

Lab 6 (4 tiết): UI Game



MỤC ĐÍCH

Bài lab này giúp sinh viên tìm hiểu và xây dựng menu cho game dựa trên các thành phần UI trên Unity.

YÊU CẦU

- Sinh viên phải làm xong Lab 5 trước khi làm lab này, dự án trong Lab 5 sẽ được dùng lại cho Lab 6 (dự án của Lab 5 đã để sẵn trong thư mục Lesson 6).
- Sinh viên đọc toàn bộ phần nội dung và thực hiện theo hướng dẫn.
- Thư mục đi kèm bài lab: Lesson 6
- Sau đó thực hiện các bài tập tương ứng.

NỘI DUNG

1. Giới thiệu

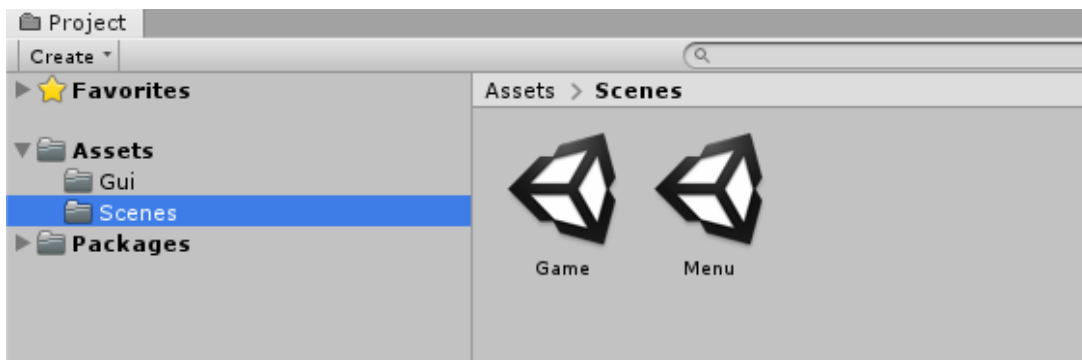
Trong bài lab này, sinh viên sẽ tạo ra một giao diện menu để vào game như hình sau:



Tương tự như các lab trước, sinh viên sử dụng thư mục Lesson 6 đính kèm.

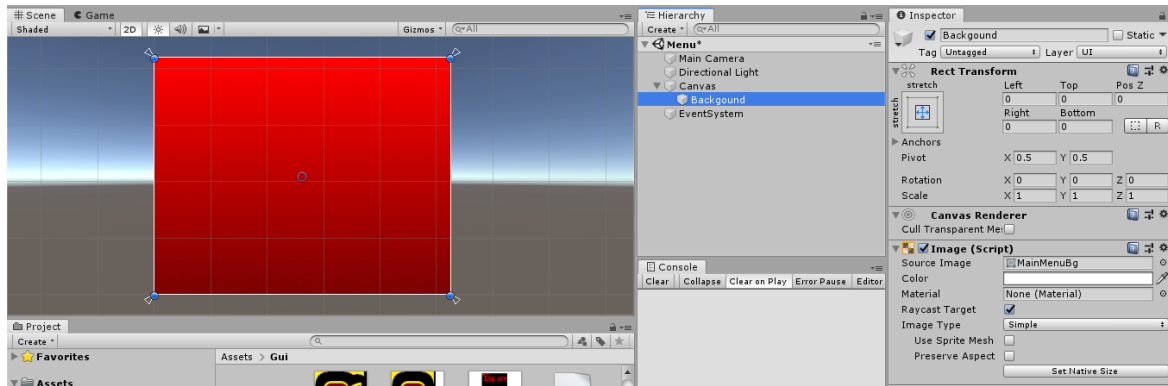
2. Tạo một Menu Scense

Kéo thư mục Gui đi kèm trong thư mục Lesson 6 vào dự án, tạo một Scense mới, đặt tên là Menu, như vậy ở đây có 2 Scense là Game và Menu:

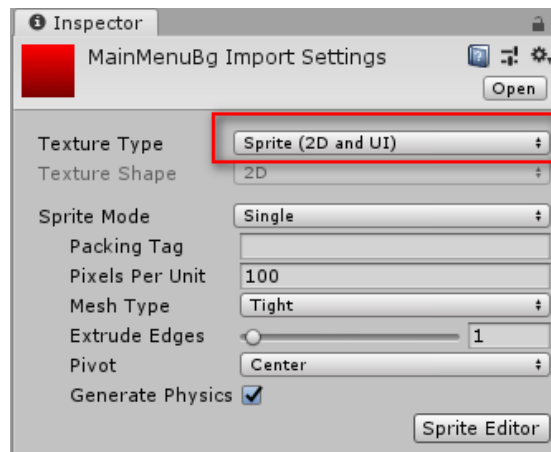


Mở scense Menu bằng cách nhấp đôi chuột lên nó. Tạo một Panel, đổi tên thành Background, gán hình nền *MainMenuBg* (đã chuyển thành sprite) cho thuộc tính Source Image như sau:

