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

****

**BÁO CÁO BÀI TẬP LỚN MÔN**

**KỸ THUẬT LẬP TRÌNH NỘI DUNG SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên sinh viên** | **MSSV** | **Mã lớp** | **Khóa** |
| Nguyễn Tất Đạt | 20172455 | 133403 | K62 |
| Hoàng Phi Long  Vũ Quang Linh | 20182651  20182640 | 133403  133403 | K63  K63 |

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

**HÀ NỘI 06/2022**

# **LỜI NÓI ĐẦU**

Nghiên cứu về nội dung số đã và đang là đề tài nghiên cứu lôi cuốn của nhiều nhà nghiên cứu hiện nay. Với sự phát triển của phần cứng tốc độ xử lý cao, hiển thị sắc nét. Khi đó yêu cầu của một kỹ sư phát triển nội dung số ngày càng cao, đòi hỏi những kiến thức nền tảng vững chắc thì mới có thể có chỗ đứng trong xã hội ngày nay. Để đạt được những nền tảng ban đầu và hiểu rõ hơn về lập trình nội dung số, chúng em đã làm đề tài bài tập lớn của môn này.

Cảm ơn thầy Phạm Văn Tiến đã hướng dẫn và giúp đỡ chúng em có thể hoàn thiện bài tập lớn này.

Đây là bảng phân chia công việc của nhóm em.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Nguyễn Tất Đạt** | **Hoàng Phi Long** | **Vũ Quang Linh** |
| **Dự án** | **Phần 1, 4** | **Phần 2,4** | **Phần 3, 4** |
| **Nội dung công việc** | Tìm hiểu vấn đề | Tìm hiểu vấn đề | Tìm hiểu vấn đề |
| Lên phương pháp triển khai | Lên phương pháp triển khai | Lên phương pháp triển khai |
| Triển khai dự án | Triển khai dự án | Triển khai dự án |
| Đưa ra và tổng hợp kết quả | Đưa ra và tổng hợp kết quả | Đưa ra và tổng hợp kết quả |

# **MỤC LỤC**

[LỜI NÓI ĐẦU 2](#_Toc107180290)

[MỤC LỤC 3](#_Toc107180291)

[DANH MỤC HÌNH ẢNH 5](#_Toc107180292)

[DANH MỤC BẢNG BIỂU 6](#_Toc107180293)

[CHƯƠNG I. LÝ THUYẾT CHUNG 7](#_Toc107180294)

[1. Blender 7](#_Toc107180295)

[1.1. Các cửa sổ cơ bản trong Blender 7](#_Toc107180296)

[1.2. Lập trình Python với Blender 8](#_Toc107180297)

[2. BlenderGIS 11](#_Toc107180298)

[2.1. Georeferencing management 11](#_Toc107180299)

[2.2. Nhập dữ liệu OSM 12](#_Toc107180300)

[3. Blender-OSM ( Open Street Map) 14](#_Toc107180301)

[4. Google Maps 3D 15](#_Toc107180302)

[5. UPBGE 18](#_Toc107180303)

[5.1. Định nghĩa 18](#_Toc107180304)

[5.2. Các tính năng 18](#_Toc107180305)

[CHƯƠNG II. TRIỂN KHAI CÔNG VIỆC 19](#_Toc107180306)

[1. Tổng hợp đối tượng 3D 19](#_Toc107180307)

[1.1. Mô hình phần chân 19](#_Toc107180308)

[1.2. Thiết lập chuyển động của chân 20](#_Toc107180309)

[1.3. Tạo phần thân và kết nối với phần chân 20](#_Toc107180310)

[1.4. Tạo đường di chuyển và hoàn thiện đối tượng 21](#_Toc107180311)

[1.5. Xây dựng hoạt ảnh 22](#_Toc107180312)

[2. Xây dựng không gian 3D 24](#_Toc107180313)

[3. Nhúng đối tượng 3D vào map 24](#_Toc107180314)

[4. Tương tác điều khiển đối tượng trong không gian 3D 25](#_Toc107180315)

[4.1. Điều khiển di chuyển 25](#_Toc107180316)

[4.2. Tạo âm thanh cho đối tượng 27](#_Toc107180317)

[4.3. Đối tượng né vật thể 28](#_Toc107180318)

[4.4. Thiết lập chuyển động 28](#_Toc107180319)

[4.5. Thiết lập góc nhìn 29](#_Toc107180320)

[CHƯƠNG III: KẾT QUẢ 30](#_Toc107180321)

[TÀI LIỆU THAM KHẢO 32](#_Toc107180322)

[SOURCE CODE 33](#_Toc107180323)

# **DANH MỤC HÌNH ẢNH**

[Hình 1.1 Cửa sổ làm việc Blender 7](#_Toc107180166)

[Hình 1.2 Mode làm việc 8](#_Toc107180167)

[Hình 1.3 Mode Scripting 8](#_Toc107180168)

[Hình 1.4 Ví dụ về scripting trong Blender 9](#_Toc107180169)

[Hình 1.5 Console tab 10](#_Toc107180170)

[Hình 1.6 Log tab 11](#_Toc107180171)

[Hình 1.7 Import OSM 12](#_Toc107180172)

[Hình 1.8 Get OSM 13](#_Toc107180173)

[Hình 1.9 Xây dựng công trình có trong map 13](#_Toc107180174)

[Hình 1.10 Properties 13](file:///E:\Tai%20lieu\20212\NDS\DM2P-Subproject_1A-2A-3-4C.docx#_Toc107180175)

[Hình 1.11 Detail’s mesh 14](#_Toc107180176)

[Hình 1.11 Lấy PID GPU 16](#_Toc107180177)

[Hình 1.13 Google map 16](#_Toc107180178)

[Hình 1.14 Render map 17](#_Toc107180179)

[Hình 1.15 Map 3D 17](#_Toc107180180)

[Hình 2.1 Mô hình chân 20](#_Toc107180182)

[Hình 2.2 Inverse Kinematic cho chân 20](#_Toc107180183)

[Hình 2.3 Thân và chân kết nối 21](#_Toc107180184)

[Hình 2.4 Follow path 22](#_Toc107180185)

[Hình 2.5 Hoàn thành tạo dáng đối tượng 3D 22](#_Toc107180186)

[Hình 2.6 Thêm khối điều khiển hướng chuyển động 23](#_Toc107180187)

[Hình 2.7 3D map Berlin tạo ra từ Google map và OSM 24](#_Toc107180188)

[Hình 2.8 Đối tượng 3D được nhúng vào 3D map 24](#_Toc107180189)

[Hình 2.9 Tool Render game trong UPBGE 25](#_Toc107180190)

