

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN ĐIỆN TỬ - VIỄN THÔNG**



***BÁO CÁO BÀI TẬP LỚN***  
**KỸ THUẬT LẬP TRÌNH NỘI DUNG SỐ**

**Đề tài: Tổng hợp mô hình đồ họa 3D (1A – 2A – 4B)**

**Giảng viên hướng dẫn: TS. Phạm Văn Tiến**

Nhóm sinh viên thực hiện:

STT	HỌ VÀ TÊN	MSSV	LỚP - KHÓA	MÃ LỚP	KÝ TÊN
1	Bùi Việt Anh	20182329	Điện tử 07 – K63	133403	
2	Lưu Bình Minh	20182683	Điện tử 06 – K63	133403	

Hà Nội, 07/2022

# MỤC LỤC

DANH MỤC HÌNH ẢNH.....	3
LỜI NÓI ĐẦU .....	4
CHƯƠNG I: TỔNG HỢP ĐỐI TƯỢNG ĐỒ HỌA 3D .....	5
I.1. XÂY DỰNG ĐỐI TƯỢNG ĐỒ HỌA 3d.....	5
I.1.1. Main character (Human).....	5
I.1.2. Dragon .....	8
I.1.3. Tree .....	8
I.2. ĐÓNG GÓI THÀNH API .....	9
CHƯƠNG II: XÂY DỰNG PHỐI CẢNH KHÔNG GIAN 3D.....	10
II.1. XÂY DỰNG PHỐI CẢNH SỬ DỤNG ADD-ON BLENDER-OSM.....	10
II.1.1. Tải và cài đặt add-on blender-osm.....	10
II.1.2. Xây dựng phối cảnh 3D.....	12
II.1.3. Hoàn thiện phối cảnh 3D .....	13
II.1.4. Một phối cảnh 3D khác .....	16
II.2. ĐÓNG GÓI THÀNH API.....	17
CHƯƠNG III: TỔNG HỢP & LẬP TRÌNH ĐIỀU KHIỂN TƯƠNG TÁC.....	19
III.1. NHÚNG CÁC ĐỐI TƯỢNG VÀ PHỐI CẢNH VÀO PROJECT UNITY .....	19
III.1.1. Chuyển động của main character .....	19
III.2. CODE XÂY DỰNG TƯƠNG TÁC GIỮA CÁC ĐỐI TƯỢNG.....	20
III.2.1. Code di chuyển main character.....	20
III.2.2. Âm thanh khi di chuyển của main character.....	21
III.2.3. Hiệu ứng của main character .....	22
III.2.4. Tương tác giữa main character và đối tượng tree.....	22
III.2.5 Toàn cảnh project và một số tương tác khác .....	23
PHỤ LỤC & MÃ NGUỒN.....	25

# DANH MỤC HÌNH ẢNH

<i>Hình 1.1: Mô hình 3D của đối tượng (main character)</i> .....	5
<i>Hình 1.2: Khung xương (rig body) của mô hình 3D</i> .....	5
<i>Hình 1.3: Ảnh xạ kết cấu mô hình 3D trên mặt phẳng 2D</i> .....	6
<i>Hình 1.4: Ảnh xạ kết cấu mô hình 3D trên mặt phẳng 2D</i> .....	6
<i>Hình 1.5: Mô hình 3D sau khi thêm chất liệu</i> .....	7
<i>Hình 1.6: Tạo animation với rig body</i> .....	7
<i>Hình 1.7: Các animation của đối tượng Main character (idle, run, jump)</i> .....	8
<i>Hình 1.8: Mô hình Dragon sau khi hoàn thiện</i> .....	8
<i>Hình 1.9: Mô hình đối tượng tree sau khi hoàn thiện</i> .....	9
<i>Hình 1.10: Cấu trúc file (.fbx) sau khi export cho đối tượng 3D</i> .....	9
<i>Hình 2.1: Giao diện Add-ons trên phần mềm Blender</i> .....	10
<i>Hình 2.2: Thực hiện install add-on blender-osm</i> .....	11
<i>Hình 2.3: Kết quả sau khi install thành công add-on blender-osm</i> .....	11
<i>Hình 2.4: Chọn khu vực sử dụng để xây dựng phối cảnh 3D</i> .....	12
<i>Hình 2.5: Thực hiện import building cho phối cảnh</i> .....	12
<i>Hình 2.6: Phối cảnh sau khi import building và terrain</i> .....	13
<i>Hình 2.7: Hoàn thiện terrain cho phối cảnh 3D (1)</i> .....	14
<i>Hình 2.8: Hoàn thiện terrain cho phối cảnh 3D (2)</i> .....	15
<i>Hình 2.9: Hoàn thiện building cho phối cảnh 3D</i> .....	15
<i>Hình 2.10: Phối cảnh 3D hoàn chỉnh.</i> .....	16
<i>Hình 2.11: Phối cảnh 3D phụ</i> .....	16
<i>Hình 2.12: Kết quả thực hiện render phối cảnh 3D phụ (1)</i> .....	17
<i>Hình 2.13: Kết quả thực hiện render phối cảnh 3D phụ (1)</i> .....	17
<i>Hình 2.14: Cấu trúc file (.fbx) sau khi export cho phối cảnh 3D</i> .....	18
<i>Hình 3.1: Kết quả sau khi nhúng các đối tượng và phối cảnh 3D vào project Unity</i> .....	19
<i>Hình 3.2: Đồ hình tương tác giữa các animation và các biến isRunning, isJumping</i> .....	19
<i>Hình 3.3: Lập trình thiết lập âm thanh khi di chuyển cho main character</i> .....	22
<i>Hình 3.4: Lập trình thiết lập hiệu ứng cho main character</i> .....	22
<i>Hình 3.5: Thiết lập thuộc tính vật cản cho tree và character</i> .....	23
<i>Hình 3.6: Lập trình cho đối tượng tree bốc cháy khi main character va chạm</i> .....	23
<i>Hình 3.7: Toàn cảnh 3D Project (1)</i> .....	24
<i>Hình 3.8: Toàn cảnh 3D Project (2)</i> .....	24

# LỜI NÓI ĐẦU

Trong khuôn khổ môn học “Kỹ thuật lập trình nội dung số”, học kỳ 20212, sau khi bàn bạc và thống nhất, nhóm chúng em đã quyết định lựa chọn đề tài bài tập lớn (DM2P – Subproject) với các nội dung:

1. Tổng hợp đối tượng đồ họa 3D theo cách tương tự như đã làm với FaceBuilder. Sau đó, có thể trang trí đối tượng này theo ý muốn (thêm màu sắc, chất liệu, chiếu sáng...). Khi trình bày đối tượng, cho phép người dùng xoay góc nhìn (bằng chuột, phím nhấn, chạm tay) và có 1 trong các tính năng tương tác với người dùng (dùng chuột, gõ phím, chạm tay...) sau:
  - 1A – Có chuyển động cục bộ (chạy, nhảy, xòe/cụp...).
2. Xây dựng phối cảnh không gian 3D theo một trong 2 cách sau:
  - 2A – Tải về bản đồ số 3D của một khu vực mở (khu đô thị, khu rừng, bãi chiến trường...) thể hiện rõ địa hình địa vật (vách núi, tòa nhà, đường sá...) sử dụng OSM. Có thể trang trí cho các chi tiết của bản đồ thêm sống động và/hoặc bổ sung thêm các đối tượng, hình ảnh giả tưởng.
3. Sử dụng các công cụ lập trình nội dung số như Blender, Unity, Google để gia công các tư liệu đầu vào nói trên, nhúng đối tượng 3D nói trên (được tạo trong Mục 1) vào phối cảnh vừa được xây dựng.
4. Sử dụng API được đóng gói trong Mục 1, lập trình cho phép người dùng điều khiển quỹ đạo chuyển động của đối tượng 3D bằng chuột hoặc phím nhấn trong bản đồ số nói trên, và trong quá trình chuyển động, đối tượng cho phép người dùng tương tác theo tính năng đã chọn. Đối tượng còn phát ra âm thanh 3D trầm bổng, to nhỏ, phỏng tạo phản xạ âm từ các vật thể xung quanh. Ngoài ra, thực hiện thêm một trong các công việc sau:
  - 4B – Đối tượng 3D có đáp ứng với một vài tình huống cụ thể, ví dụ khi người dùng hạ thấp độ cao để nó va vào cao ốc thì bốc cháy cả hai.

Nhóm chúng em đã sử dụng Blender để tổng hợp, xây dựng phối cảnh, đối tượng 3D đồng thời đóng gói và nhúng các đối tượng và phối cảnh đã tạo vào trong Unity tạo thành một project thống nhất, dưới sự hướng dẫn của thầy Phạm Văn Tiến.

Bài báo cáo của nhóm em gồm các nội dung chính sau:

- Chương I: Tổng hợp đối tượng đồ họa 3D.
- Chương II: Xây dựng phối cảnh không gian 3D.
- Chương III: Tổng hợp & Lập trình điều khiển tương tác.

# CHƯƠNG I: TỔNG HỢP ĐỐI TƯỢNG ĐỒ HỌA 3D

## I.1. XÂY DỰNG ĐỐI TƯỢNG ĐỒ HỌA 3D

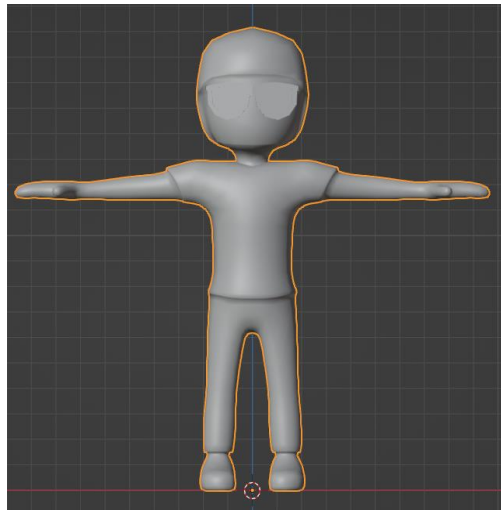
Nhóm thực hiện xây dựng 3 đối tượng đồ họa bao gồm: main character (human), dragon và tree. Trong đó, hai đối tượng dragon và tree là những đối tượng tĩnh dùng để trang trí thêm cho phối cảnh, riêng đối tượng tree sẽ là đối tượng tương tác chính với Main Character (được trình bày tại Chương III).

### I.1.1. Main character (Human)

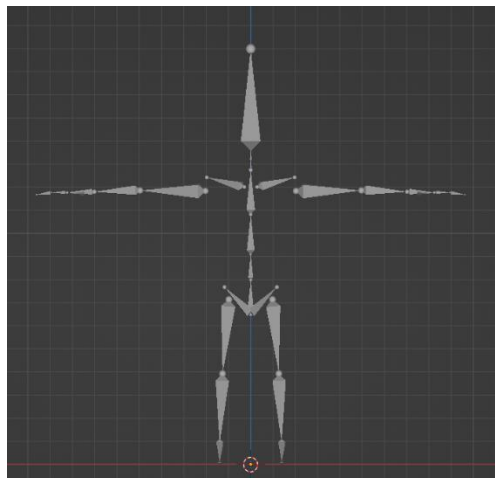
#### I.1.1.1. Xây dựng mô hình

Main character bao gồm 2 bộ phận chính:

- Mô hình 3D: Dùng để trang trí (được sử dụng để trang trí: thêm màu sắc, chất liệu...)
- Khung xương (rig body): Dùng để tạo chuyển động cục bộ (pose), làm cơ sở cho các animation của mô hình.



