

BỘ MÔN CÔNG NGHỆ PHẦN MỀM
ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÀI TẬP THỰC HÀNH
COMPUTER GRAPHICS

LÀM QUEN VỚI UNITY ENGINE

Trịnh Thành Trung

MỤC LỤC

BÀI 1 – LÀM QUEN VỚI UNITY	3
1. Chuẩn bị	3
2. Các khái niệm của Unity	3
3. Tìm hiểu về giao diện của Unity	3
4. Chạy thử demo <i>Angry Bot</i> của Unity	7
BÀI 2 – XÂY DỰNG MỘT KHUNG CẢNH ĐƠN GIẢN	8
1. Tạo một dự án mới	8
2. Thêm tài nguyên vào dự án.	9
3. Tạo các đối tượng game	9
4. Chạy thử game.	11
5. Sử dụng mô hình	11
BÀI 3 – SCRIPTING CƠ BẢN	13
1. Viết script cho Player	13
2. Viết script cho Street.....	14
3. Viết script cho Enemy	14
4. Xử lý va chạm	14
BÀI 4 – VIẾT SCRIPT BỔ SUNG	16
1. Giới hạn di chuyển của Player	16
2. Cho phép Player có khả năng tăng tốc và giảm tốc	16
3. Tính điểm	16
BÀI 5 – THÊM CÁC YẾU TỐ TRÒ CHƠI	17
BÀI 6 – THIẾT KẾ ĐỊA HÌNH	18
1. Tạo địa hình.....	18
2. Thiết kế hòn đảo	19
3. Vẽ texture.....	19
4. Đi dạo quanh hòn đảo.....	22
BÀI 7 – CÁC TÍNH NĂNG ĐỊA HÌNH NÂNG CAO	23
1. Đặt cây	23
2. Vẽ các cỏ cây hoa lá lên địa hình	24
3. Skybox	25
4. Ánh sáng.....	25
5. Nước.....	26

BÀI 1 – LÀM QUEN VỚI UNITY

1. Chuẩn bị

Download Unity tại địa chỉ <http://unity3d.com/unity/download/>

Cài Unity vào máy

2. Các khái niệm của Unity

Scene

Trong Unity, mỗi một *Scene (khung cảnh)* có thể coi như một màn chơi hoặc một vùng trong game. Một số thành phần của game cũng có thể là một Scene như Menu

Game Object

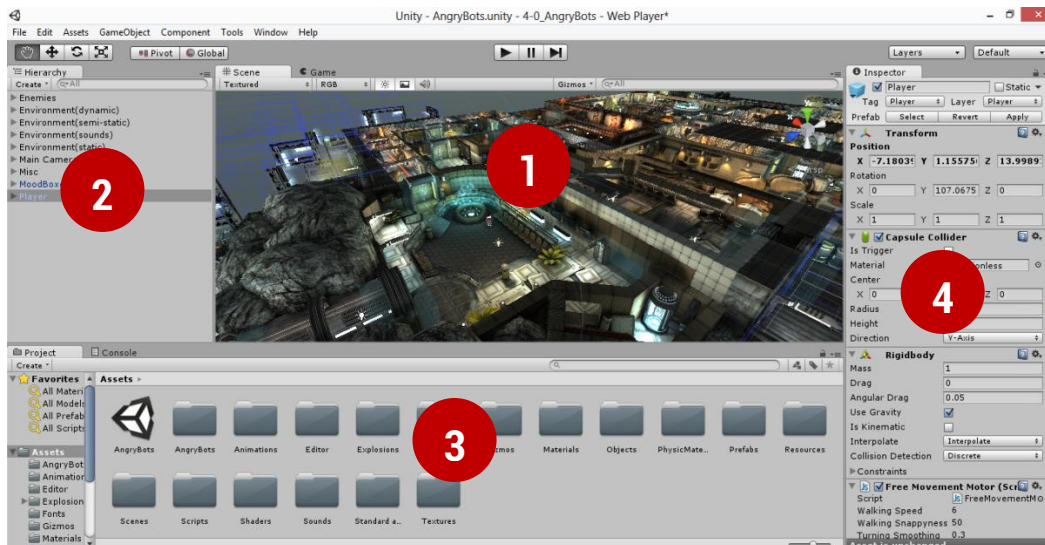
Để làm game trong Unity, chúng ta cần phải đưa vào các mô hình, texture, âm thanh... (gọi chung là các tài nguyên). Tuy nhiên để có thể sử dụng được các tài nguyên này chúng ta cần phải có các *Game Object (đối tượng game)*. Các đối tượng này luôn có thành phần Transform xác định Position (vị trí), Rotation (độ xoay) và Scale (độ phóng đại) của chúng.

Script

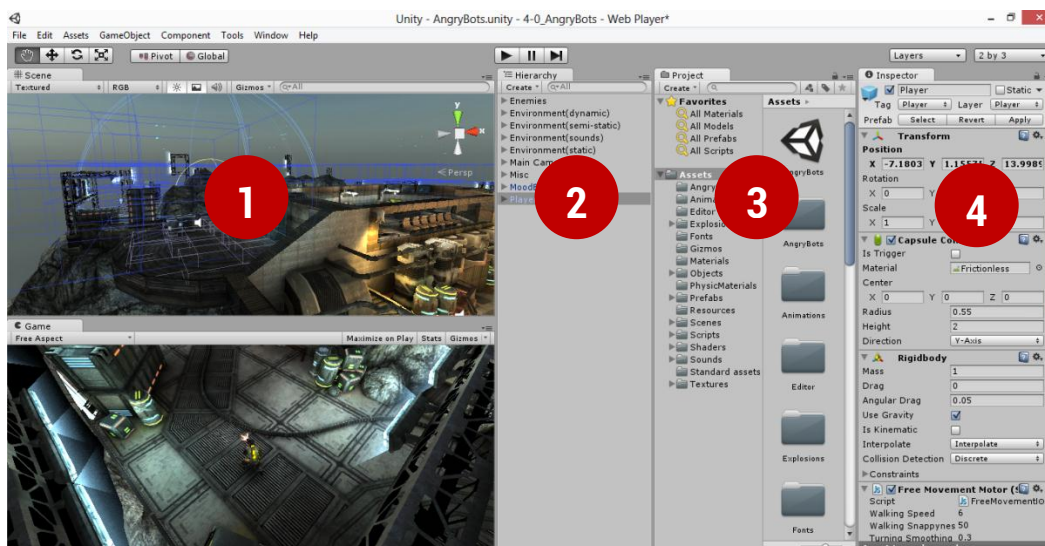
Để có thể xây dựng được game logic trong Unity, chúng ta cần phải code bằng các ngôn ngữ kịch bản mà Unity hỗ trợ, bao gồm: Javascript, C# Script và Boo Script. Thông thường bản cài đặt Unity sẽ kèm theo một công cụ soạn thảo mã, ví dụ như Mono Develop hoặc chúng ta có thể sử dụng công cụ mà mình thích.

3. Tìm hiểu về giao diện của Unity

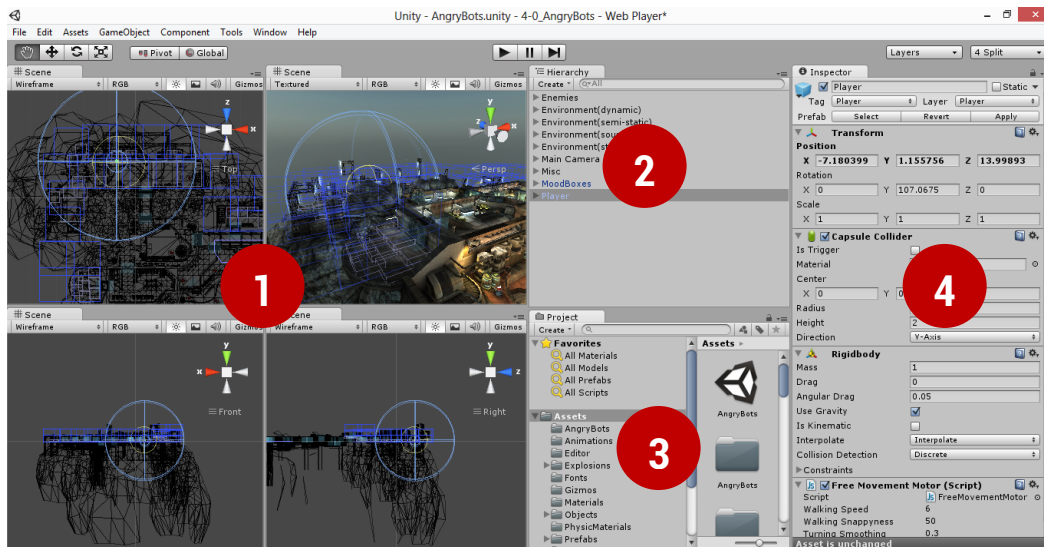
Giống như nhiều bộ công cụ phát triển khác, giao diện của Unity có các thành phần có thể dễ dàng thay đổi vị trí được. Mặc định Unity cung cấp 4 layout khác nhau, chúng ta có thể chuyển đổi qua lại bằng cách vào menu Windows | Layouts...



