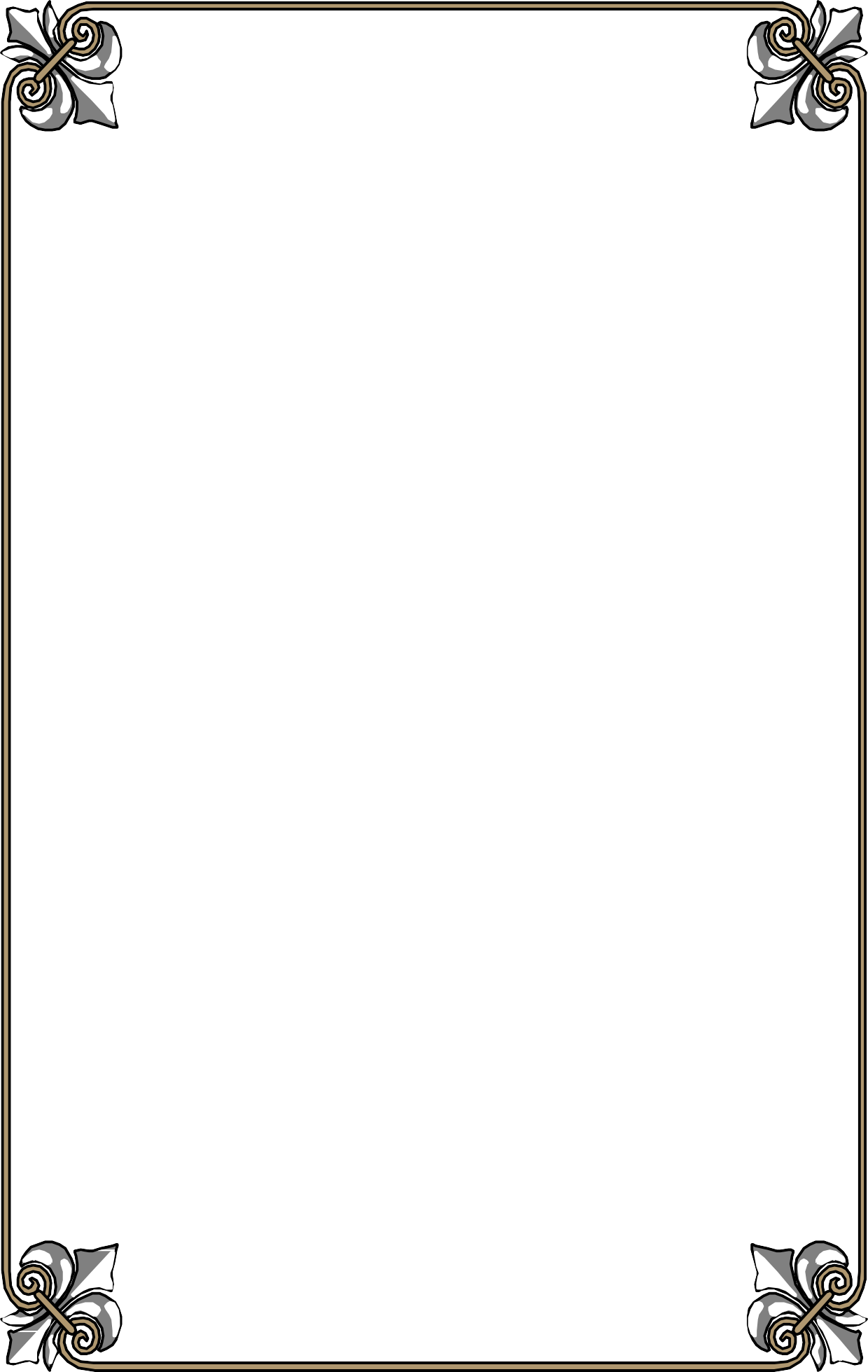
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**VIỆN ĐIỆN TỬ-VIỄN THÔNG**

**BÁO CÁO BÀI TẬP DÀI MÔN HỌC**

**Kỹ thuật lập trình nội dung số**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | TS. Phạm Văn Tiến |
| Sinh viên thực hiện: | MSSV |
| ***Nguyễn Hữu Linh***  ***Hoàng Anh Tú*** | ***20192967***  ***20172884*** |
| ***Lưu Đình Việt Ân*** | ***20186300*** |
|  |  |

*Hà Nội, 1/202**3*

# Lời cảm ơn

Trách nhiệm của bọn em là bày tỏ sự biết ơn, thừa nhận đối với những người đã dẫn dắt, tạo điều kiện cho bọn em trong quá trình thực hiện dự án.

