Làm Game 2D bằng Unity - Phần 4 - Prefab, Script và một số xử lý cơ bản
9 February 2014 at 07:37
Chào các bạn,

Ở phần trước chúng ta đã biết các kỹ thuật tạo animation và cách tạo animation bằng công cụ của unity và xây dựng mối quan hệ (máy chuyển đổi trạng thái) của một animation. Ở phần này sẽ hướng dẫn cách tạo Prefabs để tạo ra các đối tượng có xử lý giống nhau và sử dụng script để điều khiển máy trạng thái.
Trước khi đi tiếp, nếu bạn nào chưa đọc các phần trước, ta có thể tham khảo lại ở đây

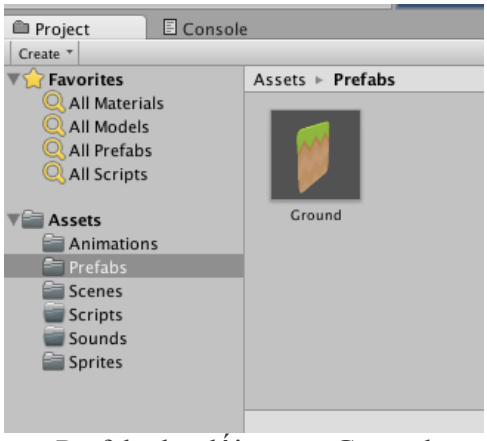
Link bài viết trước: https://www.facebook.com/notes/hội-lập-trình-viên-game-đà-nẵng/làm-game-2d-bằng-unity-phần-3-animation-và-điều-khiển-hành-động-nhân-vật-animato/233104363528500