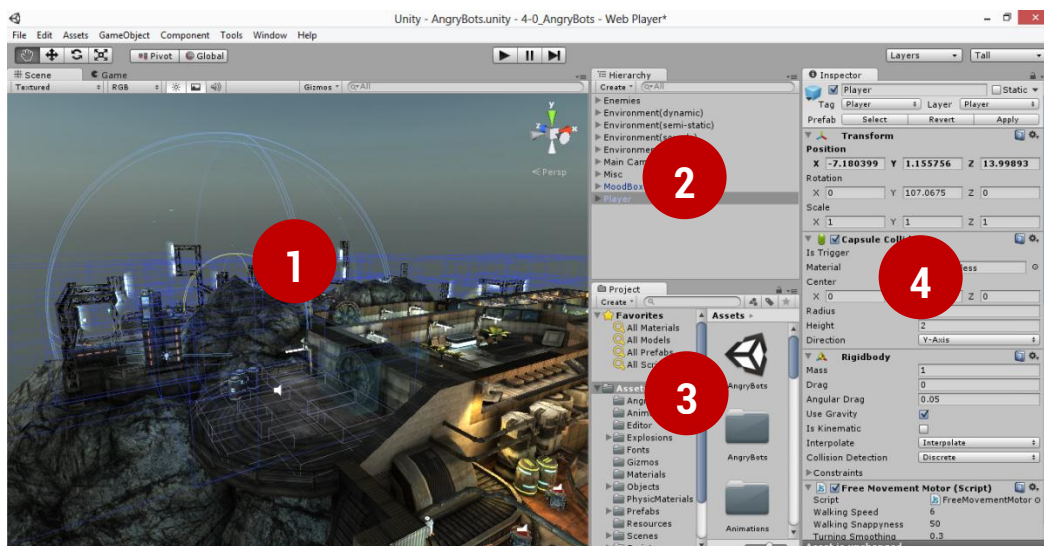
Mặc định



2 by 3



4 Split



Tall



Wide


Scene/Game window

Thực chất Scene và Game là hai cửa sổ khác nhau. Tuy nhiên cửa sổ Game chỉ được kích hoạt khi chúng ta chạy thử game. Cửa sổ Scene là cửa sổ chính mà chúng ta sử dụng để thiết kế game. Cửa sổ này hiển thị tất cả các đối tượng của khung cảnh hiện tại. Unity có khả năng tự động tạo đối tượng mới mỗi khi một tài nguyên khi được kéo thả vào cửa sổ này.




Chú ý ở phía trên cửa sổ này chúng ta có thanh công cụ



Trong đó:

-  Hand (phím tắt **Q**): Giúp chúng ta di chuyển bên trong khung cảnh. Sử dụng phím trái chuột để di chuyển tịnh tiến (pan), phím phải để quay (rotate), phím giữa để di chuyển tịnh tiến (pan) và nút cuộn để zoom.

Lưu ý: Phím phải, phím giữa và nút cuộn chuột vẫn có thể sử dụng cùng với các công cụ khác ở dưới

-  Translate (phím tắt **W**): Dùng để di chuyển đối tượng đang được chọn
-  Rotate (phím tắt **E**): Dùng để xoay đối tượng đang được chọn
-  Scale (phím tắt **R**): Dùng để phóng to và thu nhỏ đối tượng đang được chọn

Hierarchy window

Cửa sổ Hierarchy hiển thị tất cả các đối tượng của Project theo dạng cây. Chọn một đối tượng trong cửa sổ này thì đối tượng đó trên khung cảnh ở cửa sổ Scene cũng đồng thời được chọn.

Project window

Cửa sổ Project chứa tất cả các tài nguyên của game, được tổ chức theo các thư mục của dự án. Cửa sổ Project cũng cho phép chúng ta tạo ra một số các tài nguyên cơ bản (Script, Material...)

Inspector window

Cửa sổ Inspector là cửa sổ thay đổi theo ngữ cảnh, tùy thuộc vào đối tượng đang được chọn mà cửa sổ sẽ hiển thị các thành phần, thuộc tính phù hợp của đối tượng đó.

Mẹo: Ở bất kỳ cửa sổ nào khi ấn phím Space thì cửa sổ đó sẽ chuyển thành Fullscreen. Ấn phím Space lại lần nữa để trở về kích thước ban đầu.

4. Chạy thử demo *Angry Bot* của Unity

Chọn **File | Open Project**

Mở demo *Angry Bot* của Unity, thường là nằm ở thư mục `C:\Users\Public\Documents\Unity Projects\4-0_AngryBots` (Hoặc download file *1. Angry Bots.zip* tại đây)

Ấn Play để chơi thử game.

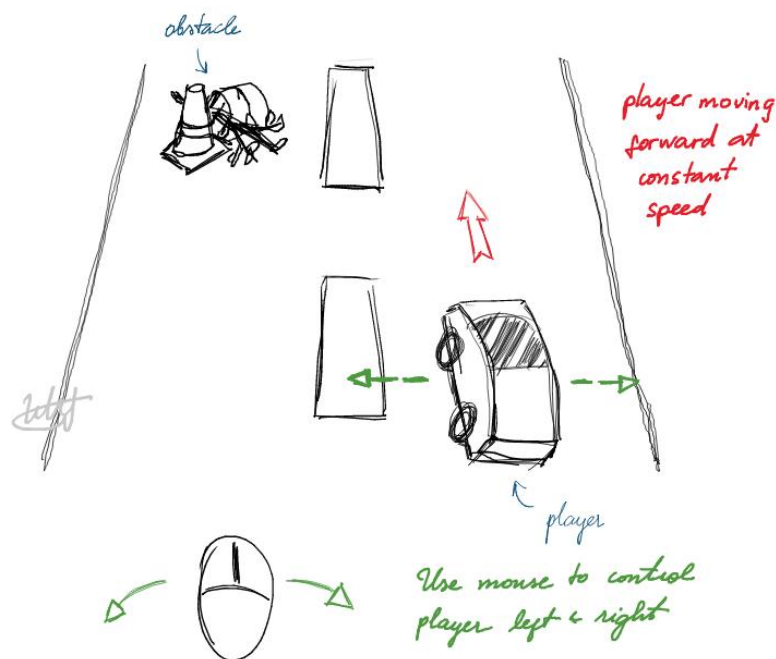
Thoát game. Tìm và chọn các đối tượng quan trọng, quan sát các giá trị trong cửa sổ Inspector.

Thay đổi vị trí, giá trị của các đối tượng game. Ấn Play để kiểm tra các thay đổi đó.