Trước tiên, bọn em muốn gửi lời cảm ơn sâu sắc nhất đến Ts. Phạm Văn Tiến. Nếu không có sự hướng dẫn của thầy trong suốt quá trình thực hiện, thì dự án của chúng em không thể hoàn thành. Chúng em học cũng hỏi thêm được rất nhiều điều trong cách làm việc của thầy qua những lần trao đổi với thầy trên lớp.

Bên cạnh đó, chúng em cũng cảm ơn các thành viên trong lớp đã nhiệt tình đưa ra góp ý và xây dựng môi trường học tập, điều này ảnh hưởng tốt đến nhiều mặt trong quá trình thực hiện dự án của bọn em.

Nguyễn Hữu Linh

Hoàng Anh Tú

Lưu Đình Việt Ân

Mục lục

[Lời cảm ơn 2](#_Toc126295194)

[Mục lục 3](#_Toc126295195)

[Danh sách hình ảnh 4](#_Toc126295196)

[I. Tải và cài đặt các phần mềm cần thiết. 5](#_Toc126295197)

[1.1. Visual Studio. 5](#_Toc126295198)

[1.2. Unity Hub. 5](#_Toc126295199)

[1.3. Blender. 5](#_Toc126295200)

[1.4. Ngôn ngữ lập trình C#. 5](#_Toc126295201)

[II. Thực hiện các yêu cầu. 6](#_Toc126295202)

[1. 1A. 6](#_Toc126295203)

[2. 2A. 8](#_Toc126295204)

[3. 3. 10](#_Toc126295205)

[4. 4B. 11](#_Toc126295206)

[Tham chiếu 13](#_Toc126295207)

# Danh sách hình ảnh

|  |  |
| --- | --- |
| Số mục | Tên mục |
| Hình 1 | Mô hình đối tượng trực thăng |
| Hình 2 | Bề mặt va chạm của đối tượng. |
| Hình 3 | Đoạn mã xử lý việc thay đổi góc nhìn |
| Hình 4 | Thành phố được xuất ra từ OSM |
| Hình 5 | Đối tượng trực thăng xuất hiện trong phối cảnh thành phố. |
| Hình 6 | Đối tượng tương tác vật lý với phối cảnh. |
|  |  |
|  |  |

# Tải và cài đặt các phần mềm cần thiết.

* 1. Visual Studio.

Visual Studio là một môi trường phát triển phần mềm mạnh mẽ của Microsoft. Nó được sử dụng rộng rãi trong cộng đồng lập trình viên để phát triển các ứng dụng cho Windows, web, di động, và nhiều nền tảng khác. Phần mềm hỗ trợ cho nhiều ngôn ngữ lập trình như C++, C#, và Visual Basic. Nó cung cấp các công cụ giúp cho việc tạo các giao diện người dùng (GUI) trực quan và có hỗ trợ cho việc tự động hóa các quy trình phát triển phần mềm.

* 1. Unity Hub.

Unity là một nền tảng phát triển game đa nền tảng và phần mềm mã nguồn mở. Nó cho phép nhà phát triển tạo các trò chơi cho các nền tảng như Windows, MacOS, iOS, Android, và Web.

Unity Hub là một công cụ quản lý Unity, cho phép người dùng quản lý các phiên bản Unity, cài đặt các add-ons và plugins, và quản lý các dự án Unity. Nó cung cấp một giao diện đơn giản và dễ sử dụng cho việc quản lý môi trường phát triển Unity của bạn.

* 1. Blender.

Blender là một phần mềm tạo hình 3D miễn phí và mã nguồn mở. Nó cho phép người dùng tạo ra các đồ họa 3D, chế tạo ra các mô hình, tạo ra các trò chơi, và thực hiện các tác vụ dựng phim và hợp tác dựng phim. Blender cung cấp một giao diện người dùng tương tác và các công cụ tạo hình và dựng hình mạnh mẽ, cho phép người dùng tạo ra những đồ họa 3D chân thực và đẹp mắt.

* 1. Ngôn ngữ lập trình C#.

C# là một ngôn ngữ lập trình được phát triển bởi Microsoft cho mục đích phát triển phần mềm trên nền .NET framework. Nó cung cấp một cấu trúc dễ sử dụng, tính tự động và tính năng tốt để hỗ trợ việc phát triển phần mềm.