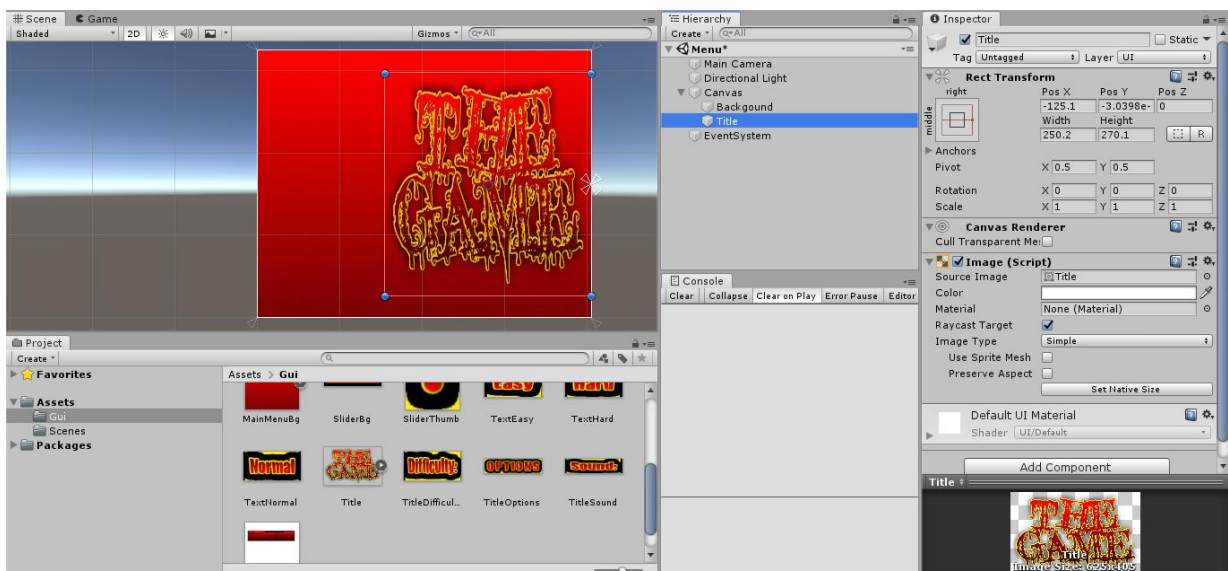
THỰC HÀNH LẬP TRÌNH GAME



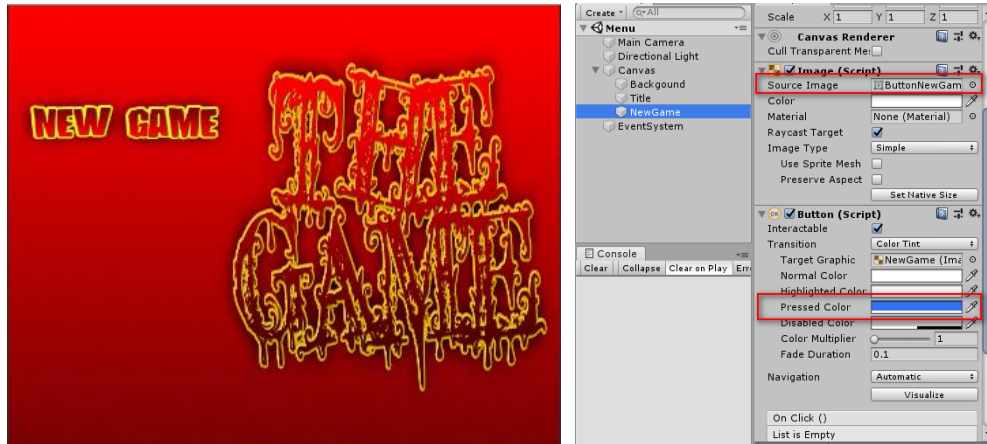
Lưu ý: Thuộc tính *Source Image* chỉ nhận Sprite, do vậy, trước khi gán hình nền, phải chuyển đổi hình thành Sprite bằng cách thiết lập thuộc tính Texture Type là Sprite 2D and UI, nhấn Apply để áp dụng.