BÀI 2 – XÂY DỰNG MỘT KHUNG CẢNH ĐƠN GIẢN

Ví dụ chúng ta có một ý tưởng game đơn giản như sau

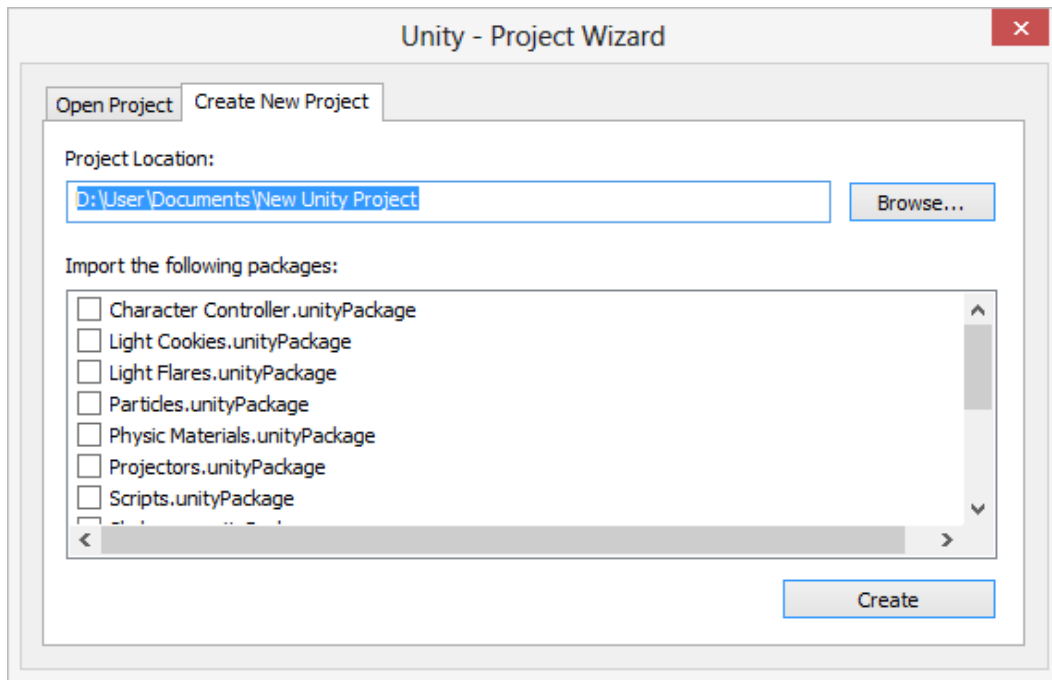
- Tên trò chơi: Đua xe
- Thể loại: Endless Running Game
- Các đối tượng game: Người chơi, chướng ngại vật, đường đi.
- Mô tả trò chơi: Người chơi luôn phóng xe về phía trước và tránh mọi vật cản. Các vật cản sinh ra ngẫu nhiên trên đường. Trò chơi kết thúc khi người chơi đâm phải một vật cản nào đó.



Sau đây chúng ta sẽ nghiên cứu để xây dựng game này

1. Tạo một dự án mới

- Chọn File | New Project...
- Chọn đường dẫn và ấn Create



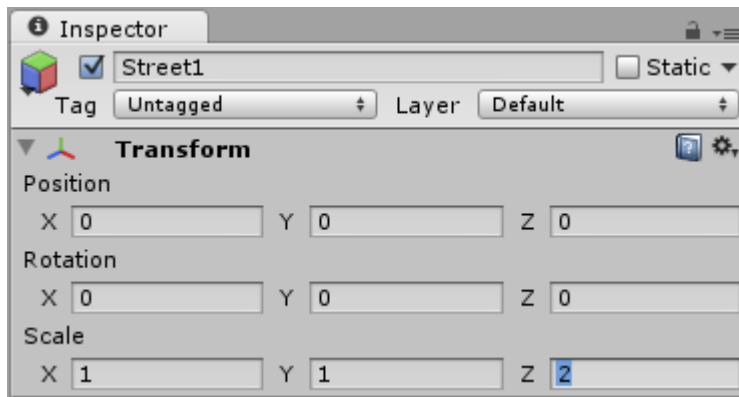
2. Thêm tài nguyên vào dự án.

Down file tài nguyên của game tại đây. Để thêm các tài nguyên này vào dự án, có hai cách:

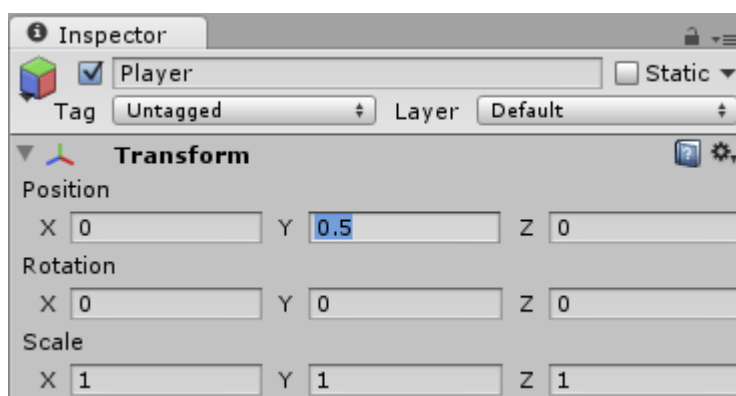
- Cách 1: Ở cửa sổ Project nhấp phải chọn Import New Assets... Lưu ý là chỉ có thể thêm một file một lần
- Cách 2: Chọn toàn bộ các tài nguyên của dự án ở cửa sổ duyệt file của Windows (có thể bao gồm cả các thư mục), sau đó kéo và thả vào cửa sổ Project của Unity

3. Tạo các đối tượng game

- Tạo đối tượng đường đi
 - Ở cửa sổ Hierarchy chọn Create | Plane (hoặc vào menu GameObject | Create Other | Plane)
 - Chọn đối tượng vừa tạo. Nhìn sang cửa sổ Inspector đổi tên đối tượng thành Street1
 - Mục Transform đặt vị trí cho Plane tại (0, 0, 0). Đặt Scale cho đường đi là (1, 1, 2)




- Tạo material cho đường đi: Kéo texture *street* vào đối tượng đường đi trên cửa sổ Scene. Unity sẽ tự động tạo ra một material mới và gán vào đối tượng đó. (Hoặc chúng ta có thể làm từng bước bằng cách Create | Material trong cửa sổ Project và gán texture vào, sau đó mới gán material cho đối tượng đường đi)
- Tương tự, tạo Street2 tại vị trí (0, 0, 20) để nối tiếp Street1
- Tạo Street3 tại (0, 0, 40) nối tiếp Street2
- Tạo đối tượng người chơi
 - Tương tự như Street ở trên, ở đây ta tạo một Cube mới
 - Đặt tên cho đối tượng là Player và đặt tại vị trí (0, 0.5, 0) để đối tượng nằm phía trên đường đi
 - Gán texture *player* cho đối tượng

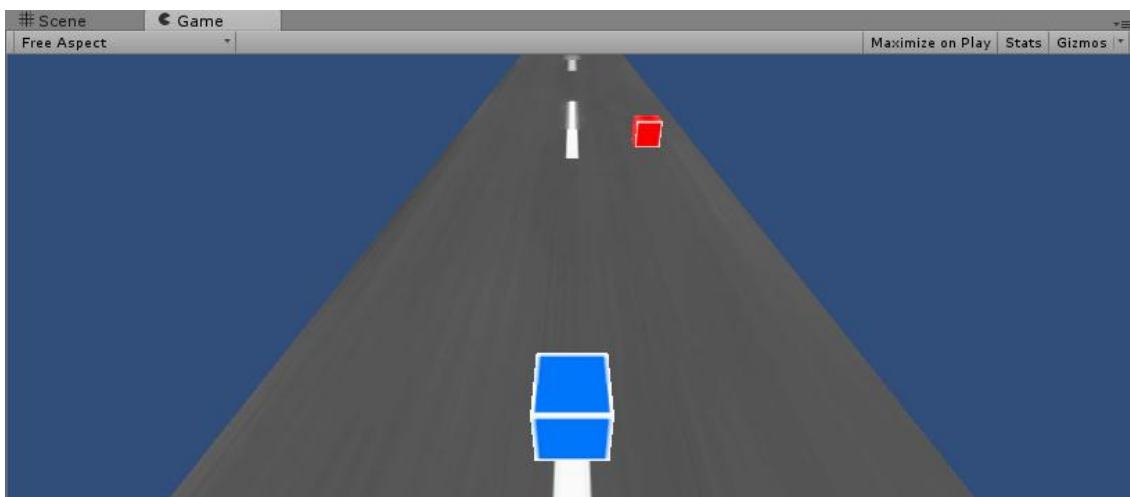


- Tạo đối tượng chương ngại vật
 - Tương tự như tạo đối tượng Player, ta cũng tạo một Cube mới, đặt ở vị trí trước mặt người chơi, độ cao Y là 0.5. Đặt tên đối tượng là Enemy

- Gán texture *enemy* cho đối tượng
- Tạo camera
 - Thông thường Unity sẽ tạo sẵn cho chúng ta một đối tượng là Main Camera. Nếu cần chúng ta có thể tạo các Camera mới
 - Di chuyển và xoay camera ở vị trí thích hợp để có thể dễ dàng nhìn thấy Player
- Tạo ánh sáng
 - Khung cảnh của chúng ta đang rất tối vì chưa đặt ánh sáng. Chúng ta có thể tạo ra các ánh sáng bằng cách Create | Light trong cửa sổ Hierarchy, hoặc để đơn giản chúng ta có thể thay đổi Ambient Light
 - Vào Edit | Render Settings. Chọn Ambient Light là màu trắng

4. Chạy thử game.

Ấn  để chạy thử kết quả



5. Sử dụng mô hình

Vào Unity Asset Store để download các mô hình thích hợp thay cho các hình khối được sử dụng ở trên. Ngoài ra chúng ta cũng có thể sử dụng các mô hình tự tạo trong Unity.