Dự án mẫu cho phần 4 ở đây: https://www.dropbox.com/sh/5gdxosk5iny0tzk/7Em4eBDizp

III. Tạo các đối tượng cơ bản
1. Game Object
2. Sprite
3. Animation
4. Prefab

Prefab cho ta tạo ra các bản sao nhanh của một đối tượng mà không cần thiết lập lại các giá trị khởi tạo của một đối tượngnào đó ngoài trừ các giá trị transform (vị trí, tỉ lệ, quay).

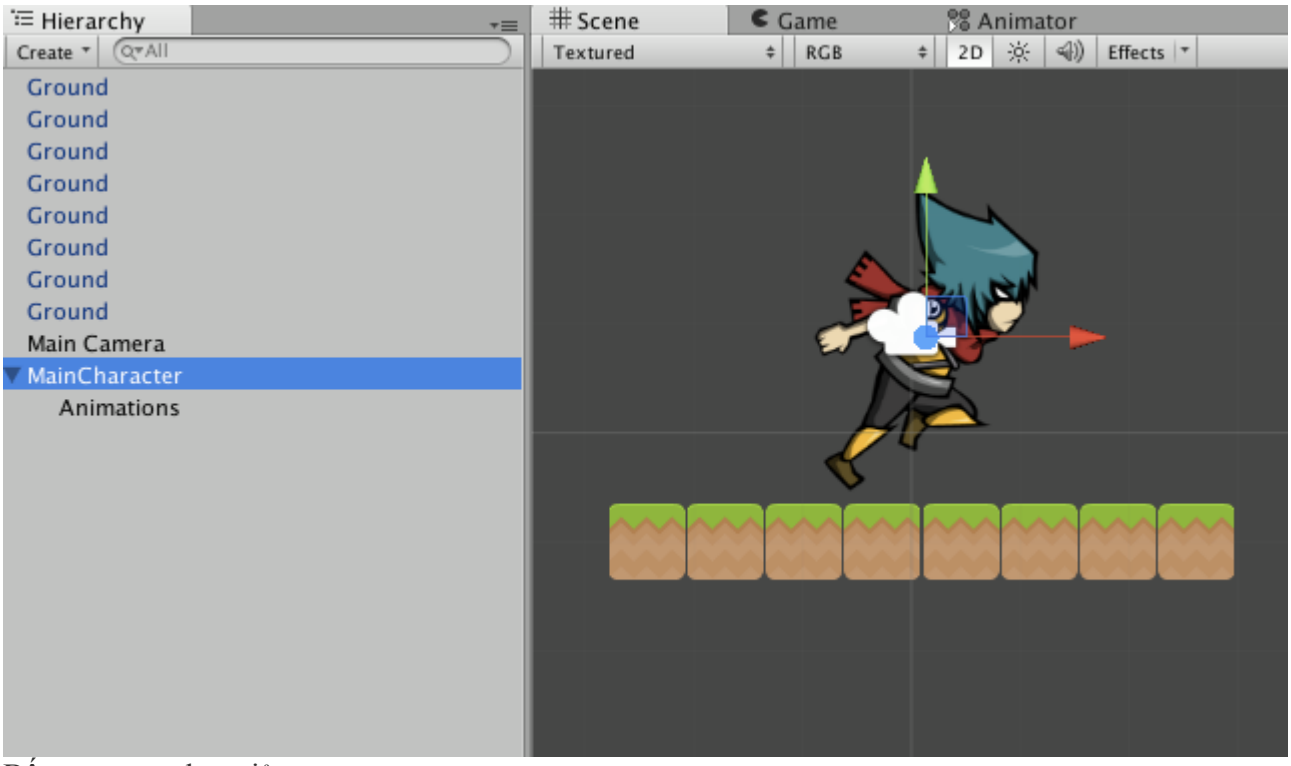
Để tạo một Prefab cho một đối tượng nàođó, ta chỉ cần kéo thả đối tượng đó ở cửa sổ Hierarchy xuống thư mục Prefab (nếu chưa có thì tạo mới) trong cửa sổ Project của chúng ta. Sau này muốn sử dụng ta chỉ việc kéo các Prefab này trở lại cửa sổ Scene.



