

Họ và tên: Hà Minh Vũ
Mã Sinh Viên: BIT23459
Lớp: 23IT SOFT 2

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

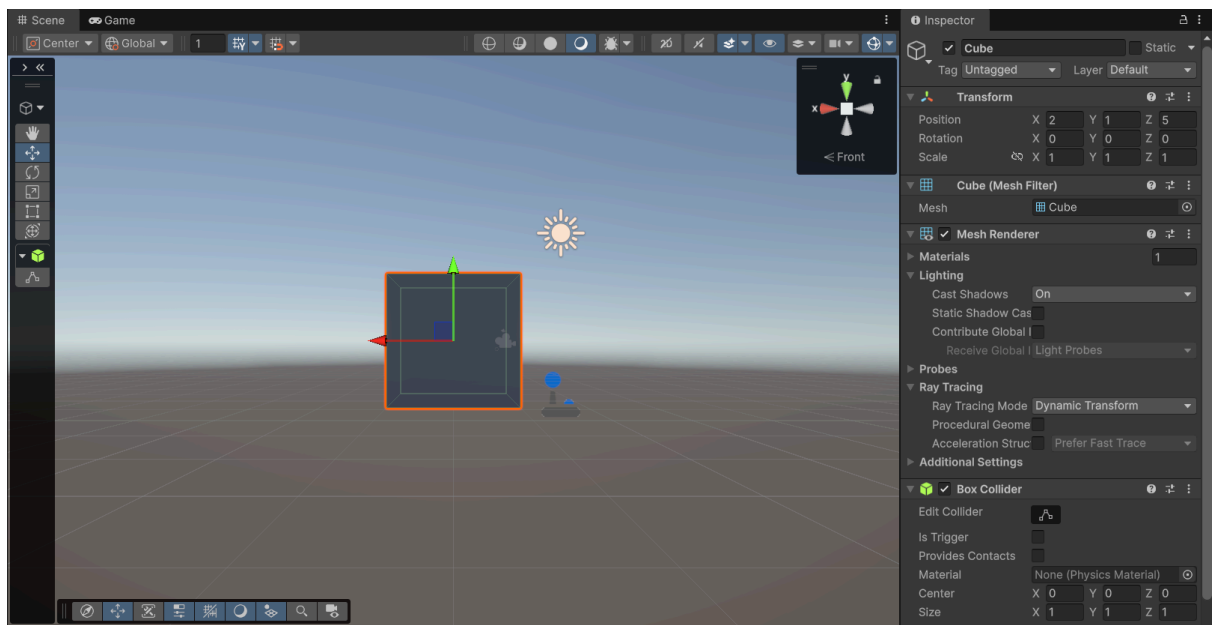
A1. Tạo Cube

- Cube được đặt tại World Position:
 - $X = 2$
 - $Y = 1$
 - $Z = 5$

A2. Gizmos trong Scene View

Khi bật Gizmos, thấy:

- Trục X: màu đỏ
- Trục Y: màu xanh lá
- Trục Z: màu xanh dương



A3. Trả lời câu hỏi

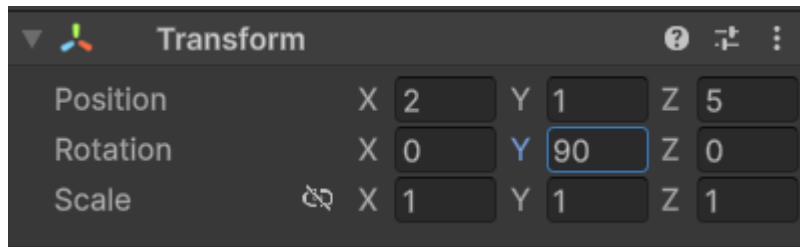
- Trục nào hướng lên trên trong Unity?
Trục Y hướng lên trên

Trục nào hướng về phía Camera?
Trục Z dương hướng ra khỏi màn hình,
Camera mặc định nhìn theo hướng -Z

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

B1. Xoay Cube

- Rotation:
Y = 90



B2. Quan sát và trả lời

-Cube quay theo chiều nào?