Việc sử dụng các mô hình trong Unity khá đơn giản: Unity hỗ trợ import mô hình từ một số các phần mềm thiết kế 3D phổ biến như Maya, 3D Studio Max, Blender, Lightwave... Việc

import các dữ liệu mô hình cũng tương tự như việc import các asset khác, chúng ta chỉ việc kéo và thả vào cửa sổ Project của Unity. Để có thể sử dụng các mô hình này chúng ta cũng có thể đơn giản kéo và thả vào cửa sổ Scene hoặc Hierarchy của Unity. Các texture hay material cũng có thể được kéo thả vào các mô hình của game.

BÀI 3 – SCRIPTING CƠ BẢN

Ý tưởng viết script của chúng ta như sau:

- Player luôn tiến về phía trước với vận tốc không đổi
- Sử dụng chuột để điều khiển xe sang trái, sang phải
- Camera luôn đi theo Player
- Enemy ngẫu nhiên xuất hiện trên đường đi. Chỉ một Enemy xuất hiện cùng lúc
- Nếu Player chạm vào Enemy thì game kết thúc

1. Viết script cho Player

Trong cửa sổ **Project** chọn **Create | C# Script**. Đặt tên cho script, ví dụ *PlayerCallback*

Chọn script *PlayerCallback*, ấn vào *Open...* trong cửa sổ **Inspector**

Như chúng ta thấy, Unity đã soạn sẵn cho ta script

```
using UnityEngine;
using System.Collections;

public class PlayerCallback : MonoBehaviour
{
    // Use this for initialization
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Ở đây chúng ta có lớp *PlayerCallback* thừa kế từ *MonoBehaviour* với các phương thức **Start()** được gọi khi đối tượng được khởi tạo, và **Update()** được gọi liên tục trong vòng đời của đối tượng.

Dưới đây là mã của phương thức **Update()** để Player luôn di chuyển về phía trước với vận tốc 0.5f

```
this.transform.position += new Vector3(Input.GetAxis("Mouse X"), 0, 0.5f);
Camera.main.transform.position += new Vector3(0, 0, 0.5f);
```

Lưu lại, ấn F8 để kiểm tra xem có lỗi nào không. Sau đó ta quay lại cửa sổ làm việc của Unity, kéo đoạn mã vừa tạo thả vào đối tượng Player

2. Viết script cho Street

Nếu ta chạy thử đoạn mã trên thì sẽ thấy xe văng thẳng ra ngoài đường. Để khắc phục chúng ta cần viết các đoạn mã cho script. Ý tưởng là khi chúng ta liên tục đưa đối tượng Street mà camera vừa đi qua đặt lên trên Street mà camera sắp chiếu tới. Như vậy là trên màn hình lúc nào cũng chỉ có 2 Street được hiển thị và chúng ta chỉ cần 3 đối tượng để xử lý mà không cần phải tạo ra quá nhiều đối tượng không dùng đến.

Đoạn mã cho ý tưởng trên như sau

```
if (Camera.main.transform.position.z >= (this.transform.position.z + 20))
{
    this.transform.position += new Vector3(0, 0, 60);
}
```

Tương tự như trên, chúng ta cũng tạo ra script mới (ví dụ *StreetCallback*) và viết mã trong phương thức **Update()**. Sau đó kéo script này vào cả 3 đối tượng Street của game.

3. Viết script cho Enemy

Đoạn mã sau giúp cho đối tượng Enemy xuất hiện ngẫu nhiên trên đường

```
if (Camera.main.transform.position.z >= (this.transform.position.z + 20))
{
    Vector3 np = this.transform.position;
    np.x = 0;
    np += new Vector3(Random.Range(-4.5f, 4.5f), 0, Random.Range(40, 60));
    this.transform.position = np;
}
```

Tạo script *EnemyCallback* và kéo script vào đối tượng Enemy của game.

4. Xử lý va chạm

Chúng ta cần viết hàm xử lý va chạm cho game. Unity đã hỗ trợ cho chúng ta tính năng xét xử lý va chạm, tuy nhiên mặc định để tối ưu hiệu năng thì chỉ có các đối tượng được cho phép thì Unity mới xử lý vật lý cho các đối tượng đó.

Để cho phép đối tượng có khả năng xử lý va chạm, ta chọn đối tượng đó rồi vào menu **Component | Physics | Rigidbody**. Ở đây ta chọn đối tượng Player.

Chọn Player, mục Inspector xuất hiện thêm một số component mới. Đánh dấu chọn vào **Is Trigger** trong component *Box Collider* để kích hoạt sự kiện *OnTriggerEnter* mỗi khi xảy ra va

chạm và bỏ mục **Use Gravity** của component *Rigid Body* vì chúng ta không sử dụng trọng lực trong game này.

Mở script *PlayerCallback*, thêm vào phương thức *OnTriggerEnter* như sau

