A person with brown hair is shown in profile, focused on a laptop screen. The background is dark blue with floating white binary code (0s and 1s) and glowing rectangular frames, creating a digital or gaming atmosphere.

Bài 5: Thêm các hiệu ứng môi trường : Water Effects; Skybox Effects; Ambient và Spatial Sounds

Mục tiêu

- Thêm hiệu ứng môi trường nước
- Thêm hiệu ứng môi trường bầu trời
- Môi trường xung quanh và không gian âm thanh
- Mass/Gravity

Thêm hiệu ứng môi trường nước

- Yêu cầu: Đã tạo xong địa hình cơ bản
- Mục đích: Tạo ra môi trường nước cho game
- Các bước thực hiện:
 1. Tạo một dự án Unity3D mới với tên dự án là **WaterWorld**
 2. Tạo một địa hình mới. Đặt các giá trị chiều dài địa hình và chiều rộng là 1000. Tạo và định vị ánh sáng trực tiếp. Thêm một Texture Hill Grass
 3. Chiều cao địa hình cần phải bắt đầu từ mặt đất, chứ không phải từ số 0, đó là mặc định cho các địa hình mới. Ta sẽ tạo ra một hiệu ứng bờ biển với cao độ địa hình không phải là đáy biển, sau đó mặt đất nên được nâng lên là chiều cao bề mặt của hòn đảo. Ta chọn Terrain-Flatten heightmap.

Thêm hiệu ứng môi trường nước

- 4. Nhấp vào bên của **Height** box và đặt một giá trị là 30 mét, và sau đó nhấn Enter để xác nhận. Click vào **Flatten** để kết thúc. Địa hình có thể được san phẳng xung quanh bằng cách lấy nghịch đảo **Raise Height** (tăng chiều cao) và đặt một hòn đảo lớn lên ở trung tâm của địa hình.
- 5. Trên cửa sổ **Inspector** của đối tượng địa hình, thành phần Script, chọn công cụ **Raise Height** để tạo nút đầu tiên trong bảy nút. Chọn **brush** đầu tiên trong bảng, và thiết lập **Brush Size** của nó là 75. Set **Opacity** của brush là 50. Thay đổi khung nhìn của mình trong scene bằng cách nhấp vào trục Y của Gizmo phần top-right

Thêm hiệu ứng môi trường nước

- 6. Sử dụng phím Shift để giảm chiều cao, vẽ xung quanh để phác thảo địa hình, tạo ra một đường bờ biển sao cho chiều cao là 0
- 7. Thêm một texture sandy vào vùng đại diện cho các bãi biển
- 8. Trong package Standard Assets đã cung cấp cho chúng ta 2 bề mặt với chất liệu là nước. Ta cần lưu với dạng prefabs. Đơn giản nhất ta mở **Standard Assets - Water** folder. Kéo prefabs **Daylight Simple Water** vào scene và thiết lập vị trí cho nó là **(500, 4, 500)** với các trục X, Y, Z tương ứng. Bạn sẽ thấy giá trị này trong component **Transform** trong cửa sổ **Inspector**.
- Để mở rộng tỷ lệ của prefabs nhằm tạo biển quanh đảo, ta tăng cả 2 trục **X** và **Z** với **Scale** là **1600**.
- Điều này làm nước ở trung tâm của bản đồ cao hơn 4 mét so với đáy biển. Nó sẽ bao phủ các góc của đảo và tạo mặt nạ thiên nhiên cho địa hình.

Thêm hiệu ứng môi trường nước

- 8. Tạo một First Person Controller trong scene và khám phá thế giới của bạn.
- 9. Sử dụng các công cụ địa hình điêu khắc (sculpting), thử một số tính năng Carving nước nội địa - ví dụ hồ, suối bằng cách giảm chiều cao mặt đất để lộ ra làn nước.
- 10. Hãy thử thêm một hiệu ứng nước trong world khác

Thêm hiệu ứng môi trường bầu trời (skybox)

