

ENGINE HUB – MATHEMATICS IN UNITY

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

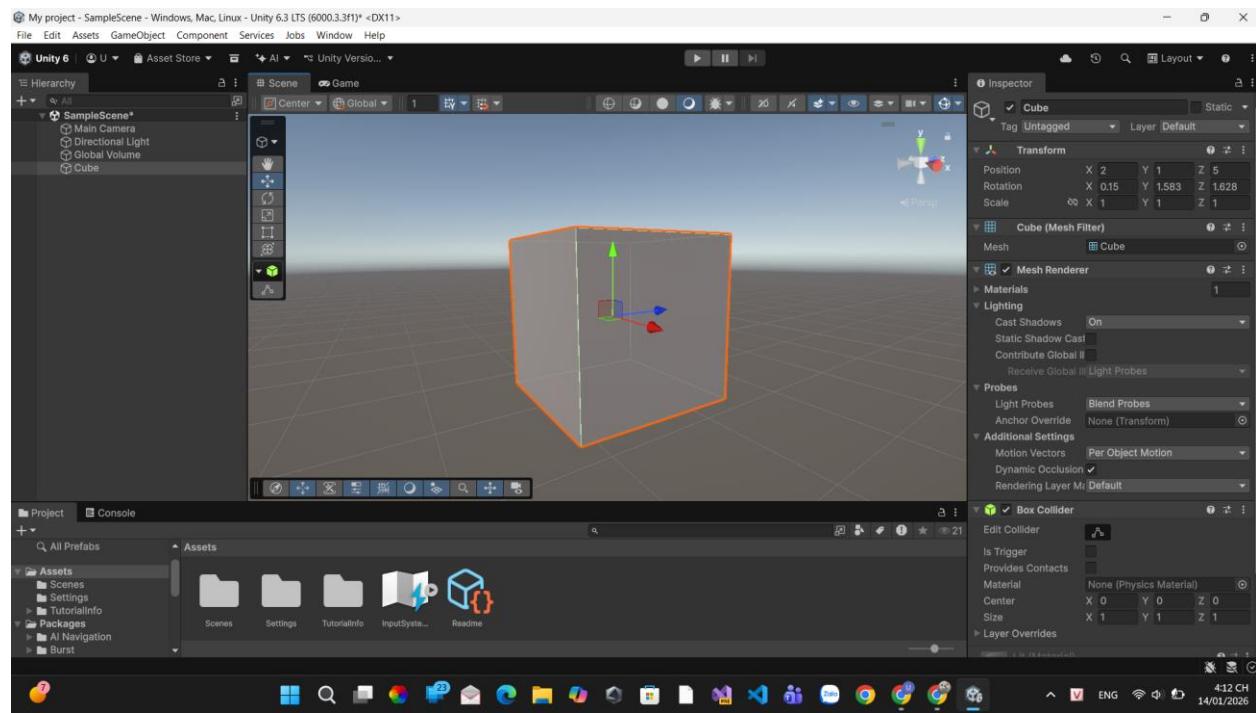
- Trục Y

- Trục Z

A3. Trả lời các câu hỏi

- Trục nào hướng lên trên trong Unity: là trục Y

- Trục nào hướng về phía Camera: là trục Z dương



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

B1. Xoay Cube với Rotation:

$$Y = 90$$

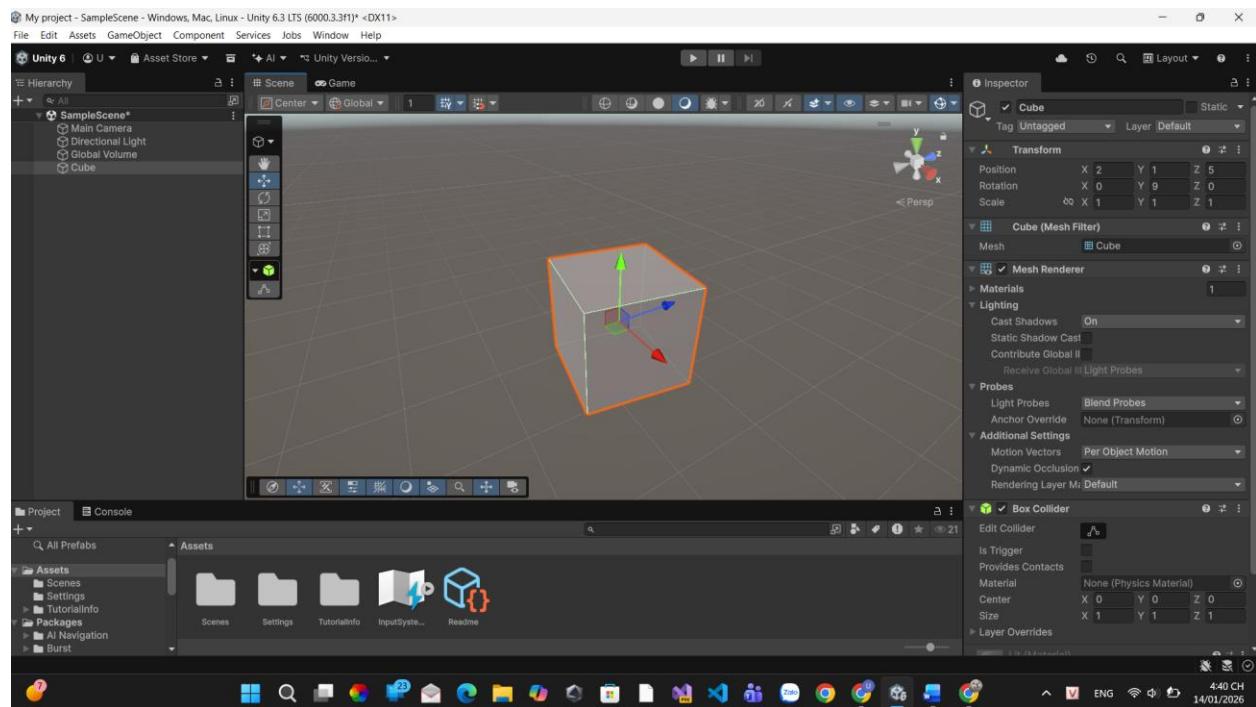
B2. Quan sát hướng quay của Cube và trả lời:

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

Trả lời:

- Cube xoay sang bên phải khi tăng Rotation Y.

- Điều này thể hiện Unity sử dụng Left-Handed Coordinate System, trong đó chiều quay dương của trục tuân theo quy tắc bàn tay trái



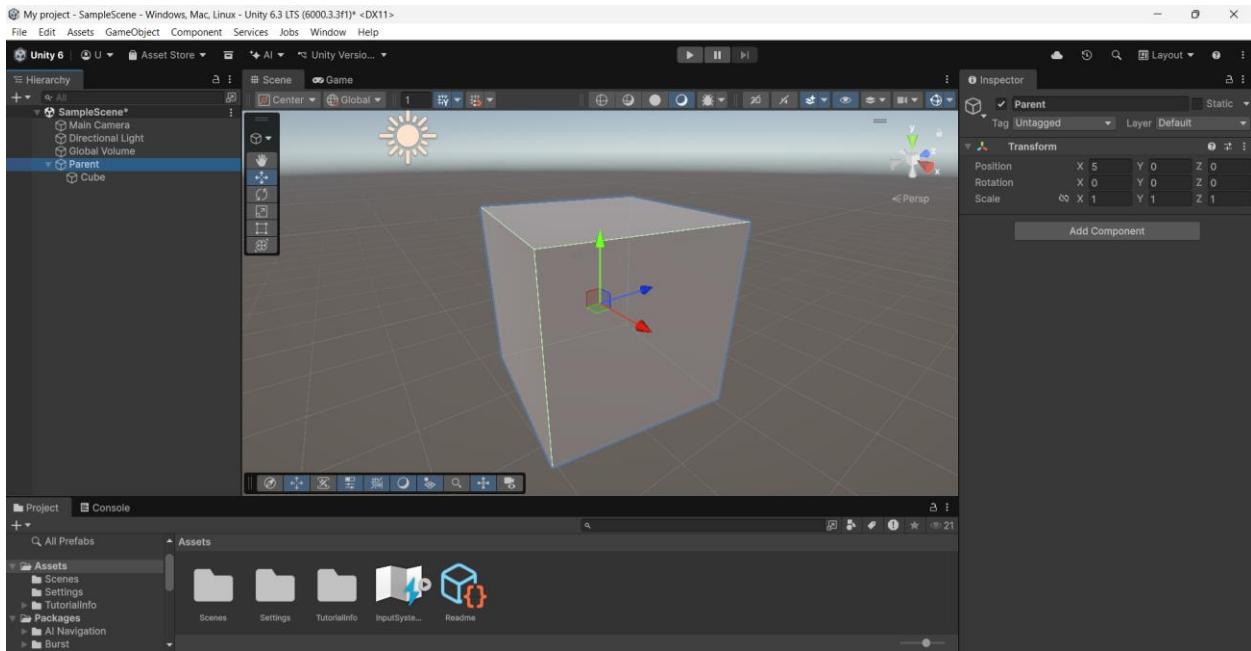
PHẦN C – LOCAL SPACE VÀ WORLD SPACE (25%)

