

# Bài 1: Hệ thống UI trong Unity




# MỤC TIÊU

## B. Tạo các đối tượng cơ bản

1. Game Object
2. Sprite
3. Animation và điều khiển hành động nhân vật
4. Prefab
5. Script và một số xử lý cơ bản
6. Thành phần vật lý và xử lý va chạm

## 7. Hệ thống UI

8. Sử dụng Particle System
  9. Chuyển đổi màn chơi
  10. Sound
  11. Design Pattern trong Game
- 

# Hệ thống UI trong Unity

- . Giao diện người dùng (User Interface – UI) là thành phần không thể thiếu đối với bất kỳ game nào.
- . UI cung cấp các thông tin trực quan cần thiết cho người chơi, giúp người chơi có cái nhìn toàn diện về các khả năng của mình (thời gian, điểm, “máu”, ...) và có chiến thuật thích hợp để vượt qua được các thử thách trong game.
- . Việc thiết kế giao diện người dùng đơn giản là sử dụng các assets có sẵn (hình ảnh, font chữ, các hiệu ứng, ...) và sắp xếp chúng theo một bố cục được Designer thiết kế.
- . Các assets này có thể được tìm thấy trên các website, Assets Store của Unity, hoặc do chính các Designer, Artist trong dự án thiết kế (thường gặp ở những dự án lớn hoặc vừa).
- . Các thành phần cơ bản trong thiết kế UI bao gồm Canvas, Text, Image, Button, ....

# Hệ thống UI trong Unity- Canvas

- . Canvas có thể hiểu là một vùng riêng cho phép chứa đựng thành phần giao diện người dùng (UI). Là một game object có một component là Canvas, và tất cả các UI phải là con của một Canvas.
- . Các thành phần UI khác khi được khởi tạo bắt buộc phải nằm trong một canvas.
- . Khi khởi tạo một thành phần UI (Text, Image, ...), Unity sẽ tự động tạo ra một canvas nếu chưa tồn tại một canvas nào trong Scene.
- . Để khởi tạo một canvas, trong cửa sổ Hierarchy, ta chọn Create → UI → Canvas. Các đối tượng UI khác cũng được khởi tạo tương tự.
- . Các đối tượng con của một Canvas sẽ được render theo thứ tự từ trên xuống dưới trong cửa sổ Hierarchy. Đối tượng nào ở trên sẽ được render trước và có thể bị che khuất bởi đối tượng phía dưới.

# Hệ thống UI trong Unity- Canvas

## Các chức năng quan trọng của Canvas

- Render mode: Có 3 tùy chọn hiển thị canvas:
  - Screen Space – Overlay. Canvas sẽ được vẽ lên layer cao nhất của màn hình và nằm trên mọi game object khác. Canvas với render mode này hoàn toàn không phụ thuộc vào camera.
  - Screen Space – Camera. Đối với mode này, ta cần chỉ định một camera cho canvas, nó sẽ được render theo camera. Nếu như không có camera được chỉ định thì canvas và các thành phần bên trong sẽ không được render.
  - World Space. Với tùy chọn này, đối tượng canvas sẽ được xem như một game object thông thường. Tùy chọn này sử dụng event camera thay vì render camera. Ngoài các chức năng như render camera, event camera còn có thêm chức năng bắt sự kiện, dựa trên thứ tự render, tọa độ z, ... của các đối tượng UI.

# Hệ thống UI trong Unity- Canvas

## Các chức năng quan trọng của Canvas

- . Render mode:
- . Đối với các tùy chọn render theo Screen Space, Unity cung cấp tính năng Pixel Perfect, tăng khả năng hiển thị sắc nét và khử vết mờ.



# Hệ thống UI trong Unity- Canvas

## Các chức năng quan trọng của Canvas

### + Rect Transform

- Tương tự như thành phần Transform trong các game object khác. Rect Transform được sử dụng để xác định kích thước, vị trí và luân chuyển trong hệ thống điều khiển giao diện người dùng
- Đối với các tùy chọn render mode Screen Space – Overlay và Screen Space – Camera, thành phần Rect Transform sẽ được khoá lại và không thể tùy chỉnh. Canvas sẽ điều chỉnh các thông số một cách tự động để phù hợp với độ phân giải màn hình game.