Cube quay theo chiều kim đồng hồ khi nhìn từ trên xuống trục Y

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

Trong Left-Handed Coordinate System:

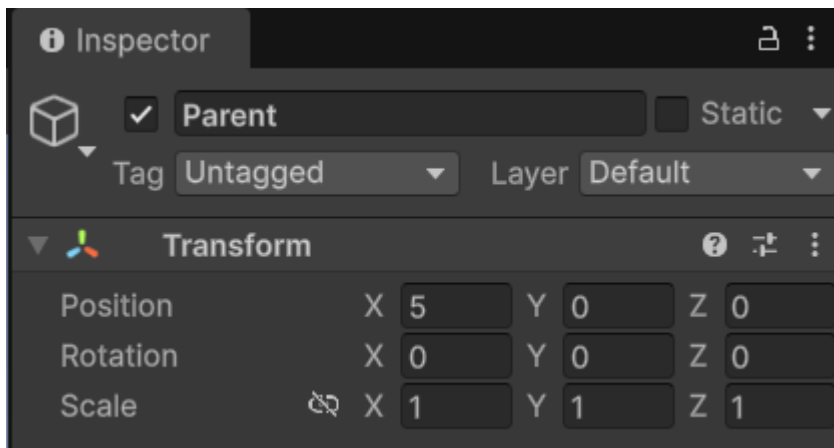
- Ngón cái tay trái: trục X
- Ngón trỏ: trục Y
- Ngón giữa: trục Z

Khi xoay dương quanh trục Y, vật thể quay theo chiều kim đồng hồ, khác với hệ Right-Handed (OpenGL).

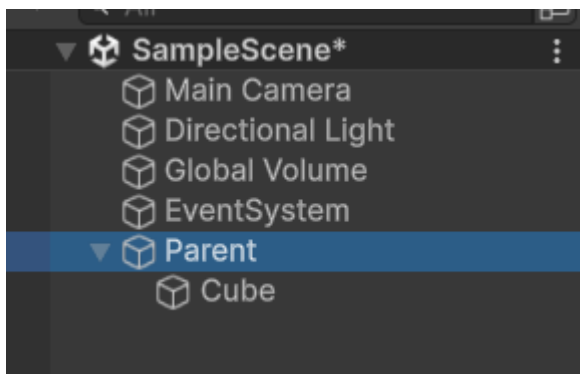
PHẦN C – LOCAL SPACE & WORLD SPACE (25%)

C1. Parent Object

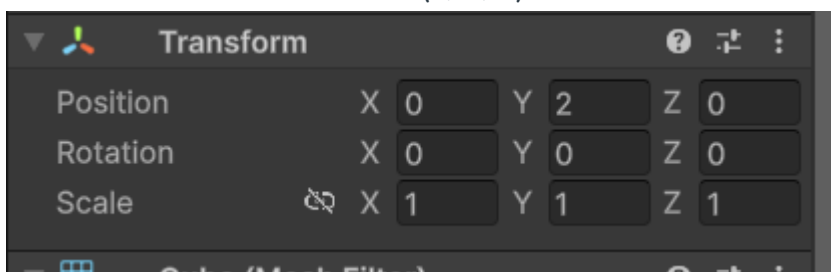
- Tên: Parent
- World Position: (5, 0, 0)



C2. Cube là con của Parent



Local Position của Cube: (0, 2, 0)



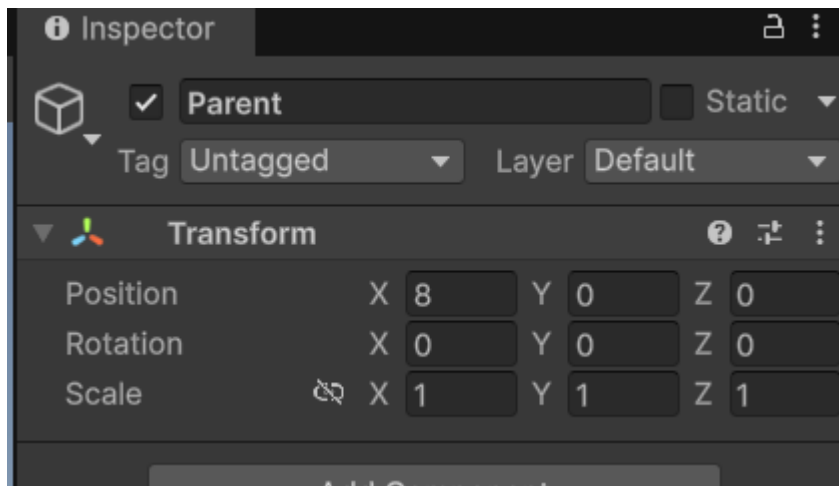
C3. Ghi lại vị trí

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

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

C4. Di chuyển Parent

- Parent → (8, 0, 0)

