**BÁO CÁO THỰC HÀNH: MATHEMATICS IN UNITY**

**Họ và tên: Phạm Khắc Đô**

**Mã sinh viên: BCS230024**

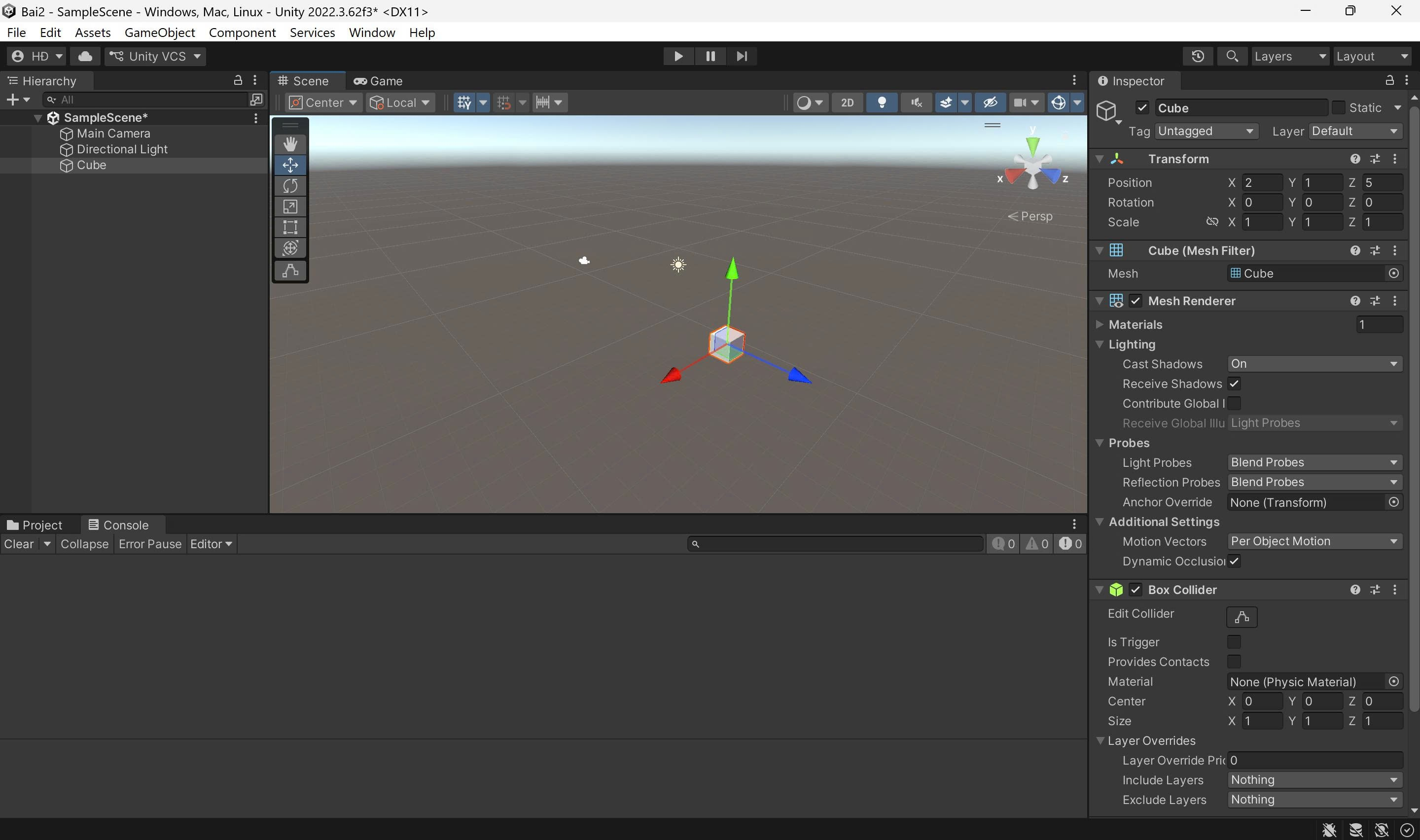
**Lớp: 23CS2**

**Môn học: Lập trình Game**

**PHẦN A:**

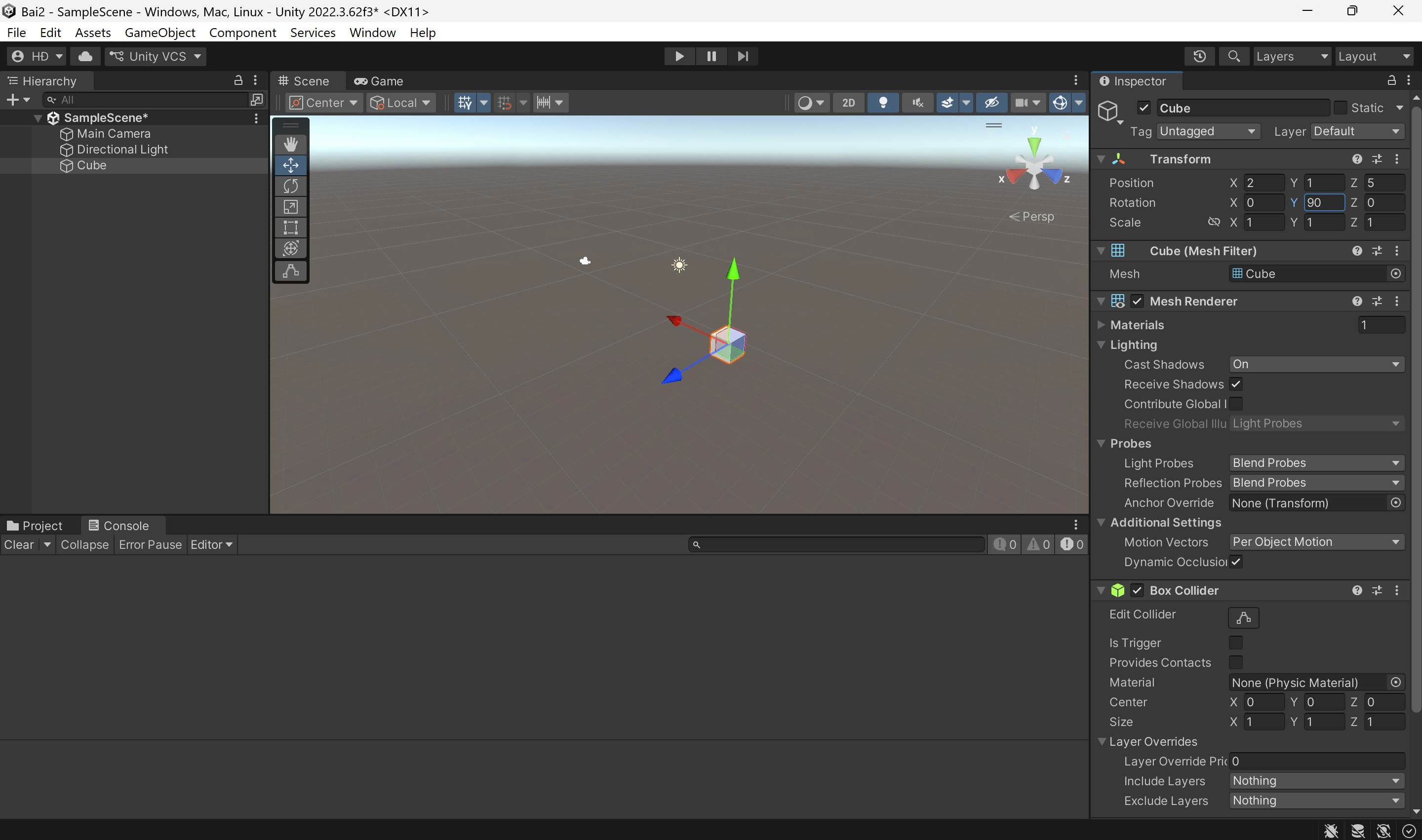
**Trục nào hướng lên trên trong Unity?**

* + **Trả lời:** Đó là trục **Y** (màu xanh lá cây).
* **Trục nào hướng về phía Camera?**
  + **Trả lời:** Đó là trục **Z** (màu xanh dương). Trong hệ tọa độ Unity, trục Z dương mặc định hướng "sâu" vào trong không gian, đối diện với hướng nhìn mặc định của Camera.