Thêm thành phần Image vào Canvas, đổi tên thành Title, và gán hình (sprite) như sau:



Bây giờ chúng ta sẽ thêm các button vào màn hình giao diện. Tạo Button mới có tên là NewGame vào trong Canvas, gán hình ảnh (ButtonNewGame), chỉnh màu cho thuộc tính Pressed Color. Chạy, nhấn chuột vào Button để xem kết quả:



Thực hiện các thao tác tương tự cho các Button Options và Quit Game:

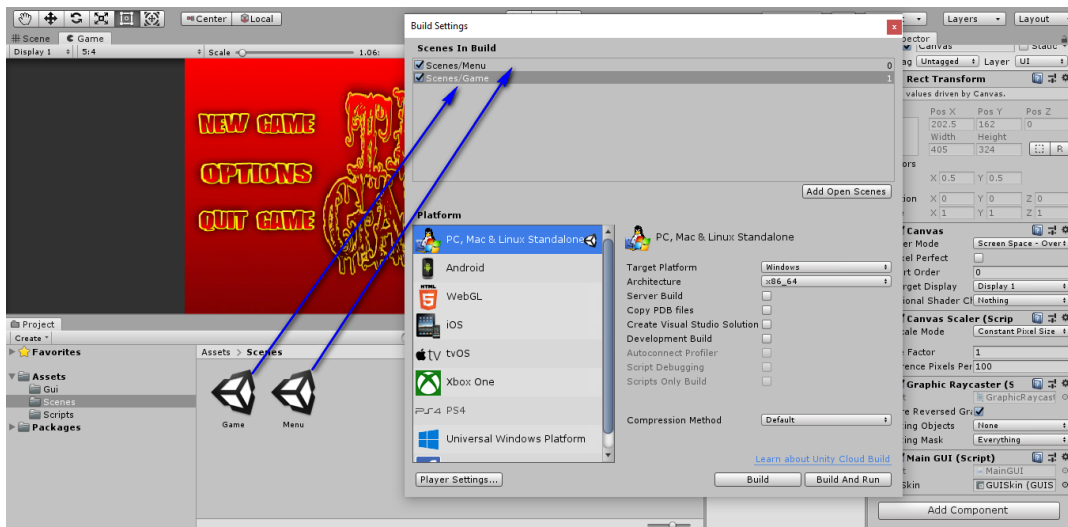


Lưu ý: Trong Unity, có một công cụ làm giao diện khá thú vị là GUISkin, các bạn có thể tìm hiểu thêm tại đây:

<https://www.youtube.com/watch?v=0padxbNDDkQ>

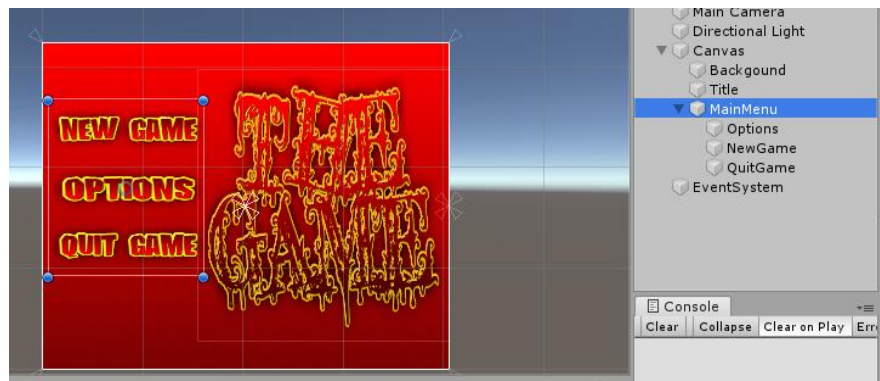
3. Xử lý New Game và Quit Game

Khi người dùng nhấn NewGame, hệ thống sẽ chạy qua scene Game. Vào File, chọn Build Settings, tìm tới thư mục Scenes trong Project, kéo các Scene vào phần Scenes In Build sao cho Menu đứng trước Game:

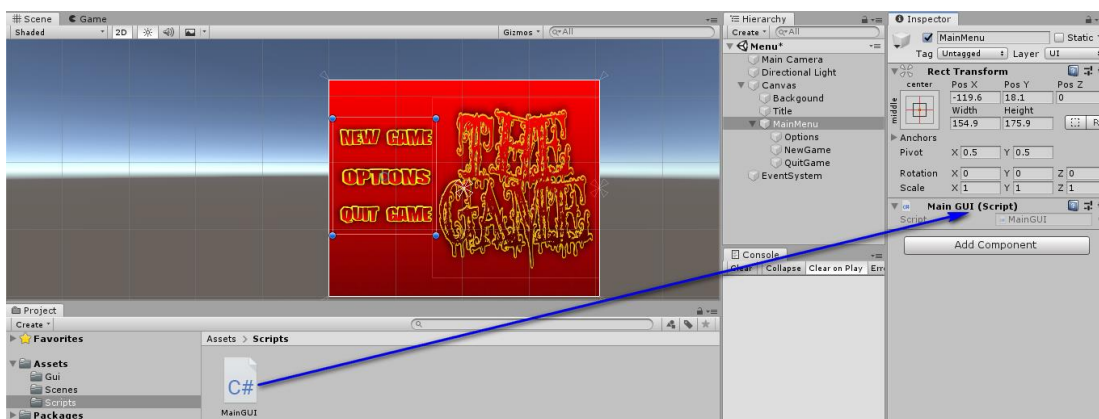


Lưu ý: Thứ tự của Menu và Scene lần lượt là 0 và 1.