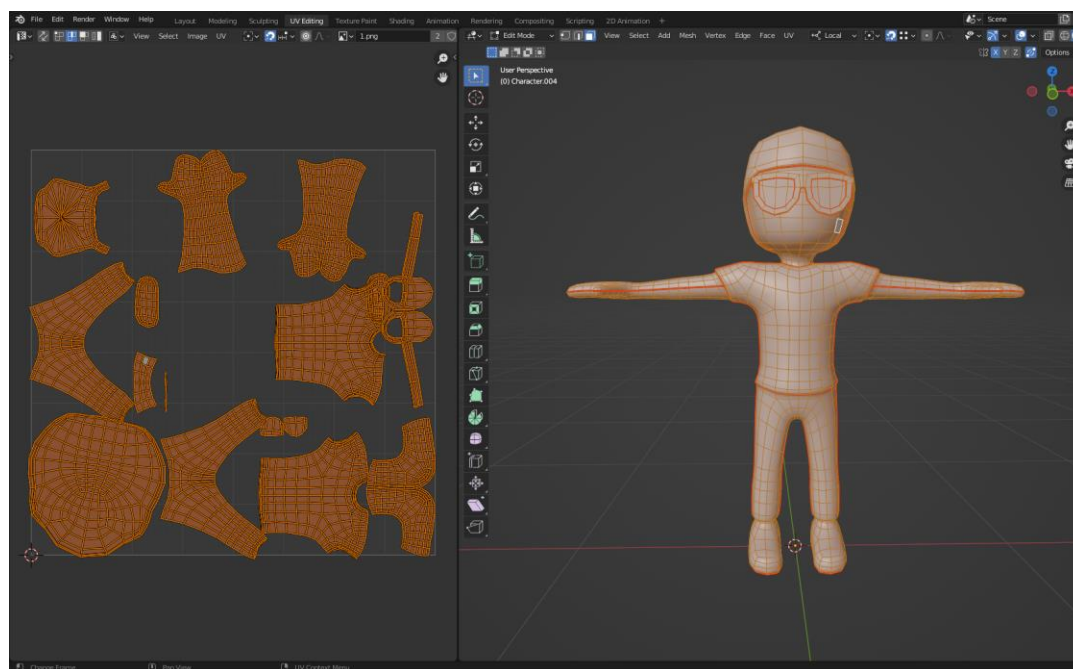
*Hình 1.1: Mô hình 3D của đối tượng (main character)*



*Hình 1.2: Khung xương (rig body) của mô hình 3D*

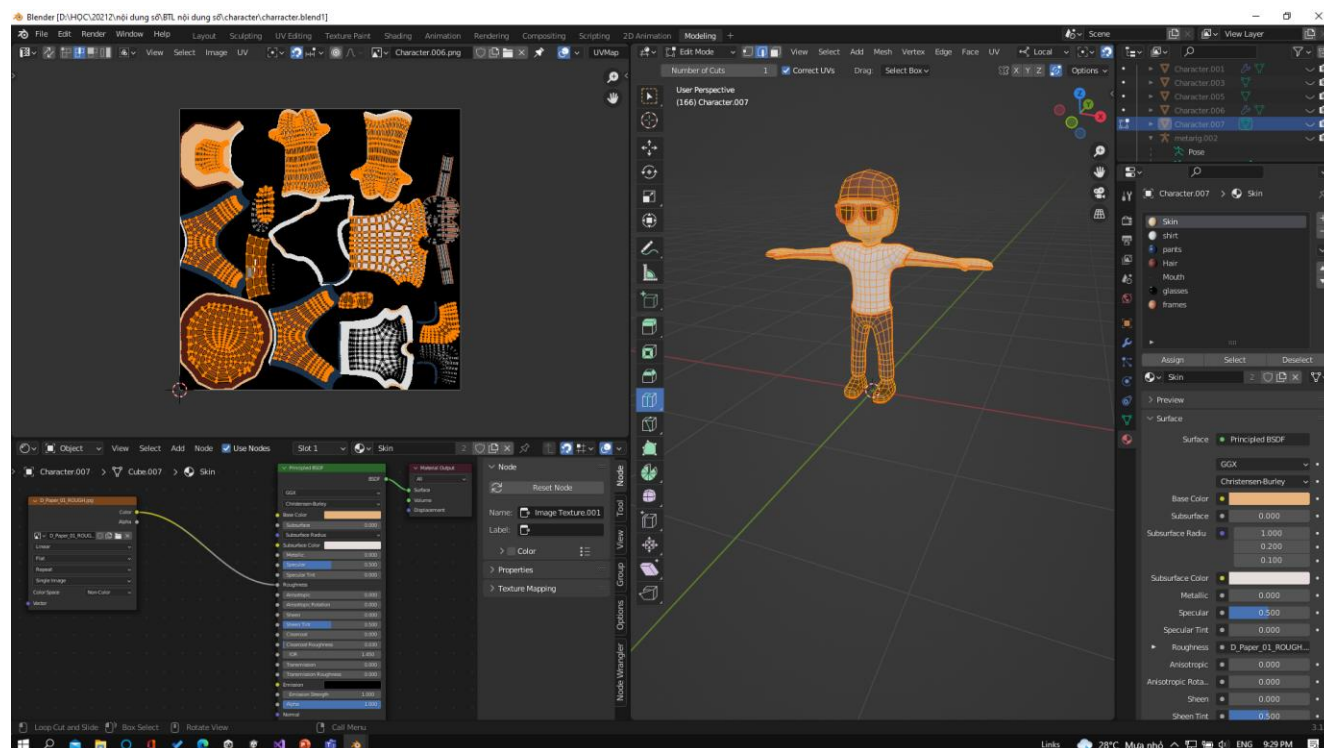
#### 1.1.1.2. Trang trí cho đối tượng

Ta sử dụng phương thức UV Editor (tab UV Editing in Blender): Thực hiện ánh xạ (mapping) kết cấu của đối tượng 3D trên mặt phẳng 2D (mặt phẳng UV), để có thể dễ dàng thay đổi và chỉnh sửa màu sắc, ngoại hình của đối tượng:



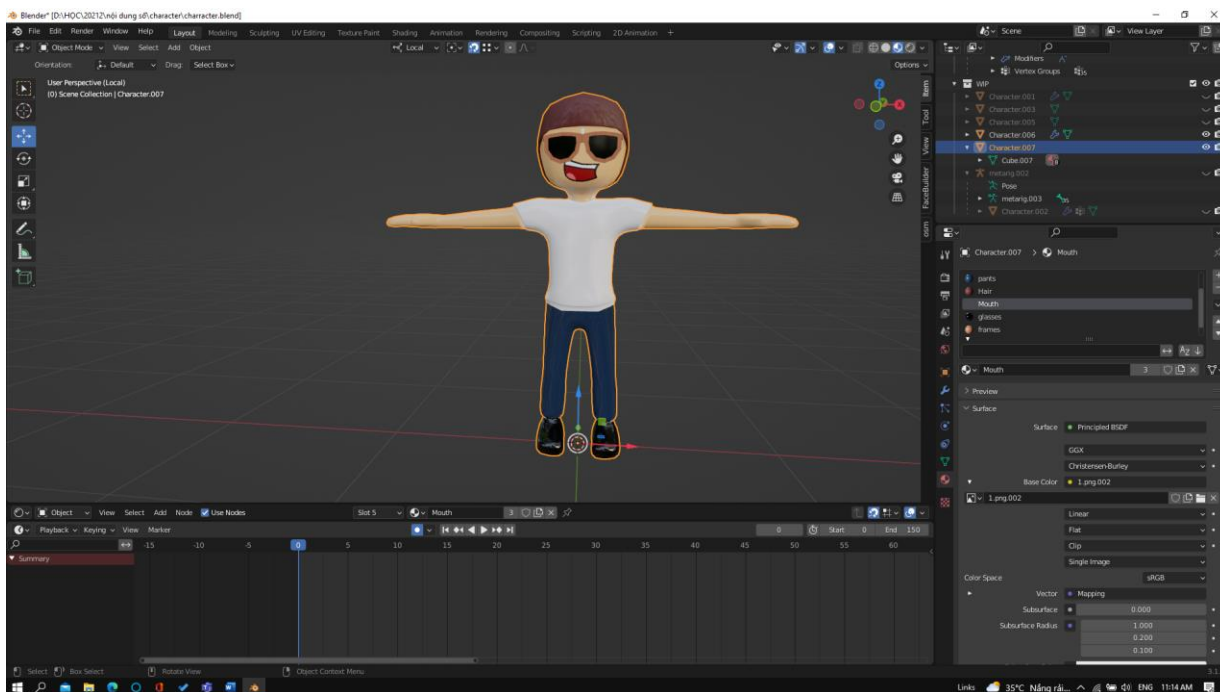
**Hình 1.3:** Ảnh xạ kết cấu mô hình 3D trên mặt phẳng 2D

Sau đó, ta sử dụng kết hợp với phương thức `Shade Editor` để thực hiện thêm màu tại các vị trí cụ thể (thực hiện trang trí màu sắc cho đối tượng).



**Hình 1.4:** Ánh xạ kết cấu mô hình 3D trên mặt phẳng 2D

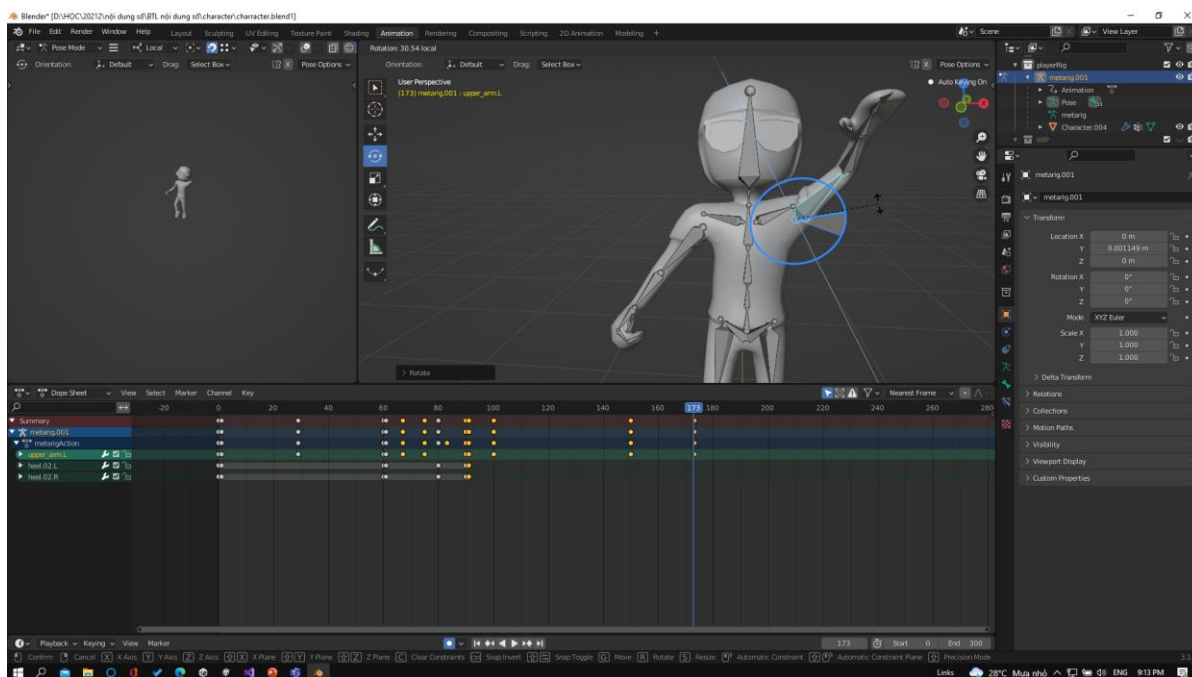
Kết quả cuối cùng sau khi thực hiện các bước trên ta thu được mô hình như hình dưới đây:



*Hình 1.5: Mô hình 3D sau khi thêm chất liệu*

#### *1.1.1.3. Tạo animation cho đối tượng*

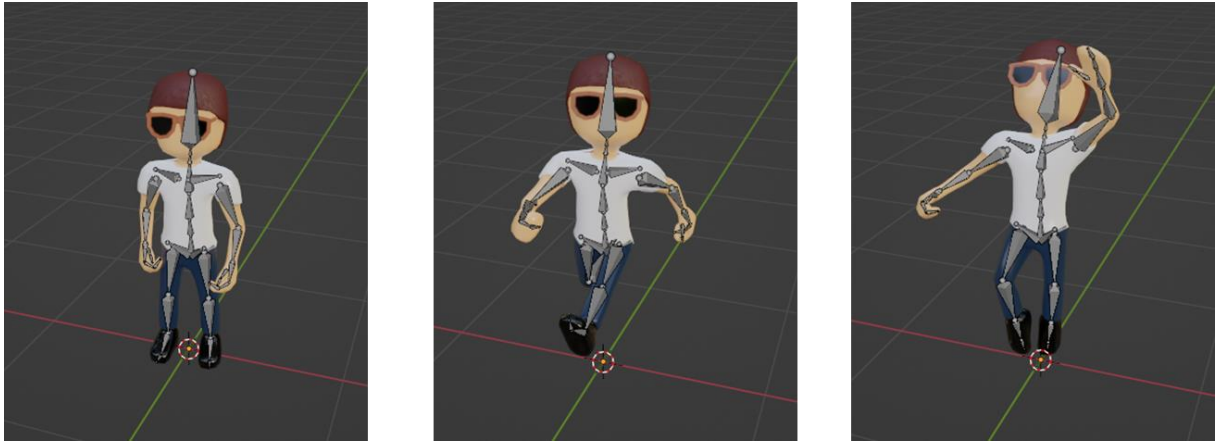
Sử dụng công cụ Pose của blender để điều chỉnh khung xương theo ý thích, từ đó tạo các dáng (pose) cho mô hình.