```
void OnTriggerEnter(Collider obj)
{
    if (obj.name == "Enemy")
    {
        Debug.Log("Crashed!");
    }
}
```

Lưu lại và chạy thử game. Chúng ta sẽ có đối tượng Player chạy thẳng về phía trước với Enemy xuất hiện ngẫu nhiên trên mặt đường. Chúng ta có thể dùng chuột để điều khiển Player. Mỗi lần Player đụng phải Enemy thì trên cửa sổ Console sẽ hiện dòng chữ “Crashed!”

BÀI 4 – VIẾT SCRIPT BỔ SUNG

Chúng ta đã xây dựng được một game đơn giản, tuy nhiên để thực sự có thể chơi được chúng ta cần phải cải thiện nhiều. Dưới đây là một số gợi ý.

1. Giới hạn di chuyển của Player

Hiện tại Player của chúng ta có thể di chuyển ra ngoài đường đi (mà như vậy thì sẽ không bao giờ đâm phải chướng ngại vật nào). Chúng ta có thể khắc phục bằng nhiều cách. Ví dụ:

- Giới hạn tọa độ x của Player, chỉ cho phép Player di chuyển trong phạm vi của đường
- Đặt thêm các đối tượng Rigidbody ở hai bên đường để cản trở di chuyển của Player

2. Cho phép Player có khả năng tăng tốc và giảm tốc

Ý tưởng: Khi bắt đầu chuyển động thì vận tốc của xe bằng 0, sau đó vận tốc của xe mới bắt đầu tăng dần đều đến một vận tốc tối đa được đặt trước. Khi đâm phải chướng ngại vật thì vận tốc của xe giảm dần về 0.

Thực hiện: Đưa vào một biến vận tốc và một biến gia tốc cho PlayerCallback.

3. Tính điểm

Ý tưởng: Khi bắt đầu chuyển động thì vận tốc của xe bằng 0, sau đó vận tốc của xe mới bắt đầu tăng dần đều đến một vận tốc tối đa được đặt trước. Khi đâm phải chướng ngại vật thì vận tốc của xe giảm đột ngột về 0, sau đó vận tốc mới từ từ tăng lên. Khi xe chạm vào vỉa hè thì vận tốc của xe giảm dần về 0.1.

Bổ sung vào game 2 yếu tố: *Fuel* và *Score*. Fuel của game là một số được đặt trước, sau đó giảm dần về 0. Score là quãng đường mà người chơi đã đi được. Khi Fuel của Player giảm về 0 thì tốc độ của xe giảm dần về 0. Khi tốc độ của xe giảm hoàn toàn về 0 thì trò chơi kết thúc và tính Final Score của người chơi.

BÀI 5 – THÊM CÁC YẾU TỐ TRÒ CHƠI

Các bạn có thể tự suy nghĩ và đưa vào các yếu tố trò chơi theo ý mình. Viết script để thực hiện các yêu cầu đó.

Gợi ý: Chúng ta có thể thêm các yếu tố khác giúp game hay hơn, ví dụ như thêm các xe chở xăng, các loại enemy khác nhau, các loại đường khác nhau... Về yếu tố đồ họa chúng ta cũng có thể bổ sung thêm các mô hình cây cối, hay các Cube bằng các mô hình ô tô...