Click phải lên Canvas, chọn Create Empty, tạo một đối tượng rộng tên là MainMenu, dùng chuột mở rộng để 3 button vừa tạo nằm trong khung của đối tượng vừa tạo (hình sau). Trên Hierarchy, cho 3 button thành đối tượng con của MainMenu:



Tạo một script có tên MainGUI và gán làm thành phần của MainMenu:



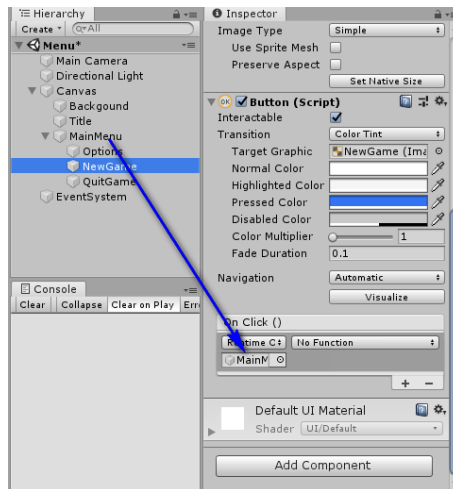
Trong lớp MainGUI tạo một phương thức dạng public có tên NewGame(), viết hàm như sau để gọi một Scene mới:

```
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class MainGUI : MonoBehaviour
5 {
6     public void NewGame()
7     {
8         SceneManager.LoadScene(1);
9     }
10 }
```

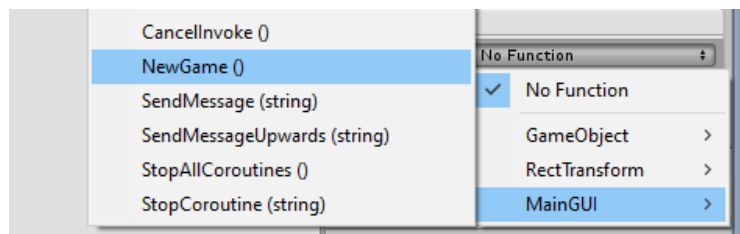
Lưu ý: Tham số 1 là index của scene Game. Ta có thể viết cách khác như sau:

```
SceneManager.LoadScene("Game");
```

Lưu lại, ra màn hình chính, trên Hierarchy, chọn button NewGame, trong phân On Click() kéo đối tượng MainMenu vào ô Object như sau:



Trong ô No Function, click chọn MainGUI, chọn NewGame():



Nhấn Play và bấm chuột vào New Game để xem kết quả.

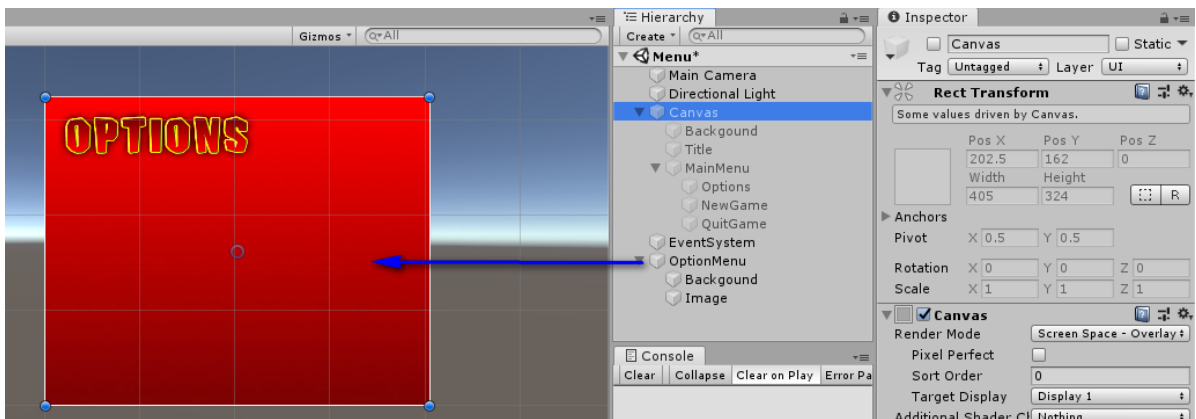
Thực hiện tương tự với nút Quit Game, mã nguồn để thoát một Game như sau:

```
// Thoát Game
0 references
public void QuitGame() => Application.Quit();
```

