

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

Tel. (+84.0236) 3736949, Fax. (84-511) 3842771
Website: <http://dut.udn.vn/khoacntt>, E-mail: cntt@dut.udn.vn



BÁO CÁO MÔN HỌC
MÔ HÌNH HOÁ HÌNH HỌC

ĐỀ TÀI :

**TRIỂN KHAI MÔ HÌNH 3D DỰA TRÊN NỀN TẢNG
WEBGL VÀ THREEJS TRÊN LOCAL WEB SERVER
(ĐỀ 1)**

HỌ TÊN SINH VIÊN	MÃ SINH VIÊN	NHÓM HP
NGUYỄN CÔNG CƯỜNG	102200013	20Nh10

CBHD : PGS.TS. Nguyễn Tấn Khôi

Đà Nẵng, 12/05/2024

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	4
1.1. CƠ SỞ LÝ THUYẾT	4
1.1.1. WebGL (Web Graphics Library)	4
1.1.2. Three.js	5
1.2. PHÁT BIỂU BÀI TOÁN	7
1.3. KẾT CHƯƠNG	7
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	8
2.1. PHÁT BIỂU BÀI TOÁN	8
2.2. PHÂN TÍCH HIỆN TRẠNG	8
2.3. PHÂN TÍCH CHỨC NĂNG	9
2.3.1. Đối tượng sử dụng	9
2.3.2. Các chức năng chính	9
2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU	9
2.5. TỔ CHỨC CHƯƠNG TRÌNH	10
2.5.1. Tổ chức thư mục	10
2.5.2. Tập tin index.html	10
2.5.3. Tập tin school.js	10
2.5.4. Tập tin webgl_animation_skinning_ik	10
2.6. Deploy ứng dụng	10
2.7. KẾT CHƯƠNG	10
CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	11
3.1. MÔ HÌNH TRIỂN KHAI	11
3.1.1. Mô hình triển khai	11
3.1.2. Các công cụ sử dụng	11
3.1.3. Cấu hình hệ thống	15
3.2. KẾT QUẢ THỰC NGHIỆM	16
3.2.1. Kịch bản 1 – Khởi động hệ thống	16
3.2.2. Kịch bản 2 – Chức năng hiển thị hoặc ẩn thanh công cụ	17
3.2.3. Kịch bản 3 – Lựa chọn dữ liệu để hiển thị theo kịch bản	17
3.2.4. Kịch bản 4 – Di chuyển đến nhiều địa điểm trong khu vực	18
3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ	19
3.4. KẾT CHƯƠNG	19

DANH SÁCH HÌNH ẢNH

Hình 1: Tổ chức thư mục

Hình 1.1: Deploy ứng dụng trên github pages

Hình 1.2: App Go Street View Photo Sphere trên CH-Play

Hình 1.3: Ảnh 360 độ được chụp từ ứng dụng

Hình 1.4: App Street View Download 360 tải về cho PC

Hình 1.5: Web chuyển đổi ảnh 360 độ panorama to cube

Hình 1.6: Kết quả web “Panorama to Cubemap” trả về

Hình 2: Khởi động hệ thống

Hình 3: Hiển thị hoặc ẩn thanh công cụ

Hình 4: Lựa chọn dữ liệu để hiện thị theo kịch bản

Hình 5: Di chuyển đến nhiều địa điểm trong khu vực

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
WebGL	Web Graphics Library
API	Application Programming Interface
URL	Uniform Resource Locator
IP	Internet Protocol
HTTP	HyperText Transfer Protocol

LỜI NÓI ĐẦU

Việc triển khai các ứng dụng 3D trên web đang trở thành một xu hướng quan trọng trong ngành công nghiệp phần mềm. Với sự phát triển của công nghệ web và nhu cầu ngày càng tăng về trải nghiệm người dùng, ứng dụng web 3D mang lại những trải nghiệm tương tác và trực quan đáng kinh ngạc.