- Hướng dẫn này giả định rằng bạn đã thực hiện các hướng dẫn về việc tạo ra và texturing một địa hình cơ bản.
- Phần này sẽ cho bạn thấy làm thế nào để tạo ra một hiệu ứng skybox.
- 1. Bắt đầu một dự án Unity3D mới và đặt tên nó Skyworld
- 2. Tạo một địa hình mới. Đặt các giá trị chiều dài độ phân giải địa hình và chiều rộng là 1000. Tạo và định vị ánh sáng trực tiếp. Thêm một texture Hill Grass. Trong việc tạo ra môi trường 3D, chân trời hoặc khoảng cách được thể hiện bằng việc bổ sung một skybox. Một skybox là một cubemap - một loạt sáu kết cấu được đặt bên trong một khối lập phương và kết xuất liên tục xuất hiện như một bầu trời xung quanh và đường chân trời. Các cubemap bao gồm các thể giới 3D liên tục và như với đường chân trời thực tế, nó không phải là một đối tượng mà người chơi có thể truy cập.

Môi trường xung quanh và không gian âm thanh

- Hướng dẫn này giả định rằng bạn đã thực hiện xong hướng dẫn về việc tạo ra một địa hình cơ bản
- Hoạt động này sẽ cho bạn thấy làm thế nào để thêm âm thanh vào môi trường; một âm thanh xung quanh được nghe ở mức âm lượng như nhau trong cả môi trường.
- Âm thanh 3D có thể được gắn vào một GameObject để cung cấp âm thanh cho cả không gian.

Môi trường xung quanh và không gian âm thanh

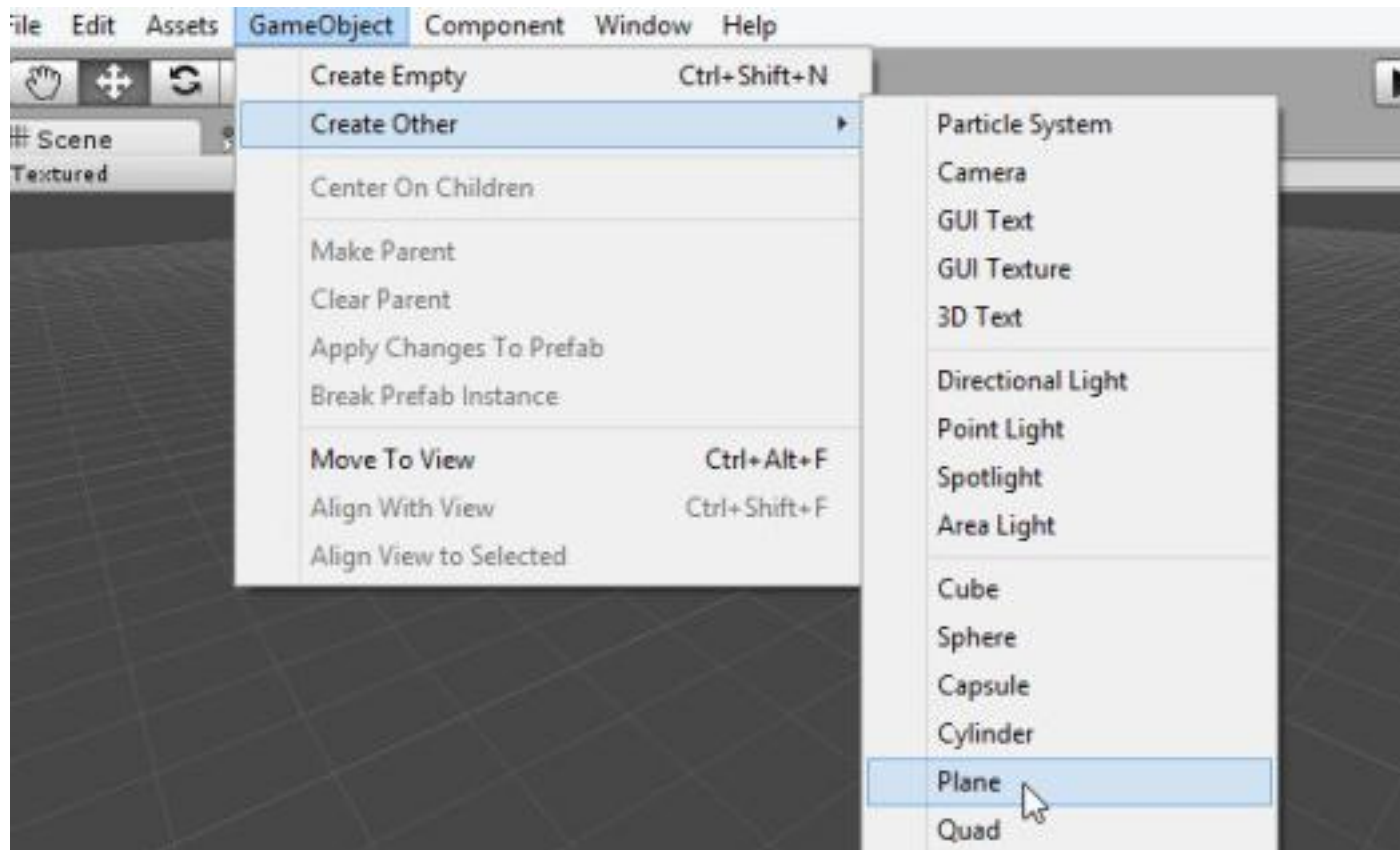
- 1. Load một dự án Unity 3D từ một dự án trước.
- 2. Tải về hai đoạn âm thanh được cung cấp `ambience.aif` và `beacon.aiff`. Import cả 2 tập tin này
- 3. Chọn các clip âm thanh `ambience.aiff` trong cửa sổ Project và đảm bảo âm thanh 3D không được đánh dấu (**unchecked**). Kéo và thả các clip âm thanh `ambience.aiff` từ cửa sổ dự án vào Scene. Cửa sổ Inspector sẽ hiển thị các thành phần âm thanh vừa được thêm vào. Hãy chắc chắn rằng **Play On Awake** được **checked** và **Loop** được **checked**. Chạy kịch bản, bạn sẽ nghe thấy một hiệu ứng âm thanh xung quanh môi trường.