[Hình 2.10 Logic điều khiển cho đối tượng 26](#_Toc107180191)

[Hình 2.11 Logic phát ra âm thanh cho đối tượng 27](#_Toc107180192)

[Hình 2.12 Logic chuyển động nhảy cho đối tượng 28](#_Toc107180193)

[Hình 2.13 Logic xoay hướng cho Direction Controller 28](#_Toc107180194)

[Hình 2.14 Logic chuyển động nhún nhảy của phần than 28](#_Toc107180195)

[Hình 2.15 Logic chuyển động chuột điều khiển camera 29](#_Toc107180196)

# **DANH MỤC BẢNG BIỂU**

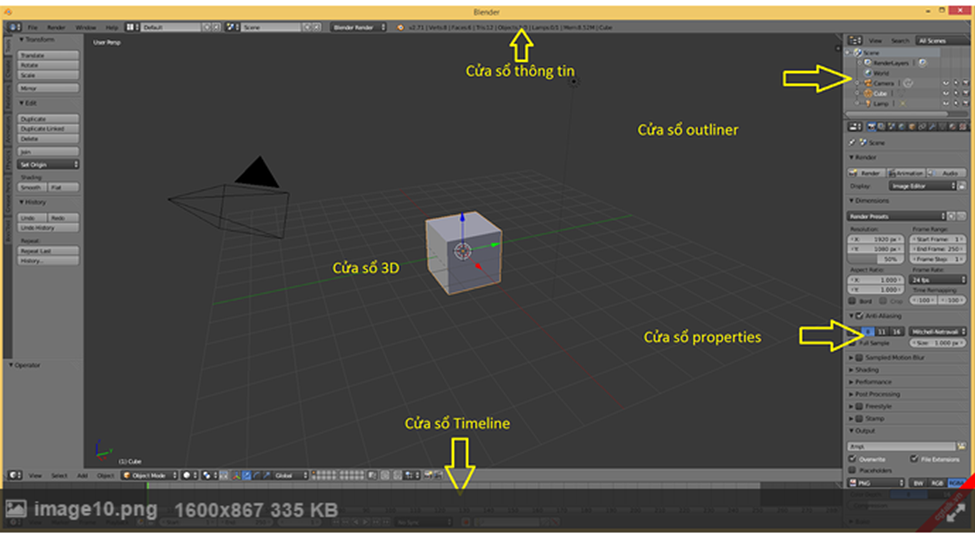
[**Bảng 1 Bảng phân chia công việc** 19](#_Toc107180121)

# **CHƯƠNG I. LÝ THUYẾT CHUNG**

* 1. Blender

Blender là phần mềm thiết kế đồ họa 3D có thể sử dụng hoàn toàn miễn phí, người dùng có thể sử dụng phần mềm này để dựng phim hoạt hình, làm kỹ xảo, mẫu in 3D và video game. Blender còn hỗ trợ cửa sổ terminal, cửa sổ soạn thảo giúp người dùng có thể lập trình python trong Blender kết hợp với các thư viện như BGE, UPBGE, ... ta có thể thao tác với các đối tượng bằng dòng lệnh.

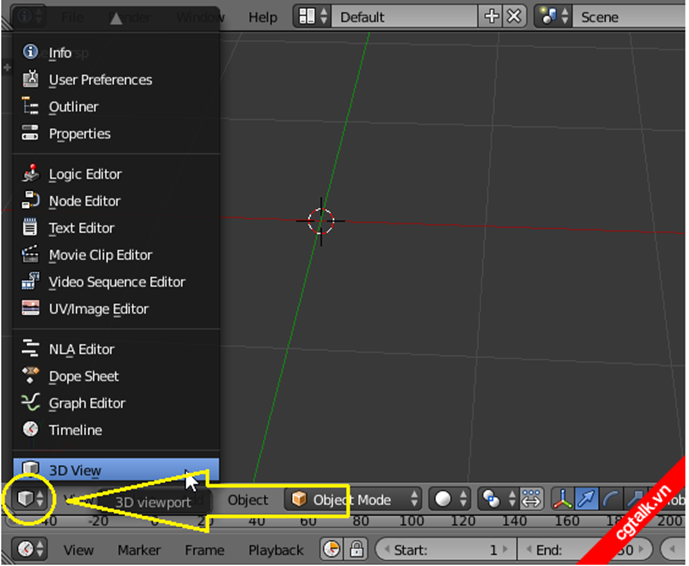
* + 1. Các cửa sổ cơ bản trong Blender



Hình 1.1 Cửa sổ làm việc Blender

* Cửa sổ thông tin (info): chứa các thông tin giao diện và các lệnh với file
* Của sổ 3D view: khu vực hiện thị đối tượng 3D
* Của sổ Outliner: quản lý đối tượng theo tên
* Của sổ Properites: Các thuộc tính của chương trình và thuộc tính đối tượng
* Cửa sổ Timeline: làm việc với animation

Nội dung trong các cửa sổ được quy định bởi biểu tượng ở góc trái ( trên hoặc dưới) Có thể thay đổi các biểu tượng để cửa sổ hiển thị nội dung mong muốn