Trong lập trình game và nội dung số, C# được sử dụng rộng rãi vì nó cung cấp một cấu trúc tốt để xử lý các tác vụ phức tạp, cung cấp các tính năng tốt cho việc xử lý logic và xử lý sự kiện, và hỗ trợ việc tương tác với các thiết bị ngoại vi. Nó cũng được sử dụng với một số nền tảng phát triển game, chẳng hạn như Unity, để tạo ra các trò chơi và nội dung số.

# Thực hiện các yêu cầu.

1. 1A.
2. Yêu cầu.

Tổng hợp đối tượng đồ họa 3D là máy móc, hình nhân... Sau đó, có thể trang trí đối tượng này theo ý muốn (thêm màu sắc, chất liệu, chiếu sáng...). Đối tượng này thay đổi hình dạng, vị trí khi có một trong các tương tác sau của người dùng, bên cạnh thao tác chuột và phím nhấn: Tiếng nói.

Đối tượng 3D với các thuộc tính và tính năng vừa được tạo sẽ được tái sử dụng, nhúng vào trong các phối cảnh không gian cụ thể. Do đó, hãy đóng gói mã nguồn thành API để hỗ trợ người sáng tạo nội dung có thể lập trình sau này (ví dụ, trong Python, C#, C++...).

1. Thực hiện.

Xây dựng đối tượng: Trực thăng.