Prefab cho đối tượng Ground

Bây giờ ta sẽ kéo thả Prefabs này tạo thành một đường thẳng, và đặt lại vị trí đối tượng MainCharacter ở vị trí nằm trên đường thẳng này như hình vẽ.

Chú ý: là đặt vị trí của đối tượng MainCharacter, chứ không phải đối tượng con Animation của MainCharacter. Vị trí của đối tượng Animation bây giờ vẫn là (0, 0) so với đối tượng Maincharacter (vì là đối tượng con) nghĩa là đối tượng Animation này nằm tại tâm của đối tượng MainCharacter.



Bố cục game đơn giản

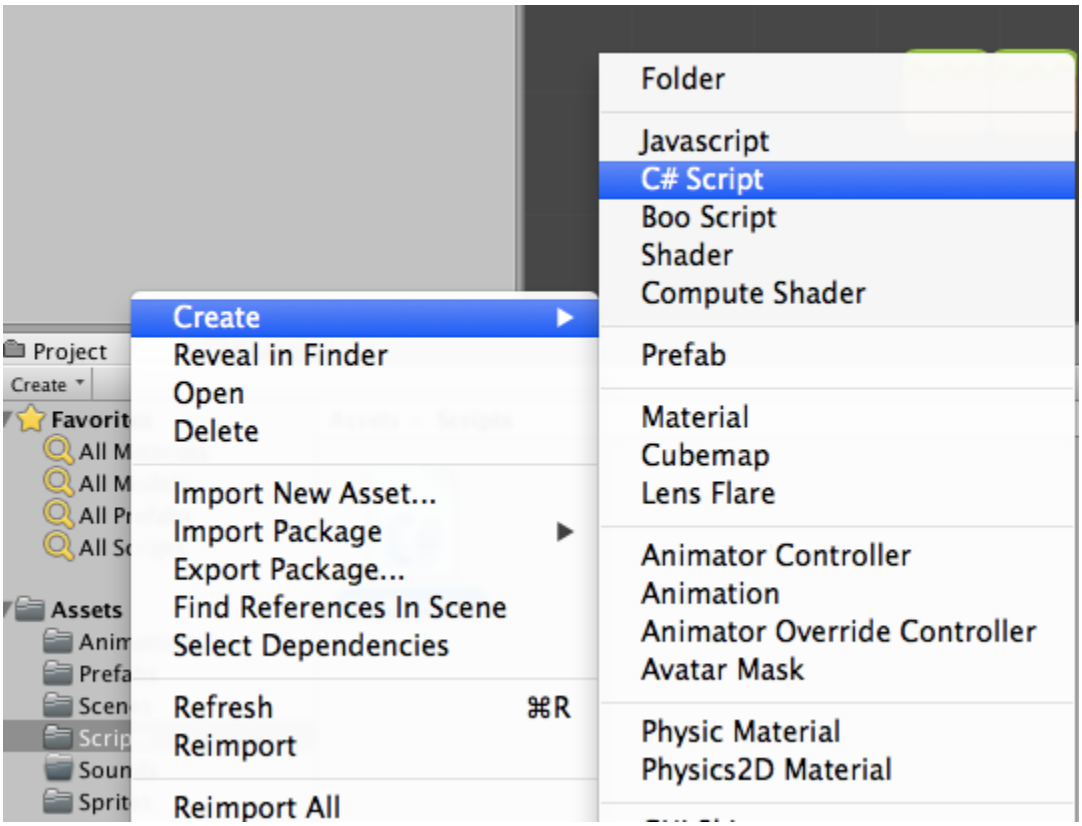
Tip: Để thiết lập giá trị y của các Ground bằng nhau, ta chọn tất cả rồi thiết lập giá trị y ở cửa sổ Inspector.

5. Script và điều khiển máy trạng thái

a. Script

Script là một tập tin (cũng là Game Component) chứa các mã điều khiển cho một đối tượng nào đó trong game, được viết bằng C# hay Javascript hoặc BOO. (Trong bài hướng dẫn này, chúng ta sẽ sử dụng C#).

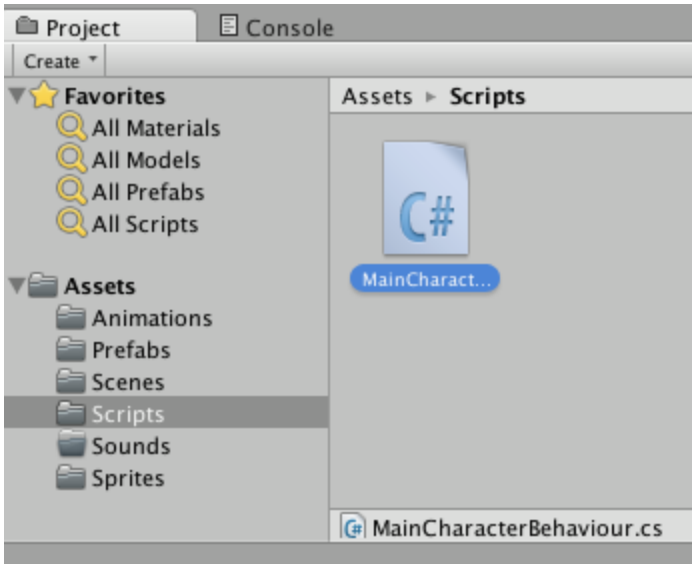
Để tạo script ta click chuột phải ở thư mục Scripts/Create/C# Script



Tạo Script

Ta tiến hành đổi tên file Script và tên class thành MainCharacterBehavior (Behaviour script của đối tượng MainCharacter).

Chú ý: tên lớp và tên file script phải giống nhau.