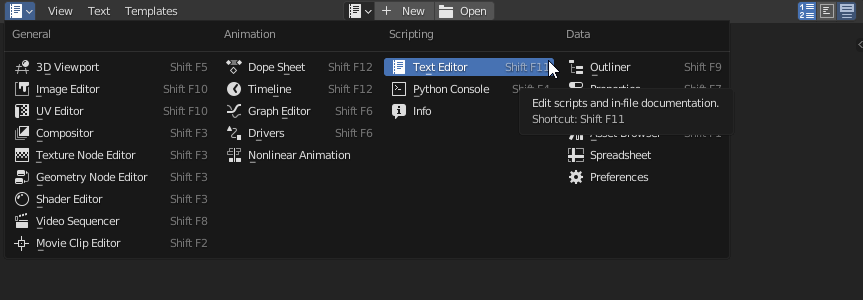


Hình 1.2 Mode làm việc

* + 1. Lập trình Python với Blender

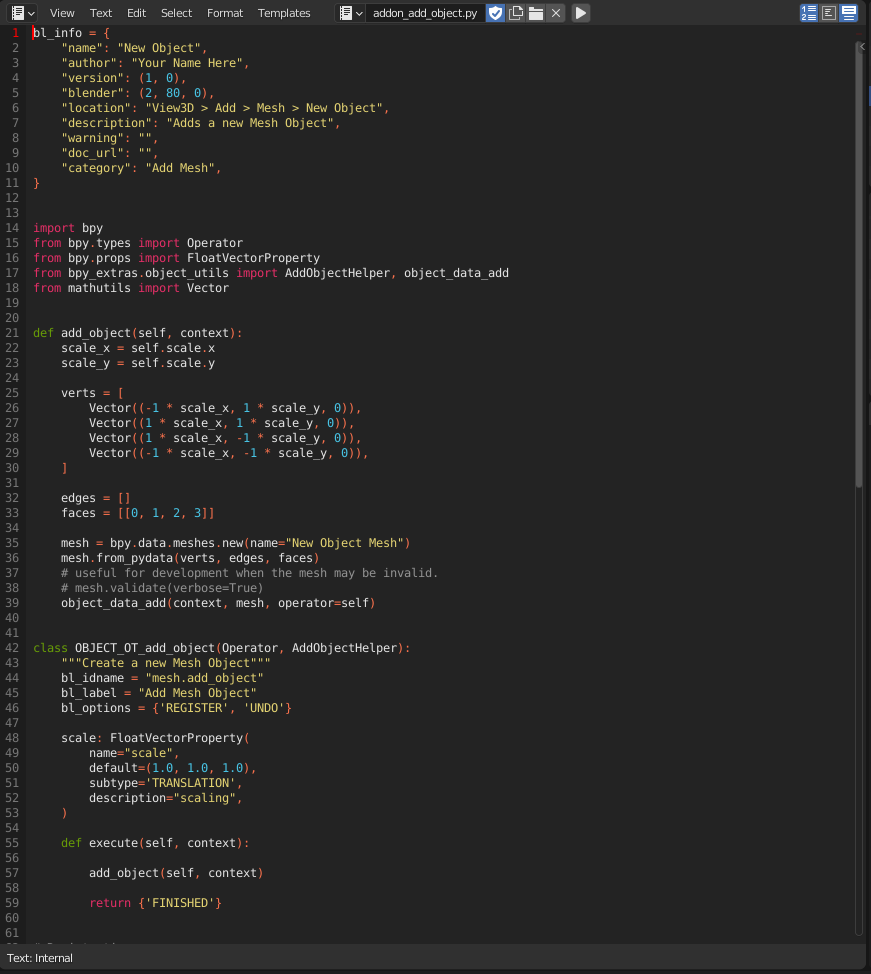
**Cửa sổ soạn thảo code**

Để hiển thị được cửa sổ soạn thảo, ta nhấn vào biển tượng góc trên bên trái của của sổ, sau đó ta chọn loại cửa sổ là Text Editor



Hình 1.3 Mode Scripting

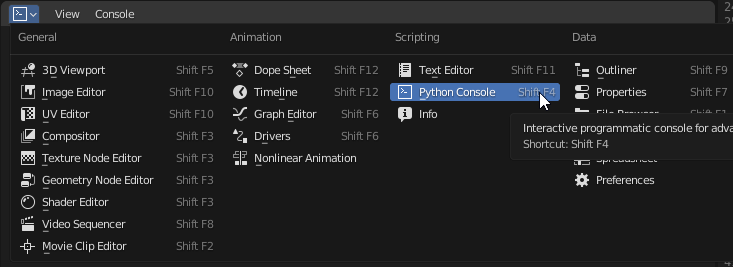
Với cửa sổ này ta có thể soạn thảo code ngay bên trong Blender, mà không cần dùng đến trình soạn thảo ở bên ngoài.

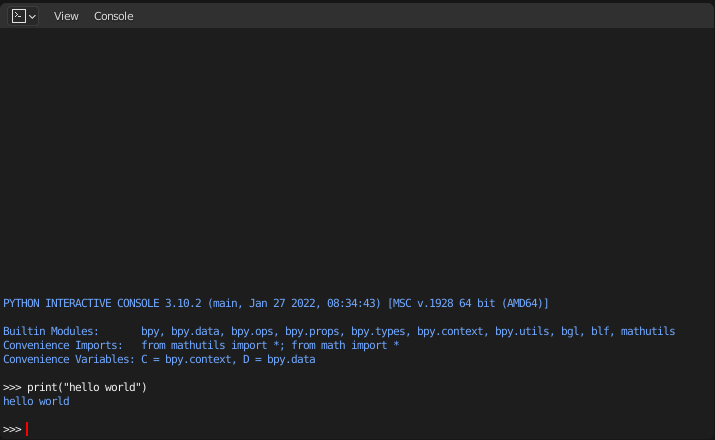


Hình 1.4 Ví dụ về scripting trong Blender

**Cửa sổ Terminal**

Để sử dụng cửa sổ terminal trong Blender ta làm tương tự và chọn Python Console

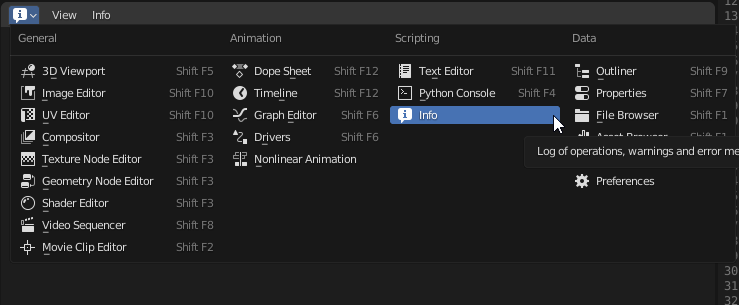


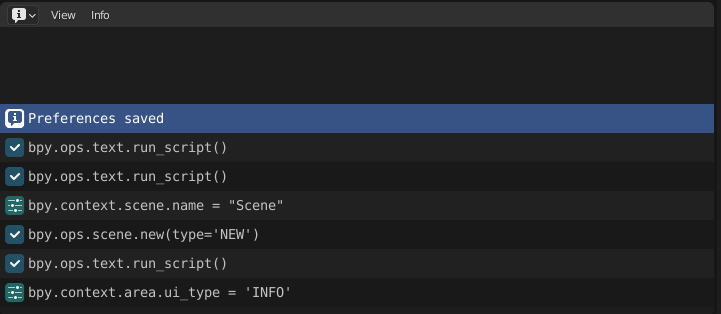


Hình 1.5 Console tab

**Cửa sổ infor**

Cửa sổ này dùng để hiện thị các lệnh khi ta thao tác trên blender ( kể cả thao tác trên UI Blender hay thông qua code)





Hình 1.6 Log tab

* 1. BlenderGIS

Blender gis là một addon trong trong phần mềm Blender, nó là cầu nối giữa Blender với dữ liệu địa lý ( geographic )

Với BlenderGIS chúng ta có thể lựa chọn khu vực trên bản đồ thế giới trực tiếp trong Blender hoặc import bản đồ (vệ tinh), bản đồ dịch chuyển và hình học như các tòa nhà bằng các file có định dạng như: Shapefile vector, raster image, geotiff DEM, OpenStreetMap xml.