*Hình 1.6: Tạo animation với rig body*

Sau đó ta thực hiện insert KeyFrame để tạo tương tác giữa các dáng, sau khi thực hiện xong ta thu được animation cho đối tượng.



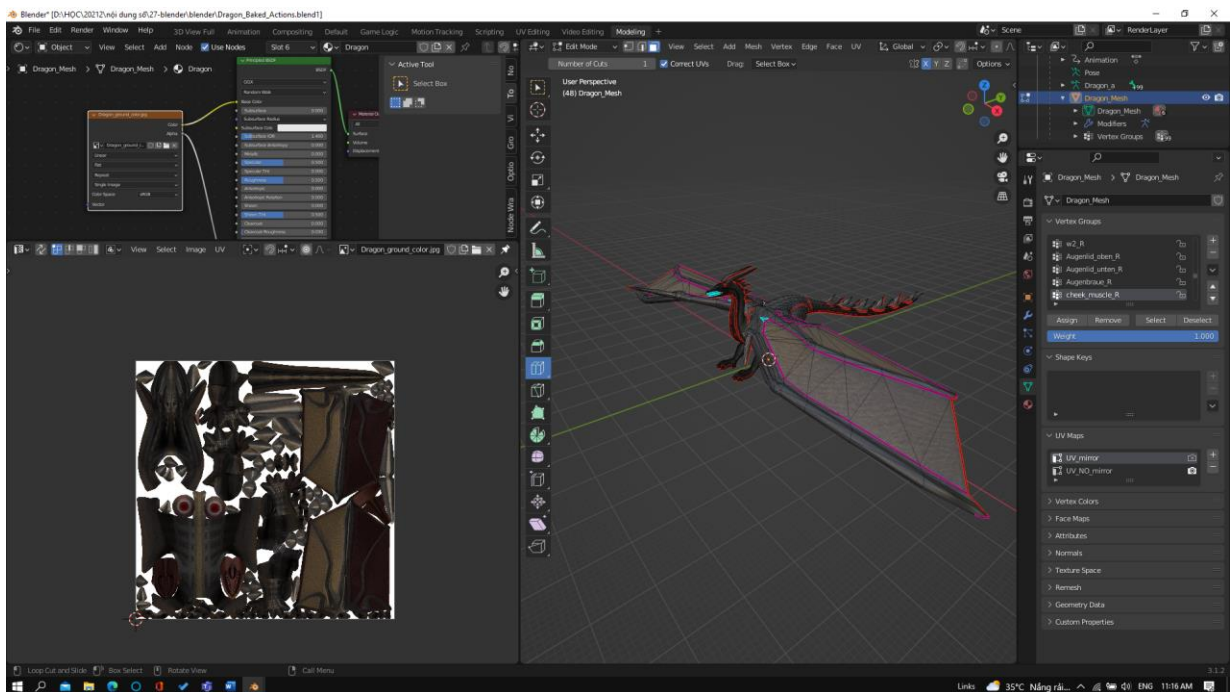


*Hình 1.7: Các animation của đối tượng Main character (idle, run, jump)*

## I.1.2. Dragon

### I.1.2.1. Xây dựng mô hình dragon & tạo animation

Thực hiện các bước tương tự như ở phần tạo main character và thực hiện tạo animation bay cơ bản cho mô hình dragon, ta thu được kết quả như hình dưới:

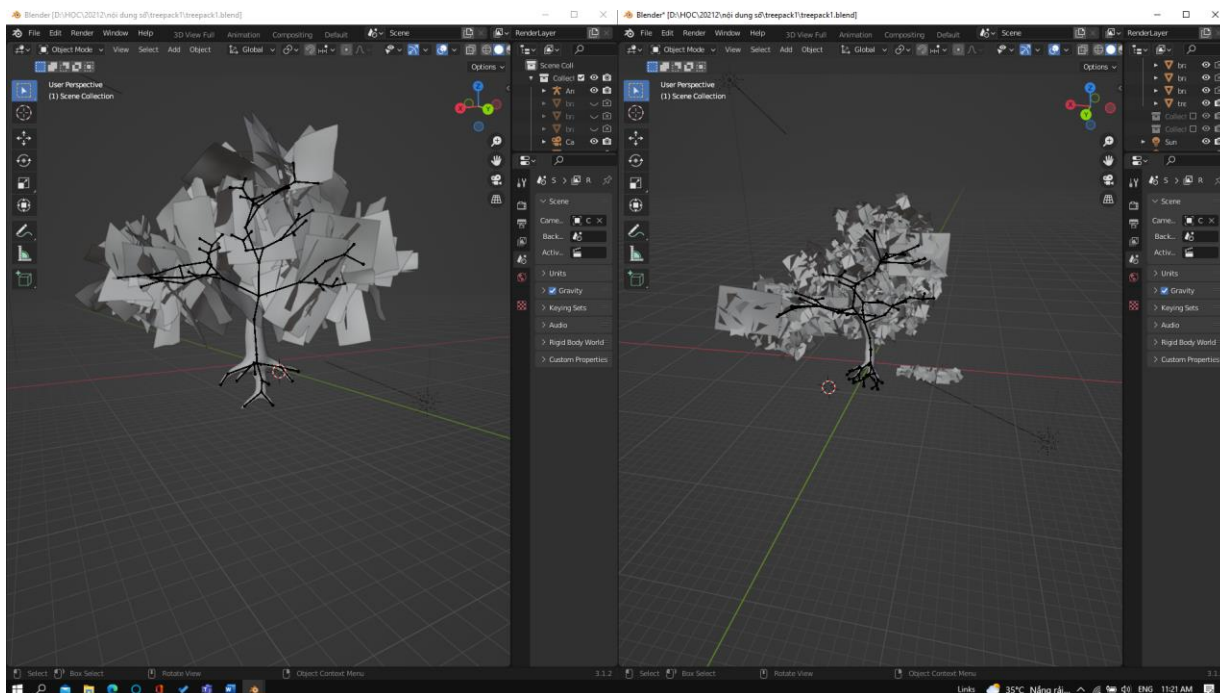


*Hình 1.8: Mô hình Dragon sau khi hoàn thiện*

## I.1.3. Tree

### I.1.3.1. Xây dựng mô hình tree



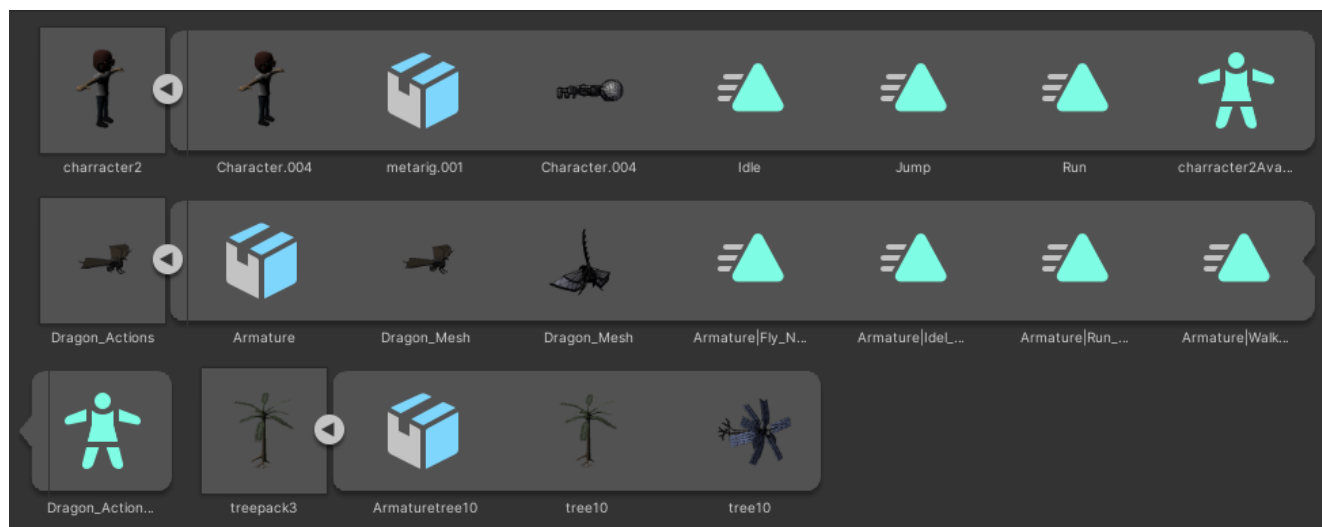


*Hình 1.9: Mô hình đối tượng tree sau khi hoàn thiện*

## I.2. ĐÓNG GỐI THÀNH API

Các đối tượng 3D được đóng gói thông qua tính năng export file (.fbx) trong Blender.

Các file này sẽ mang theo các thuộc tính (mesh, animation, avatar) của đối tượng, được sử dụng để lập trình tương tác sau này.



*Hình 1.10: Cấu trúc file (.fbx) sau khi export cho đối tượng 3D*

## CHƯƠNG II: XÂY DỰNG PHỐI CẢNH KHÔNG GIAN 3D

Để xây dựng phối cảnh không gian 3D, nhóm em lựa chọn sử dụng OSM để tải về bản đồ số 3D của một khu vực mở (khu đô thị), trong đó có thể hiện rõ địa hình địa vật (tòa nhà, đường sá...).