C1. Tạo một Empty GameObject tên là “Parent” tại vị trí:
(5, 0, 0)

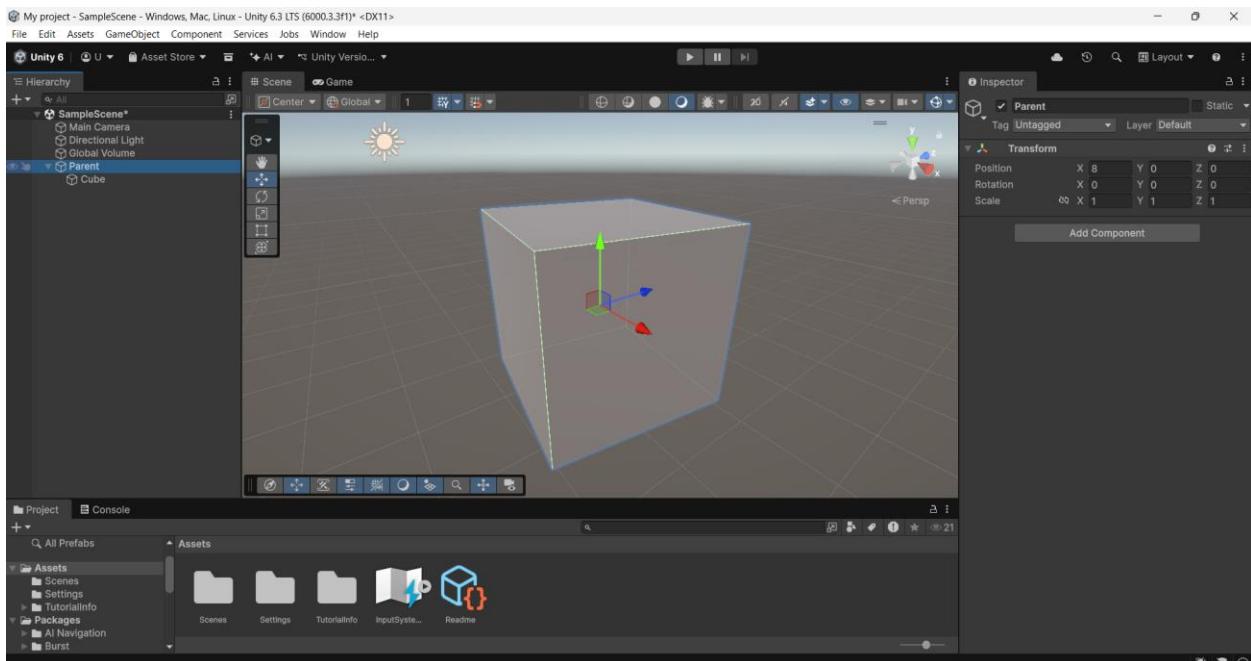
C2. Đặt Cube làm con của Parent và thiết lập:
Local Position của Cube = (0, 2, 0)

C3. Ghi lại:

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



C4. Di chuyển Parent sang vị trí:
 $(8, 0, 0)$



C5. Trả lời các câu hỏi:
- Local Position của Cube có thay đổi không?
- World Position của Cube thay đổi như thế nào?