* + 1. Georeferencing management

Để lưu trữ thông tin tham chiếu địa lý, BlenderGIS tạo một số thuộc tính tùy chỉnh trong scene:

**Hệ quy chiếu tọa độ (CRS)** còn được gọi là hệ quy chiếu không gian

Bản đồ thường được xây dựng thông qua phép chiếu bản đồ để biến đổi tọa độ góc trên bề mặt trái đất thành tọa độ phẳng và cho phép thể hiện một lãnh thổ giảm thiểu sự biến dạng hoặc giữ nguyên một số đặc tính nhất định.

Số phân định tham chiếu không gian ( Spatial Reference Identifier SRID) là một giá trị duy nhất được sử dụng để xác định rõ ràng hệ quy chiếu tọa độ. SRID có dagj AUTHORITY: CODE

**Scene origin geolocation**

Blender và hầu hết các phần mềm 3d không thể xử lý tốt các đối tượng cố nguồn gốc cảnh ở xa. Hơn nữa Blender hoạt động với các tọa độ dưới dạng độ chính xác đơn (float 32). Mặt khác dữ liệu địa lý được lưu trữ trên một hệ thống chiếu ( thường tính bằng mét ) cho giá trị tọa độ cao. Tọa độ được chiếu phải được biểu diễn dưới dạng độ chính xác kép (float 64).

Do đó ta cần tọa Mesh gần với Scene origin. Để tránh mất độ chính xác của tham chiếu địa lý, các thuộc tính tùy chỉnh được sử dụng để theo tọa độ Scene origin.

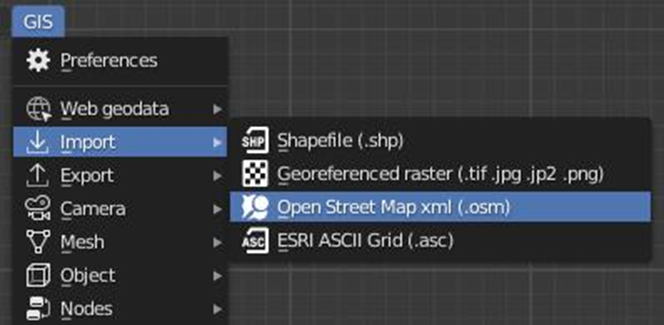
**Map scale denominator**

Đây là một thuộc tính cần thiết nếu muốn xây dựng một bản đồ bao trùm một lãnh thổ rộng lớn mà vẫn duy trì một cảnh hợp lý trong Blender.

* + 1. Nhập dữ liệu OSM

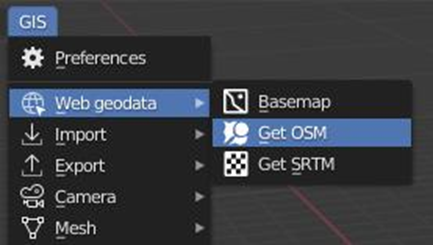
Có 2 cách nhập dữ liệu OSM:

Import trực tiếp file xml OSM: ta có thể lấy xml bằng công cụ xuất trên giao diện web OSM [www.openstreetmap.org](http://www.openstreetmap.org)



Hình 1.7 Import OSM

Sử dụng chức năng GET OSM để truy vấn trực tiếp từ Blender:



Hình 1.8 Get OSM

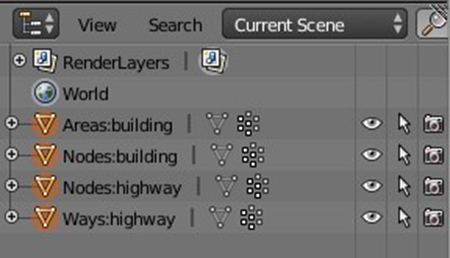
Chức năng này chỉ hiệu quả nếu cảnh được tham chiếu địa lý, chẳng hạn sử dụng addon basemaps.

**Operator options**

Importer cho phép lọc các đặc tính bằng loại (node, way hoặc area) và bằng tags

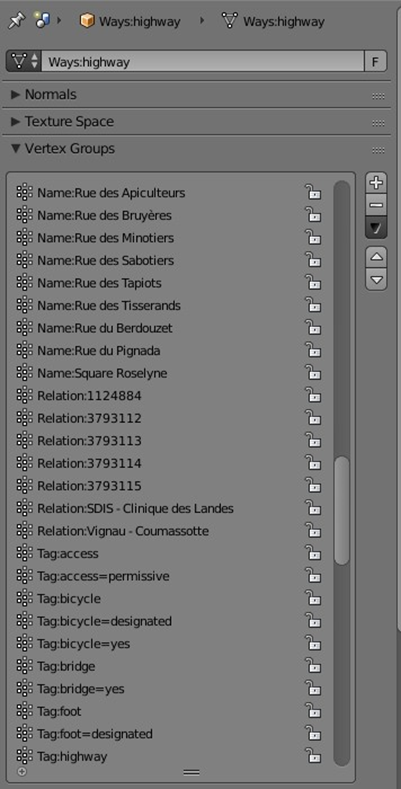


Hình 1.9 Xây dựng công trình có trong map

Các feature và filter tag sẽ có một mesh riêng cho từng loại

Các Mesh này chứa vertex group để truy xuất tên, tag và relations:

Hình 1.10 Properties



Hình 1.11 Detail’s mesh

* 1. Blender-OSM ( Open Street Map)

Là một addon cho phép chúng ta có thể tải về và impor OpenStreetMap và dữ liệu địa hình trên toàn cầu

**Các vật thể có thể import từ OSM:**

Tòa nhà: Chiều cao và số tầng của tòa nhà để tạo ra cảnh cuối cùng. Có đa dạng các kiểu hình dạng mái nhà: mái bằng, hình chóp, hình tròn, mái vòm ,.... Nếu địa hình được cung cấp, các tòa nhà sẽ tự động được đặt trên địa hình đó.

Nước ( Sông, hồ ) : Import dưới dạng đa giác. Đường bờ biển và các đại dương được import dưới dạng cạnh. Nếu địa hình được cung cấp, các da giác cho các đối tượng nước sẽ được chiếu trên địa hình.