hoặc cũng có thể như thế này

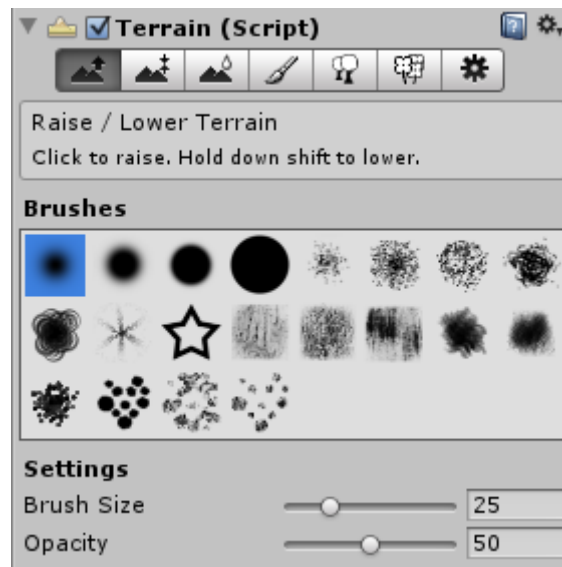


BÀI 6 – ĐỊA HÌNH








1. Tạo địa hình

Để tạo địa hình trong Unity, ta chọn menu **GameObjects | Create Others | Terrain**

Chọn Terrain vừa tạo, ta thấy cửa sổ Inspector như sau



Trong đó

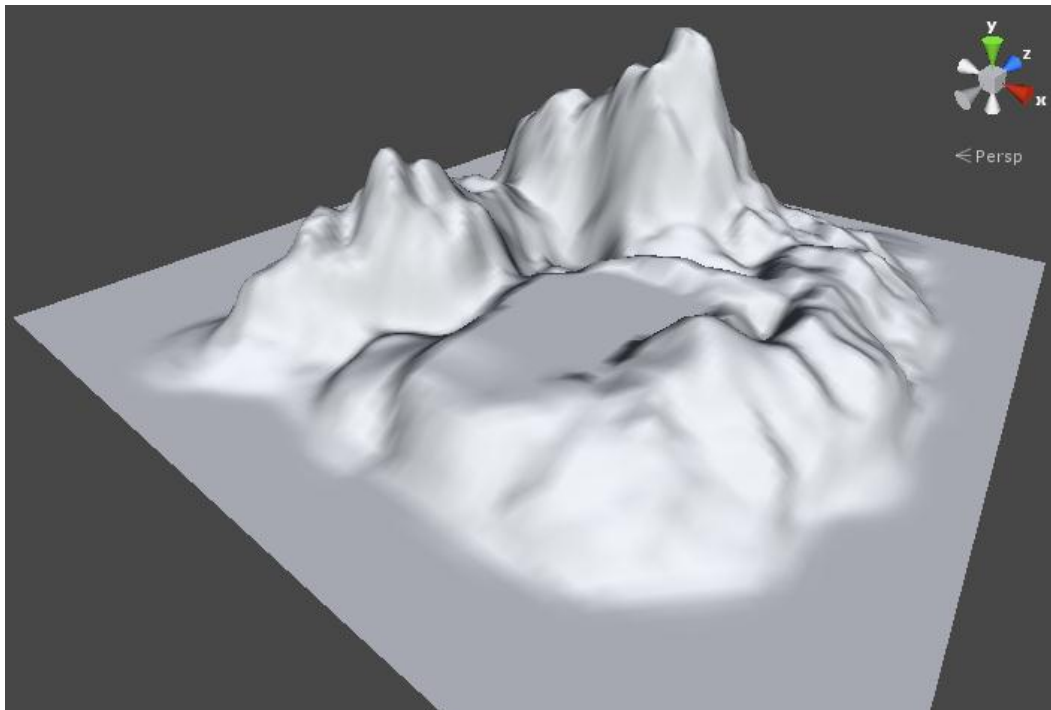
-  (Raise / Lower Terrain): Sử dụng để nâng cao bề mặt địa hình. Sử dụng kết hợp với phím Shift để hạ địa hình
-  (Paint Height): Sử dụng phím Shift và ấn vào một độ cao nào đó để lấy mẫu, sau đó ta có thể sử dụng công cụ này để nâng độ cao đến bằng mẫu đó
-  (Smooth Terrain): Làm mịn bề mặt địa hình, giúp địa hình các vùng lân cận có độ cao tương đồng hơn
-  (Paint Texture): Vẽ texture lên địa hình
-  (Place Trees): Đặt cây lên bề mặt địa hình
-  (Paint Details): Đặt các chi tiết nhỏ như cỏ cây hoa lá lên bề mặt địa hình
-  (Terrain Settings): Thay đổi các thiết lập của địa hình.

2. Thiết kế hòn đảo

Để làm quen với các công cụ thiết kế địa hình của Unity, chúng ta sẽ tiến hành thiết kế một hòn đảo.


Dựng địa hình

Trước tiên chúng ta sử dụng công cụ **Raise / Lower Terrain**, **Smooth Terrain** và **Paint Height** và tạo hình một hòn đảo dạng như hình vẽ dưới đây



3. Vẽ texture

Unity cung cấp cho chúng ta một số các texture và các tài nguyên khác cho việc thiết kế địa hình. Để sử dụng những tài nguyên này chúng ta chọn menu **Assets | Import Package | Terrain Assets**. Chúng ta cũng có thể sử dụng các texture tự tạo nếu muốn.

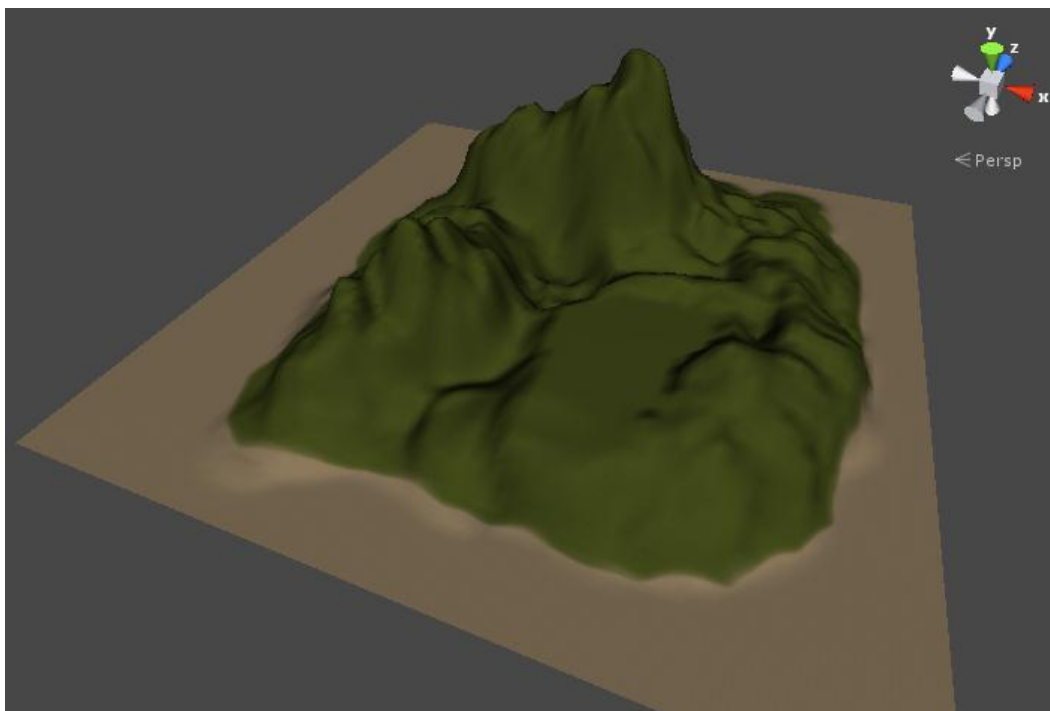
Chọn công cụ **Paint Texture** , ấn vào button **Edit Textures** và chọn **Add Texture...** Sau đó chúng ta lần lượt thêm vào các texture *Grass (Hill)*, *Good Dirt*, *Grass & Rock* và *Cliff (Layered Rock)*. Mục Normal map chúng ta để trống vì ở đây chúng ta không sử dụng Normal map. Chúng ta để nguyên giá trị size của texture là (15, 15), trừ *Cliff (Layered Rock)* chúng ta chọn size là (70, 70).



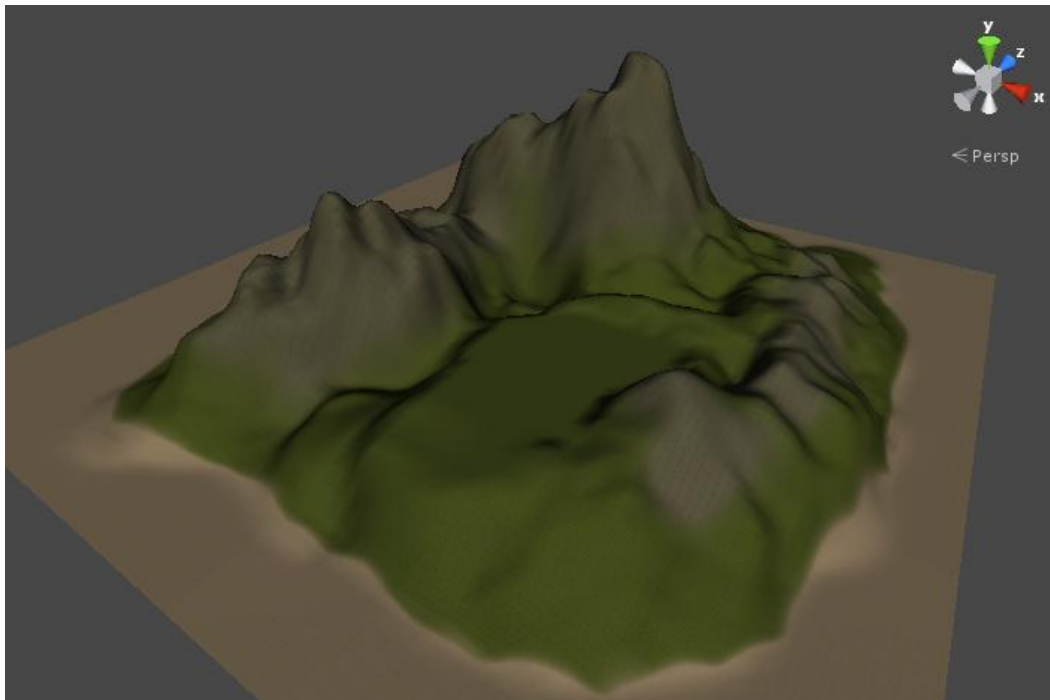
Ngay sau khi chúng ta vừa thêm vào texture đầu tiên thì toàn bộ địa hình của chúng ta sẽ được phủ texture đó. Vì địa hình của chúng ta phần lớn là cỏ xanh nên để cho dễ dàng các bạn nên thêm texture *Glass (Hill)* đầu tiên.

Tiếp theo các bạn chọn texture *Good Dirt* để tô vào những vùng xung quanh hòn đảo bằng Brush thích hợp. Lưu ý là các bạn có thể chỉnh Brush size, Opacity và Target Strength của Brush.