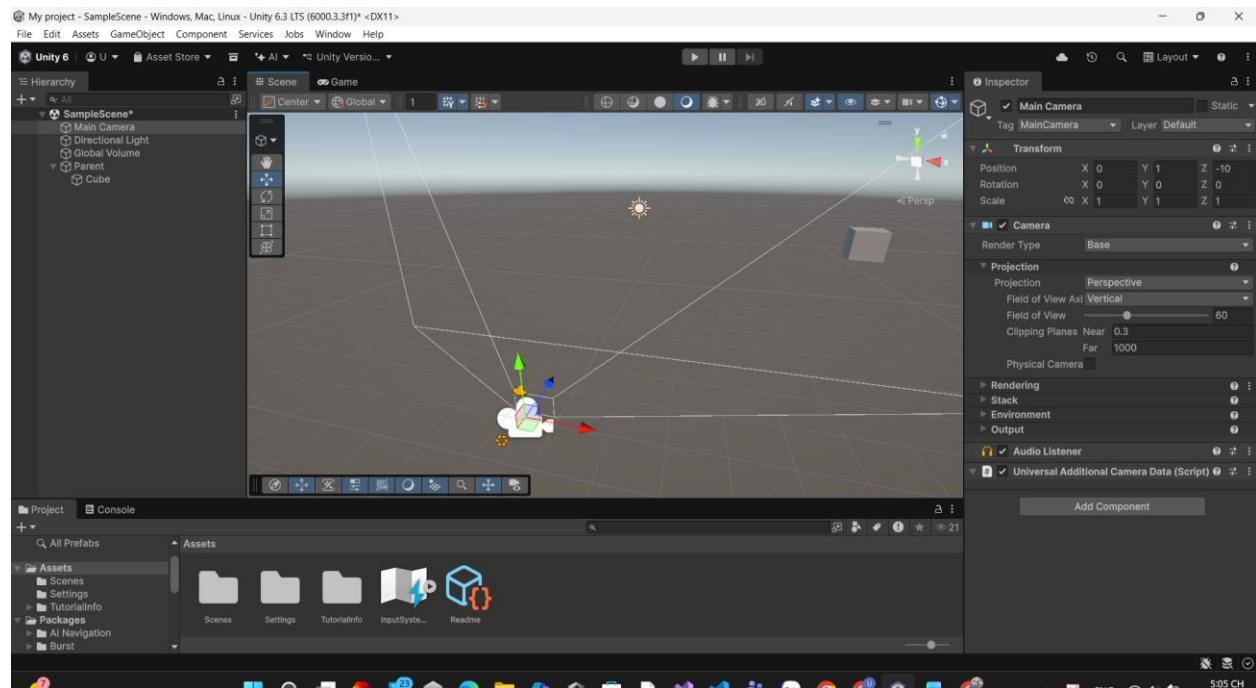
Trả lời:

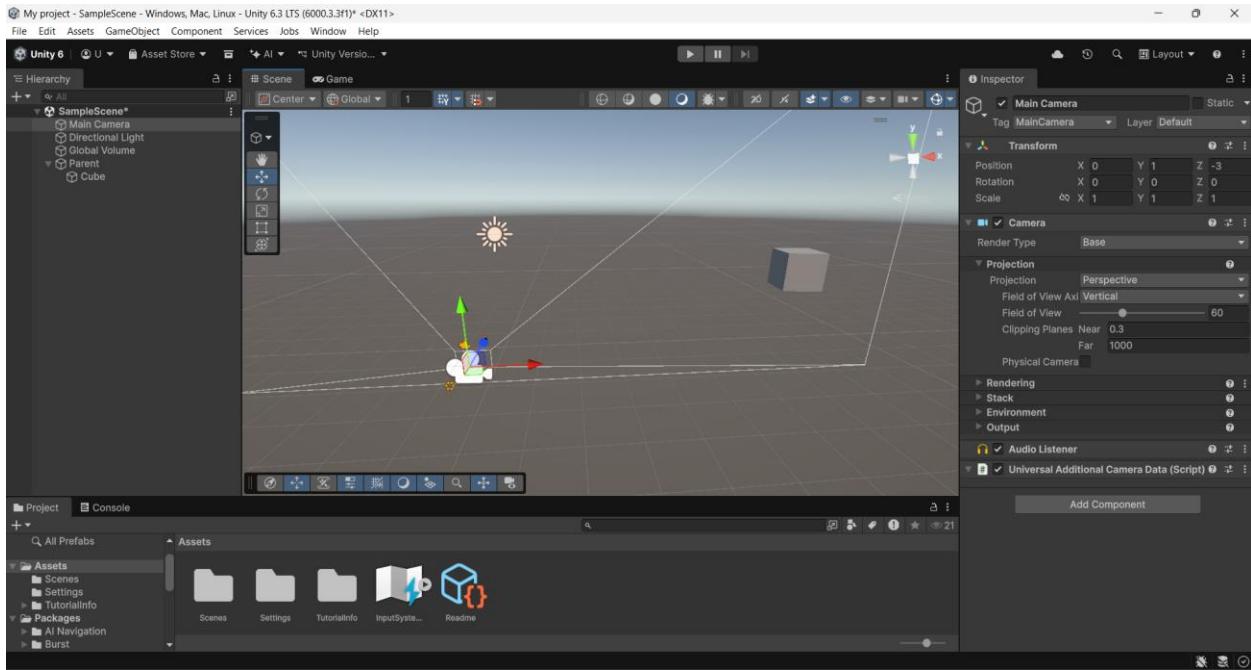
-Local Position của Cube: không thay đổi