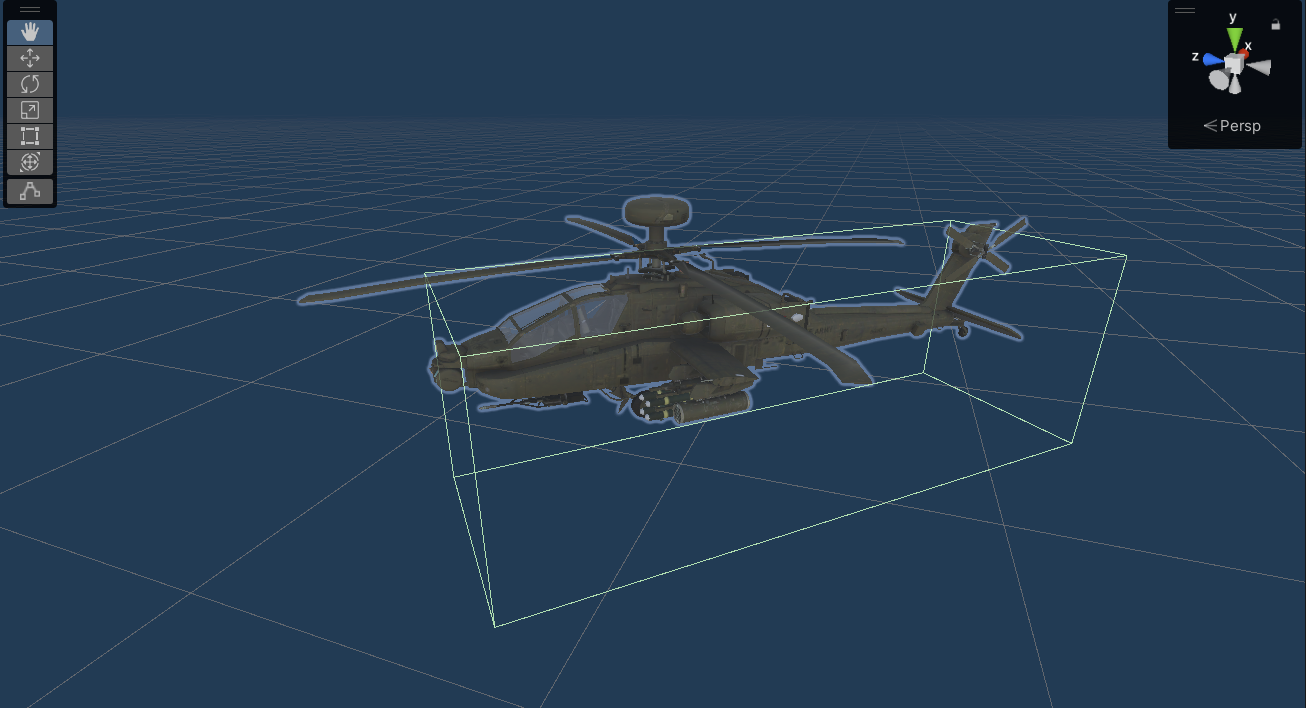


*Hình 1: Mô hình đối tượng trực thăng.*

Người dùng nhấn (lần lượt hoặc đồng thời) các phím điều hướng (trên/xuống/phải/trái) và các phím t/T, c/C để cung cấp dữ liệu vào. Các đại lượng như lực nâng, lực tiến, lực lùi, lực rẽ, độ nghiêng sẽ được tính toán thông qua dữ liệu bên trên để áp dụng vào điều khiển, từ đó đối tượng có thể thay đổi vị trí trong một không gian cụ thể. Để thêm tính thực tế, đối tượng có chịu những ảnh hưởng vật lý như va chạm, khối lượng.

Trong không gian 3 chiều, đối tượng là một vật (có khối lượng m). Để đối tượng có thể di chuyển trong không gian, ta cần tác động vào vật để vật chuyển động theo quy tắc của đối tượng trực thăng trong thực tế.

* Trực thăng bay lên: vật (có khối lượng m) cần được động bởi một lực F theo phương hướng lên. Lực F hướng lên này, tối thiểu phải lớn hơn lực hướng xuống được tạo bởi vật (có khối lượng m) và gia tốc trọng trường (trong cấu hình vật lý của bối cảnh). Mỗi lần nhấn phím t/T, ta sẽ tăng một lượng gia tốc nhỏ cho lực F. Đến khi lực F đủ lớn thì vật sẽ từ từ được nhấc lên khỏi mặt đất. Mô phỏng này giống với hành vi của đối tượng trực thăng trong thực tế.
* Chuyển động của dao quạt gió: chi tiết dao quạt gió sẽ được chuyển động nhanh dần (tỉ lệ với gia tốc) quanh trục cố định để mô phỏng lưỡi dao quạt gió trong thực tế.
* Trực thăng tiến về phía trước: Khi vật không ở trên mặt đất (khoảng cách với mặt đất > 0), để vật di chuyển về phía trước thì ta cần một lực F1 về phía trước để kéo vật di chuyển. Tương tự khi vật lùi về phía sau. Đồng thời, mô hình trực thăng sẽ ngả vể phía trước hoặc phía sau.
* Trực thăng rẽ sang trái hoặc phải: Để vật có thể rẽ, ta cần một momen quay để mô phỏng thực tế. Đồng thời, mô hình trực thăng sẽ nghiêng về bên phải hoặc bên trái.