### II.1. XÂY DỰNG PHỐI CẢNH SỬ DỤNG ADD-ON BLENDER-OSM

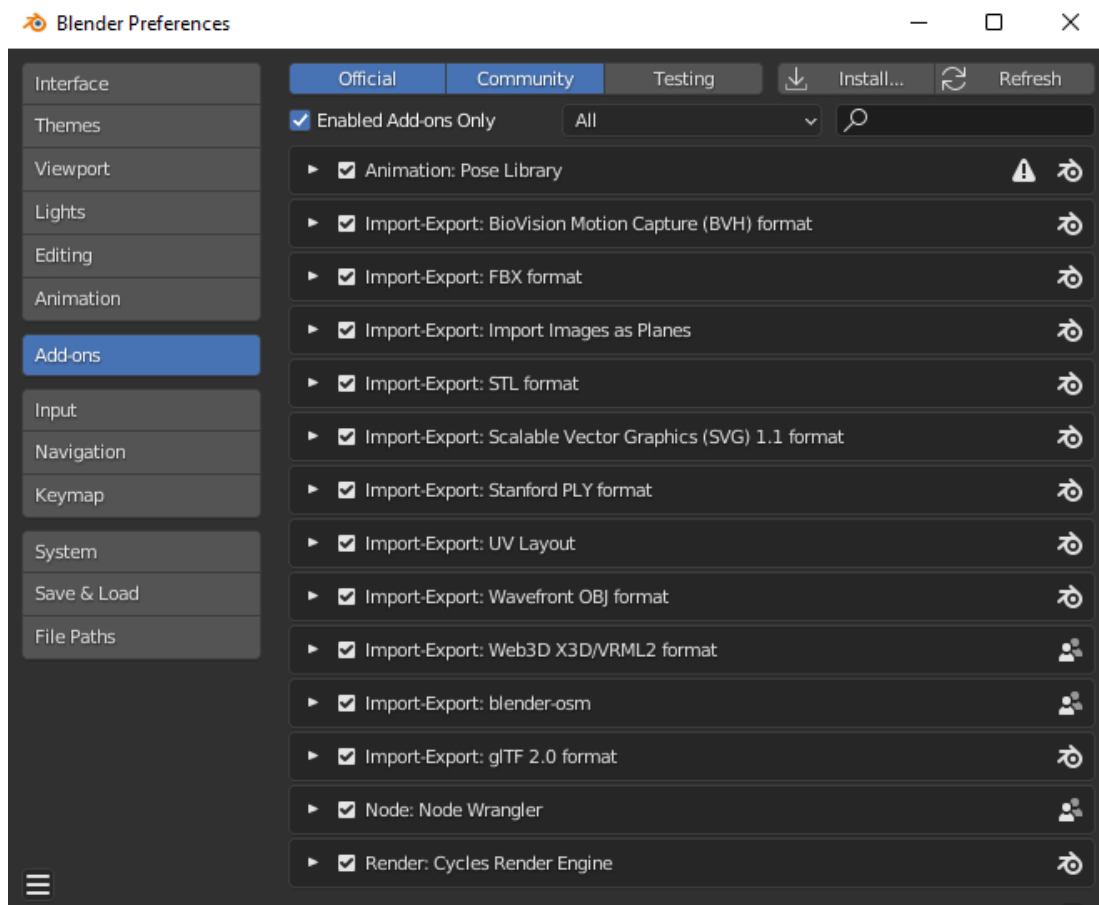
Open Street Map được viết tắt OSM (tạm dịch: "Bản đồ đường sá mở") là một dịch vụ bản đồ thế giới với nội dung mở trên Internet. Open Street Map ra đời nhằm mục đích cung cấp dữ liệu địa lý do nhiều người cùng cộng tác với nhau trên hệ thống wiki. Nó còn được gọi là "Wikipedia của bản đồ".

Nhóm em lựa chọn sử dụng OSM do OSM có cơ sở dữ liệu bản đồ trực tuyến, lưu dưới định dạng vector. Có thể sử dụng trên nhiều phần mềm khác nhau như JOSM, Mercator..., và có thể dễ dàng chuyển đổi thành các định dạng khác theo ý muốn.

#### II.1.1. Tải và cài đặt add-on blender-osm

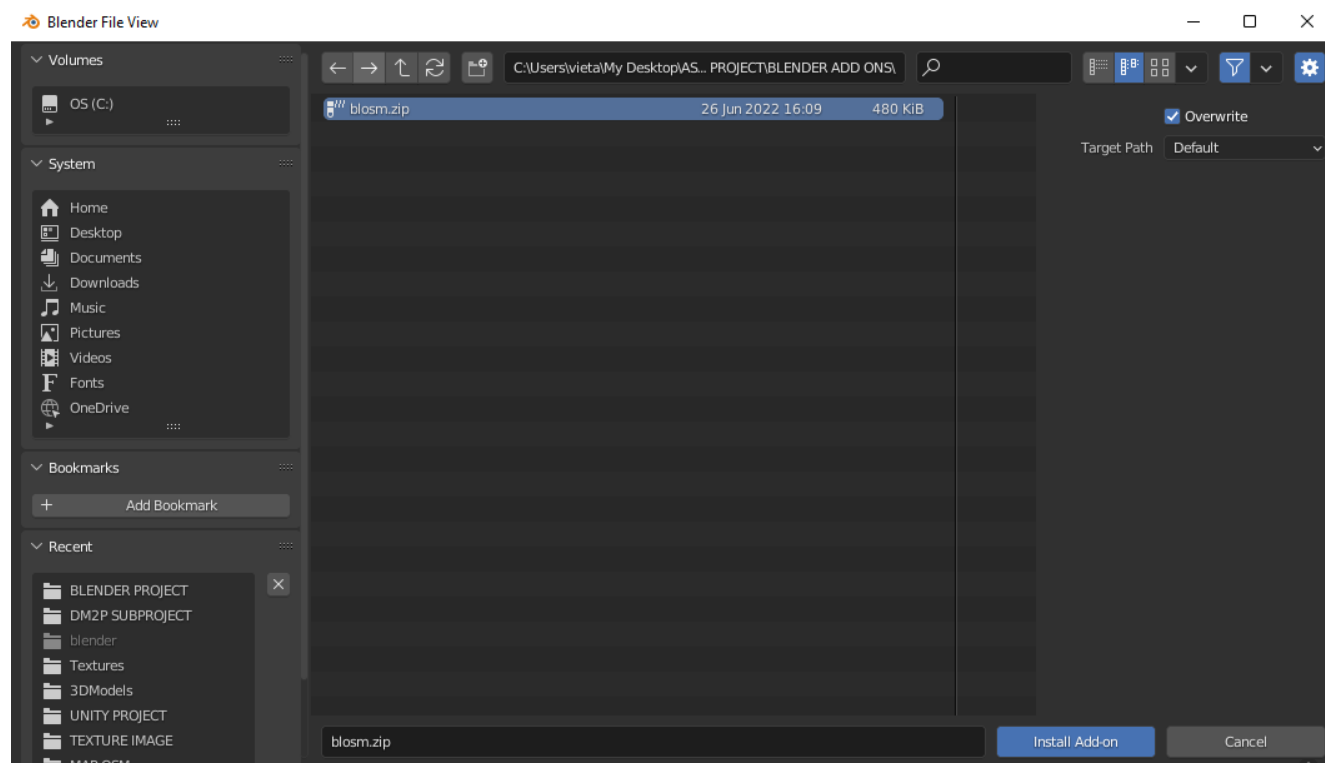
Vào trang web <https://prochitecture.gumroad.com/l/blender-osm> và thực hiện download add-on blender-osm (file tải về có dạng blom.zip).

Trong Blender, chọn Edit → Preference → Add-ons → Install:



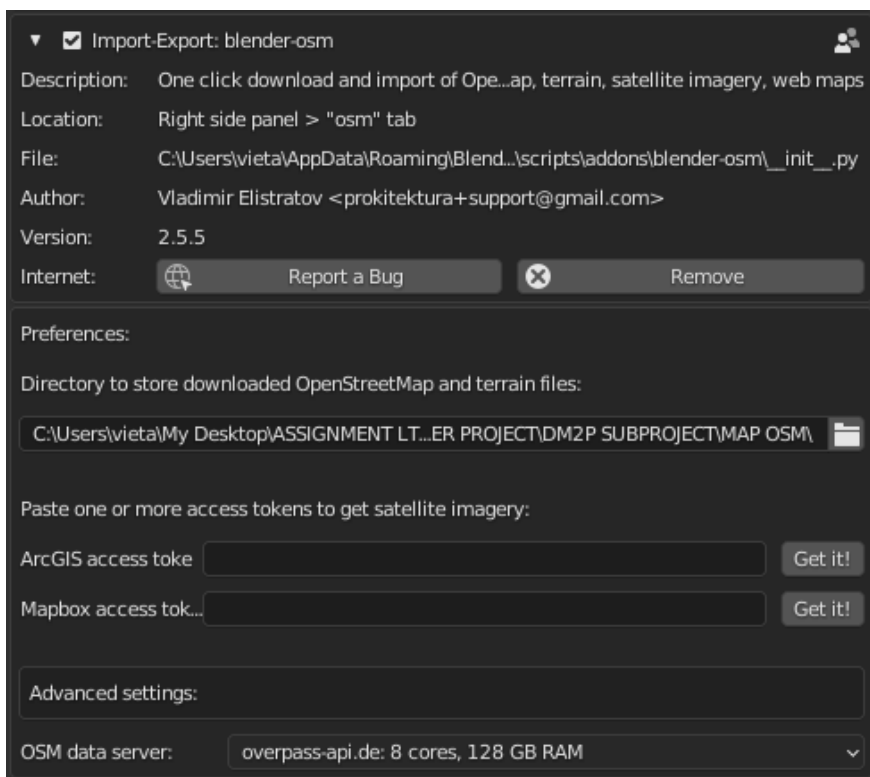
Hình 2.1: Giao diện Add-ons trên phần mềm Blender

Sau đó chọn đến nơi file add-on blender-osm vừa tải về, chọn Install Add-on để cài đặt blender-osm.



*Hình 2.2: Thực hiện install add-on blender-osm*

Kết quả sau khi thực hiện install thành công như hình dưới:



*Hình 2.3: Kết quả sau khi install thành công add-on blender-osm*

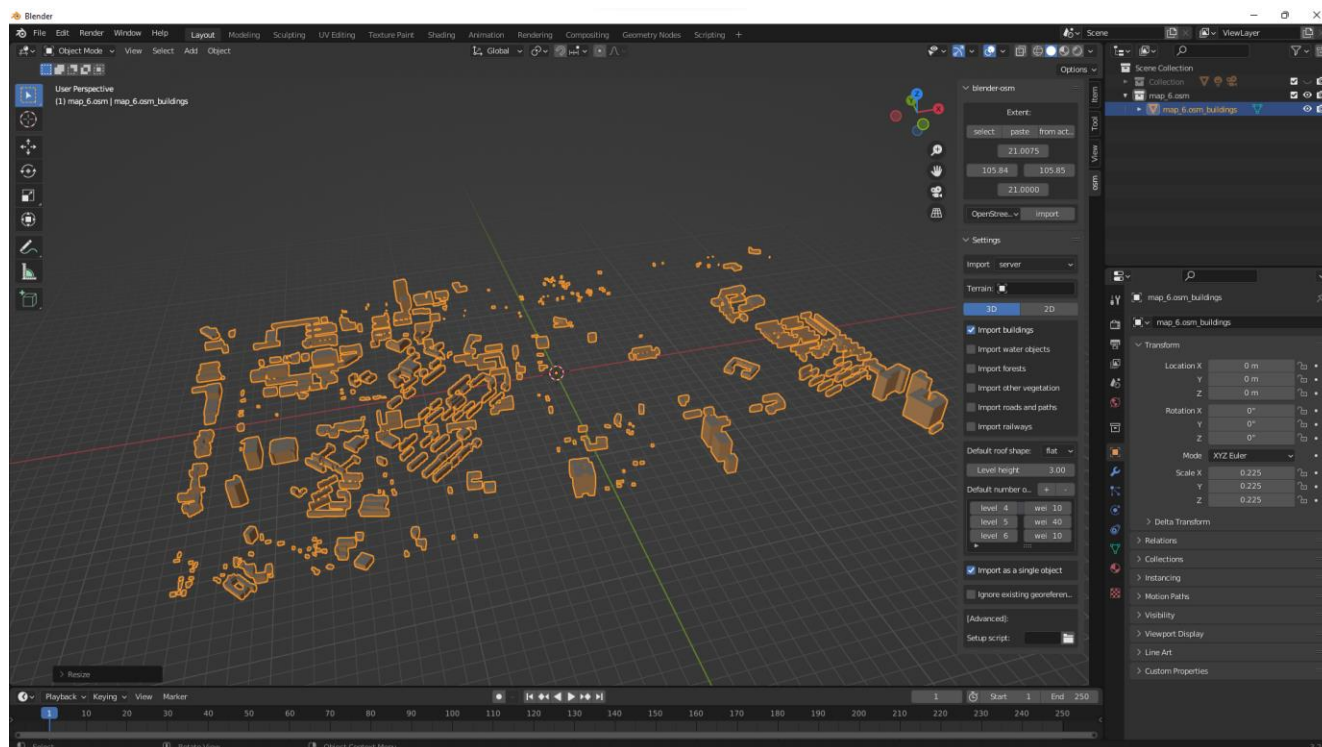
## II.1.2. Xây dựng phối cảnh 3D

Chọn vùng muốn xây dựng trên bản đồ (tọa độ trường Đại học Bách Khoa Hà Nội):



*Hình 2.4: Chọn khu vực sử dụng để xây dựng phối cảnh 3D*

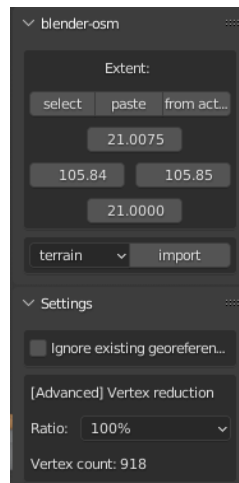
Tiếp theo ta thực hiện xây dựng (import) building cho phối cảnh (nhóm chỉ lựa chọn import building để giảm kích thước của mô hình), sau đó scale thu nhỏ toàn bộ map vừa import để có thể dễ dàng quan sát toàn bộ map và chỉnh sửa (phím tắt: A (chọn toàn bộ map) + S (scale các đối tượng được chọn)), kết quả thu được như hình dưới:



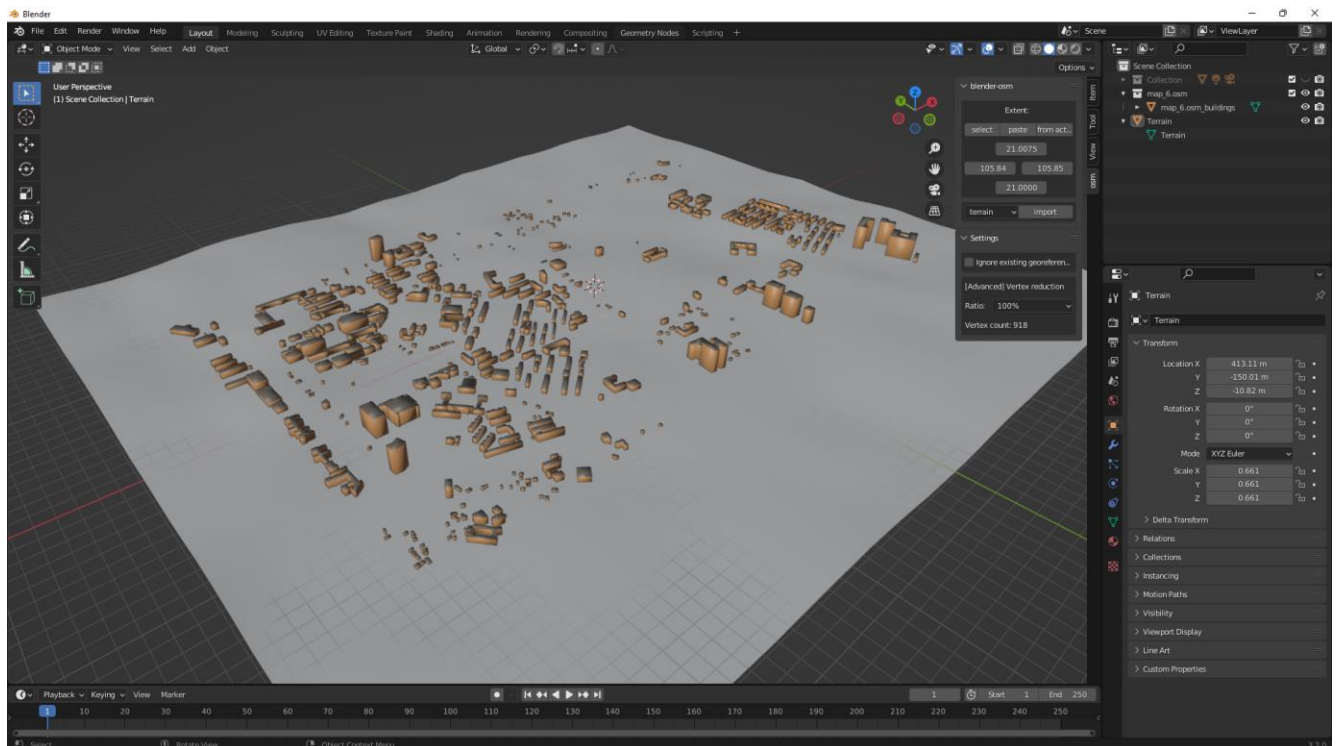
*Hình 2.5: Thực hiện import building cho phối cảnh*



Ta lựa chọn import terrain cho phối cảnh:



Sau khi scale (phím tắt S) và dịch chuyển terrain (phím tắt G) để khớp với building, ta thu được kết quả như hình:

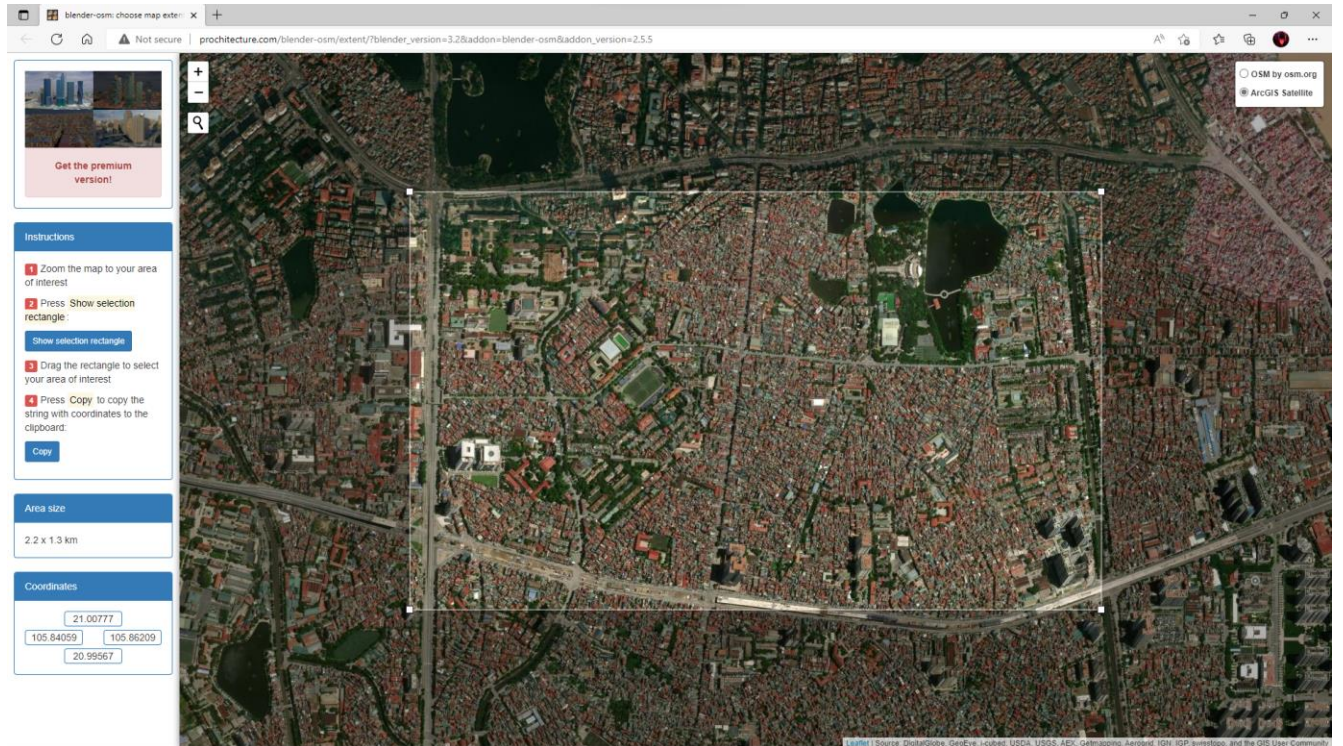


*Hình 2.6: Phối cảnh sau khi import building và terrain*

## II.1.3. Hoàn thiện phối cảnh 3D

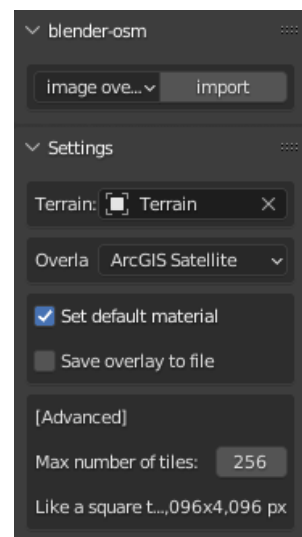
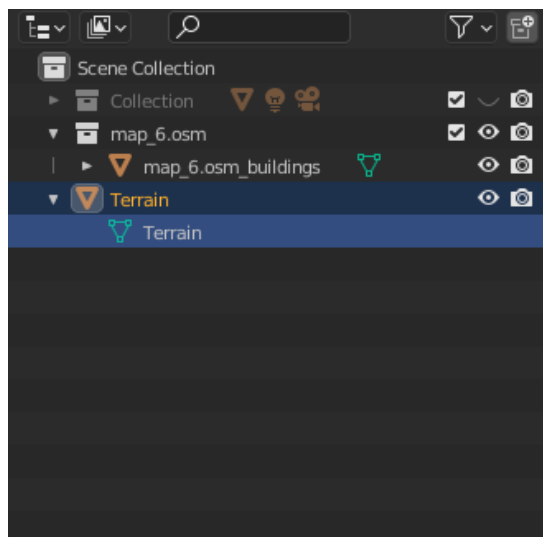
### II.1.3.1. Terrain

Ta sử dụng ArcGIS Satellite như hình vẽ:



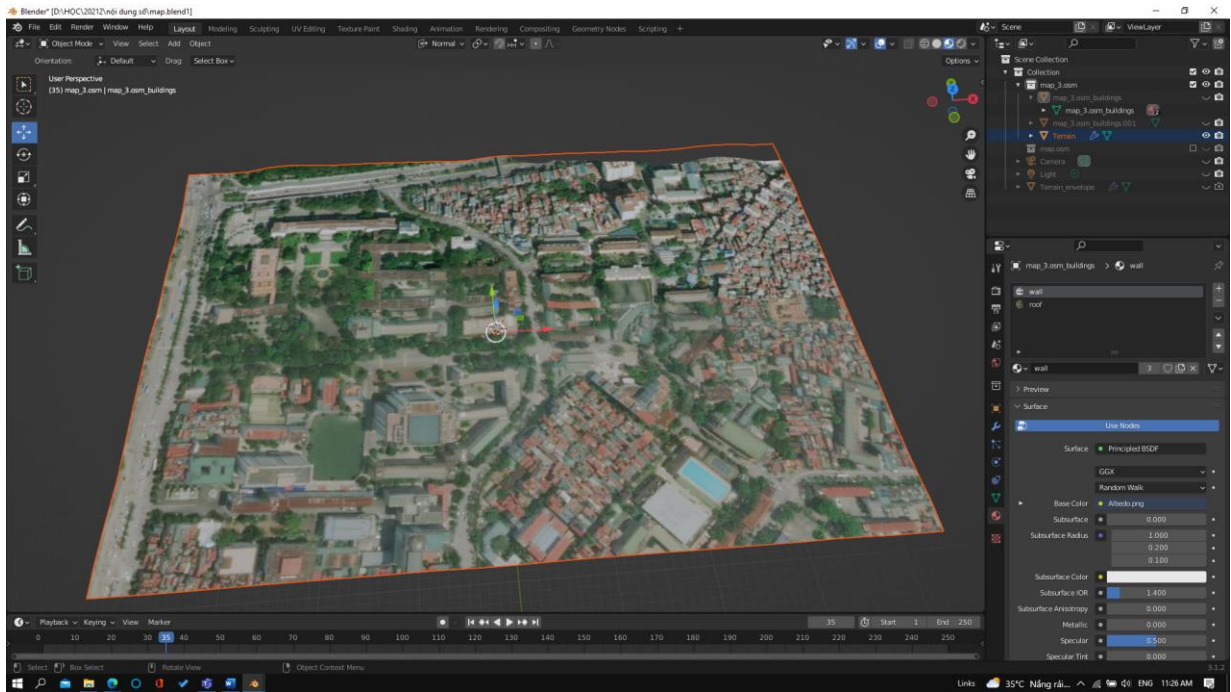
*Hình 2.7: Hoàn thiện terrain cho phối cảnh 3D (1)*

Trong Blender, sau khi chọn đối tượng terrain, ta lựa chọn image overlay và thực hiện import:



Kết quả thu được như hình vẽ:

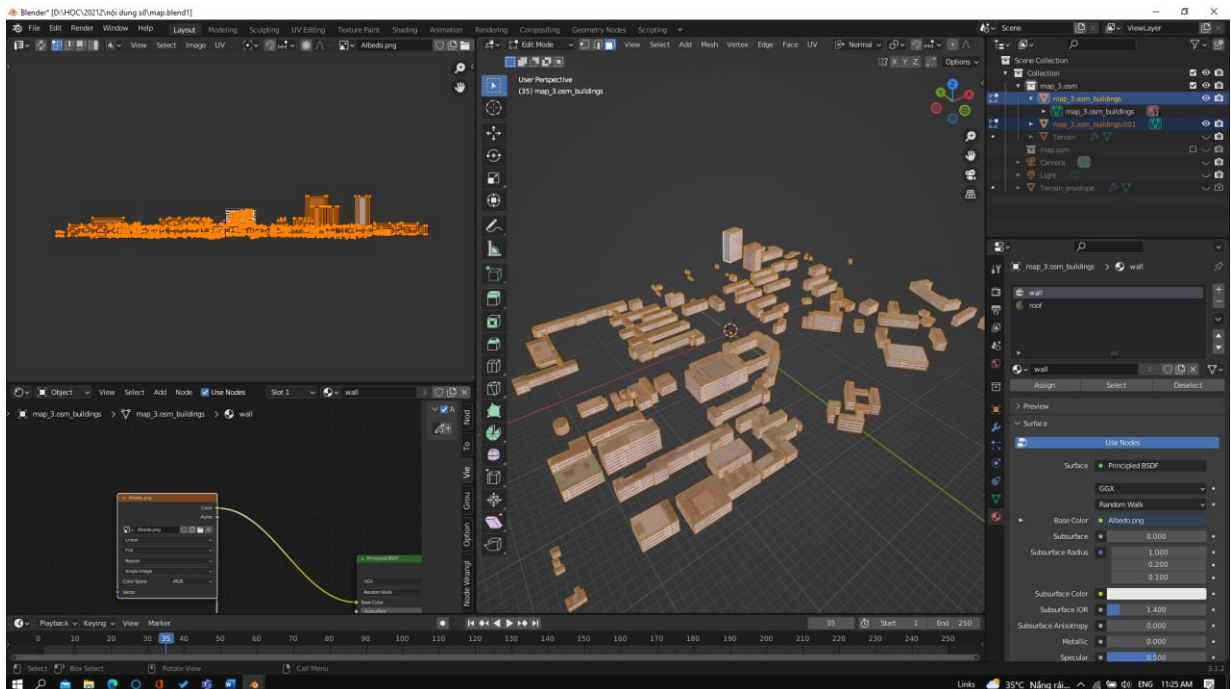




*Hình 2.8: Hoàn thiện terrain cho phối cảnh 3D (2)*

### II.1.3.2. Building

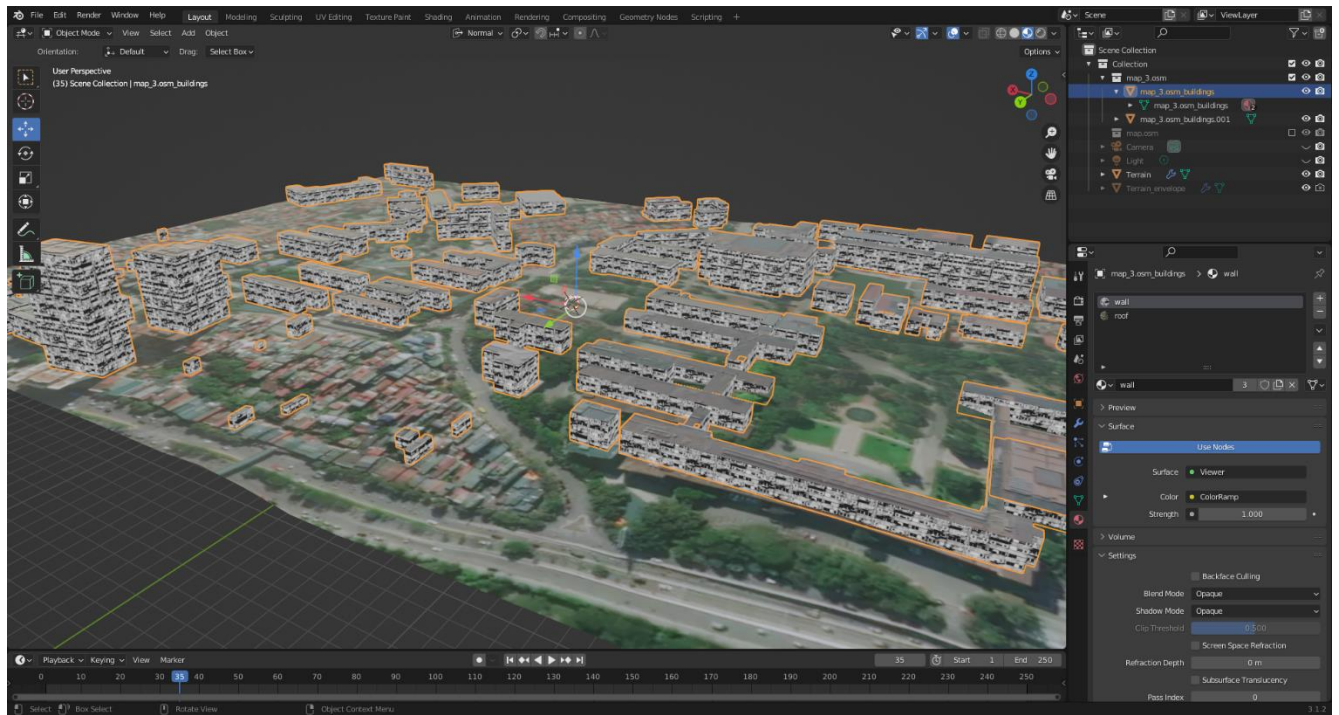
Sử dụng phương thức Shade Editor để thực hiện trang trí cho các tòa nhà (building), nhóm sử dụng 2 material là wall (tường) và roof (mái nhà) để trang trí. Kết quả thu được như hình vẽ:



*Hình 2.9: Hoàn thiện building cho phối cảnh 3D*

### II.1.3.3. Phối cảnh 3D sau khi hoàn thiện

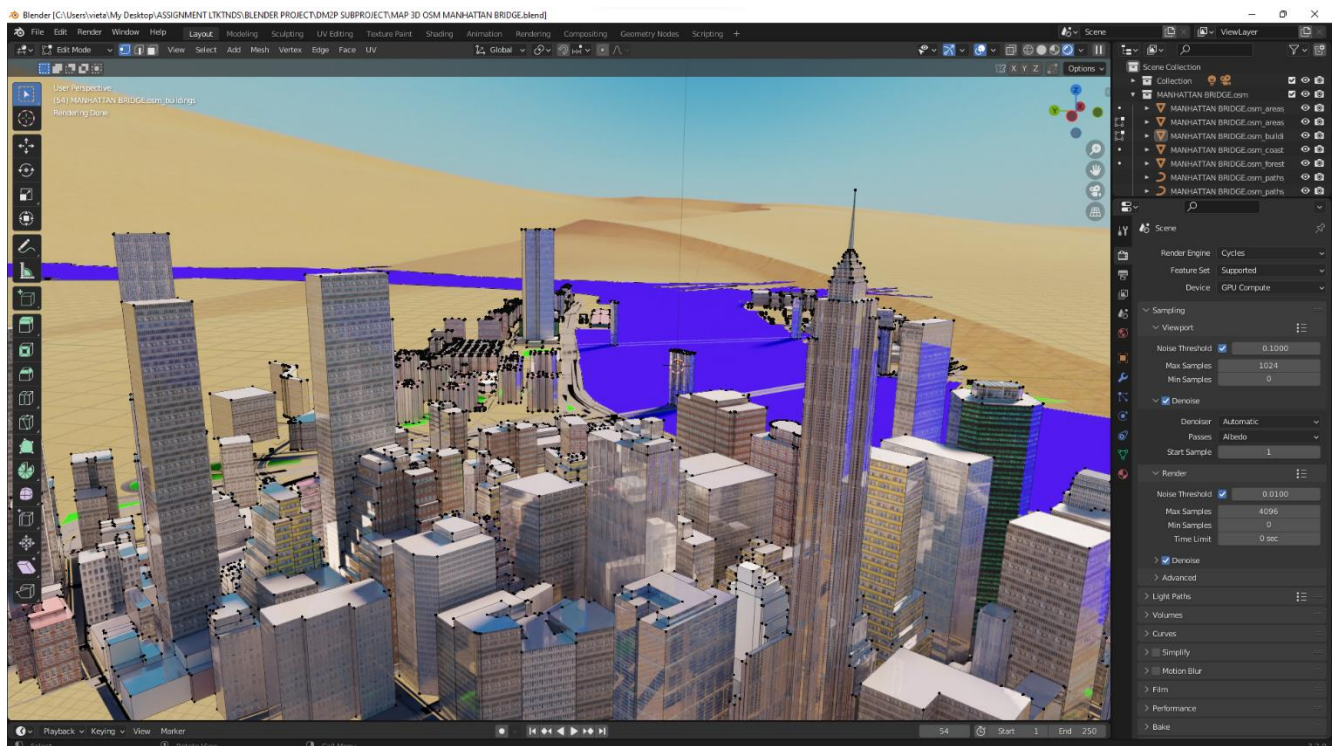
Sau khi hoàn thiện xây dựng và trang trí, ta thu được phối cảnh 3D:



*Hình 2.10: Phối cảnh 3D hoàn chỉnh.*

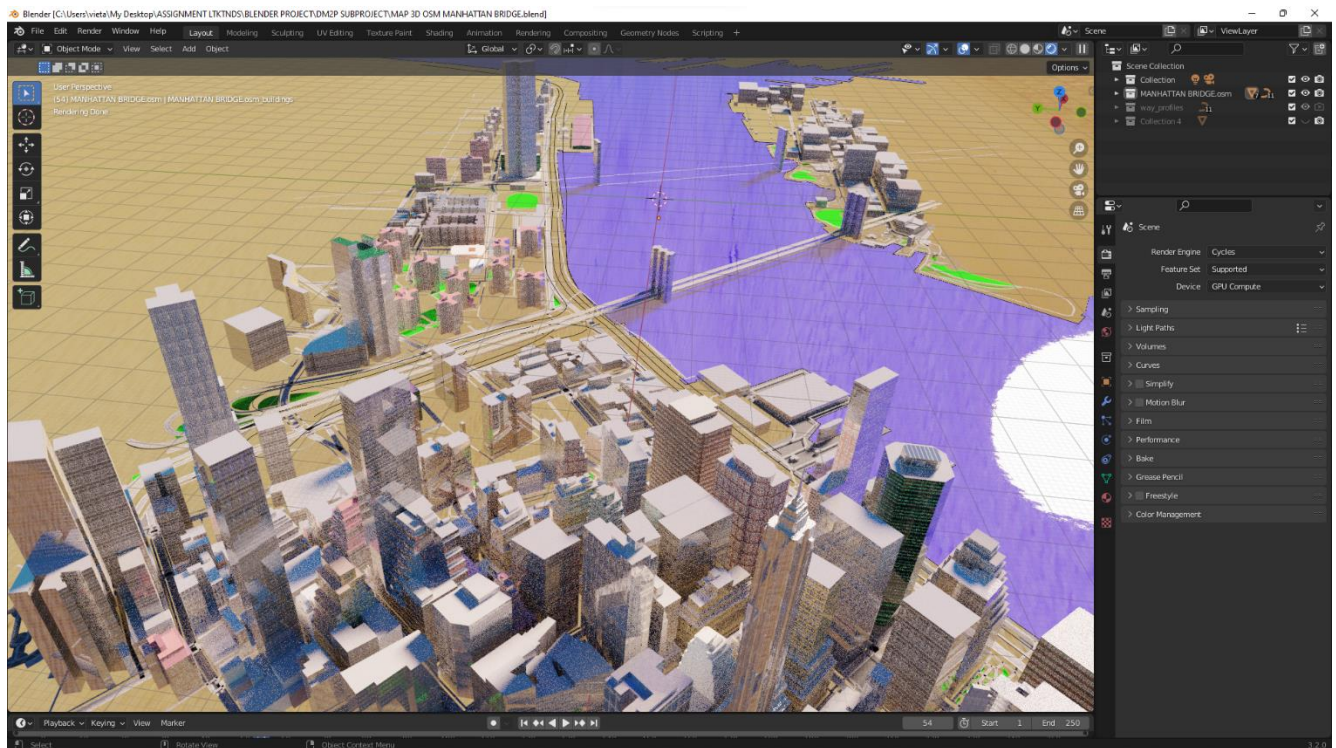
## II.1.4. Một phối cảnh 3D khác

Nhóm em có tạo một phối cảnh 3D hoàn thiện khác, tuy nhiên do file khá nặng nên nhóm em không sử dụng phối cảnh 3D này cho phần tiếp theo.