Đề tài này tập trung vào việc triển khai các mô hình 3D sử dụng WebGL và ThreeJS trên Local Web Server, với mục tiêu tạo ra một ứng dụng web 3D động. Bằng cách sử dụng các ví dụ cụ thể từ ThreeJS và thực hiện các bổ sung và thay đổi, đề tài mong muốn đem lại cái nhìn sâu sắc về khả năng của công nghệ này trong việc xây dựng các ứng dụng web 3D phong phú và đa dạng.

Đây là đề tài phục vụ cho môn học “Mô hình hoá hình học”. Do đó mục đích chính của đề tài là giúp sinh viên thực hành, tìm hiểu, áp dụng cách tạo các đối tượng 3D trên web. Từ đó, sinh viên phát triển khả năng sáng tạo, tư duy, nghiên cứu các ứng dụng của 3D vào trong các dự án của mình.

Lời cuối, em xin cảm ơn thầy Nguyễn Tấn Khôi đã chia sẻ cho chúng em những tài liệu liên quan đến đề tài, cùng như những kiến thức được thầy giảng dạy giúp chúng em dễ dàng tiếp cận với đề tài và hoàn thiện hơn đề tài của mình.

MỞ ĐẦU

1. Tổng quan về đề tài

Hiện nay, công nghệ 3D đang ngày càng phát triển và được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như đồ họa máy tính, game, mô phỏng, kiến trúc, y học,... Trong đó, WebGL và Three.js là hai công nghệ tiên tiến cho phép hiển thị đồ họa 3D trực tiếp trên trình duyệt web mà không cần cài đặt bất kỳ phần mềm hỗ trợ nào. Đây là một bước tiến lớn trong việc đơn giản hóa quá trình phát triển và triển khai ứng dụng 3D trên nền web.

Tuy nhiên, việc triển khai các mô hình 3D phức tạp trên nền web vẫn còn gặp một số thách thức như hiệu suất, tối ưu hóa, tương tác người dùng, và khả năng mở rộng. Đề tài này tập trung vào việc giải quyết những vấn đề này bằng cách nghiên cứu và phát triển một hệ thống cho phép triển khai mô hình 3D phức tạp trên nền tảng web sử dụng WebGL và Three.js.

2. Mục đích và ý nghĩa của đề tài

2.1. Mục đích

Mục đích của đề tài là:

- Tạo ra một ứng dụng web 3D động trên nền tảng WebGL và ThreeJS, có khả năng chạy trên Local Web Server.
- Phân tích và hiểu rõ cấu trúc, chức năng của ví dụ cụ thể từ ThreeJS, đặc biệt là ví dụ Đề 1.
- Mở rộng và bổ sung các chức năng của ứng dụng để đáp ứng các yêu cầu cụ thể của đề tài, bao gồm việc thay thế hình ảnh và tạo bộ dữ liệu mới.
- Nghiên cứu và đánh giá hiệu suất, khả năng mở rộng và sử dụng công cụ WebGL và ThreeJS trong việc xây dựng ứng dụng web 3D.

2.2. Ý nghĩa

Đề tài mang lại ý nghĩa sau:

- Nâng cao hiểu biết về công nghệ web 3D: Thực hiện đề tài giúp tăng cường kiến thức và kỹ năng về WebGL và ThreeJS, hai công nghệ quan trọng trong việc phát triển ứng dụng web 3D.
- Ứng dụng thực tiễn trong giáo dục và công nghiệp: Việc xây dựng và triển khai một ứng dụng web 3D có thể áp dụng rộng rãi trong giáo dục, quảng cáo, thương mại điện tử và nhiều lĩnh vực khác.
- Nghiên cứu và đánh giá công nghệ: Đề tài cung cấp cơ hội để nghiên cứu và đánh giá các công nghệ mới và phát triển trong lĩnh vực web 3D, từ đó đưa ra những nhận xét và đề xuất cải tiến.
- Tạo ra sản phẩm thực tế: Xây dựng một ứng dụng web 3D có thể được coi là một sản phẩm thực tế, từ đó thể hiện khả năng thiết kế và phát triển ứng dụng của nhóm nghiên cứu.