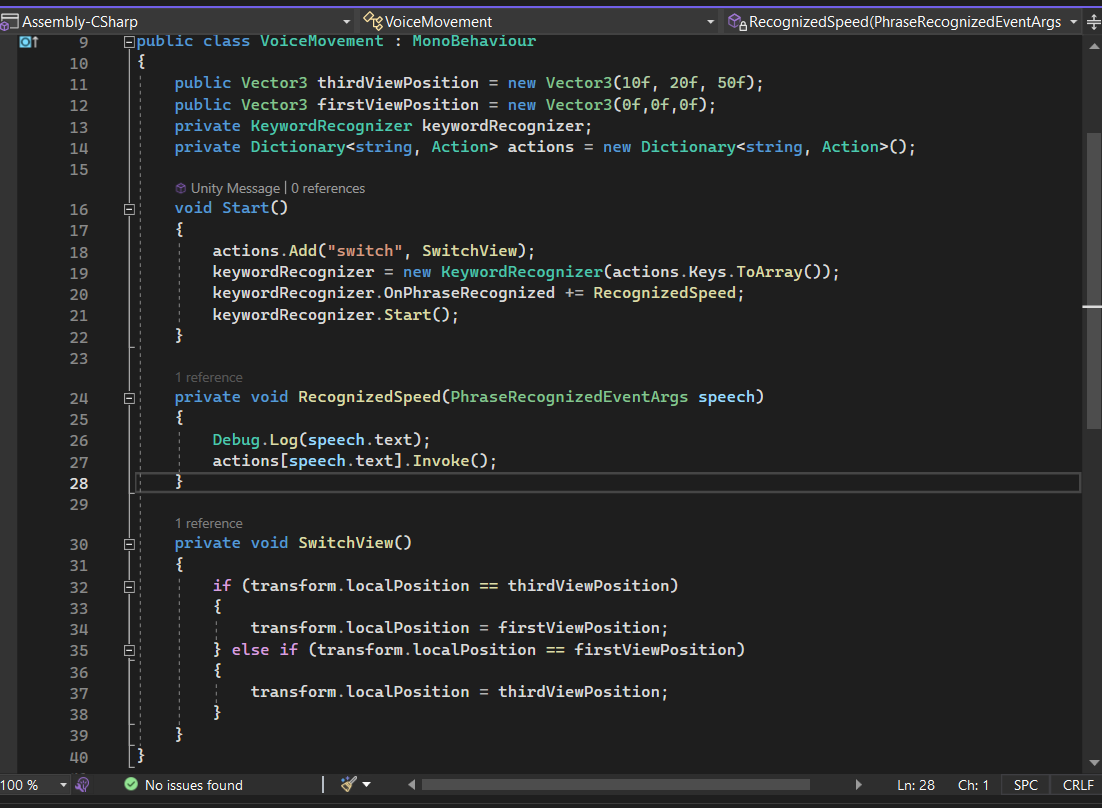
*Hình 2: Bề mặt va chạm của đối tượng.*

Vật có thể tác động vật lý đến vật khác, ta cần tạo thêm bề mặt cho mô hình. Bề mặt này sẽ là nơi xảy ra va chạm hoặc tương tác vật lý với các vật/bối cảnh khác.

Muốn nhìn thấy vật trong bối cảnh, ta cần bổ sung thêm một góc quay từ đằng sau. Khi có thêm góc quay, ta sẽ dễ dàng quan sát được các hành động của đối tượng trong không gian. Hoặc có thể nhìn ra không gian từ góc nhìn của vật. Nói cách khác, đây là góc nhìn thứ nhất và góc nhìn thứ ba. Góc nhìn là một phần của đối tượng, ta sẽ thêm tính năng chuyển góc quay dựa thông qua mệnh lệnh từ âm thanh.

Về phần mềm Unity, tính năng chuyển đổi đóng gói trên các nền tảng (Build Platform) có hỗ trợ tính năng điều khiển thông qua giọng nói này. "Build platform" trong Unity Hub là một tùy chọn cho phép biên dịch và xuất bản mã nguồn của một dự án Unity sang một nền tảng hoặc hệ điều hành khác nhau, chẳng hạn như Windows, MacOS, iOS, Android hoặc WebGL. Khi tạo một Build, Unity sẽ sử dụng các tài nguyên, thư viện và mã nguồn của dự án để tạo ra một bản định dạng có thể chạy trên nền tảng đã chọn. Việc sử dụng Build Platform trong Unity Hub giúp bạn dễ dàng quản lý và xuất bản các Build cho dự án của mình một cách thuận tiện. Nền tảng được sử dụng ở đây là “Universal Windows Platform”. Ta sẽ cấu hình dự án cho phép sử dụng microphone.

Sau khi cấu hình, ta cần thêm file mã nguồn để điều khiển góc nhìn và kích hoạt hành động mỗi khi có tín hiệu.



*Hình 3: Đoạn mã xử lý việc thay đổi góc nhìn.*

Khi khởi chạy, mỗi lần người dùng nói “switch” thì góc quay sẽ chuyển vị trí từ góc nhìn thứ nhất sang góc nhìn thứ ba. Nói thêm lần nữa để chuyển tiếp về vị trí thứ nhất.

1. 2A.
2. Yêu cầu.