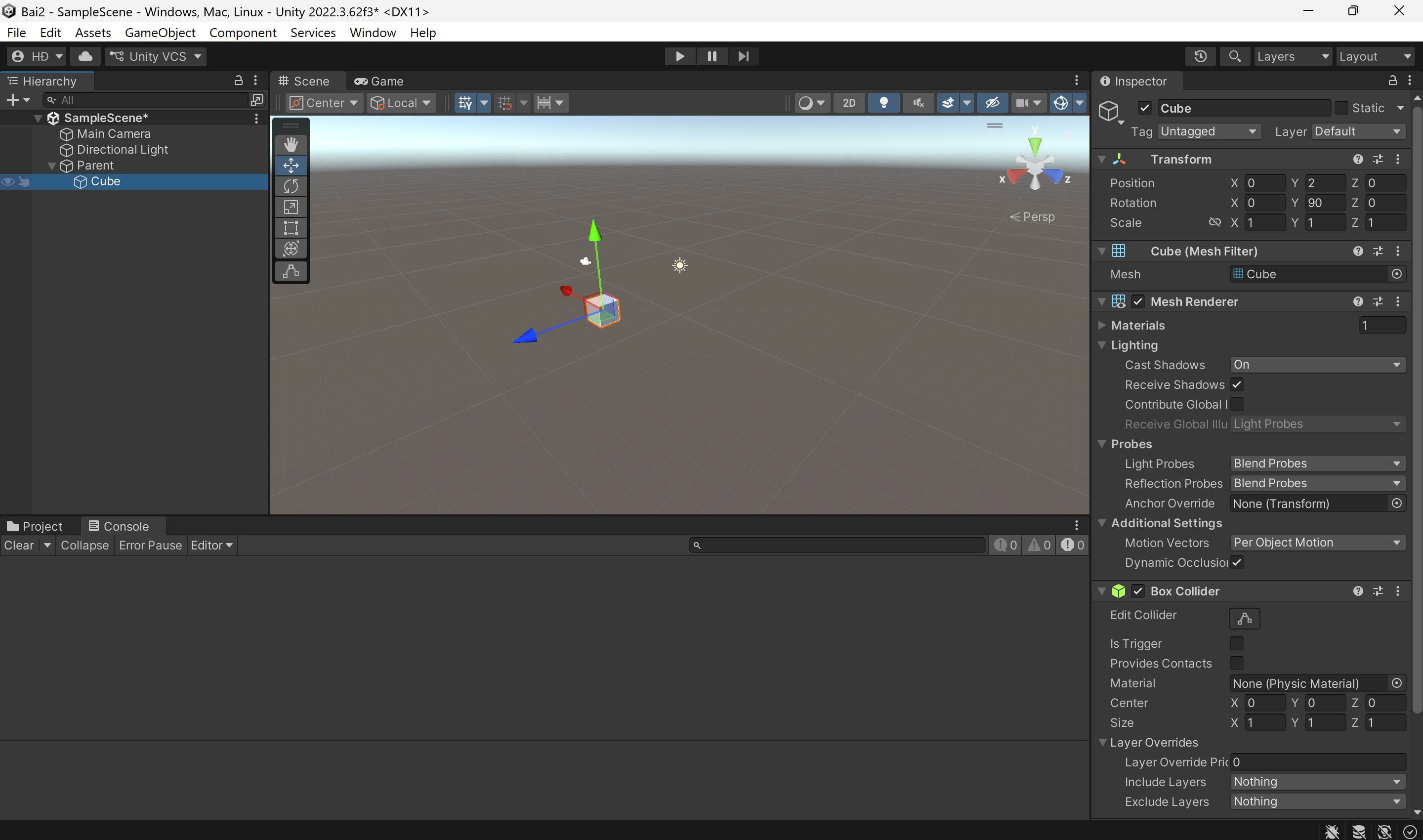
**PHẦN B:**

* **Cube quay theo chiều nào?**
  + **Trả lời:** Khi nhập $Y = 90$, Cube quay theo **chiều kim đồng hồ** nếu nhìn từ trên đỉnh trục Y xuống.
* **Điều này thể hiện Unity sử dụng Left-Handed Coordinate System như thế nào?**
  + **Trả lời:** Unity sử dụng hệ tọa độ bàn tay trái. Theo quy tắc này, nếu bạn nắm bàn tay trái sao cho ngón cái trỏ theo hướng dương của trục xoay (trục Y hướng lên), thì chiều khum của các ngón tay còn lại chính là chiều xoay dương (chiều kim đồng hồ).