3. Phương pháp thực hiện

Thu thập dữ liệu từ nhiều nguồn khác nhau cho không gian 3D từ ảnh 360 độ. Các nguồn thu thập gồm, google map, tự chụp ảnh 360 bằng ứng dụng “Google Street View”, nguồn ảnh từ internet khác.

Sau đó sử dụng “Panorama to Cubemap” (tại đường link: <https://jaxry.github.io/panorama-to-cubemap/>) để thực hiện chia ảnh 360 độ thành các mặt của hình lập phương không gian 3D.

Sử dụng mã nguồn mở của ThreeJS để làm ví dụ thực hiện phân tích các thành phần của ThreeJS, hiểu cách tổ chức đối tượng và cách thức nó hoạt động, từ đó hiểu và làm việc với ThreeJS dễ dàng hơn. Tiến hành đưa ảnh cube vào ThreeJS để xây dựng hình ảnh 360 độ cùng các chức năng khác.

Đánh giá và cải thiện trải nghiệm người dùng thông qua các kỹ thuật tương tác, giao diện người dùng phù hợp. Kết hợp với kiểm thử và đánh giá hiệu suất của hệ thống trên các thiết bị và môi trường khác nhau.

3. Bố cục của đề án

Đề án bao gồm các nội dung sau:

Mở đầu

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích thiết kế hệ thống

Chương 3: Triển khai và đánh giá kết quả

Kết luận và hướng phát triển.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. CƠ SỞ LÝ THUYẾT

1.1.1. WebGL (Web Graphics Library)

WebGL (Web Graphics Library) là một tiêu chuẩn mã nguồn mở cho phép các trình duyệt web hiển thị đồ họa 3D tương tác trực tiếp trên trang web, mà không cần phải cài đặt plugin hay phần mềm hỗ trợ nào. WebGL được phát triển bởi Khronos Group và dựa trên OpenGL ES (OpenGL for Embedded Systems), cung cấp một API lập trình đồ họa 3D ở cấp thấp cho các ứng dụng web.

Với WebGL, các nhà phát triển web có thể tạo ra các ứng dụng đồ họa 3D tương tác cao, như trò chơi, ứng dụng thực tế ảo (VR), mô phỏng khoa học, và các ứng dụng đa phương tiện khác trên nền web. Điều này mở ra một không gian mới cho các ứng dụng web, vượt ra ngoài các tính năng truyền thống của HTML, CSS và JavaScript.

WebGL hoạt động bằng cách sử dụng bộ xử lý đồ họa (GPU) của máy tính hoặc thiết bị di động để thực hiện các tính toán và vẽ đồ họa 3D. Khi một ứng dụng WebGL được chạy, nó gửi các lệnh vẽ lên GPU thông qua API WebGL. GPU sau đó thực hiện các tính toán phức tạp để tạo ra hình ảnh 3D và gửi kết quả trở lại cho trình duyệt.

WebGL sử dụng ngôn ngữ shader để mô tả cách xử lý và vẽ dữ liệu đồ họa. Có hai loại shader chính: vertex shader (xử lý vị trí và hình dạng của các đối tượng) và fragment shader (xử lý màu sắc và texture của các đối tượng).

Rendering pipeline trong WebGL bao gồm các bước chính sau:

- Chuẩn bị dữ liệu: Dữ liệu đồ họa (geometry, texture, ...) được đưa vào GPU thông qua bộ đệm (buffer).
- Xử lý vertex: Vertex shader xác định vị trí cuối cùng của từng đỉnh (vertex) trên màn hình.
- Tính toán ảnh chiếu (rasterization): Các đỉnh được biến đổi thành các điểm ảnh (pixel) trên màn hình.
- Xử lý pixel: Fragment shader xác định màu sắc và độ sáng của từng pixel.
- Hiển thị ảnh cuối cùng lên màn hình.