Các khu vực thực vât ( rừng, cỏ, cây bụi). Import dưới dạng đa giác

Đường bộ, đường dẫn và đường sắt: import có chiều rộng. Chúng được biểu diễn bằng các đường cong Blender với một thông in của đối tượng

Địa hình: Blender-osm tải xuống và import dữ liệu địa hình thế giới thực với phạm vi toàn cầu và độ phân giải khoảng 30 mét

* 1. Google Maps 3D

Ta có thể import được bản đồ từ google map vào Blender thông qua một addon Maps Models Importer, addon này cho phép import file với định dạng .cdr, để tải được file .rdc này về từ google map ta phải sử dụng công cụ Render Doc

**Các bước thực hiện**

**Tạo một shortcut chrome**

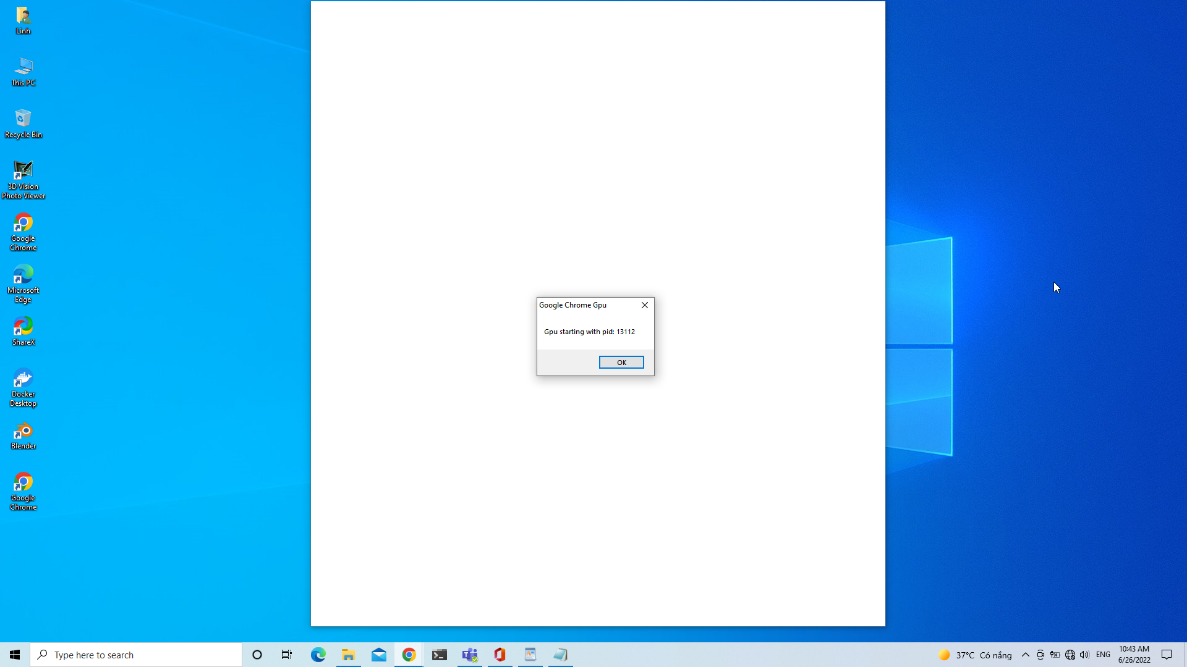
Chuột phải vào short cut vừa tạo > chọn properties > chèn đoạn mã bên dưới vào Target

C:\Windows\System32\cmd.exe /c "SET RENDERDOC\_HOOK\_EGL=0 && START "" ^"C:\Program Files\Google\Chrome\Application\chrome.exe^" --disable-gpu-sandbox --gpu-startup-dialog"

**Lấy Module từ google map**

Tắt hết chrome đang bật

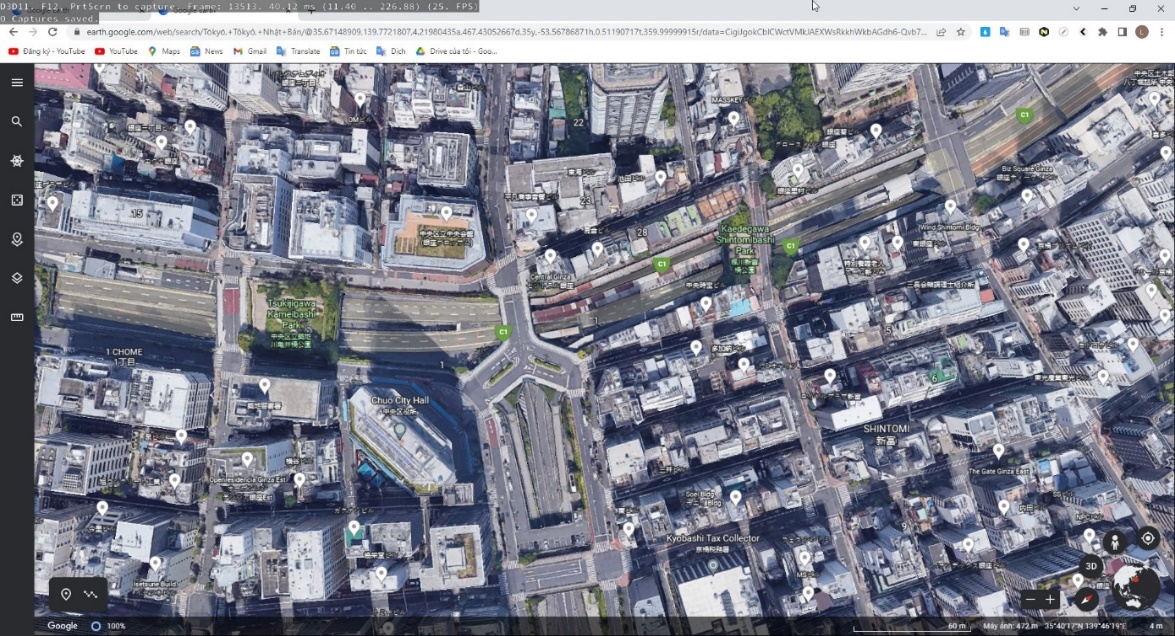
Mở chrome bằng shortcut vừa tạo > nó sẽ hiện ra mã PID GPU