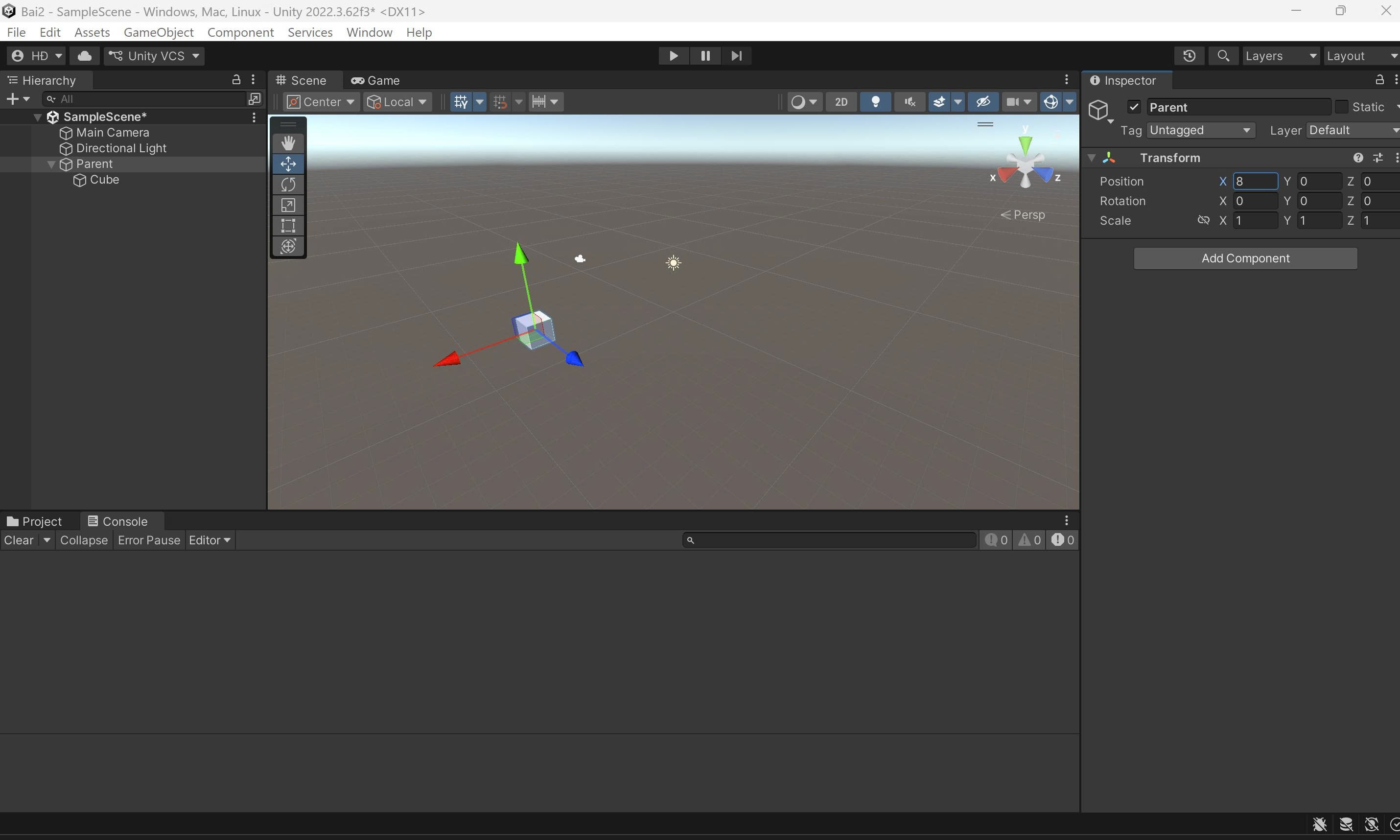


**PHẦN C:**

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

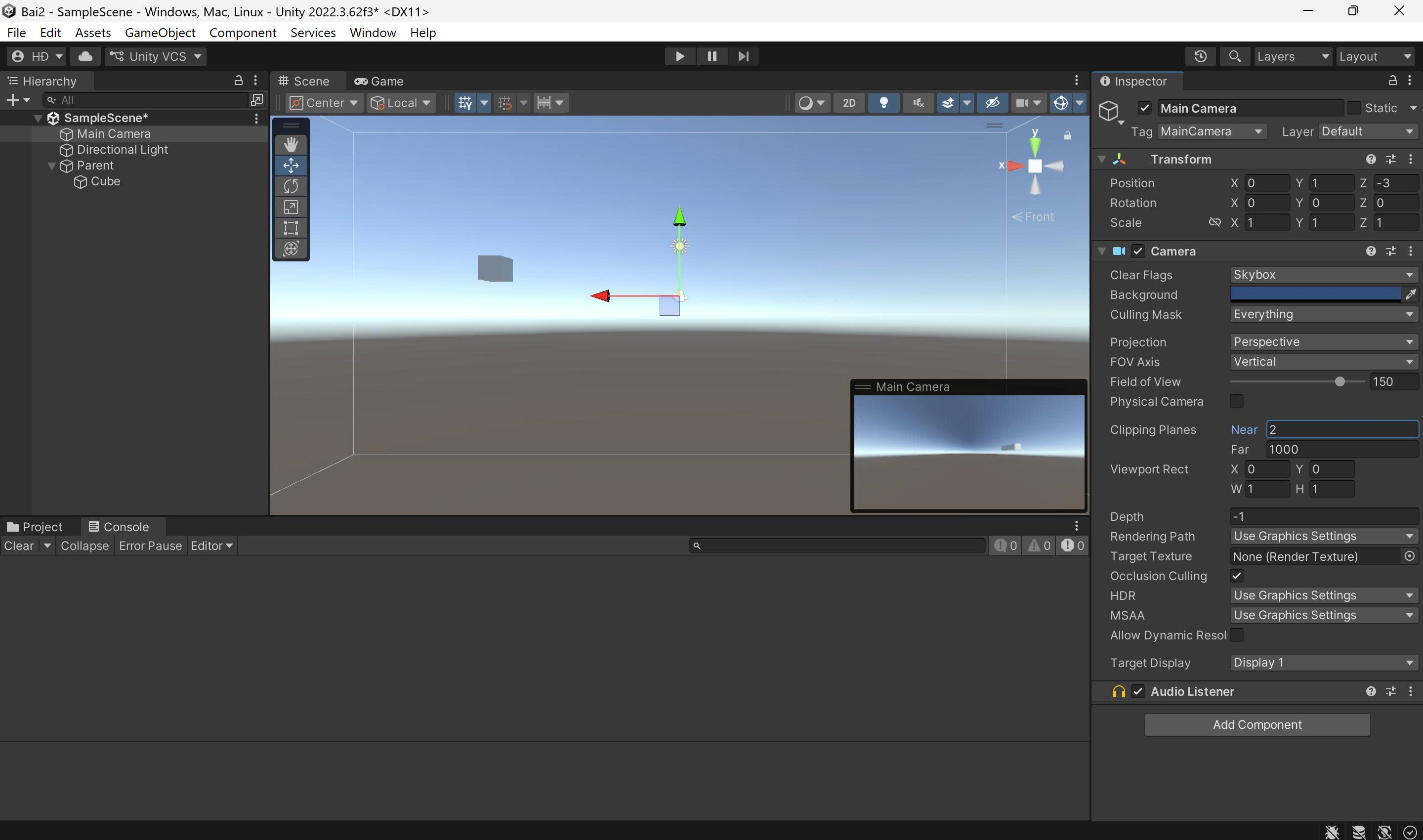
****

* **Local Position của Cube có thay đổi không?**
  + **Trả lời:** **Không thay đổi**. Local Position vẫn là $(0, 2, 0)$ vì vị trí của nó so với vật cha (Parent) không hề bị tác động.
* **World Position của Cube thay đổi như thế nào?**