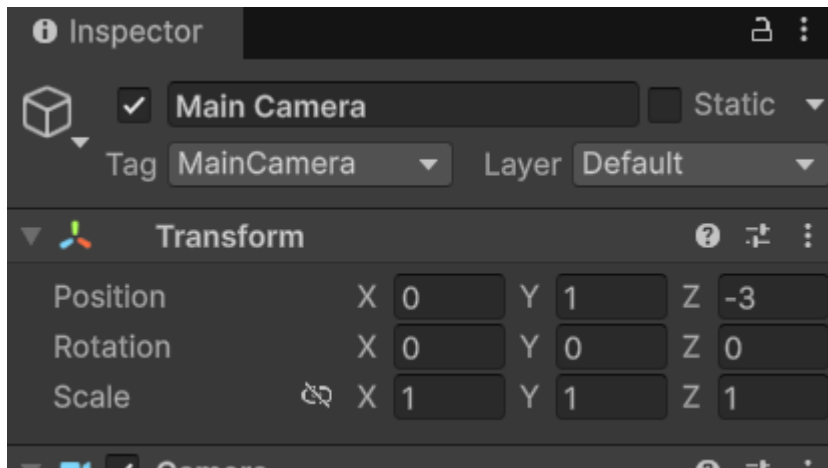


- Local Position của Cube có thay đổi không?
Không thay đổi – vẫn là: (0, 2, 0)
 - World Position của Cube thay đổi thế nào?
Trở thành: (8, 2, 0)
- Vì Cube luôn giữ vị trí tương đối so với Parent

PHẦN D – GRAPHICS PIPELINE (20%)

D1. Di chuyển Camera

- Camera di chuyển dọc trục Z từ -10 đến -3



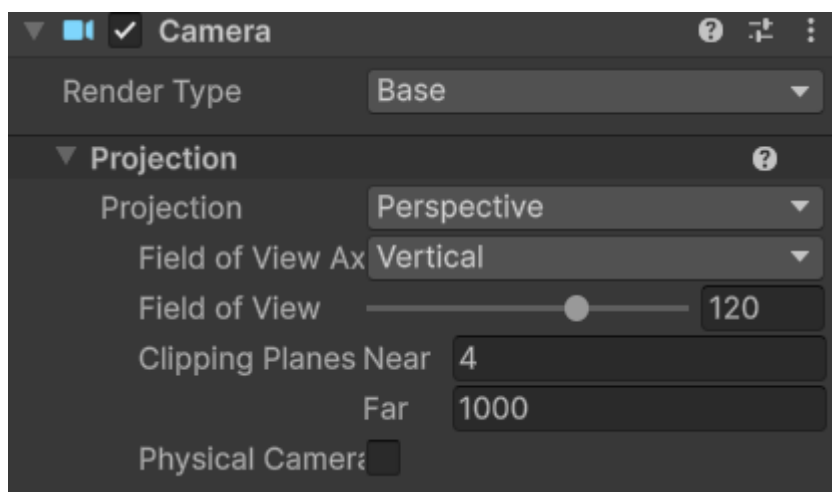
D2. Thay đổi Camera Settings

- Field of View (FOV)

Thay đổi FOV của camera từ 60 thành 120

- Near Clip Plane

Thay đổi Clipping Planes Near từ 0.3 thành 4



Trả lời câu hỏi

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

Do:

- Camera tiến gần → object to hơn
- Camera lùi xa → object nhỏ hơn
- FOV lớn → hiệu ứng wide-angle, object nhỏ
- FOV nhỏ → object trông to hơn

- Đây là bước Projection trong Graphics Pipeline (3D → 2D)

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

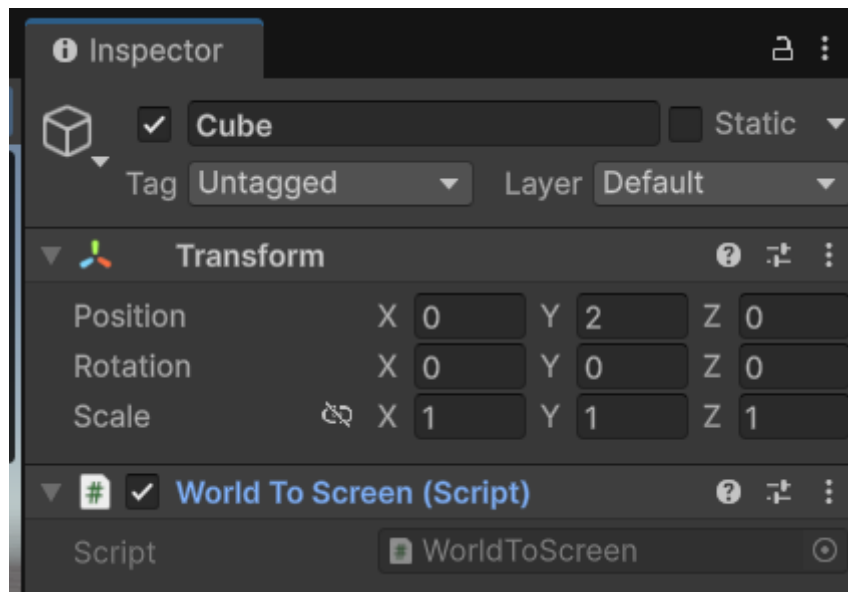
Vì object:

- Nằm ngoài Near Clip Plane
- Nằm ngoài Far Clip Plane
- Nằm ngoài View Frustum của Camera

PHẦN E – SCREEN SPACE (20%)

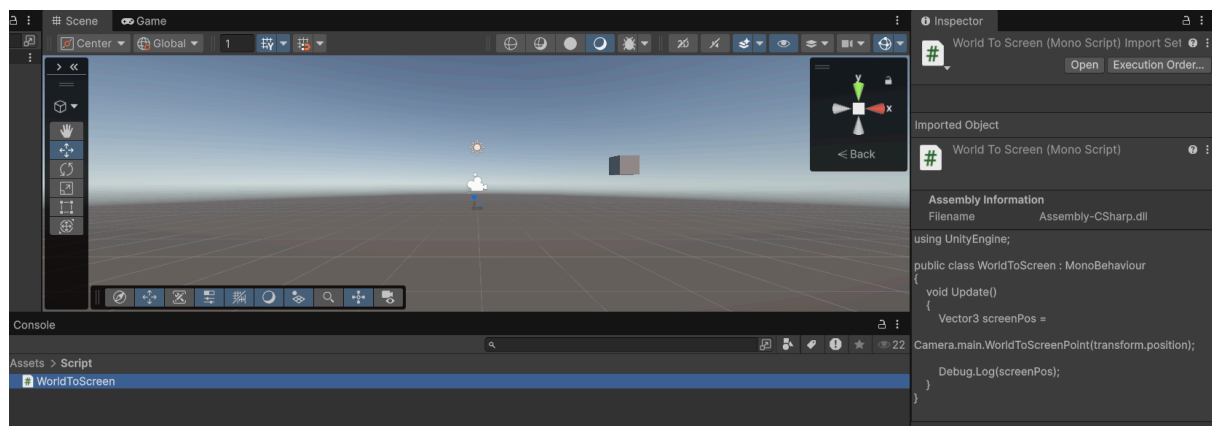
E1 & E2. Script WorldToScreen

Gán Script



Script chuyển đổi:

World Space → Screen Space



E3. Ghi lại Screen Position

(Giả sử độ phân giải 1920×1080)

- ♦ Cube ở giữa màn hình
Screen Position $\approx (960, 540, z)$
- ♦ Cube ở góc dưới bên trái
Screen Position $\approx (0, 0, z)$

(z là khoảng cách từ Camera)

E4. Trả lời câu hỏi

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	Screen Space
Không gian 3D	Không gian 2D
Dùng đơn vị m (unit)	Dùng đơn vị Pixel
Phụ thuộc vào Scene	Phụ thuộc vào độ phân giải

Nhận xét cá nhân:

Thông qua bài thực hành này, em đã hiểu rõ hơn về cách Unity sử dụng hệ tọa độ trong không gian 3D và cách một đối tượng trong World Space được hiển thị lên màn hình 2D thông qua Graphics Pipeline. Trước đây em chỉ đặt và di chuyển object theo cảm tính, nhưng sau bài này em đã hiểu rõ ý nghĩa của từng trục X, Y, Z cũng như hướng mặc định của Camera trong Unity.

Việc thực hành Left-Handed Coordinate System giúp em phân biệt rõ sự khác nhau giữa Unity và một số hệ đồ họa khác. Khi xoay object, có thể giải thích được vì sao vật thể quay theo chiều đó thay vì chỉ quan sát kết quả. Điều này giúp em tự tin hơn khi làm việc với Rotation và Camera.

Phần Local Space và World Space giúp em hiểu được mối quan hệ giữa Parent và Child Object. em nhận ra rằng Local Position 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 phức tạp hoặc nhân vật trong game.

Ngoài ra, phần thực hành Camera và Screen Space giúp em hiểu rõ vai trò của Field of View, Clip Plane và cách chuyển đổi từ World Space sang Screen Space

bằng script. em thấy rõ mối liên hệ giữa vị trí object trong Scene và giá trị pixel hiển thị trên màn hình, điều này rất hữu ích cho việc làm UI, minimap hoặc hiển thị thông tin trong game.

Nhìn chung, bài thực hành này giúp em nắm vững kiến thức nền tảng về không gian tọa độ trong Unity, tạo tiền đề quan trọng cho các bài học và dự án game 3D sau này.