Môi trường xung quanh và không gian âm thanh

- 4. Tạo một Cube GameObject và đặt tên cho nó là SonicBeacon. Chọn GameObject SonicBeacon trong cửa sổ Hierarchy, các thuộc tính xuất hiện trong cửa sổ **Inspector**. Kéo và thả clip âm thanh SonicBeacon.aiff vào cửa sổ **Inspector**. Chọn clip âm thanh SonicBeacon.aiff trong cửa sổ project và đảm bảo âm thanh 3D được chọn trong cửa sổ **Inspector**. Clip âm thanh bây giờ được gắn vào khối lập phương và bất cứ nơi nào của khối được gán âm thanh sẽ phát ra âm thanh tương tự

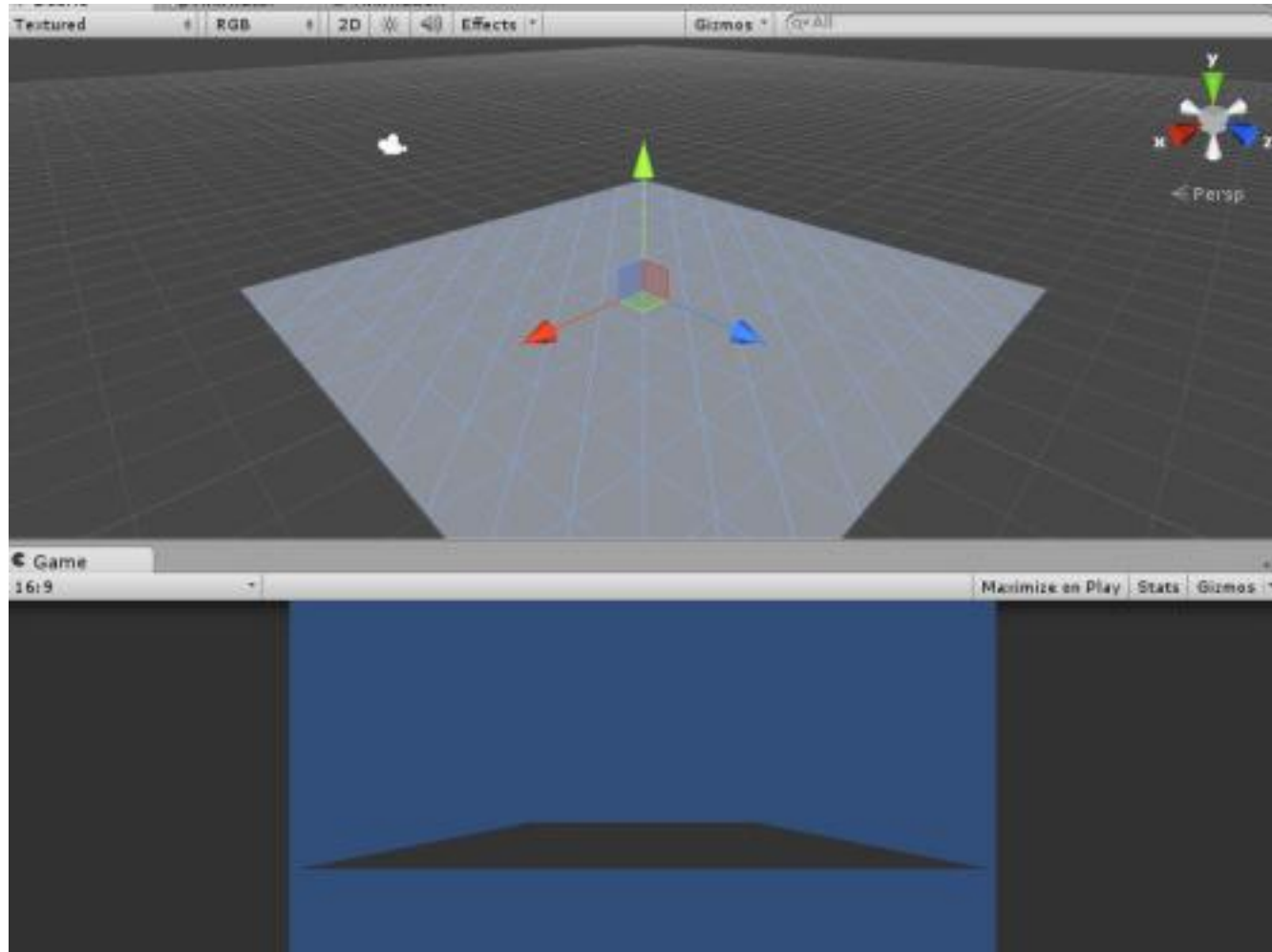
Khái niệm về Mass / Gravity

- Việc đầu tiên là chúng ta cần tạo 1 **scene**, mặc định sẽ có cho chúng ta rồi
- Tiếp theo ta cần tạo một Plane, làm như hình dưới