  + **Trả lời:** World Position thay đổi từ $(5, 2, 0)$ sang **$(8, 2, 0)$**. Trong Unity, vị trí thế giới của vật con được tính bằng tổng vị trí của vật cha và vị trí local của chính nó: $World = Parent(8,0,0) + Local(0,2,0) = (8,2,0)$.



**PHẦN D:**

* **Vì sao object trông to hoặc nhỏ hơn dù không đổi vị trí?**
  + **Trả lời:** Do thay đổi thông số **Field of View (FoV)** của Camera. FoV nhỏ giống như việc "zoom" ống kính lại gần, khiến object chiếm nhiều diện tích hơn trong vùng nhìn (Frustum) và hiển thị to hơn trên màn hình.
* **Vì sao object có thể biến mất khỏi màn hình?**
  + **Trả lời:** Có hai lý do chính:
    1. Vật nằm ngoài vùng nhìn thấy của Camera (Frustum).
    2. Vật nằm gần hơn khoảng cách **Near Clip Plane** (mặt phẳng cắt gần), khiến Camera không thực hiện vẽ (render) đối tượng đó.



**PHẦN E:**

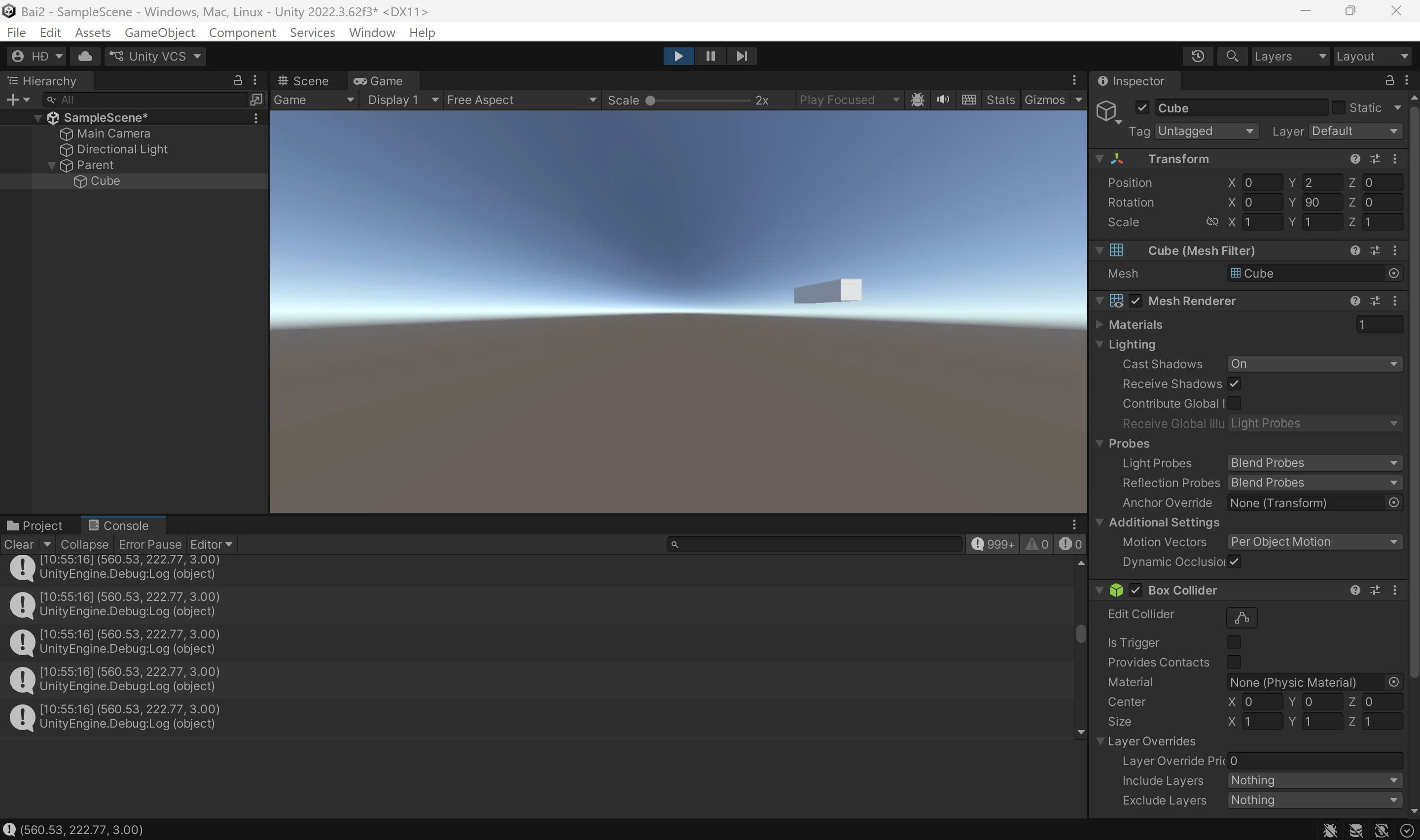
**1. Gốc tọa độ của Screen Space nằm ở đâu?**