Hình 1.11 Lấy PID GPU

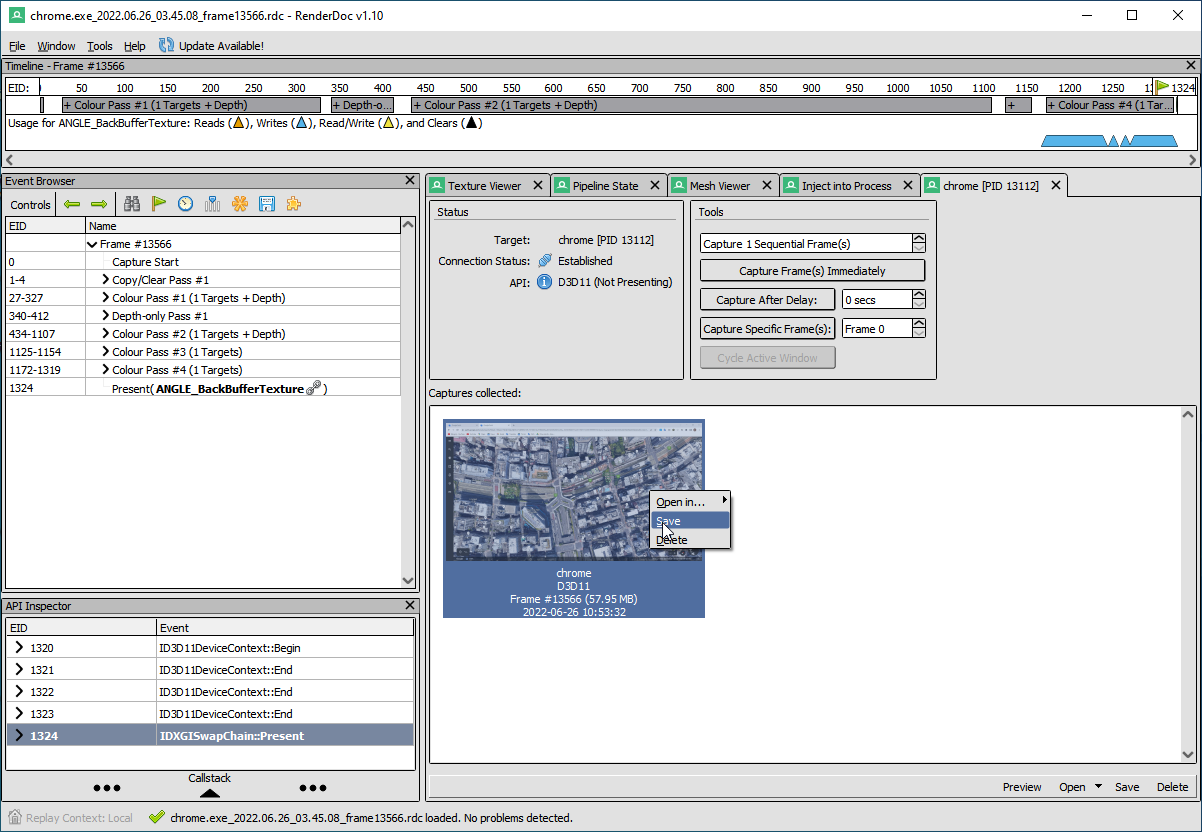
Mở Render Doc, vào file > inject into project > tìm chrome có PID GPU > click đúp chọn

Vào google chrome truy cập đường dẫn <https://www.google.com.vn/intl/vi/earth/> sau đó chọn địa điểm muốn lấy



Hình 1.13 Google map

Mở render doc chọn capture Frame

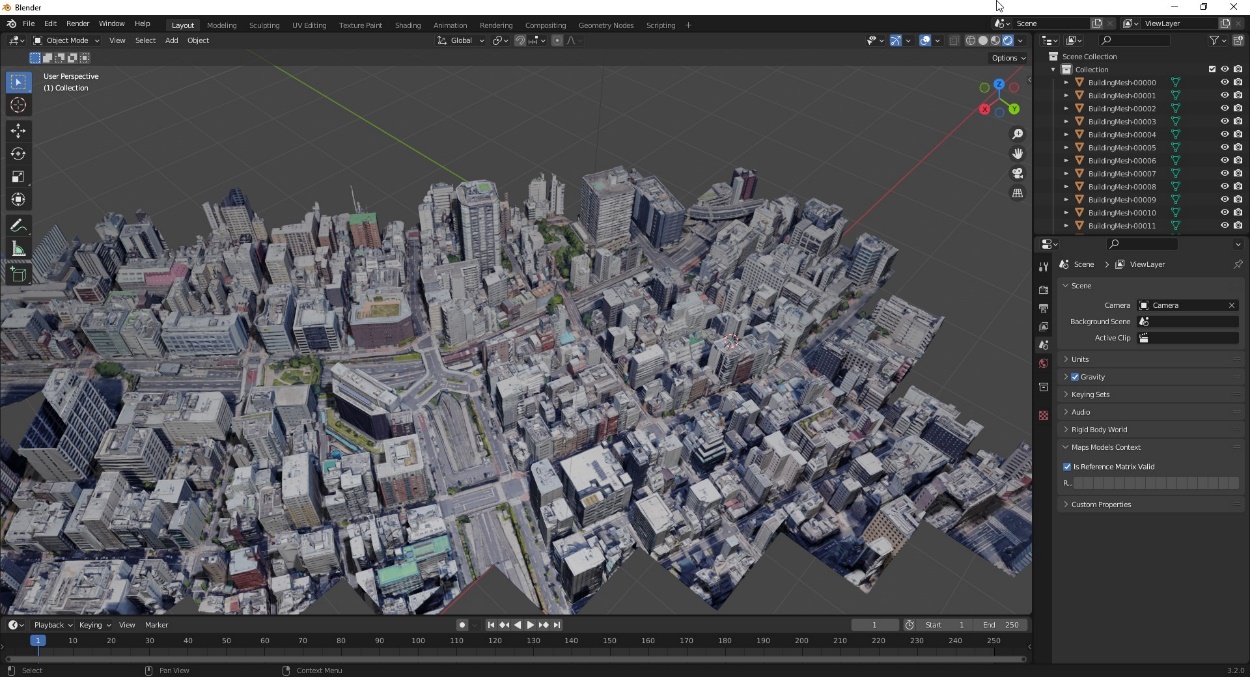


Hình 1.14 Render map

Click đúp vào hình thu nhỏ

Click chuột phải, chọn save, File này có định dạng .rdc

Sau đó mở Blender có cài addon Maps Models Importer, vào File > import > Google Map Capture > chọn file .rdc vừa lưu



Hình 1.15 Map 3D

* 1. UPBGE
     1. Định nghĩa

UPBGE là một mã nguồn mở, là công cụ tạo game 3D được tách ra từ Blender Game Engine và được triển khai trên chính Blender. Quy trình làm việc thống nhất này chính là điểm mạnh chính của nó, và chính vì vậy, ta có thể tạo ra trò chơi từ khi bắt đầu đến khi kết thúc mà không cần thoát Blender.