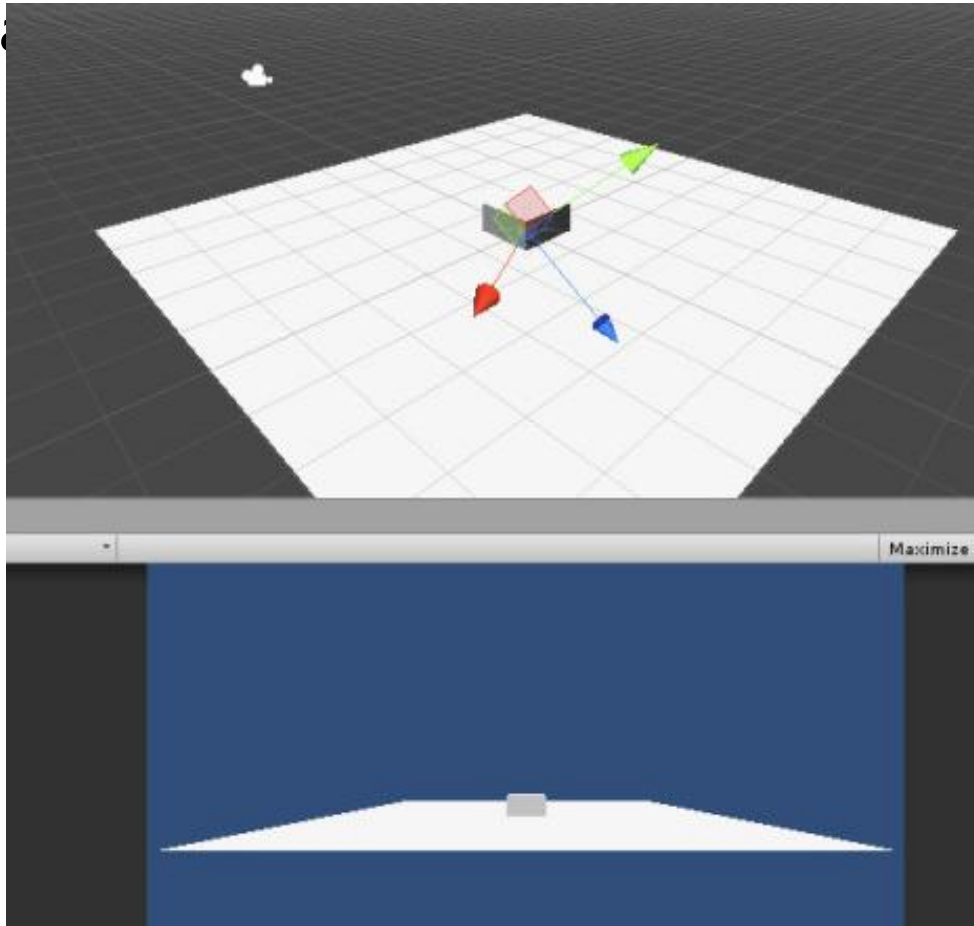
Khái niệm về Mass / Gravity

- Sau khi tạo xong thì bạn có được hình như dưới:



Khái niệm về Mass / Gravity

- Tiếp theo là ta tạo một Cube tương tự như bước tạo Plane, sau khi tạo xong cube đồng thời ta thêm một nguồn sáng: Nhấn Game Object -> Create Other -> Directional



Nguồn sáng

- Hầu như chúng ta phải thêm nguồn sáng cho các bối cảnh của Game.
- Có ba kiểu nguồn sáng cơ bản, mỗi nguồn sáng khác nhau sẽ thay đổi sắc thái của game rất nhiều.



Nguồn sáng

■ *Dirictional Light:*

- Có thể hiểu là với nguồn sáng này ta có thể định hướng chiếu cho nó, các tia chiếu là song song.

■ *Point Light:*

- là điểm sáng, thường dùng để tạo các điểm sáng nhỏ trên các Character, hay các cảnh trong Game theo đúng ý tưởng của chủ nhân.

■ *Spotlight:*

- Đèn tại chỗ chỉ chiếu theo một chiều hình nón, thường dùng là hiệu ứng chiếu cho đèn các loại xe, đèn pin, ánh sáng hắt vào object trong các cảnh...