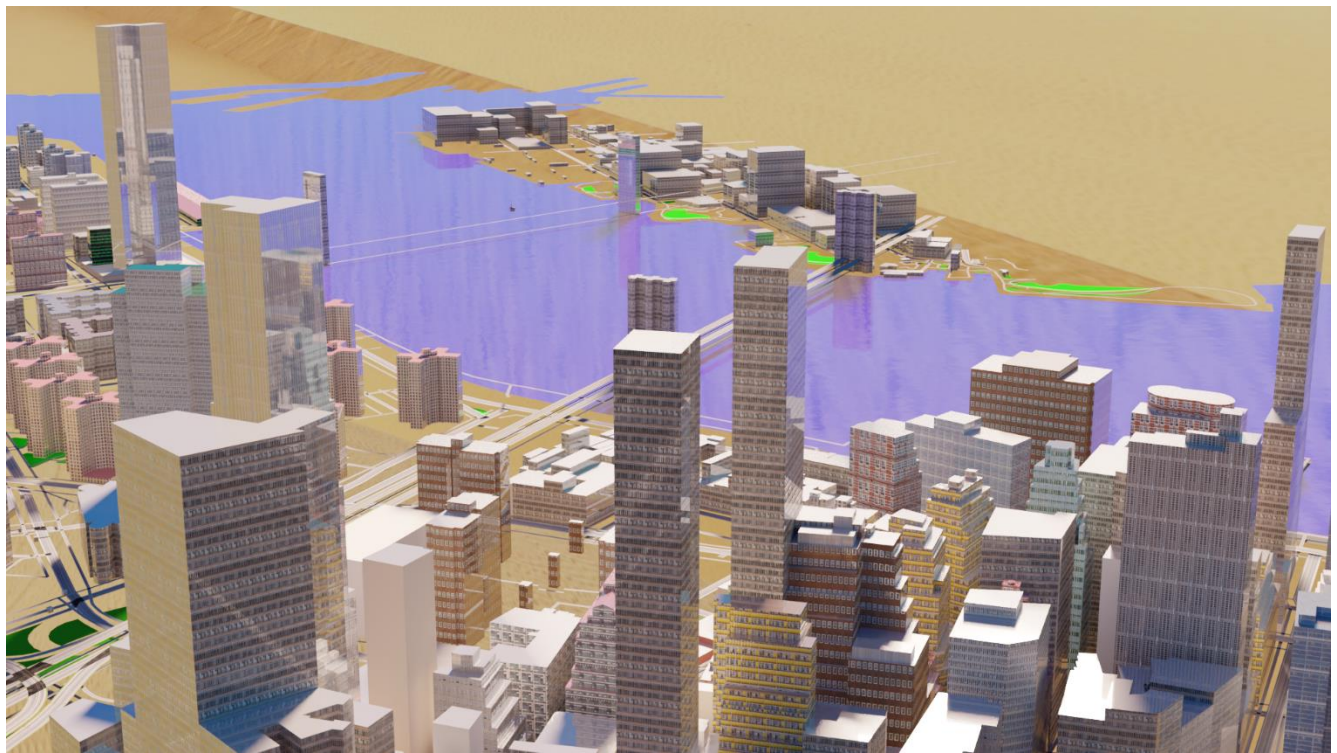


*Hình 2.11: Phối cảnh 3D phụ*





*Hình 2.12: Kết quả thực hiện render phối cảnh 3D phụ (1)*

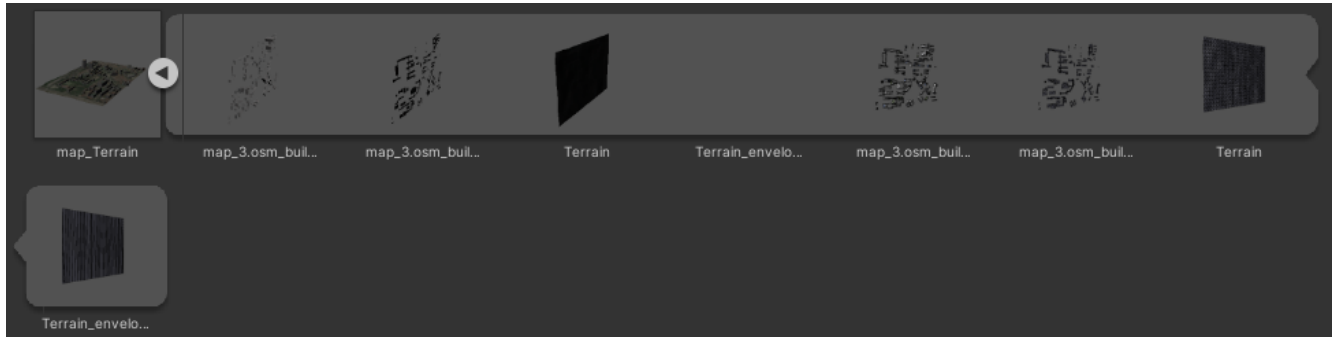


*Hình 2.13: Kết quả thực hiện render phối cảnh 3D phụ (1)*

## II.2. ĐÓNG GÓI THÀNH API

Phối cảnh 3D được đóng gói thông qua tính năng export file (.fbx) trong Blender.

Các file này sẽ mang theo các thuộc tính (mesh, material) của phối cảnh, được sử dụng để lập trình tương tác sau này.

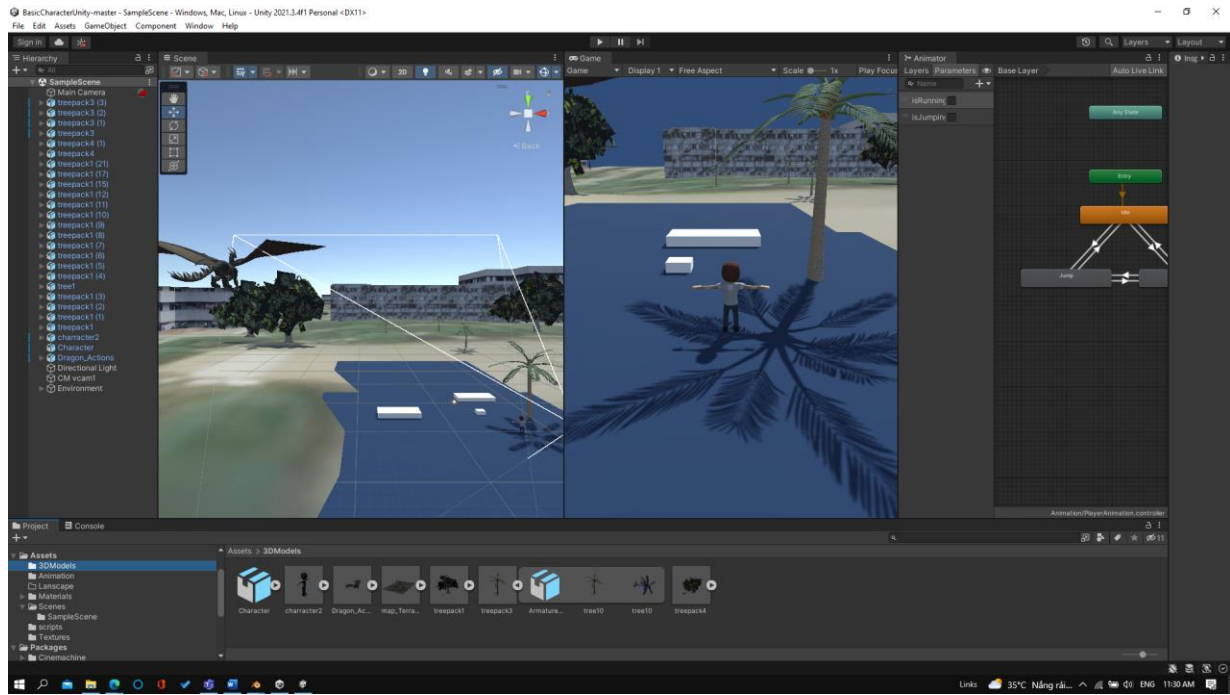


**Hình 2.14:** Cấu trúc file (.fbx) sau khi export cho phối cảnh 3D

# CHƯƠNG III: TỔNG HỢP & LẬP TRÌNH ĐIỀU KHIỂN TƯƠNG TÁC

## III.1. NHÚNG CÁC ĐỐI TƯỢNG VÀ PHỐI CẢNH VÀO PROJECT UNITY

Tất cả các đối tượng và phối cảnh 3D sau khi được export thành dạng file .fbx, được nhúng vào project unity để lập trình tương tác giữa các đối tượng.

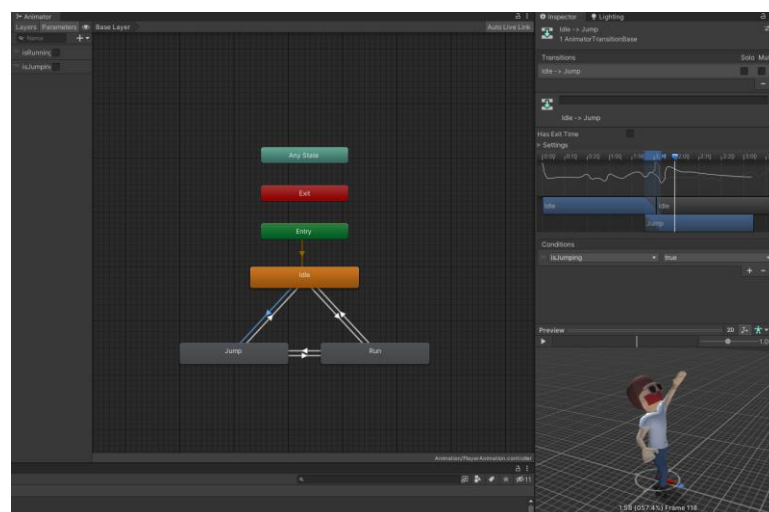


*Hình 3.1: Kết quả sau khi nhúng các đối tượng và phối cảnh 3D vào project Unity*

### III.1.1. Chuyển động của main character

Với các animation được đóng gói từ trước, sử dụng công cụ Animator:

- Tạo 2 biến isRunning và isJumping thiết lập điều kiện chuyển đổi giữa các animation.
- Xây dựng đồ hình tương tác giữa các animation kèm các thông số liên quan.



*Hình 3.2: Đồ hình tương tác giữa các animation và các biến isRunning, isJumping*



## III.2. CODE XÂY DỰNG TƯƠNG TÁC GIỮA CÁC ĐỐI TƯỢNG

Sử dụng API được đóng gói trong Mục 1., lập trình cho phép người dùng điều khiển quỹ đạo chuyển động của đối tượng 3D bằng chuột hoặc phím nhấn trong bản đồ số nói trên, và trong quá trình chuyển động, đối tượng cho phép người dùng tương tác theo tính năng đã chọn.

