 2D
- + Đơn vị pixel
- + Phụ thuộc độ phân giải
- + Dùng cho UI, hiển thị

NHẬN XÉT CÁ NHÂN

Thông qua bài tập này, em hiểu rõ hơn cách Unity tổ chức không gian 3D và cách một đối tượng được hiển thị lên màn hình 2D thông qua Camera và Graphics Pipeline. Việc phân biệt Local Space, World Space và Screen Space giúp em tránh được nhiều lỗi khi lập trình game, đặc biệt trong việc xử lý Camera, UI và các hiệu ứng tương tác. Bài tập cũng giúp em hiểu rõ sự khác biệt giữa Left-Handed và Right-Handed Coordinate System – một kiến thức nền tảng quan trọng khi làm việc với các engine và công cụ đồ họa khác nhau.