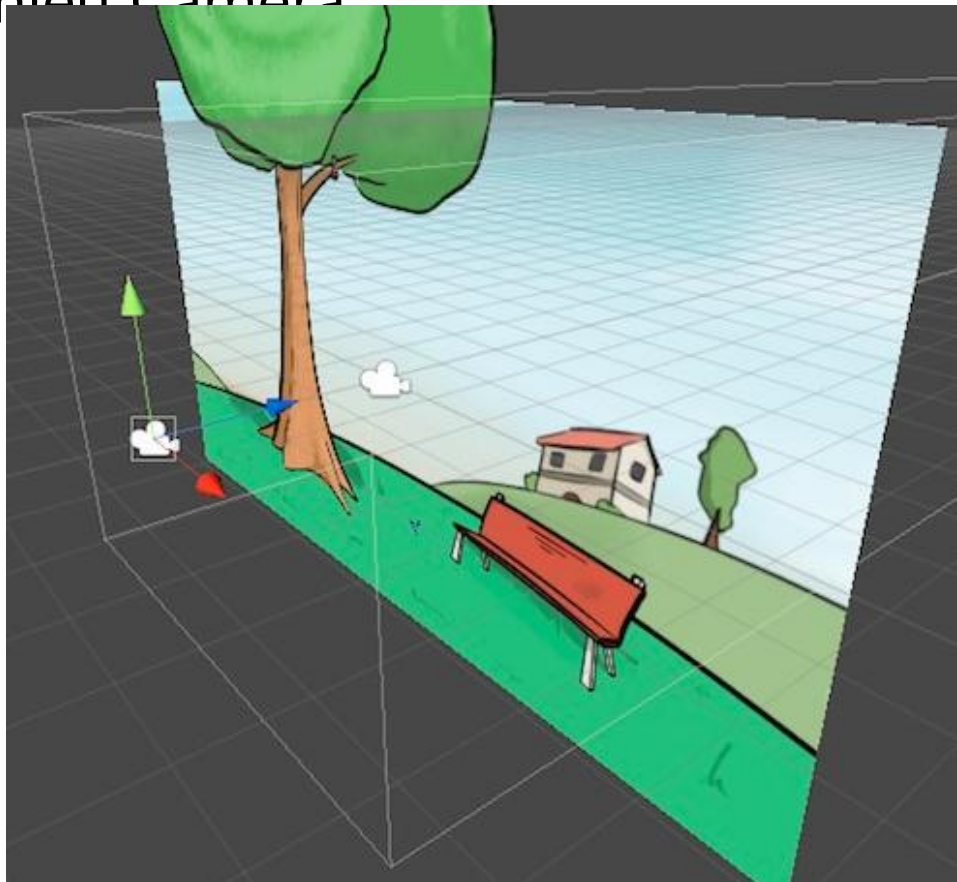
Camera với địa hình

- Như Camera được sử dụng trong các bộ phim , Camera trong Unity cũng được dùng để hiển thị game trên thế giới cho người chơi. Nó cũng được coi là một Game Object trong Unity. Bạn có thể xoay, di chuyển.... tùy chỉnh nó theo ý tưởng của bạn.



Camera với địa hình

- Camera được sử dụng để hiển thị cảnh trong game, chúng ta có thể làm cho game của mình trở nên độc đáo hơn nhờ tùy chỉnh Camera. Trong một cảnh, chúng ta có thể có một hoặc rất nhiều Camera



Đặc điểm của Camera

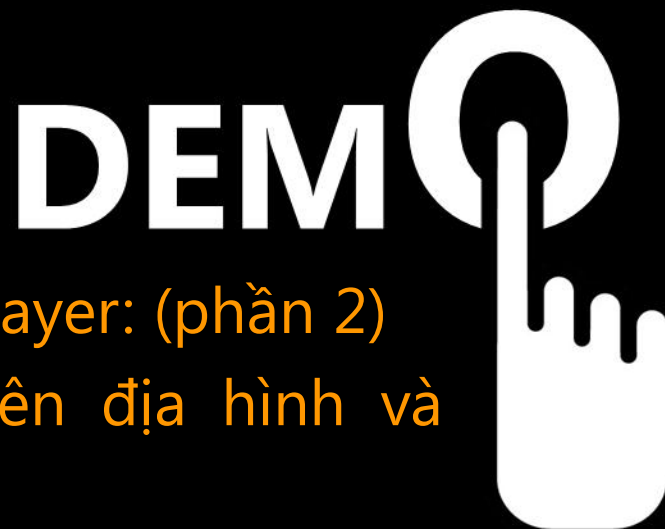
- Clear Flags: Xác định các bộ phận mà màn hình sẽ bị xóa. Thuận tiện khi sử dụng nhiều máy ảnh và để vẽ nhiều đối tượng khác nhau. Không xóa nó sẽ hiển thị màu đen
- Background: Màu nền cho phần màn hình còn lại
- Culling Mask: Chỉ định các lớp đối tượng của bạn trong Inspector. Cho phép hoặc bỏ qua các đối tượng được hiển thị trong Camera

Đặc điểm của Camera

- Projection:
 - Perspective: Camera hiển thị các đối tượng theo phối cảnh tròn vện.