Kết quả thu được tương tự như hình sau:

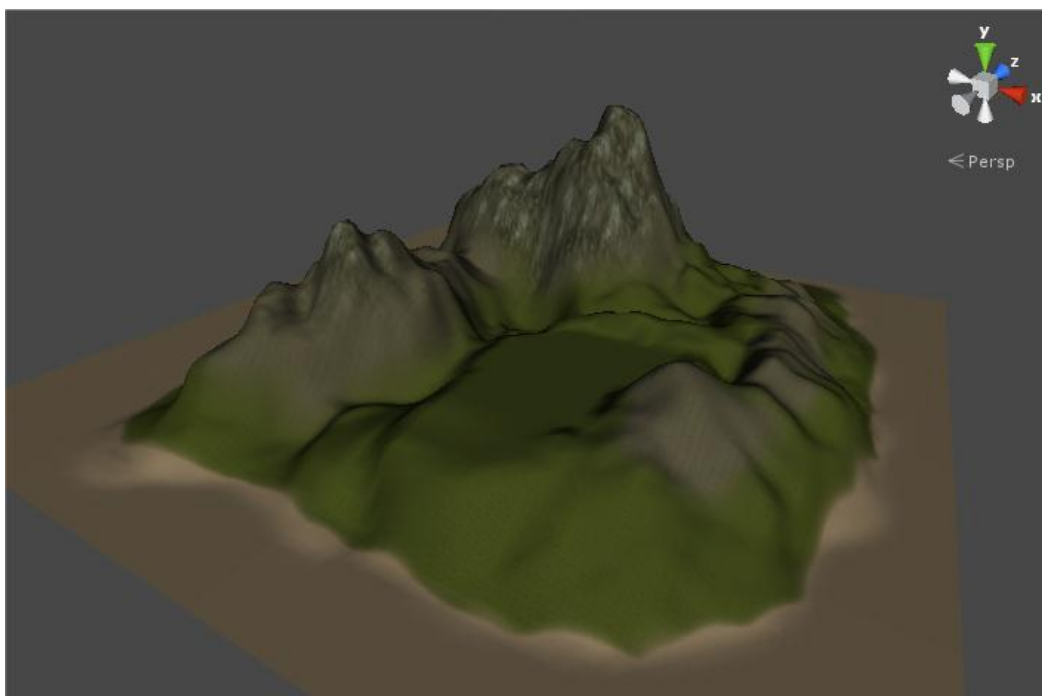


Tiếp theo, chúng ta sẽ sử dụng texture *Grass&Rock* để vẽ ở những vùng núi cao, theo tư tưởng là ở các vùng núi cao cỏ sẽ khó mọc hơn vùng đồng bằng.



Cuối cùng, chúng ta sẽ dùng texture *Cliff (Layered Rock)* để vẽ lên đỉnh núi cao nhất.

Chúng ta có được một hòn đảo (tương đối) hoàn chỉnh như sau



4. Đi dạo quanh hòn đảo

Như vậy hòn đảo của chúng ta đã tương đối hoàn tất. Giờ chúng ta có thể đi dạo một vòng quanh hòn đảo để kiểm tra những gì chúng ta đã xây dựng.


Chọn menu **Assets | Import Package | Character Controller**. Sau đó ở cửa sổ *Project* vào thư mục *Standard Assets/Character Controllers* chọn *First Person Controller* và thả vào khung cảnh. Chỉnh vị trí mong muốn cho đối tượng vừa được tạo, lưu ý là phải cao hơn bề mặt địa hình.

Để tránh xung đột ta nên xóa đối tượng *Main Camera* của game đi.

Chạy game và chúng ta đã có thể đi dạo vòng quanh hòn đảo.

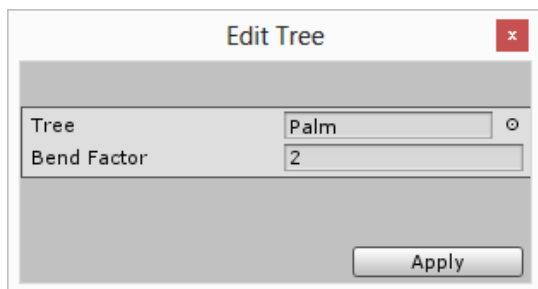
BÀI 7 – CÁC TÍNH NĂNG ĐỊA HÌNH NÂNG CAO

1. Đặt cây

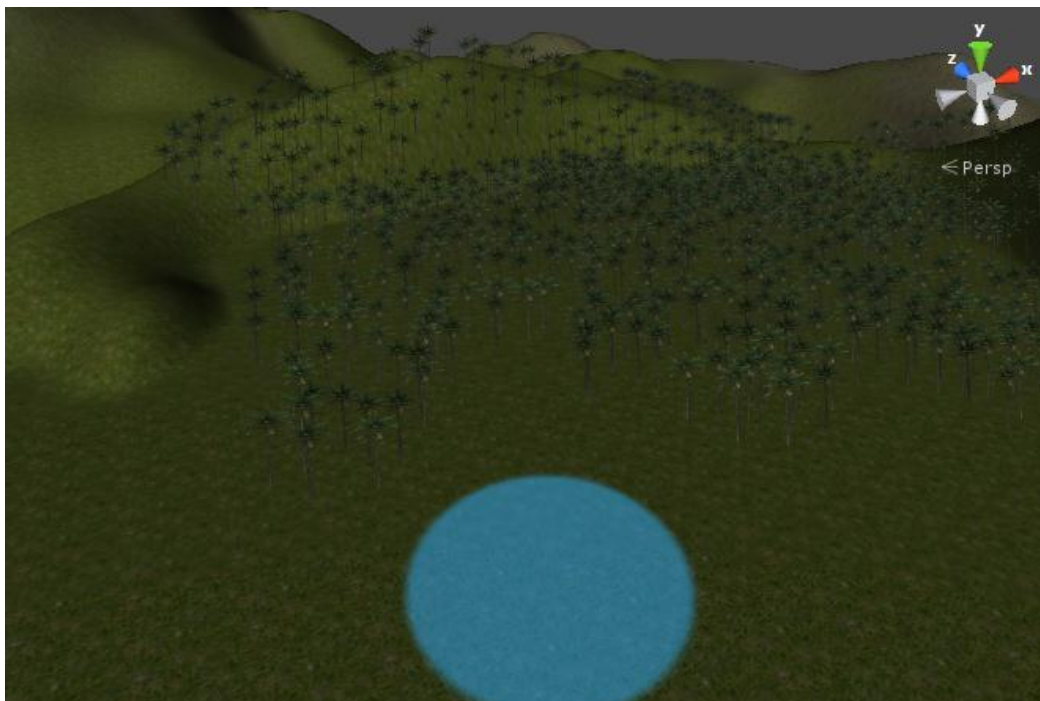
Chọn địa hình vừa tạo, ấn vào công cụ **Place Tree** 

Ấn vào **Edit Trees...** và chọn **Add Tree**


Trong cửa sổ Add Tree hiện ra, ấn vào vòng tròn bên cạnh mục Tree và chọn mô hình cây bất kỳ. Ở đây ta chọn mô hình *Palm* có sẵn trong Terrain Asset. Ta cũng có thể thay đổi giá trị *Bend Factor*, là độ tác động của gió lên độ nghiêng của cây (ví dụ: 2)



Việc đặt cây lên địa hình cũng tương tự như việc vẽ texture lên địa hình. Chú ý khi vẽ là nếu zoom quá xa thì Unity sẽ tự động không hiển thị lên cửa sổ Scene, do đó khi đặt cây lên địa hình nên zoom vào một khoảng cách nhất định

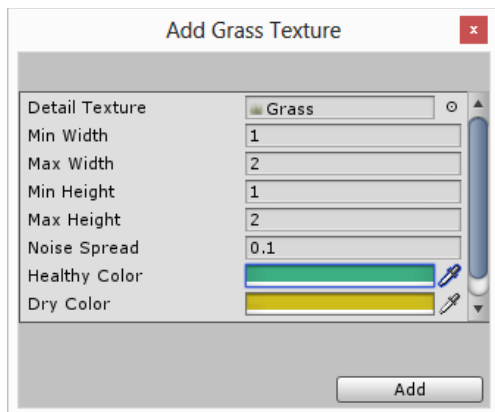


2. Vẽ các cỏ cây hoa lá lên địa hình

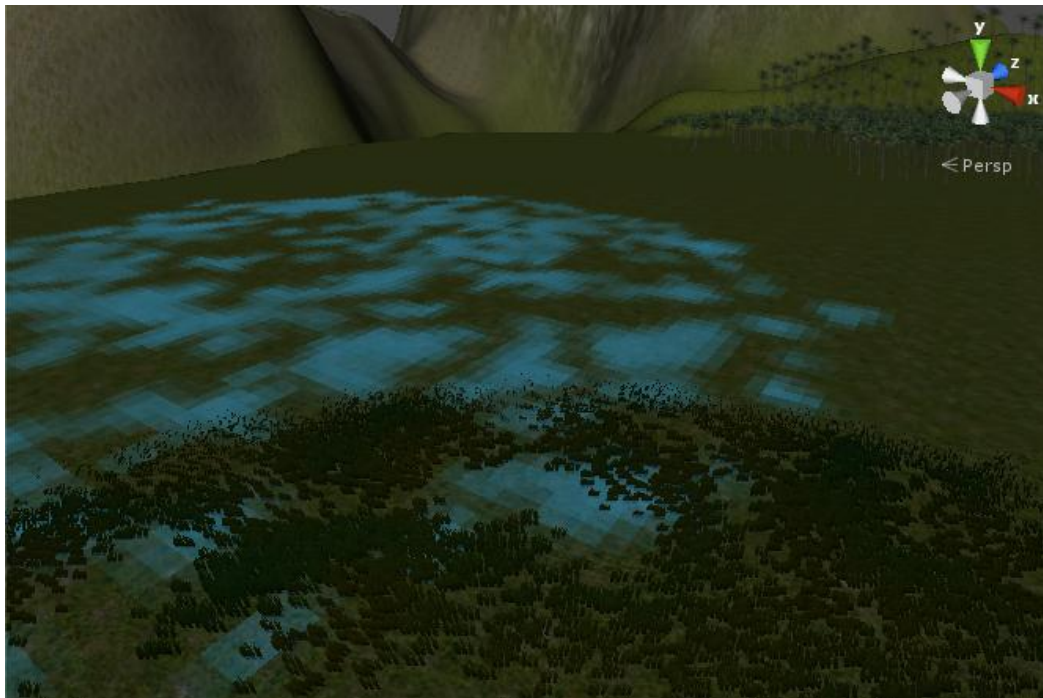
Chọn công cụ **Paint Details**  trong thanh công cụ của địa hình. Cũng tương tự như vẽ cây, chúng ta có thể dùng công cụ này để vẽ các chi tiết nhỏ như cỏ, hoa, lá... lên bề mặt địa hình. Các chi tiết này sử dụng công nghệ *billboarding*, thực chất là các texture 2D nhưng luôn quay về phía người chơi tạo cảm giác 3D.

Tương tự, ta ấn vào **Edit Details...** và chọn **Add Grass Texture**.