# Hệ thống UI trong Unity - Canvas

## Các chức năng quan trọng của Canvas


### Graphic Raycast

- Hỗ trợ bắt sự kiện.
- Khi nhận được tín hiệu (mouse click, touch, ...), một tia nhìn tại vị trí tương tác sẽ được tạo ra. Bằng cách này, chúng ta dễ dàng xác định được đối tượng mà người chơi muốn tương tác, thông qua tọa độ z của đối tượng.



# Hệ thống UI trong Unity- Rect Transform

## Rect Transform:

- . Rect Transform là một thành phần mới mà được dùng để thay thế cho thành phần Transform trên tất cả các UI mới mà Unity bổ sung ở phiên bản 4.6.
  - . Thành phần Transform của một game object trong Unity có 3 yếu tố là position – căn chỉnh kích thước, rotation – điều chỉnh độ xoay và scale dùng căn chỉnh tỷ lệ của một đối tượng trong Scene.
- 

# Hệ thống UI trong Unity- Rect Transform

- . Position cho thấy khoảng cách bằng pixel từ các điểm anchor đến pivot, dọc theo trục X và Y, Z là khoảng cách dọc trục Z local (chiều sâu) và thường được để ở giá trị 0. Width và Height cho biết kích thước của UI theo pixel.
- . Rotation: thì xoay object, thường thì trong 2D mình chỉ sử dụng xoay trục Z.
- . Scale, bạn phải hiểu được sự khác nhau giữa scale và size. Size thay đổi kích thước của nó, không ảnh hưởng đến UI object bên trong, còn scale sẽ canh chỉnh lại tỷ lệ của object và tất cả các thành phần con. Mình thấy resize một yếu tố UI thì tốt hơn là scale nó. Scale thường được sử dụng cho các hiệu ứng động hay các mục đích đặc biệt khác

# Hệ thống UI trong Unity- Rect Transform

## Rect Transform:

- . Rect Transform thể hiện một hình chữ nhật được xác định bởi chiều rộng và chiều cao liên quan đến một điểm tâm của nó (gọi là pivot).
- . Pivot là điểm trụ của object, thông thường mặc định sẽ là ở tâm của object, nếu bạn xoay UI thì nó sẽ xoay quanh điểm tâm.
- . Anchor(điểm neo): một Rect Transform có thể được anchored tới đối tượng cha của nó, nếu cha của nó cũng có một thành phần Rect Transform.
- . Anchor cho phép chúng ta di chuyển hoặc kéo dài UI dựa trên position và size của thành phần Rect Transform của UI cha. Cần phải chú ý rằng tất cả các UI sau cùng cũng phải trở thành là một con của một Canvas.
- . Theo một cách nào đó thì UI con sẽ được neo vào UI cha dựa vào thành phần Rect Transform, tất cả UI đều có Rect Transform.

# Hệ thống UI trong Unity- Rect Transform


- . Anchor: Trong scene, neo được nhận biết bởi bốn hình tam giác nhỏ, mà theo mặc định, được nhóm lại với nhau ở trung tâm của thành phần rect transform UI cha.
- . Cụ thể về Anchor:
  - Rect transform có thể được neo vào parent object nếu parent object cũng có một thành phần Rect Transform.
  - Trong scene view: neo được đại diện bởi bốn hình tam giác nhỏ, mỗi tam giác đại diện cho một neo. Và mỗi neo kết hợp với một góc của rect transform tương ứng. (chụp hình)
  - Neo có thể được di chuyển bằng cách kéo thả trên Gizmo tam giác, Nhấp ở trung tâm của một nhóm các hình tam giác sẽ di chuyển các hình tam giác như là một nhóm. Nhấp vào hình tam cá nhân sẽ di chuyển một neo đậu tại một thời điểm.
- . Neo có một mối quan hệ cố định với rect transform của nó, nó đang được neo. Neo có mối quan hệ linh hoạt với Rect Transform của object cha, mà họ đang neo đậu.