 - Orthographic: Hiển thị các đối tượng như một thể thống nhất, không theo nghĩa của phối cảnh(Perspective).
- Size: Kích thước quan sát của Camera khi chọn phép chiếu là Orthographic.
- Field of view: Chiều rộng của góc nhìn Camera. Được đo bằng độ dọc theo trục Local Y.

Đặc điểm của Camera

- Depth:
 - Vị trí Camera có một giá trị lớn hơn sẽ được hiển thị lên đầu. Tức là cái nào có giá trị lớn hơn thì nó sẽ được ưu tiên được hiển thị lên màn hình.
- Rendering path: Tùy chọn cho việc xác định phương pháp vẽ những gì sẽ được hiển thị lên Camera.
 - Use Player Setting: Sử dụng cài đặt Player
 - Vertex Lit: tất cả các đối tượng được đưa ra bởi máy ảnh sẽ được trả lại như đối tượng Vertex-Lit
 - Forward: tất cả các đối tượng sẽ được trả về như một tài liệu



Xây dựng scene Player: (phần 2)
Character chạy trên địa hình và
ăn coin

Cài đặt Player_Get

- Tìm đối tượng game trong Scene

```
GameObject a = GameObject.Find("05_GameManager");  
if(a!=null)_gm = a.GetComponent<Game_Manager>();
```

```
GameObject b = GameObject.Find("Black_Screen");  
if(b!=null)_fade = b.GetComponent<Fade>();
```