* + 1. Các tính năng
* Được tích hợp trong công cụ Blender 3D.
* Trình kết xuất đồ họa của UPBGE là trình kết xuất khung nhìn thời gian thực của Blender, gọi là Eevee.
* Hỗ trợ viết kịch bản và logic.
* Có hệ thống hoạt ảnh, vật lý, OpenXR, trình soạn thảo đa nền tảng, điều hướng, âm thanh, và triển khai đa nền tảng.

# **CHƯƠNG II. TRIỂN KHAI CÔNG VIỆC**

Chương này sẽ bao gồm nội dung công việc đã tiến hành, bao gồm các mục sau:

* Tổng hợp đối tượng 3D có các chuyển động cực bộ
* Xây dựng không gian 3D từ mã nguồn mở Google Maps và Open Street Maps
* Nhúng đối tượng 3D đã tổng hợp ở mục 1 vào không gian 3D ở mục 2
* Sử dụng API của đối tượng 3D ở mục 1, cho phép người dừng điều khiển chuyển động của đối tượng. Trong quá trình điều khiển, đối tượng cho phép người dùng tương tác với các tính năng đã chọn. Đối tượng có thể phát ra âm thanh 3D, có thể né chướng ngại vật

1. Tổng hợp đối tượng 3D

Đối tượng tổng hợp là 1 sinh vật 4 chân, có các di chuyển cục bộ, chuyển hướng, nhún nhảy.

1.1. Mô hình phần chân

Để có được chuyển động uyển chuyển thì mô hình phần chân cần có khung xương và các khớp nối. Ta sẽ tạo ra các joints được nối với nhau và bọc nó bằng da bao phủ xong quanh để tạo nên 1 chân:

Chart, radar chart

Description automatically generated

Hình 2.1 Mô hình chân

1.2. Thiết lập chuyển động của chân

Ta sẽ dùng 1 kỹ thuật tên là Inverse Kinematic để điều khiển chuyển động cục bộ của toàn bộ phần chân. Kỹ thuật này sẽ tính toán mối liên hệ của khớp nối điểm cuối với các khớp nối liên quan để khi ta điều khiển khớp nối điểm cuối các khớp nối còn lại sẽ có chuyển động phù hợp theo sau. Ta sẽ gọi khối điều khiển này là IK.

**Chart, radar chart

Description automatically generated**

Hình 2.2 Inverse Kinematic cho chân

1.3. Tạo phần thân và kết nối với phần chân

Phần thân có thể là bất kỳ khối hình nào. Vì mục tiêu đề tài chưa nhắm đến chuyển động uốn dẻo của phần thân nên ta sẽ mô hình phần thân 1 cách đơn giản. Phần thân sẽ là 1 khối lập phương có kích thước phù hợp với phần chân đã tạo ra. Để làm cho đối tượng trông tự nhiên hơn thì ta sẽ tạo 1 khối cầu ghép nối giữa chân và thân.

Câu hỏi đặt ra là 2 khối thân và chân tách rời thì sao có thể đồng bộ được chuyển động. Vì vậy ta cần phải tạo mối liên hệ giữa hai khối này. Phần chân sẽ là bậc con và phần thân là bậc cha, mối quan hệ này sẽ được lí giải sau ở mục điều hướng chuyển động. Lúc này chuyển động của phần thân sẽ kéo theo thay đổi hình dạng của phần chân, còn phần chân sẽ có liên kết với phần thân nhưng sẽ không kéo theo chuyển động của thân.

A picture containing text

Description automatically generated

Hình 2.3 Thân và chân kết nối

1.4. Tạo đường di chuyển và hoàn thiện đối tượng

Phần chân sẽ chuyển động theo khối IK và theo 1 hình nửa vòng tròn, giống với quỹ đạo thông thường của chân. Tạo 1 path và sau đó đặt cho IK di chuyển theo path đó “Follow path”.

A white lamp on a tile floor

Description automatically generated with low confidence

Hình 2.4 Follow path

Phần chân sẽ chuyển động theo khối IK và theo 1 hình nửa vòng tròn, giống với quỹ đạo thông thường của chân. Tạo 1 path và sau đó đặt cho IK di chuyển theo path đó “Follow path”.

A picture containing text, floor, outdoor object

Description automatically generated

Hình 2.5 Hoàn thành tạo dáng đối tượng 3D

1.5. Xây dựng hoạt ảnh

Tổng quan về hoạt động của đối tượng sẽ là chuyển động của các chân và chuyển động của toàn bộ đối tượng so với không gian.

* Để tạo chuyển động các chân ta tạo ra các keyframe về offset của IK di chuyển theo đường Follow path. Khi đó ta thấy các chân của đối tượng cùng chuyển động giống nhau và trông sẽ rất kì cục. Do đó ta cần đặt keyframe của các chân khác nhau để có chuyển động theo ý muốn.
* Để chuyển động theo không gian, đối tượng cần chuyển động toàn bộ các khối trên cơ thể 1 cách đồng bộ. Ta sẽ tạo 1 khối lập phương emty (gọi là Cube) làm bậc cha và kết nối với tất cả các phần của đối tượng với bậc là con. Sau đó chỉ việc đặt các keyframe cho Cube để có thể đối tượng chuyển đông theo Cube.

Mặt khác ta muốn đối tượng của thể đổi hướng chuyển động (từ x sang y, y sang x, …) mà các chân cũng chuyển động theo hướng đó. Ta sẽ tạo ra 1 khối Empty→ Sphere làm khối điều hướng (gọi là Direction Controller). Khối này sẽ là khối cha kết nối với tất cả Follow path để khi Direction controller đối hướng sẽ kéo theo sự đổi hướng của các Follow path. Tã sẽ đặt các Follow path sao chép Rotation của khối Direction Controller.

Diagram

Description automatically generated

Hình 2.6 Thêm khối điều khiển hướng chuyển động

Từ đây đối tượng 3D của đã được hoàn thiện. Ta có thể đặt các keyframe phù hợp cho phân cảnh của mình.

* 1. Xây dựng không gian 3D

Khung cảnh được trích xuất cho đề tài sẽ là 1 góc của thành phố Berlin. Các vật thể có trong map được tạo ra sẽ chỉ gồm những kiến trúc tòa nhà và đường xá.

A map of a city

Description automatically generated with low confidence

Hình 2.7 3D map Berlin tạo ra từ Google map và OSM

* 1. Nhúng đối tượng 3D vào map

Từ Blender có map đã tạo, ta sẽ Append đối tượng đã tổng hợp vào map. Bối cảnh của nhóm đặt sẽ có 2 đối tượng 3D, 1 to 1 nhỏ.