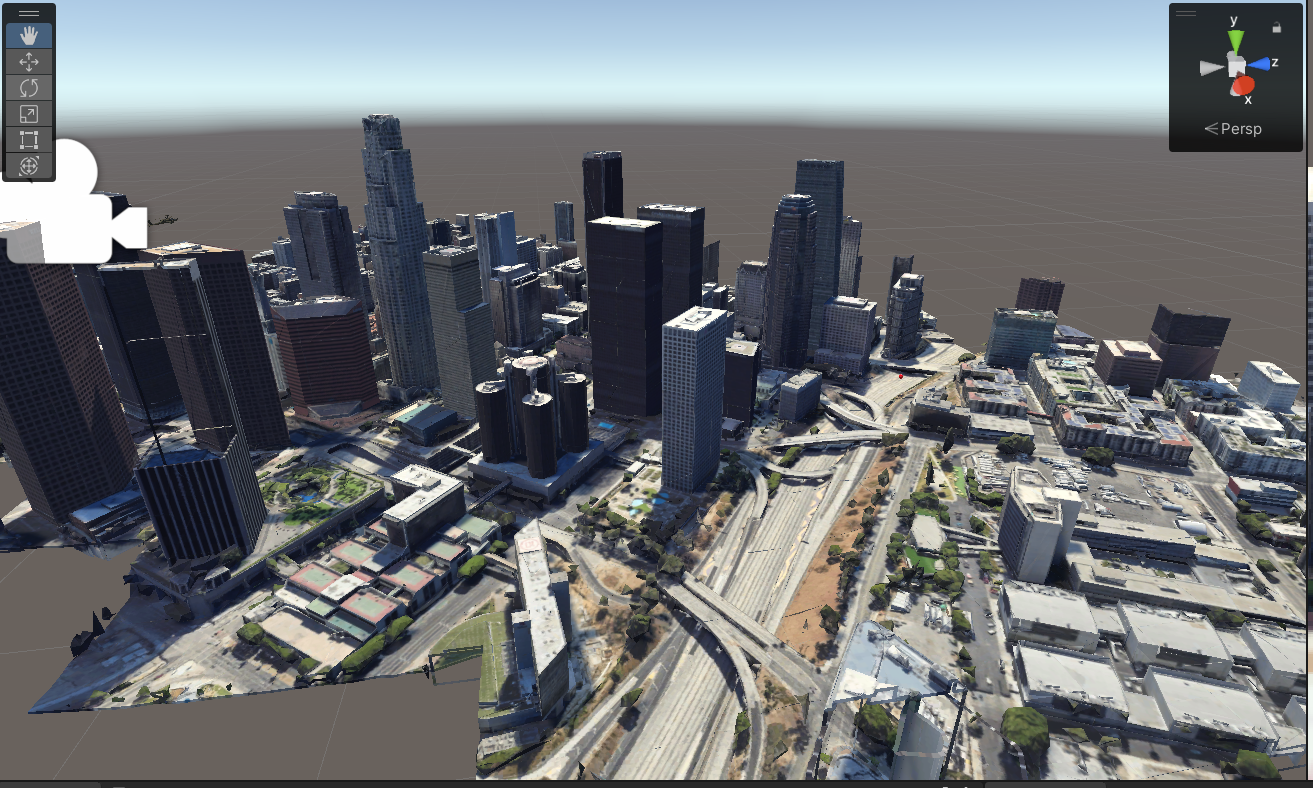
Xây dựng phối cảnh không gian 3D: Tải về bản đồ số 3D của một khu vực mở (khu đô thị, khu rừng, bãi chiến trường...) thể hiện rõ địa hình địa vật (vách núi, tòa nhà, đường sá...) sử dụng OSM hoặc Google Maps. Có thể trang trí cho các chi tiết của bản đồ thêm sống động và/hoặc bổ sung thêm các đối tượng, hình ảnh giả tưởng.

1. Thực hiện.

Từng bước thực hiện yêu cầu:

* Tải xuống và cài đặt JOSM (Java OpenStreetMap Editor) trên máy tính của bạn. ( Về Java OpenStreetMap: JOSM là trình chỉnh sửa mã nguồn mở, miễn phí dành cho dữ liệu OSM.)
* Khởi chạy JOSM và chọn tùy chọn "Tải xuống đối tượng" từ menu Tệp. Nhập tọa độ cho khu vực bạn muốn tải xuống hoặc sử dụng bản đồ tích hợp để chọn khu vực.
* Nhấp vào nút "Tải xuống" để bắt đầu quá trình tải xuống. Dữ liệu sẽ được lưu trữ ở định dạng dữ liệu nội bộ của JOSM.
* Sau khi dữ liệu được tải xuống, bạn cần xuất dữ liệu đó sang định dạng mà Unity có thể hiểu được. Để thực hiện việc này, hãy chọn "Xuất" từ menu Tệp, sau đó chọn "Xuất dưới dạng .obj".
* Đặt tên cho bản xuất của bạn, chọn vị trí lưu và nhấp vào "Lưu". Dữ liệu đã xuất sẽ được lưu trữ ở định dạng tệp .obj.
* Bắt đầu Unity và tạo một dự án mới.
* Nhập tệp .obj bạn vừa xuất vào Unity bằng cách đi tới Nội dung -> Nhập gói -> Gói tùy chỉnh, sau đó chọn tệp .obj bạn đã lưu trước đó.
* Trong Unity, chọn tệp .obj đã nhập trong bảng Nội dung và kéo tệp đó vào bảng Cảnh. Bản đồ sẽ được hiển thị trong chế độ xem Cảnh.
* Để tùy chỉnh giao diện của bản đồ, bạn có thể thêm vật liệu, họa tiết và ánh sáng cho bản đồ.
* Cuối cùng, bạn có thể xây dựng và chạy dự án của mình để xem bản đồ đang hoạt động.