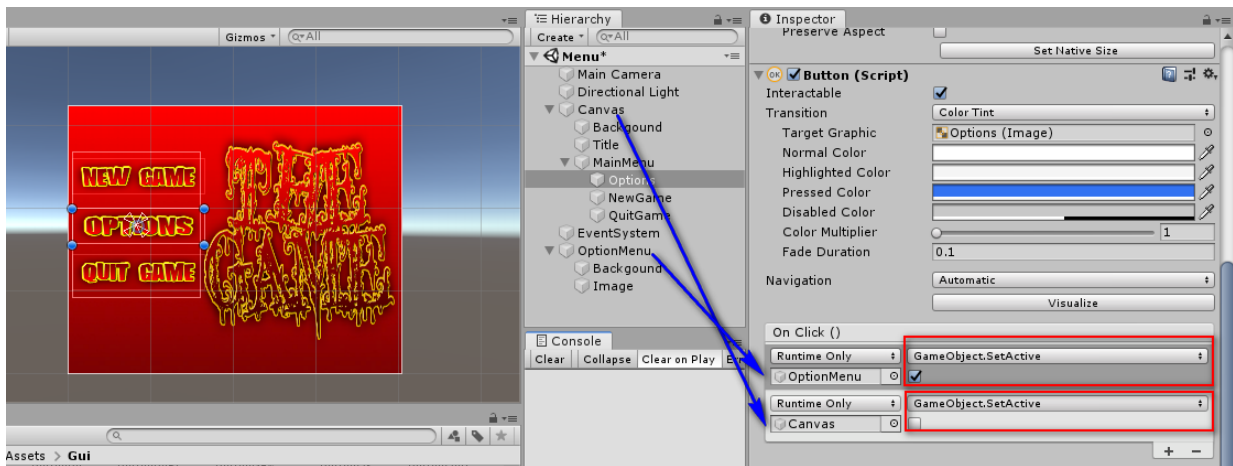
Chạy Game, nhấn nút để xem kết quả.

4. Tạo màn hình Options

Màn hình Options là màn hình khi người dùng nhấn nút Options. Nhân đôi Canvas vừa tạo, đổi tên thành OptionMenu, set Unactive của Canvas, và thiết kế OptionMenu như sau:



Active lại Canvas, chọn Button Options, chọn thành phần On Click(), sau đó tạo ra 2 sự kiện, kéo lần lượt OptionMenu và Canvas, thiết lập SetActive như sau:



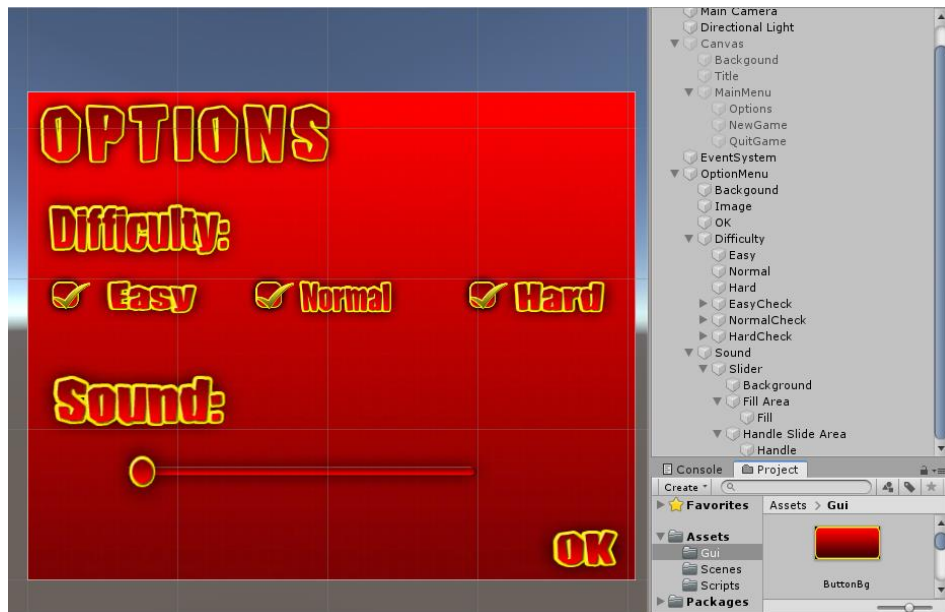
Chạy và bấm vào nút Options để xem kết quả.