# Hệ thống UI trong Unity- Rect Transform

## Vị trí Anchor

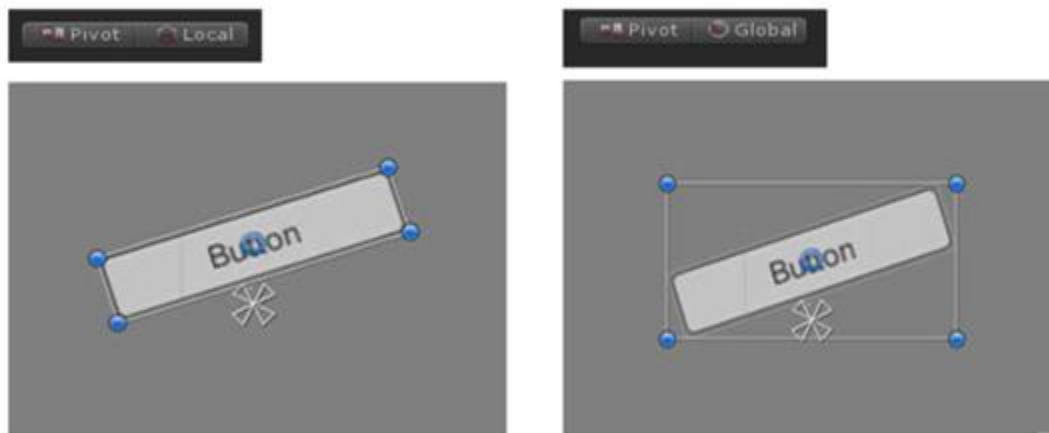
- Khi bạn thiết lập một neo, khoảng cách giữa neo và góc liên kết của rect transform là một giá trị cố định.
- Vị trí của neo là ở bên trong Rect Transform của đối tượng cha, tuy nhiên nó có thể cân đối tùy theo kích thước object cha và bị ảnh hưởng bởi thay đổi kích thước trong các Rect Transform của đối tượng cha.
- Vị trí của neo cân đối với đối tượng cha là một số phần trăm kích thước của đối tượng cha dọc theo trục X và Y, (chụp hình Anchor).

# Hệ thống UI trong Unity- Rect Transform

- . Để thực hiện các thiết lập nhanh chóng và dễ dàng Rect Transform bao gồm cửa sổ neo Presets.
  - . Điều này có thể được truy cập bằng cách nhấp vào Nút neo Presets.
  - . Trong Scene, để thao tác với một UI, cách tốt nhất là sử dụng Rect Tool, với phím tắt là Shift + T.
  - . Với Rect tool được chọn có thể để di chuyển, thay đổi kích thước và xoay bất kỳ UI nào bằng cách giữ chuột và kéo, Giữ phím Shift sẽ buộc các UI thay đổi kích thước tương ứng.
  - . Kéo chuột ở gần bất kỳ góc nào của hình chữ nhật để xoay.
- 

# Hệ thống UI trong Unity- Rect Transform

- . Global and Local space trên thanh công cụ).



- . Khi chỉnh sửa một UI trên local space, hình chữ nhật và xử lý phải sắp thẳng hàng, (phủ vừa đủ) với UI.
- . Còn trong global space, hình chữ nhật và xử lý phải được sắp thẳng hàng với tổng thể (hay tổng thể thôi) trên toàn cầu, và hình chữ nhật cho thấy giới hạn của UI. Điều này phát hiện rõ nhất khi chúng ta xoay một UI.

# Hệ thống UI trong Unity- Text

- . UI Text được sử dụng để hiển thị các thông tin trên màn hình như Score, Lives, Times, ...
- . Một số game sử dụng các texture riêng để hiển thị thông tin thay cho text, tuy nhiên text vẫn là lựa chọn phổ biến hơn vì tính đơn giản và dễ thao tác.
- . Để khởi tạo một text, trong cửa sổ Hierarchy, ta chọn Create → UI → Text.



# Hệ thống UI trong Unity- Text

**✚** Sau khi khởi tạo một đối tượng Text, cửa sổ Inspector có dạng như sau:

- Rect Transform: quản lý vị trí, kích thước, góc quay, ...của Text.
- Text: lưu trữ chuỗi ký tự cần hiển thị ra màn hình.
- Unity hỗ trợ một số chức năng để tùy biến chuỗi như font, kiểu chữ (thường, in đậm, in nghiêng, ...), cỡ chữ, màu chữ, ...
- Tùy chọn Best Fit sẽ tự động điều chỉnh kích thước font chữ phù hợp với kích thước được quy định trong Rect Transform.

THANKS FOR  
WATCHING!