Tiếp theo, ta chọn *Detail Texture* là *Grass*, chọn màu sắc *Healthy Color* và *Dry Color* cho phù hợp với màu sắc của địa hình



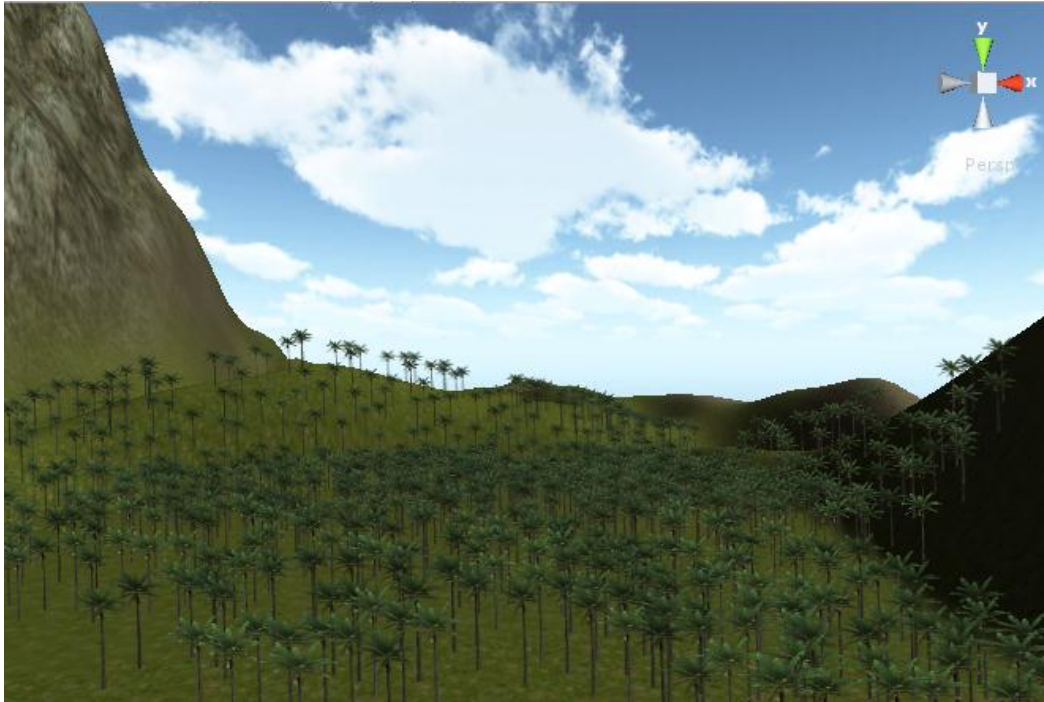
Để cỏ mọc rải rác và đồng thời cũng giảm thiểu tiêu tốn tài nguyên của máy tính, ta nên chọn Brush thưa hơn khi vẽ cỏ lên địa hình. Tương tự như khi vẽ cây cối, Unity sẽ không hiển thị các chi tiết này trừ phi chúng ta zoom vào thật gần.



3. Skybox

Chọn menu **Assets | Import Package | Skyboxes**

Chọn menu **Edit | Render Settings...** Ở cửa sổ *Inspector* ấn vào *Skybox Material* và chọn skybox phù hợp



4. Ánh sáng

Địa hình của chúng ta vừa tạo là cảnh ngoài trời, do đó chúng ta sẽ sử dụng *Directional Light* để làm nguồn ánh sáng chính cho khung cảnh. Với các loại ánh sáng khác chúng ta cũng làm tương tự.

Chọn menu **GameObject | Create Other | Directional Light**

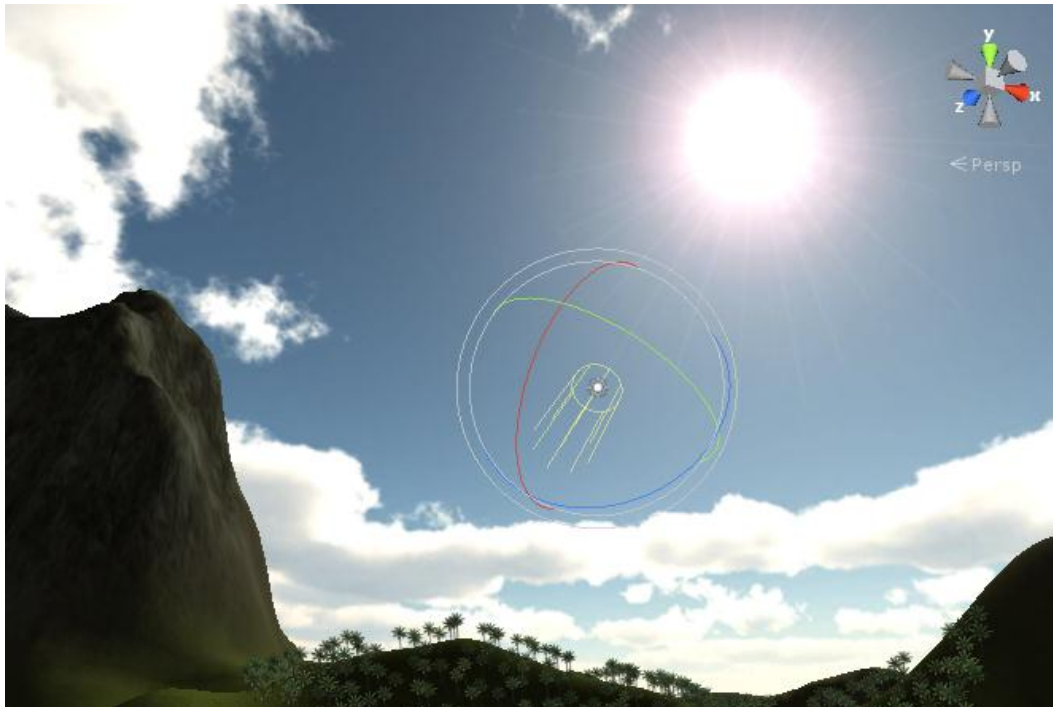
Ta có thể chọn đối tượng ánh sáng này và rotate hướng ánh sáng cho phù hợp. Mặc dù ta cũng có thể thay đổi vị trí của đối tượng này nhưng việc này không có tác dụng trong game vì ánh sáng chiếu của *Directional Light* là ánh sáng song song.

Tạo Flare cho ánh sáng

Hiệu ứng light flare là hiệu ứng được sử dụng khá nhiều trong game; tức là khi người chơi nhìn vào nguồn ánh sáng thì những quầng ánh sáng mô phỏng ánh sáng chiếu vào ống kính máy quay sẽ được tạo ra.

Để sử dụng hiệu ứng này, trước tiên ta phải import các tài nguyên tạo ánh sáng. Chọn menu **Assets | Import Package | Light Flares**

Sau đó ta chọn nguồn ánh sáng của chúng ta, như vừa tạo ở trên là *Directional Light*. Trong cửa sổ *Inspector* ta ấn vào mục *Flare* và chọn là **Sun**. Lưu ý là nếu sử dụng sun flare thì phải đặc biệt lưu ý đến hướng của ánh sáng chiếu vào cho phù hợp với skybox mình sử dụng, đặc biệt là đối với các skybox có hình ảnh mặt trời.



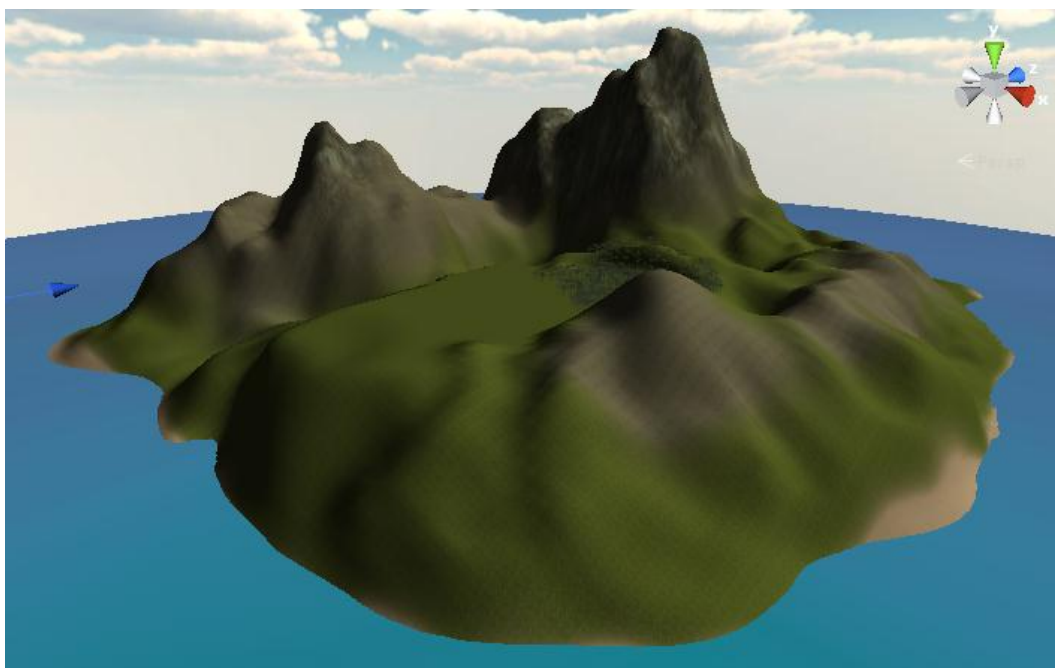
5. Nước

Chúng ta tiếp tục sử dụng địa hình mà chúng ta đã tạo để tạo biển bao quanh hòn đảo.

Chọn menu **Asset | Import Package | Water (Basic)**

Chọn *Daylight Simple Water* trong thư mục *Standard Assets/Water (Basic)* và thả vào cửa sổ Scene. Chỉnh vị trí sao cho độ cao y cao hơn mặt đất và tâm đặt tại chính giữa địa hình. Sau đó phóng to đối tượng Water để bao trùm cả địa hình.

Kết quả thu được như hình dưới



Ta có thể chạy thử game để kiểm tra kết quả