Trên OptionMenu, thiết lập Button OK để khi nhấn button này thì quay về màn hình Menu:



5. Thiết kế giao diện cho chức năng Options

Trong OptionMenu, dùng chức năng Image để thiết kế giao diện là các chữ như Difficulty, Easy, Normal, Hard, Sound; Sử dụng thành phần UI\Toggle để tạo các nút dạng bật, tắt; Sử dụng thành phần UI\Slider để tạo thanh trượt cho âm thanh. Sinh viên tự tìm hiểu các thuộc tính, gán các Sprite để được giao diện như hình sau:

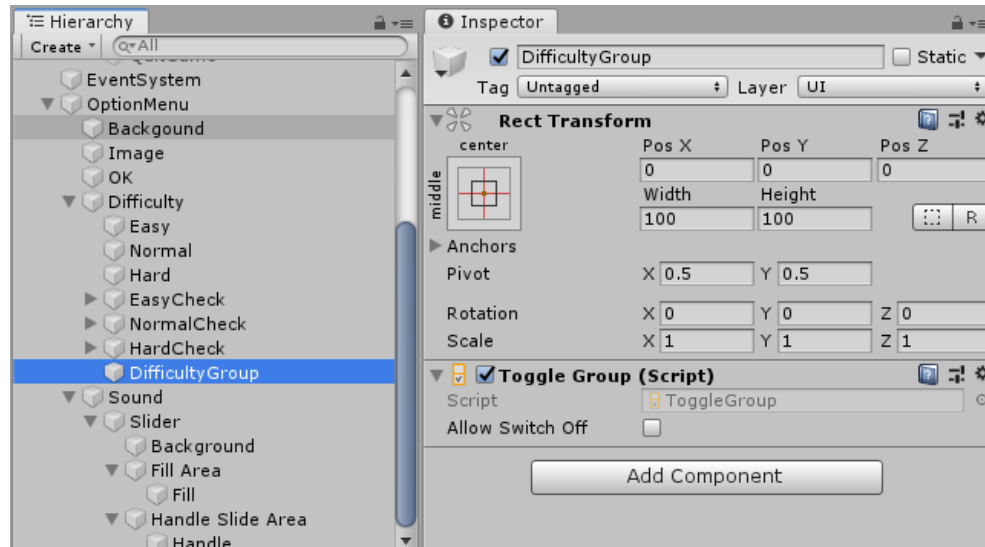


Lưu ý: - Các thành phần Easy, Normal, Hard là con của Difficulty; Tương tự như vậy với thành phần Slider và Sound.

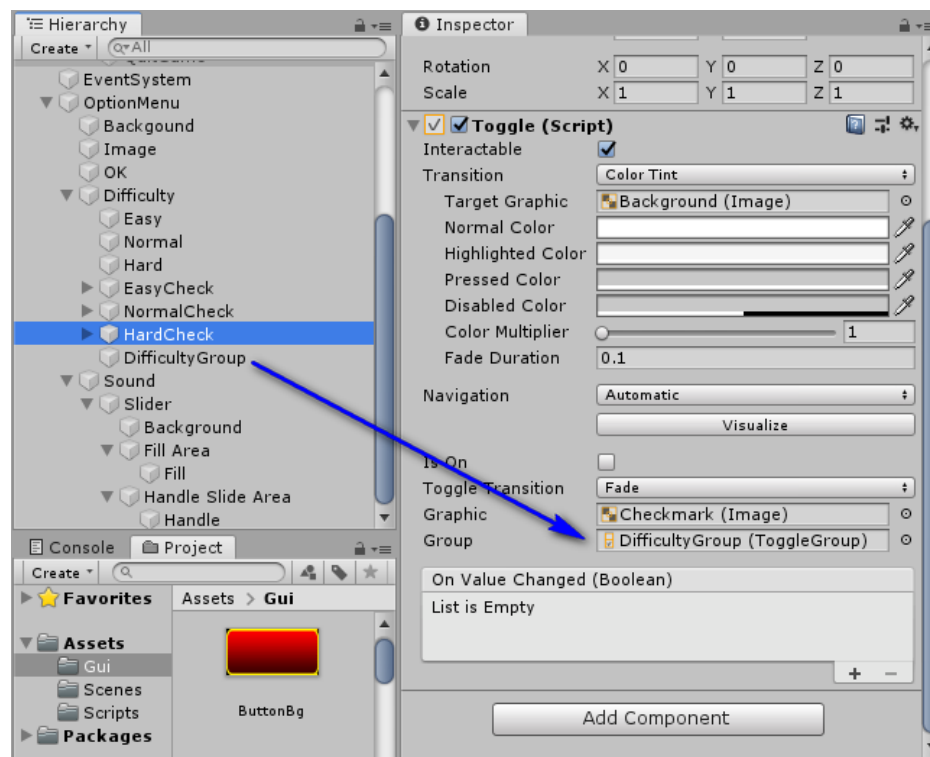
- Sinh viên xem bên Hierarchy để biết cách thiết kế các thành phần.