Cài đặt Player_Get

- Sự kiện Trigger

```
void OnTriggerEnter (Collider Get)
{
    //nếu là tag "coin"
    if (Get.tag == "coin") {
        Get.gameObject.SetActive (false);
        Get_Coin_Count += 1;
        if (_gm != null)
            _gm.GETCOIN ();
        if (_SP != null)
            _SP.SoundPlay (1);
    }
}
```

Cài đặt Player_Get

- Sự kiện Trigger

```
//nếu là tag "DeathZone"
if (Get.tag == "DeathZone") {
    Debug.Log ("Die");
    if (_PM.status != PlayerMoveStatus.Die) {
        _PM.status = PlayerMoveStatus.Die;
        //Handheld.Vibrate ();
        this.gameObject.rigidbody.AddForce (0, -50f, 0);
        if (_gm != null)
            _gm.GAMEOVER ();

        if (_SP != null)
            _SP.SoundPlay (2);
        if (_fade != null)
            _fade.FadeOut ();
    }
}
```


Cài đặt Sound_Player

- Player sound theo hành động của character:

```
public AudioClip[] Sound;

public void SoundPlay(int SoundNumber){
    audio.clip = Sound[SoundNumber];
    audio.Play();
}
```

Cài đặt Sprite_Animation

- Khai báo các biến cần sử dụng

```
public Texture[] Run_Image;  
int _now_run_ani_count;  
public Texture[] Jump_Image;  
int _now_jump_ani_count;  
public Texture[] D_Jump_Image;  
int _now_d_jump_ani_count;  
public float Ani_Play_Speed;  
float Ani_Play_Time;  
public bool _run;  
public bool _jump;  
public bool _d_jump;
```

Cài đặt Sprite_Animation

- Hàm Run_ANI

```
void RUN_ANI_ing ()  
{  
    _now_run_ani_count += 1;  
    if (_now_run_ani_count >= Run_Image.Length) {  
        _now_run_ani_count = 0;  
    }  
}
```

Cài đặt Sprite_Animation

- Hàm Jump_ANI

```
void JUMP_ANI_ing ()
{
    _now_jump_ani_count += 1;
    if (_now_jump_ani_count >= Jump_Image.Length) {
        Run_Play ();
        return;
    }
    this.gameObject.renderer.material.mainTexture = Jump_Image [_now_jump_ani_count];
}
```

Cài đặt Sprite_Animation

- Hàm D_JUMP_ANI

```
void D_JUMP_ANI_ing ()
{
    _now_d_jump_ani_count += 1;
    if (_now_d_jump_ani_count >= D_Jump_Image.Length) {
        Run_Play ();
        return;
    }
    this.gameObject.renderer.material.mainTexture = D_Jump_Image [_now_d_jump_ani_count];
}
```

Cài đặt Sprite_Animation

- Hàm Run_Play

```
public void Run_Play ()  
{  
    _run = true;  
    _jump = false;  
    _d_jump = false;  
    _now_run_ani_count = 0;  
}
```

Cài đặt Sprite_Animation

- Hàm Jump_Play

```
public void Jump_Play ()  
{  
    _run = false;  
    _jump = true;  
    _d_jump = false;  
    _now_jump_ani_count = 0;  
}
```

Cài đặt Sprite_Animation

- Hàm D_Jump_Play

```
public void D_Jump_Play ()  
{  
    .....  
    _run = false;  
    _jump = false;  
    _d_jump = true;  
    _now_d_jump_ani_count = 0;  
  
}
```


Cài đặt Sprite_Animation

■ Hàm Update

```
void Update ()
{
    Ani_Play_Time += Time.deltaTime;

    if (Ani_Play_Time >= Ani_Play_Speed) {
        Ani_Play_Time = 0;
        if (_run == true) {
            RUN_ANI_ing ();
        }

        if (_jump == true) {
            JUMP_ANI_ing ();
        }

        if (_d_jump == true) {
            D_JUMP_ANI_ing ();
        }
    }
}
```

Tổng kết

- Thêm hiệu ứng môi trường nước
- Thêm hiệu ứng môi trường bầu trời
- Môi trường xung quanh và không gian âm thanh



FPT POLYTECHNIC

THANK YOU!

www.poly.edu.vn