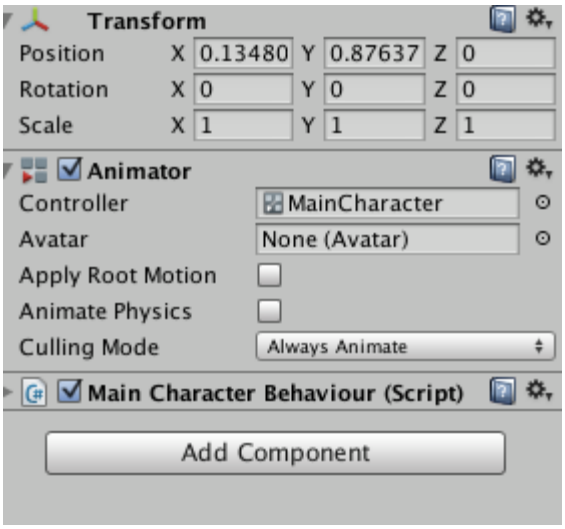
MainCharacterBehaviour

Tip: Chúng ta chú ý đến việc đặt tên các hàm, biến, đối tượng rõ ràng, phù hợp với chức năng của chúng, điều này sẽ giúp chúng ta kiểm soát code dễ dàng, đọc code dễ hiểu, dễ debug, dễ dàng phán đoán ra lỗi, và đặt biệt là khi làm nhóm. Tránh đặt tên viết tắt, sai lệch với chức năng, phải viết comment dài dòng gây rối code và khó hiểu cho chúng ta...

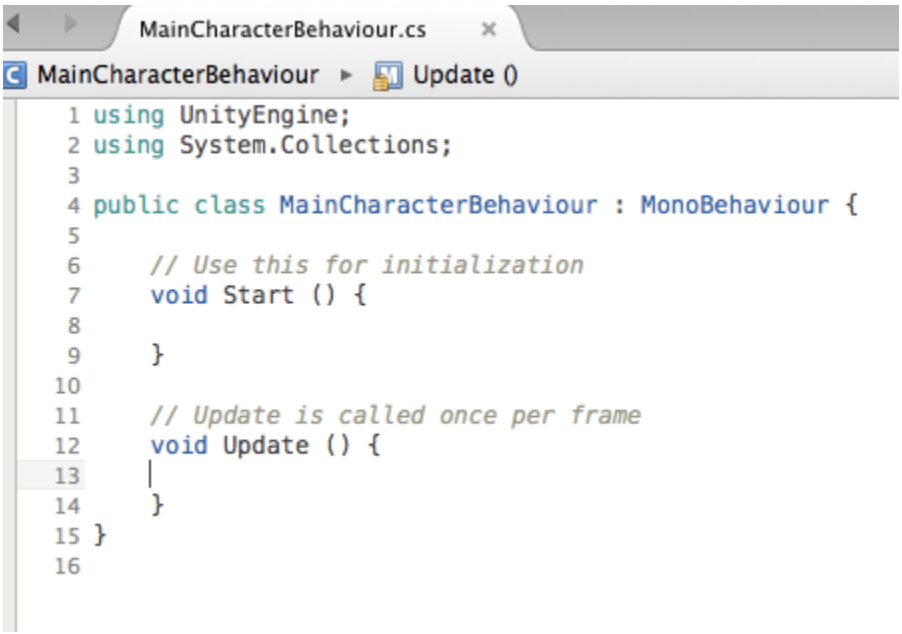
Khi chúng ta code, đầu óc còn minh mẩn, tỉnh táo, hoặc vừa mới viết nên có thể nhớ hết những gì mình viết, nhưng với một dự án vừa phải hoặc sau một ngày làm việc mệt nhọc, hoặc khi gặp những lỗi vớ vẩn đau đầu mà tìm hoài không ra lỗi thì đây mới chính là lúc cách trình bày code chúng ta lên tiếng.