6. Chọn lựa các độ khó, dễ cho game

Trong OptionMenu, tạo một đối tượng Empty, đặt tên là DifficultyGroup. Click chuột vào DifficultyGroup, thêm thành phần là ToggleGroup:



Lần lượt click chuột vào các Toggle Easy, Hard, Normal và kéo ToggleGroup vào thuộc tính Group:



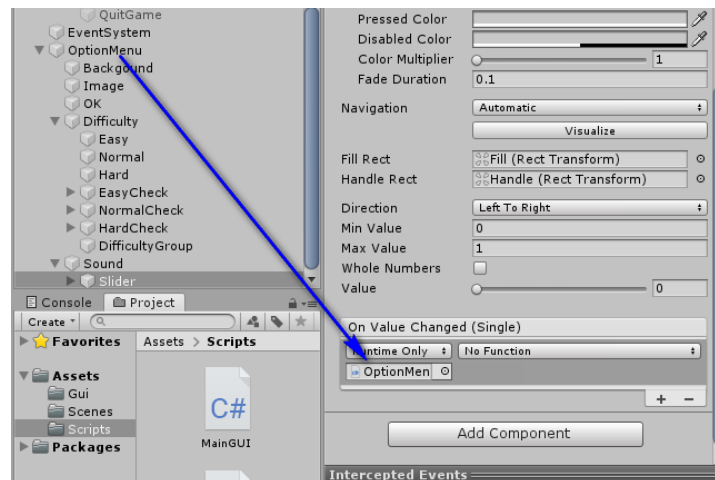
Chạy và xem kết quả.

7. Điều khiển âm thanh cho game

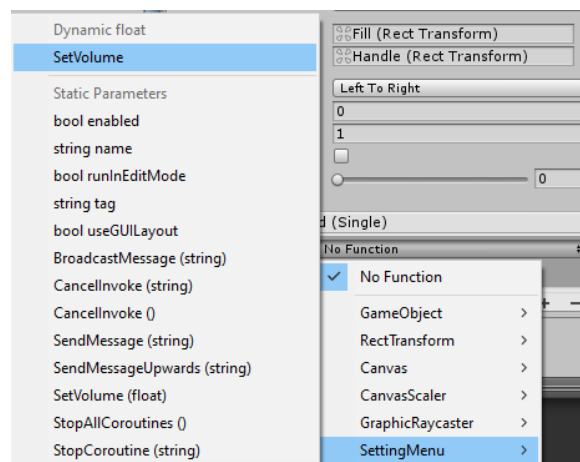
Tạo một Script có tên SettingMenu và gán vào OptionMenu, trong lớp SettingMenu, viết như sau:

```
public class SettingMenu : MonoBehaviour
{
    0 references
    public void SetVolume(float volume)
    {
        Debug.Log(volume);
    }
}
```

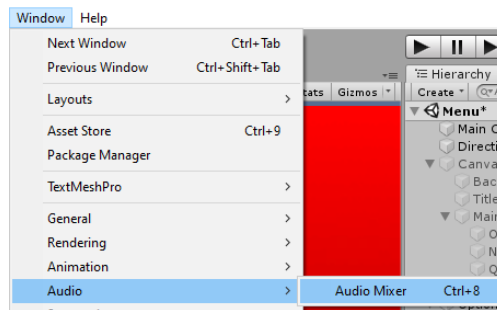
Chọn đối tượng Slider, tạo sự kiện và gán OptionMenu như sau:



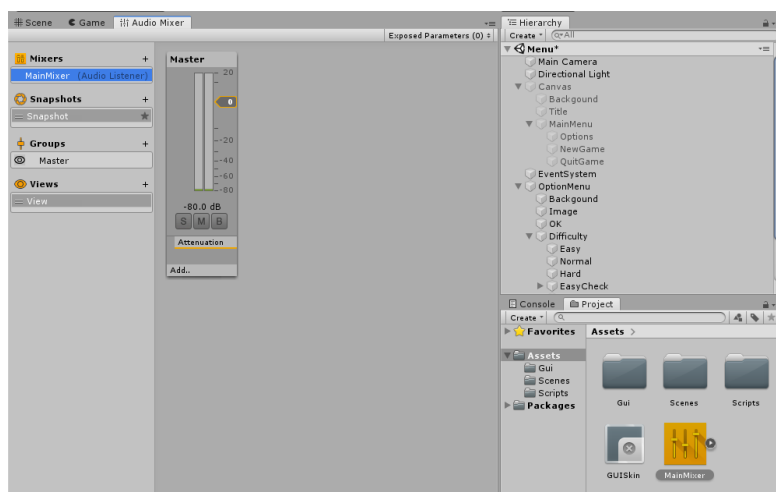
Trong phần No Function, chọn SettingMenu, chọn biến SetVolume trong Dynamic float:



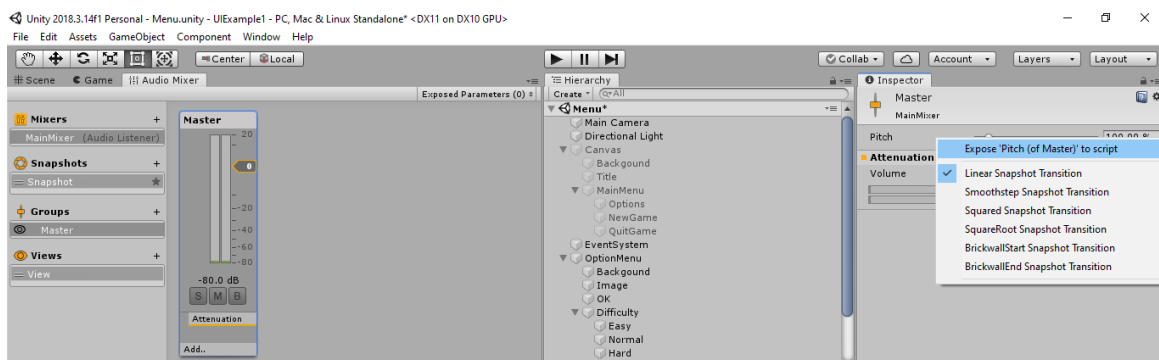
Tạo một AudioMixer bằng cách vào Window\Audio\Audio Mixer:



Thẻ Audio xuất hiện, nhấn vào dấu + để tạo một Mixer mới tên là Main Mixer:



Chọn đối tượng Master, right click lên thành phần bên phải, chọn *Expose 'Pitch (of Master)' to script*:



Khi đó, Exposed Parameter (0) sẽ trở thành Exposed Parameter (1), chọn và đổi tên thành volume:

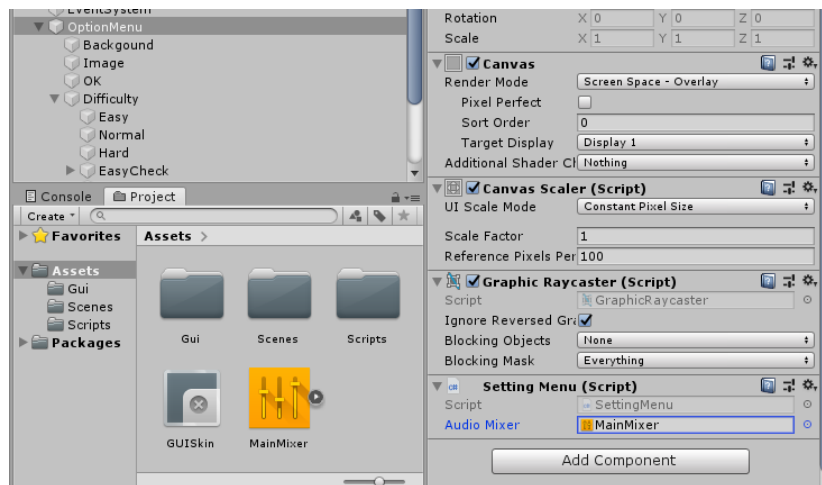


Thay đổi code trong class SettingMenu như sau:

```
using UnityEngine;
using UnityEngine.Audio;

0 references
public class SettingMenu : MonoBehaviour
{
    public AudioManager audioMixer;
    0 references
    public void SetVolume(float volume)
    {
        audioMixer.SetFloat("volume", volume);
    }
}
```

Chọn đối tượng OptionMenu, gán MainMixer cho script:



Cho một Audio vào Game, chạy và xem kết quả.

BÀI TẬP

Bài 1.

Viết Script để từ màn hình Game, khi nhấn nút Esc thì quay về màn hình Menu.

Bài 2.

Dùng Photoshop, tạo ra 2 nút OK và Cancel, tạo một giao diện (như OptionsGui) để khi nhấn nút thoát thì hỏi lại người dùng là có chắc chắn là sẽ thoát không. Chọn OK thì thoát, chọn Cancel thì trở lại Menu chính.

Bài 3. Viết script để khi người dùng thoát khỏi Game, hệ thống sẽ giữ lại các giá trị về Volume hoặc độ khó đã chọn.

Gợi ý: Có thể dùng lớp PlayerPrefs:

<https://www.youtube.com/watch?v=BgxbCej0G0g>

Bài 4.

- Thiết lập các tham số về tốc độ, số lượng Enemy để khi người dùng chọn các độ khó của game thì tăng các giá trị này lên.
- Thiết lập âm thanh nền cho game, khi người dùng tùy chỉnh âm thanh thì giá trị của Volume sẽ lớn nhỏ theo yêu cầu.

Bài 5*. Sinh viên xem video sau đây (có thể tham khảo một số video khác):

<https://www.youtube.com/watch?v=YOaYQrN1oYQ>

- Thiết lập tham số về độ phân giải cho Game;
- Thiết lập tham số về Graphics cho game.
- Thiết lập HealthBar cho nhân vật chính để khi bị Zombie chạm thì bị mất máu, đến khi hết máu thì nhân vật chết.

--Hết--