**Trong Unity, Screen Space (Không gian màn hình) là không gian 2D tương ứng với màn hình hiển thị .**

* **Vị trí gốc (0, 0): Nằm ở góc dưới cùng bên trái của màn hình.**
* **Trục tọa độ:**
  + **Trục X tăng dần sang bên phải.**
  + **Trục Y tăng dần lên phía trên.**
* **Giới hạn: Tọa độ tối đa là (Screen.width, Screen.height góc trên cùng bên phải sẽ là (1920, 1080).**

**2. Sự khác biệt giữa Screen Space và World Space**

| **Đặc điểm** | **World Space (Không gian thế giới)** | **Screen Space (Không gian màn hình)** |
| --- | --- | --- |
| **Bản chất** | Là không gian 3D nơi các vật thể tồn tại. | Là không gian 2D để hiển thị hình ảnh lên màn hình. |
| **Gốc tọa độ** | Nằm ở tâm của "thế giới" (0, 0, 0). Trong ảnh, khối Cube đang ở vị trí Y = 2. | Nằm ở góc dưới bên trái màn hình điện thoại/máy tính của người chơi. |
| **Đơn vị đo** | Tính bằng **Meter** (Mét). | Tính bằng **Pixel** (Điểm ảnh). |
| **Sự thay đổi** | Cố định trong thế giới game. | Thay đổi tùy thuộc vào độ phân giải màn hình của thiết bị. |



**NHẬN XÉT CÁ NHÂN**

**Qua bài tập Unity này, em đã học được những nội dung quan trọng sau:**

1. Hiểu rõ hơn về hệ trục tọa độ Left-Handed trong Unity, đặc biệt là quy tắc xoay đối tượng theo trục Y.
2. Nắm được cách xác định hướng xoay của đối tượng thông qua việc quan sát Cube xoay khi thay đổi giá trị Rotation.
3. Phân biệt rõ Local Space và World Space, nhất là khi làm việc với các đối tượng có quan hệ cha – con (Parent – Child).
4. Biết cách quản lý vị trí và chuyển động của nhóm đối tượng một cách khoa học hơn trong không gian 3D.
5. Hiểu được nguyên lý chuyển đổi tọa độ từ không gian 3D sang màn hình 2D thông qua script WorldToScreen.cs.
6. Rèn luyện kỹ năng đọc lỗi, sửa lỗi và hiểu cấu trúc code C# trong Unity.