Kết quả:



*Hình 4: Thành phố được xuất ra từ OSM.*

1. 3.
2. Yêu cầu.

Sử dụng các công cụ lập trình nội dung số như Blender, Unity, Google, Adobe, X3D, Maya, OpenCV, v.v. 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.

1. Thực hiện.

Để nhúng đối tượng 3D đã tạo (asset) vào phối cảnh bên trên:

* Trong bảng Nội dung, nhấp chuột phải và chọn "Nhập gói" hoặc chọn "Nội dung" từ menu trên cùng rồi chọn "Nhập gói".
* Chọn gói muốn nhập từ trình khám phá tệp xuất hiện.
* Trong cửa sổ Nhập gói Unity, chọn nội dung bạn muốn nhập rồi nhấp vào "Nhập".
* Giờ đây, đối tượng sẽ được thêm vào bảng Nội dung và ta có thể sử dụng chúng trong dự án của mình.



*Hình 5: Đối tượng trực thăng xuất hiện trong phối cảnh thành phố.*

1. 4B.
2. Yêu cầu.

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 tiếng nói hoặc cử chỉ bên cạnh chuột và bàn phím, và trong quá trình chuyển động, đối tượng cho phép người dùng tương tác. 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: Đối tượng 3D biết tự né các chướng ngại vật.

1. Thực hiện.

- Tạo một dự án Unity mới hoặc mở một dự án hiện có.

- Trong Unity Editor, chọn GameObject mà bạn muốn áp dụng vật lý. Bạn có thể tạo một GameObject mới từ bảng phân cấp hoặc chọn một GameObject hiện có.

- Trong bảng Thanh tra, thêm thành phần Rigidbody vào GameObject. Để thực hiện việc này, hãy nhấp vào nút "Thêm thành phần", chọn "Vật lý" rồi chọn "Thân cứng".

- Định cấu hình các thuộc tính Rigidbody để phù hợp với hành vi vật lý mong muốn. Có thể đặt khối lượng của đối tượng, xác định xem nó có bị ảnh hưởng bởi trọng lực hay không và chỉ định lực kéo và lực kéo góc của đối tượng.

- Nếu ta muốn áp dụng một lực hoặc xung lực cụ thể cho đối tượng, ta có thể thêm một tập lệnh vào GameObject để tính toán và áp dụng lực đó. (Ở đây, ta sẽ dùng kịch bản được xây dựng trong yêu cầu 1)

- Nếu ta muốn đối tượng va chạm với các đối tượng khác trong cảnh, có thể thêm thành phần Máy va chạm vào GameObject. Để thực hiện việc này, nhấp vào nút "Thêm thành phần", chọn "Vật lý" rồi chọn "Máy va chạm hộp", "Máy va chạm hình cầu", "Máy va chạm viên nang" hoặc bất kỳ loại máy va chạm nào khác phù hợp với nhu cầu.

- Định cấu hình các thuộc tính của Máy va chạm để phù hợp với hành vi va chạm mong muốn. Có thể đặt kích thước và hình dạng của máy va chạm, xác định xem nó có phải là trình kích hoạt hay không và chỉ định vật liệu đại diện cho các thuộc tính vật lý của đối tượng.

- Khi Ta đã thêm Rigidbody và Collider vào GameObject của mình, ta có thể chạy cảnh của mình và quan sát hoạt động mô phỏng vật lý.



*Hình 6: Đối tượng tương tác vật lý với phối cảnh.*

# Tham chiếu

1. <https://learn.unity.com/tutorial/intro-to-the-unity-physics-engine-2019-3?signup=true#5f7cefdbedbc2a00243033f0>
2. <https://digitalopus.ca/site/bullet-physics-in-unity-3d/>
3. <https://game-ace.com/blog/3d-modeling-in-unity/>
4. <https://learnosm.org/en/beginner/start-osm/>
5. <http://barankahyaoglu.com/dev/openstreetmap-in-unity3d/>