--

Với mỗi script tạo ra, ta cần chỉ định script này thuộc về đối tượng nào. Ta thực hiện bằng cách: Chọn đối tượng MainCharacter ở cửa sổ Hierarchy, ở cửa sổ Inspector chọn Add Component/Scripts/Main Character Behaviour.cs ở danh sách. Sau khi add:



Kích đúp để mở file script ta sẽ thấy như sau:



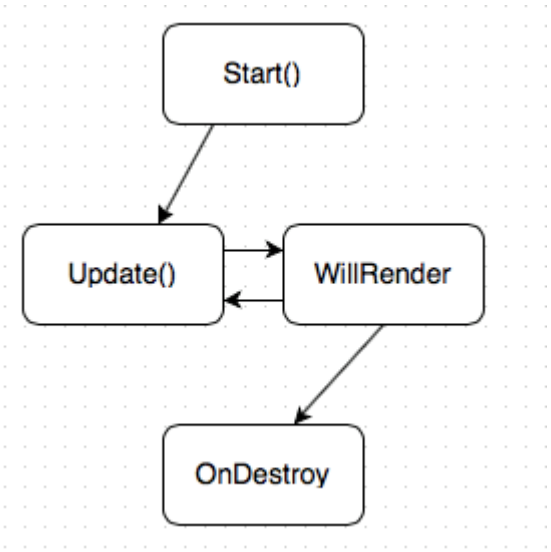
Cấu trúc script mặc định

Vòng đời của một script hay vòng đời của một game object các bạn có thể tham khảo chi tiết tại https://docs.unity3d.com/Documentation/Manual/ExecutionOrder.html. Mới bắt đầu chúng ta cần quan tâm đến một số hàm cơ bản sau:

+ **Start**(): Được gọi 1 lần đầu tiên sau khi khởi tạo đối tượng, trước khi vào Update.

+ **Update**(): Được gọi liên tục sau mỗi frame, sau Start.
+ **OnWillRenderObject():** Được gọi liên tục sau một frame sau Update. Thường thì ta sẽ ít đụng đến hàm này.
+ **OnDestroy**(): Được gọi khi đối tượng bị huỷ.

Biểu đồ:



- Ta có thể khai báo các thuộc tính trong file script, các thuộc tính này sẽ được hiển thị ở cửa sổ Inspector

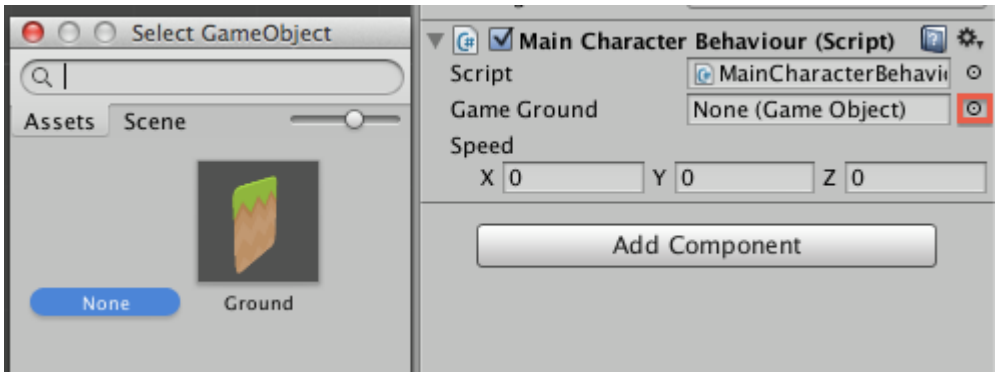
Ở MainCharacterBehaviour:

```
public class MainCharacterBehaviour : MonoBehaviour {

    public GameObject gameGround;
    public Vector3 speed;
    // Use this for initialization
    void Start () {

    }
```

Ở Inspector:



Đối với thuộc tính là GameObject chúng ta có thể nhấn vào nút đỏ, sau đó chọn prefabs cho đối tượng ở bảng mới hiện ra.