-World Position của Cube: (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
- Near Clip Plane

D3. Trả lời:

- Vì sao object trông to hoặc nhỏ hơn dù không đổi vị trí?
- Vì sao object có thể biến mất khỏi màn hình?

Trả lời:

- Object trông to hoặc nhỏ hơn vì Field of View thay đổi góc nhìn phô cản (Perspective Projection)
- Object có thể biến mất khỏi màn hình khi nằm trước Near Clip Plane hoặc nằm ngoài vùng nhìn của Camera

PHẦN E – SCREEN SPACE (20%)

E1. Tạo script WorldToScreen.cs với nội dung:

```
using UnityEngine;
```

```
public class WorldToScreen : MonoBehaviour
```

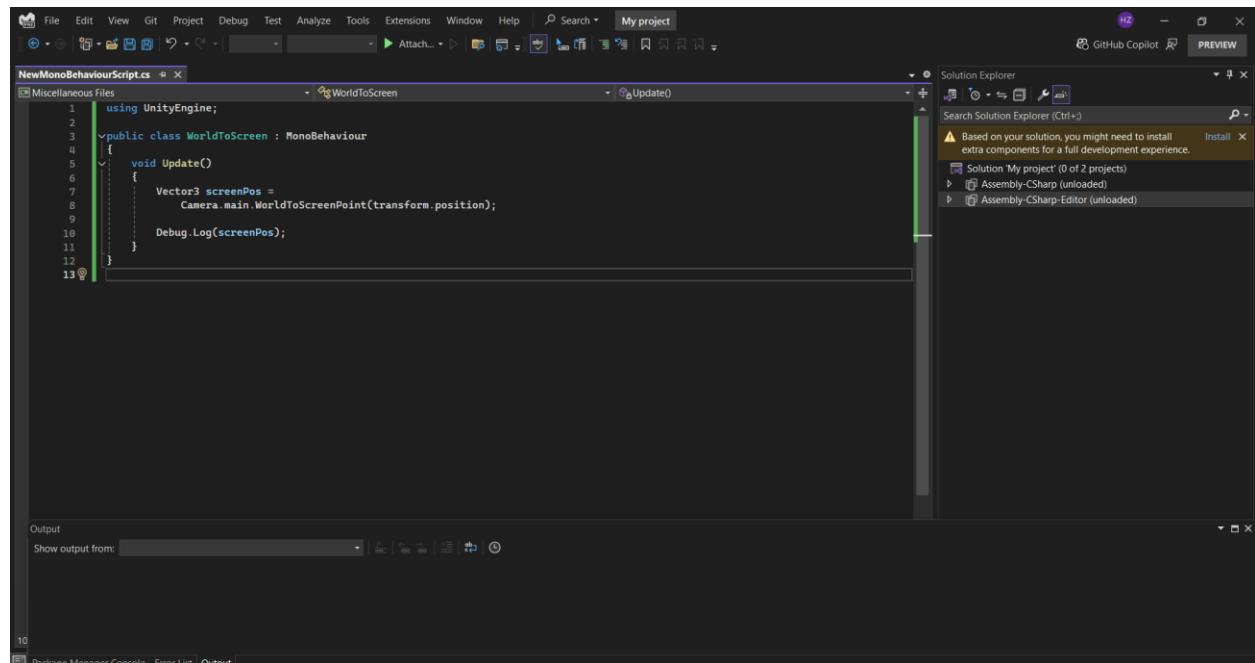
```
{
```

```
    void Update()
```

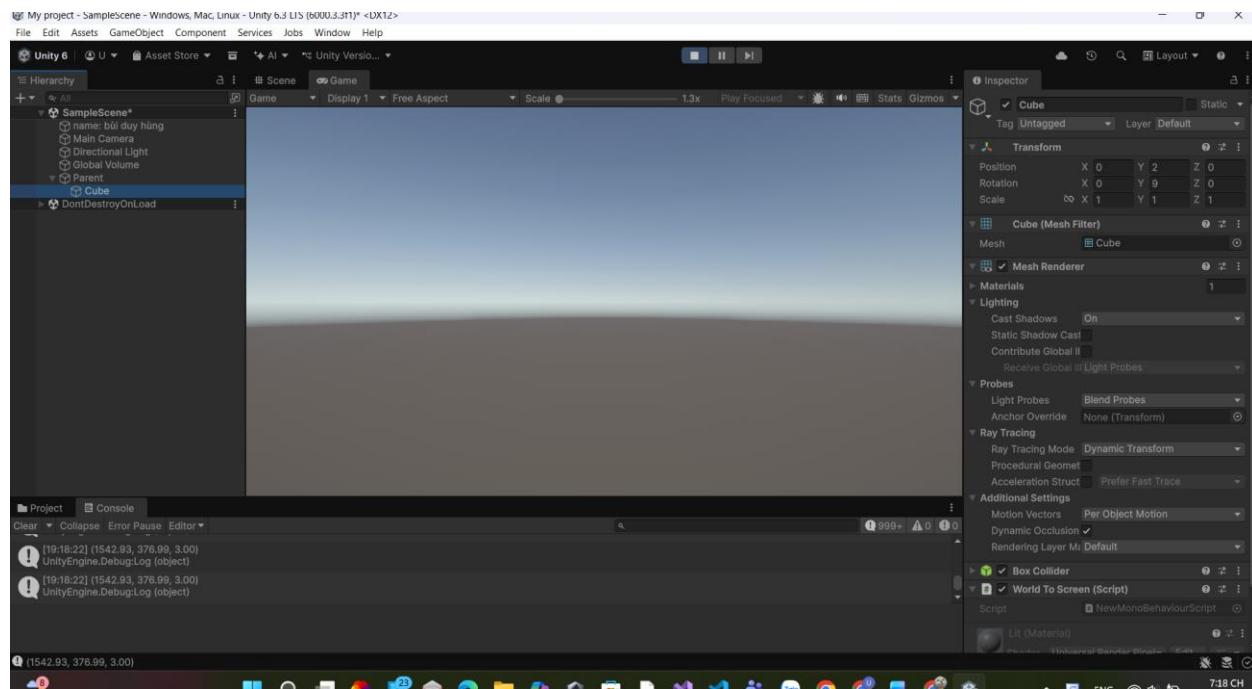
```
{
```

```
    Vector3 screenPos =
        Camera.main.WorldToScreenPoint(transform.position);
```

```
        Debug.Log(screenPos);
    }
}
```



E2. Gắn script vào Cube và chạy game



E3. Ghi lại:

- Screen Position khi Cube ở giữa màn hình:

$$\text{Screen Position} \approx (\text{ScreenWidth} / 2, \text{ScreenHeight} / 2)$$

- Screen Position khi Cube ở góc dưới bên trái:

$$\text{Screen Position} \approx (0, 0)$$

E4. Trả lời:

- Góc tọa độ của Screen Space nằm ở đâu?
- Screen Space khác World Space như thế nào?

Trả lời:

- Góc tọa độ của Screen Space nằm ở góc dưới bên trái màn hình
- Screen Space là tọa độ 2D theo pixel, còn World Space là tọa độ 3D trong không gian thật

Nhận xét cá nhân

Qua bài thực hành này, em hiểu rõ hơn cách Unity sử dụng hệ tọa độ 3D để xác định vị trí object và cách Camera chuyển đổi từ World Space sang Screen Space thông qua việc thao tác trực tiếp với trục X, Y, Z giúp em phân biệt rõ hệ tọa độ trong Unity và mối quan hệ giữa Local Space và World Space. Ngoài ra, việc thay đổi các thông số của Camera giúp em hiểu vì sao kích thước object trên màn hình có thể thay đổi dù vị trí không đổi, về phần chuyển đổi World Space sang Screen Space giúp em nắm được cách Unity hiển thị object từ không gian 3D lên màn hình 2D, phục vụ tốt cho việc học và làm game sau này.