A picture containing text, outdoor, aircraft

Description automatically generated

Hình 2.8 Đối tượng 3D được nhúng vào 3D map

Các đối tượng 3D và tòa nhà trên map cần được cài đặt lại cại các rằng buộc vật lí để có thể tương tác với nhau:

* Nền đất (Plane) cần đặt là Static để cố định và không thể tác động
* Các tòa nhà có thể dịch chuyển hoặc phá hủy, tuy nhiên nhóm sẽ đặt các tòa nhà là các khối cố định làm vật cản cho đối tượng 3D
* Đối được 3D được đặt ở dạng Character có thể chuyển động và tương tác, loại va chạm sẽ là Capsule để phù với đối tượng.
  1. Tương tác điều khiển đối tượng trong không gian 3D

Để có thể sử dụng game engine thì ta sẽ sử dụng framework upbge 0.3g. Mọi mô phỏng và lập trình sẽ thực hiện trên môi trường này. Quá trình mô phỏng sẽ được bắt đầu khi ta Render game:

A screenshot of a phone

Description automatically generated with low confidence

Hình 2.9 Tool Render game trong UPBGE

* + 1. Điều khiển di chuyển

Có 2 cách để có thể điều khiển được đối tượng 3D của ta: sử dụng công cụ Logic Brick Editors (LBE) hoặc lập trình đối tượng với python với các API được cung cấp bởi Blender Game Engine (bge).

Nhóm đã sử dụng cả 2 cách:

* Đối tượng 3D nhỏ được thiết lập điều khiển bằng LBE:

A picture containing text, electronics, calculator, black

Description automatically generated

Hình 2.10 Logic điều khiển cho đối tượng

* Đối tượng 3D nhỏ được thiết lập điều khiển bằng python:

class Movement(bge.types.KX\_PythonComponent):

args = OrderedDict([

('Speed', 0.0)

])

def start(self, args):

self.speed = args['Speed']

self.char = bge.constraints.getCharacter(self.object)

def update(self):

if W.active:

self.object.applyMovement([0,self.speed,0], 1)

if S.active:

self.object.applyMovement([0,-self.speed,0], 1)

if A.active:

self.object.applyMovement([-self.speed,0,0], 1)

if D.active:

self.object.applyMovement([self.speed,0,0], 1)

if SPACE.active:

self.char.jump()

import bge

from collections import OrderedDict

from mathutils import Vector

if not hasattr(bge,'\_\_component\_\_'):

scene = bge.logic.getCurrentScene()

key = bge.logic.keyboard.inputs

ms = bge.logic.mouse.inputs

W = key[bge.events.WKEY]

A = key[bge.events.AKEY]

S = key[bge.events.SKEY]

D = key[bge.events.DKEY]

SPACE = key[bge.events.SPACEKEY]

lmb = ms[bge.events.LEFTMOUSE]

rmb = ms[bge.events.RIGHTMOUSE]

* + 1. Tạo âm thanh cho đối tượng

Đối tượng sẽ được thiết lập để mỗi khi thay đổi vị trí tọa độ xyz sẽ phát ra 1 đoạn âm thanh

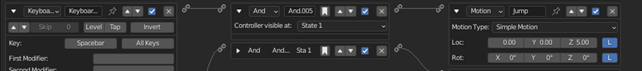
Graphical user interface, application

Description automatically generated

Hình 2.11 Logic phát ra âm thanh cho đối tượng

* + 1. Đối tượng né vật thể

Các vật thể và đối tượng đã được thiết lập ở mục 3 để cho chúng ko thể đi xuyên qua nhau, các vật thể sẽ chặn hướng di chuyển của đối tượng 3D. Nhóm đã tạo 1 nút điều khiển “Spacebar” để khi gặp chướng ngại vật, người dùng có thể khiến đối tượng 3D nhảy qua vật thể



Hình 2.12 Logic chuyển động nhảy cho đối tượng

* + 1. Thiết lập chuyển động

Để chuyển động của đối tượng trở nên mượt mà, ta cần thiết lập chuyển động của đối tượng thay đổi ứng với mỗi nút điều khiển

* Thiết lập cho Direction Controller:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 2.13 Logic xoay hướng cho Direction Controller

* Thiết lập cho Body:

A screenshot of a computer

Description automatically generated with medium confidence

Hình 2.14 Logic chuyển động nhún nhảy của phần than

* + 1. Thiết lập góc nhìn

Nhóm đặt camera gắn với đối tượng để ta có thể trải nhiệm góc nhìn thứ 3 đối với đối tượng. Ta có thể chuyển động quay camera bằng chuột để thay đổi sang các góc nhìn bên cạnh



Hình 2.15 Logic chuyển động chuột điều khiển camera

CHƯƠNG III: KẾT QUẢ

* Đã tổng hợp được đối tượng 3D có các chuyển động cực bộ
* Đã xây dựng không gian 3D từ mã nguồn mở Google Maps và Open Street Maps
* Đối tượng 3D đã được nhúng vào không gian 3D
* Đối tượng 3D đã có thể điều khiển được:
* Di chuyển theo không gian Oxy bằng bàn phím: W A S D
* Nhảy lên các địa hình cao bằng nút Spacebar
* Phát ra âm thanh khi di chuyển
* Nhún nhảy thân mỗi khi di chuyển

**TỔNG KẾT**

Qua quá trình học tập nghiên cứu và hoàn thành đề tài, nhóm đã học được nhiều kỹ năng và kiến thức thú vị về một lĩnh vực trái ngành. Những trò chơi và thước phim kỹ xảo từ trước đến này đối với nhóm là những thứ xa vời và không thể hiều được nhưng đề tài đã đem đến cho nhóm 1 cái nhìn gần gũi hơn về các kỹ thuật về đồ họa này.

Với độ mở của đề tài thì đề tài hoàn toàn có thể mở rộng thêm nữa. Ngoài những chuyển động cục bộ đơn gian đối tượng tổng hợp có thể được tương tác thêm với môi trường hoặc những đối tượng 3D khác. Trong thời gian rảnh thì đây là 1 đề tài thú vị để ta mày mò và tự làm cho mình 1 sản phẩm mà ta cảm thấy hứng thú như này.

Em xin chân thành cảm ơn thầy Phạm Văn Tiến đã hướng dẫn em có thể hoàn thành đề tài.

TÀI LIỆU THAM KHẢO

[1] <https://upbge.org>

[2] <https://github.com/eliemichel/MapsModelsImporter/releases/tag/v0.5.0-rc1>

[3] <https://github.com/vvoovv/blender-osm/wiki/Documentation>

[4] <https://github.com/domlysz/BlenderGIS/wiki>

SOURCE CODE

<https://github.com/zViolett/Game3D_Project_1A-2A-3-4C>