b. Một số xử lý cơ bản

- Truy cập vào đối tượng game hiện tại thông qua this.gameObject
- Thay đổi vị trí, tỉ lệ, quay đối tượng thông qua gameObject.transform(.position, .scale, .rotate)
- Để huỷ một đối tượng game Destroy(GameObject) (*)
- Để tạo một prefab trong quá trình thực thi game: Instantiate(gameObject, Vector3, Quaternion) (**)
- Input.GetKeyDown(keyCode), GetKey(keyCode), GetKeyUp(keyCode) kiểm tra xem một key được bấm, được giữ, được thả ra hay không ?
- Input.GetAxis ("Horizontal") trả về giá trị số thực trong khoảng -1..1 nếu có sự kiện các key right hoặc left được bấm (key ngang).
- Input.GetAxis ("Vertical") trả về giá trị số thực trong khoảng -1..1 nếu có sự kiện các key up hoặc down được bấm (key dọc).
- OnMouseDown, OnMouseUp, OnMouseDrag các hàm xử lý chuột.
- gameObject.GetComponentComponentName>() get Game Component được đính kèm trong gameObject hiện tại có thể là: Animator, Transform ...

Chú ý: (*), (**) thường được sử dụng trong trường hợp như một nhân vật bắn súng.

Ví dụ dưới đây là một xử lý đơn giản khi bấm nút left, right là đối tượng sẽ đi qua đi về bằng cách thay đổi vị trí của đối tượng:

```
public class MainCharacterBehaviour : MonoBehaviour {

    public GameObject gameGround;
    public Vector3 speed;
    // Use this for initialization
    void Start () {
        speed = new Vector3 (5, 0, 0);
    }

    // Update is called once per frame
    void Update () {
        if (Input.GetKey (KeyCode.LeftArrow))
        {
            gameObject.transform.Translate (-speed * Time.deltaTime);
        }
        if (Input.GetKey(KeyCode.RightArrow))
        {
            gameObject.transform.Translate (speed * Time.deltaTime);
        }

    }
```

c. Điều khiển chuyển đổi trạng thái

Ta sẽ set trạng thái mặc định của Animation là Idle. Các thiết lập trạng thái mặc định đã đề cập trong phần trước.

Tiếp theo, ta sẽ khai báo một animator để tham chiếu đến Animator Component của đối tượng MainCharacter.

```
public GameObject gameGround;
public Vector3 speed;
protected Animator animator;
// Use this for initialization
void Start () {
    speed = new Vector3 (5, 0, 0);
    animator = gameObject.GetComponent<Animator>();
}
```

Ở hàm update ta sẽ xử lý để chuyển đổi state như sau:

```
//Switch states
if (Input.GetKey(KeyCode.LeftArrow) || Input.GetKey(KeyCode.RightArrow))
{
    Debug.Log("Running...");
    animator.SetBool("isIdle", false);
    animator.SetBool("isRunning", true);
}
else
{
    Debug.Log("Idle...");
    animator.SetBool("isRunning", false);
    animator.SetBool("isIdle", true);
}
```

Tiếp theo ở phần xử lý di chuyển khi nhấn key như đã đề cập trước, ta sẽ xử lý thêm phần xoay đối tượng theo chiều x, khi nhấn nút Left/Right như sau:

```
Vector3 localScale = transform.localScale;
//Translation$
if (Input.GetKey (KeyCode.LeftArrow))
{
    gameObject.transform.Translate (-speed * Time.deltaTime);
    if (localScale.x > 0)
    {
        localScale.x *= -1.0f;
    }
}

if (Input.GetKey(KeyCode.RightArrow))
{
    gameObject.transform.Translate (speed * Time.deltaTime);
    if (localScale.x < 0)
    {
        localScale.x *= -1.0f;
    }
}
transform.localScale = localScale;
```

Nhấn nút play và nhấn thử nút Left, Right ta sẽ thấy kết quả. Tương tự ta có thể xử lý và thay đổi máy trạng thái cho các trạng thái khác.

Như vậy là ta đã biết được cách tạo và sử dụng Script và Prefabs để điều khiển, xử lý các đối tượng trong game. Phần tiếp theo mình sẽ đề tập đến thành phần vật lý, cách xử lý va chạm trong Unity, một trong những thành phần rất quan trọng để làm game.

Link phần tiếp theo: https://www.facebook.com/notes/hội-lập-trình-viên-game-đà-nẵng/làm-game-2d-bằng-unity-phần-5-thành-phần-vật-lý-và-xử-lý-va-chạm/246753422163594