### III.2.1. Code di chuyển main character

Main character có hai hành động (di chuyển) chính là chạy (movement) và nhảy (jumping), được thể hiện trong đoạn code dưới đây, trong đó:

- Input.GetAxisRaw("Vertical") trả về: -1 ~ "S", 1 ~ "W", 0 ~ "không có phím nào được nhấn".
- Input.GetAxisRaw("Horizontal") trả về: -1 ~ "A", 1 ~ "D", 0 ~ "không có phím nào được nhấn".
- Các giá trị movementSpeed, rotationSpeed, jumpsSpeed, gravity được thiết lập dưới dạng có thể thay đổi trực tiếp trên Unity.
- movementSpeed: thiết lập tốc độ di chuyển của nhân vật.
- rotationSpeed: thiết lập tốc độ di quay của nhân vật.
- jumpSpeed: thiết lập tốc độ (độ cao) khi nhân vật thực hiện nhảy.
- gravity: thiết lập giá trị gia tốc trọng trường.
- Bên cạnh đó sử dụng thêm biến (boolean) playerGrounded (xác định giá trị thông qua hàm isGrounded()) để kiểm tra xem nhân vật hiện có đang ở trên mặt đất hay không.
- Các câu lệnh thực hiện hành động của main character được thực hiện ở hàm Update() – được gọi khi mỗi một frame mới được xây dựng.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(CharacterController), typeof(Animator))]
public class PlayerController : MonoBehaviour{
    private CharacterController characterController;
    private Animator animator;

    [SerializeField]
    private float movementSpeed, rotationSpeed, jumpSpeed, gravity;

    private Vector3 movementDirection = Vector3.zero;
    private bool playerGrounded;
```



```

// Start is called before the first frame update
void Start() {
    characterController = GetComponent<CharacterController>();
    animator = GetComponent<Animator>();
}

// Update is called once per frame
void Update() {
    playerGrounded = characterController.isGrounded;

    //movement
    Vector3 inputMovement = transform.forward * movementSpeed *
        Input.GetAxisRaw("Vertical");
    characterController.Move(inputMovement * Time.deltaTime);
    transform.Rotate(Vector3.up * Input.GetAxisRaw("Horizontal") * rotationSpeed);

    //jumping
    if (Input.GetButton("Jump") && playerGrounded) {
        movementDirection.y = jumpSpeed;
    }
    movementDirection.y -= gravity * Time.deltaTime;

    characterController.Move(movementDirection * Time.deltaTime);

    //animations
    animator.SetBool("isRunning", Input.GetAxisRaw("Vertical") != 0);
    animator.SetBool("isJumping", !characterController.isGrounded);
}
}

```

### III.2.2. Âm thanh khi di chuyển của main character

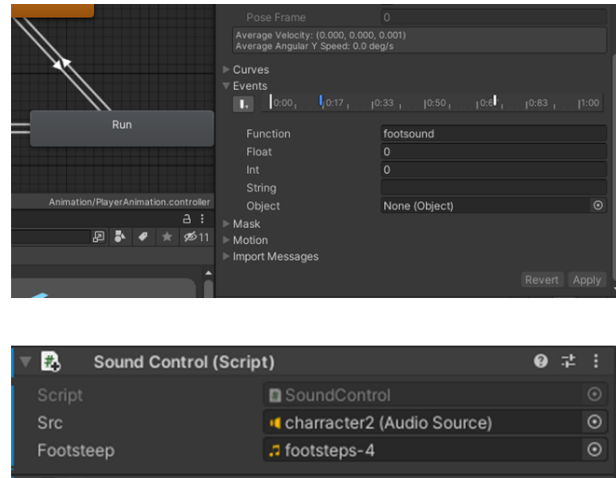
Đối tượng phát ra âm thanh 3D (tiếng bước chân) khi di chuyển. Lập trình thiết lập chạy file “footsteps-4” khi hàm “footsound” được kích hoạt:

```

public class SoundControl : MonoBehaviour
{
    public AudioSource src;
    public AudioClip footsteps;

    void footsound()
    {
        src.clip = footsteps;
        src.Play();
    }
}

```



*Hình 3.3: Lập trình thiết lập âm thanh khi di chuyển cho main character*

### III.2.3. Hiệu ứng của main character

Lập trình cho đối tượng bốc cháy (bấm phím "1") hoặc phát ra đốm sáng (bấm phím "3").

```

void Update()
{
    if (Input.GetKeyDown("1"))
    {
        CatchFire.Play();
    }
    if (Input.GetKeyDown("2"))
    {
        CatchFire.Stop();
    }
    if (Input.GetKeyDown("3"))
    {
        Shiny.Play();
    }
    if (Input.GetKeyDown("4"))
    {
        Shiny.Stop();
    }
}

```



*Bốc cháy*

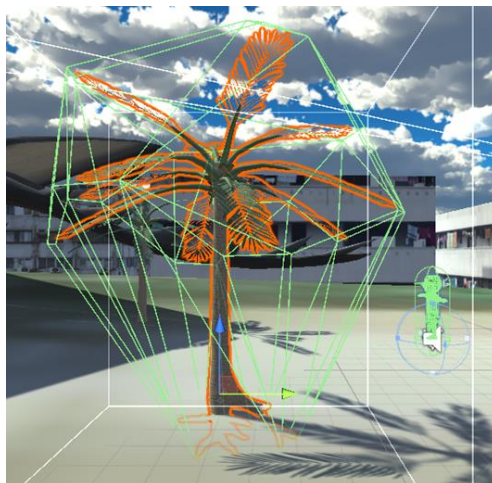


*Phát ra đốm sáng*

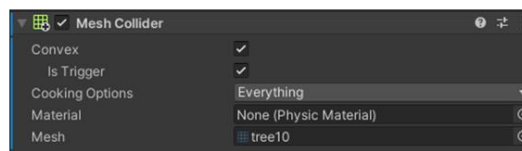
*Hình 3.4: Lập trình thiết lập hiệu ứng cho main character*

### III.2.4. Tương tác giữa main character và đối tượng tree

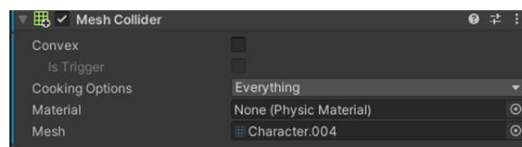
Với các mesh được đóng gói từ trước, sử dụng công cụ Mesh Collider thiết lập thuộc tính vật cản cho Main Character và "tree".



Hình dáng thuộc tính vật cản



Thiết lập cho tree



Thiết lập cho Character

**Hình 3.5:** Thiết lập thuộc tính vật cản cho tree và character

Lập trình cho đối tượng tree bốc cháy khi va chạm với main character, và thêm âm thanh khi cháy tương tự phần III.2.2 cho đối tượng tree:

```
private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag("Player"))
    {
        CatchFire.Play();
        Destroy(gameObject, 8);
    }
}
```



Trước khi va chạm

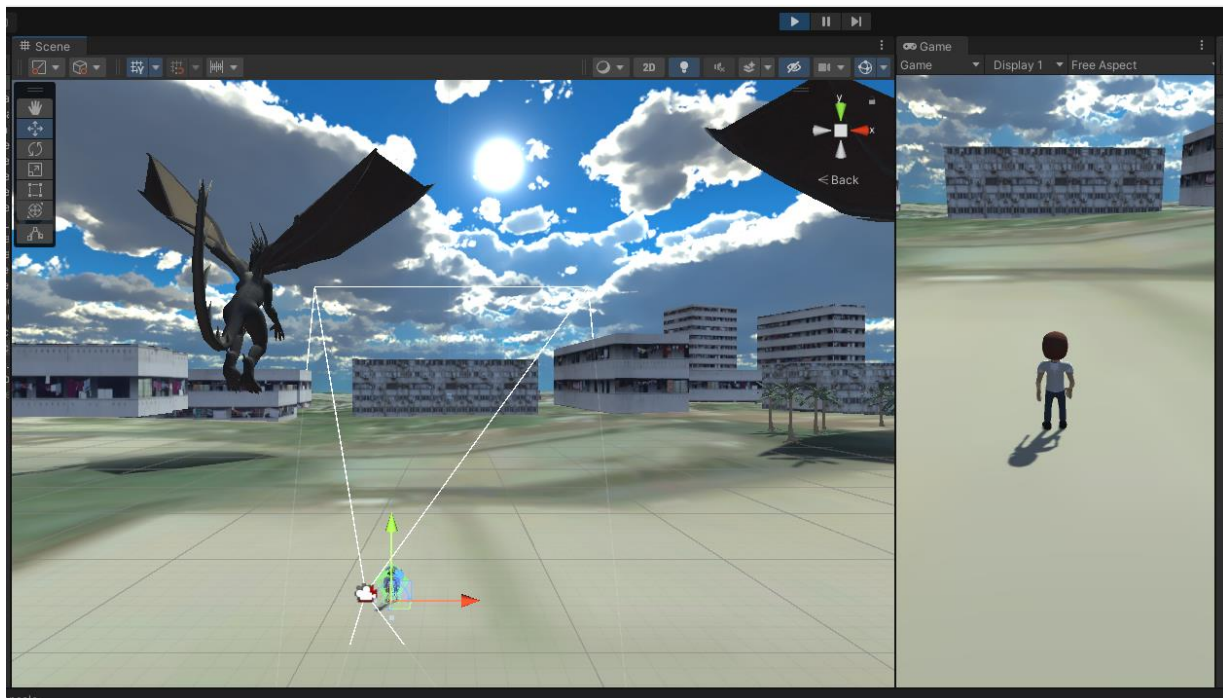


Sau khi va chạm

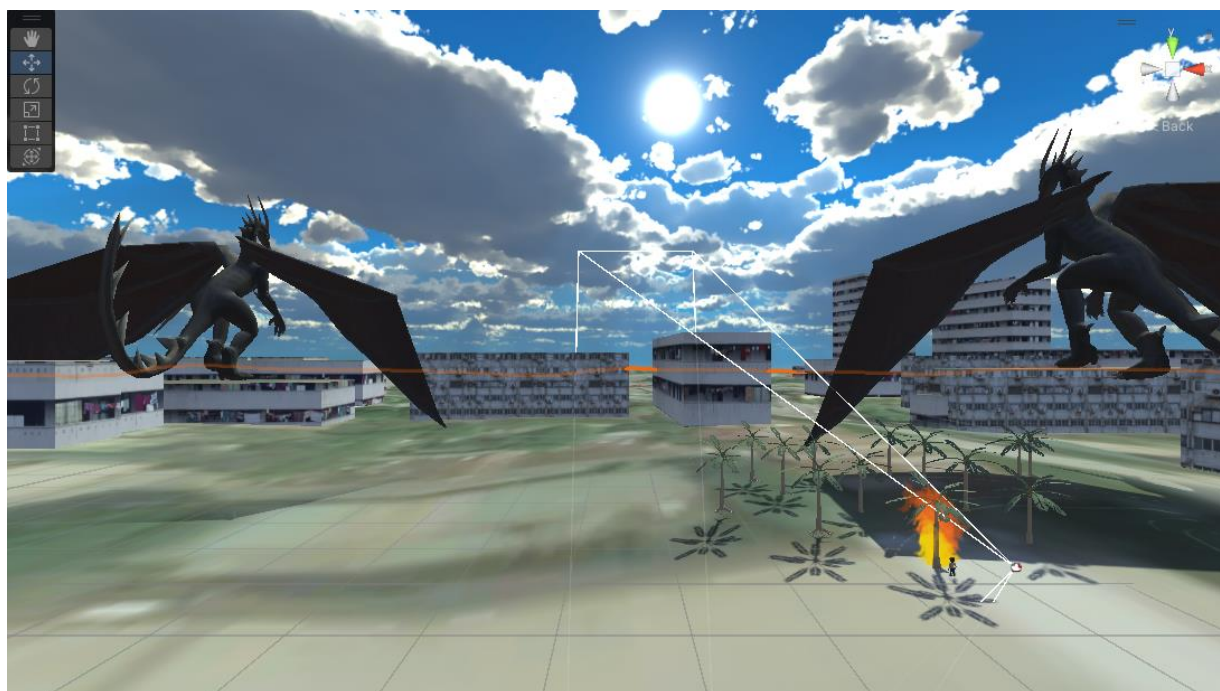
**Hình 3.6:** Lập trình cho đối tượng tree bốc cháy khi main character va chạm

### III.2.5 Toàn cảnh project và một số tương tác khác

Đối tượng dragon được thiết lập luôn ở trạng thái bay khi khởi động chương trình. Dưới đây là toàn cảnh project:



*Hình 3.7: Toàn cảnh 3D Project (1)*



*Hình 3.8: Toàn cảnh 3D Project (2)*

## PHỤ LỤC & MÃ NGUỒN

Project sử dụng Blender 3.2 và Unity 2021.3.2f1.

Link GitHub project: [https://github.com/BruceLuuM/3D\\_Project](https://github.com/BruceLuuM/3D_Project).

Nội dung đầu ra: [OUTPUT DM2P SUPPROJECT.mkv](#).