WebGL hoạt động như một phần mềm con của trình duyệt web, được tích hợp với JavaScript thông qua các đối tượng và phương thức trong API WebGL. Để sử dụng WebGL, trình duyệt phải tạo một context WebGL liên kết với một phần tử canvas HTML. Sau đó, các lệnh WebGL có thể được gửi đến context này để vẽ đồ họa 3D lên canvas.

Đồng thời, JavaScript cũng đóng vai trò quan trọng trong việc quản lý dữ liệu, xử lý sự kiện và tương tác với các đối tượng 3D trong ứng dụng WebGL. Điều này cho phép WebGL được tích hợp hoàn toàn với các tính năng web khác và tạo ra các ứng dụng đồ họa 3D tương tác phong phú.

1.1.2. Three.js

ThreeJS là một thư viện JavaScript mã nguồn mở mạnh mẽ, được sử dụng để tạo và hiển thị đối tượng 3D trực tiếp trong trình duyệt web mà không cần sự hỗ trợ của bất kỳ plugin nào khác.

Các khái niệm cơ bản:

- **Render Loop:** ThreeJS sử dụng một vòng lặp render để liên tục cập nhật và vẽ các đối tượng 3D lên màn hình. Điều này đảm bảo rằng các đối tượng được hiển thị một cách liên tục và mượt mà trên trình duyệt web.
- **Scene (Cảnh):** Là không gian 3D chứa tất cả các đối tượng và ánh sáng. Ba thành phần chính của một cảnh là camera, ánh sáng và các đối tượng.
- **Camera (Máy ảnh):** Xác định góc nhìn và phạm vi của người dùng. ThreeJS cung cấp các loại camera khác nhau như `PerspectiveCamera` và `OrthographicCamera`.
- **Geometry (Hình học):** Định nghĩa hình dạng của đối tượng. ThreeJS cung cấp một loạt các hình học tiêu chuẩn như `SphereGeometry`, `BoxGeometry`, và `PlaneGeometry`.
- **Material (Vật liệu):** Xác định cách mà một đối tượng được vẽ trên màn hình. ThreeJS hỗ trợ nhiều loại vật liệu như `MeshBasicMaterial`, `MeshPhongMaterial`, và `MeshStandardMaterial`.

- **Light (Ánh sáng):** ThreeJS hỗ trợ các loại ánh sáng khác nhau như AmbientLight, DirectionalLight, PointLight, và SpotLight để tạo ra các hiệu ứng ánh sáng phức tạp và chân thực.
- **Texture (Chất liệu):** Cho phép sử dụng hình ảnh hoặc video làm texture cho các đối tượng, tạo ra các hiệu ứng phong phú và đa dạng.
- **Animation (Hoạt hình):** ThreeJS hỗ trợ việc tạo và điều khiển các hoạt hình 3D, bao gồm chuyển động, quay, co giãn, và hiệu ứng khác.

Áp dụng vào dự án:

- Sử dụng thư viện ThreeJS để tạo ra cảnh 3D, được import từ các module cụ thể như three, OrbitControls, LightProbeHelper, và LightProbeGenerator.
- Khởi tạo Renderer, Scene và Camera:
 - Renderer được tạo để vẽ cảnh 3D lên màn hình.
 - Scene là không gian chứa tất cả các đối tượng và ánh sáng.
 - Camera xác định góc nhìn và phạm vi của người dùng trong cảnh 3D.
- Sử dụng CubeCamera và LightProbe:
 - CubeCamera được sử dụng để tạo các hình ảnh của môi trường xung quanh, còn được gọi là cube map, được sử dụng làm ánh sáng môi trường cho cảnh.
 - LightProbe được sử dụng để lưu trữ thông tin ánh sáng môi trường từ cube map, giúp tạo ra hiệu ứng chiếu sáng phản ánh trên các đối tượng 3D.

1.2. PHÁT BIỂU BÀI TOÁN

Mục tiêu và phạm vi của bài toán như sau:

- Mục tiêu: Xây dựng một ứng dụng web 3D động sử dụng ThreeJS, triển khai trên Local Web Server, dựa trên ví dụ Đề 1 từ ThreeJS examples.
- Phạm vi: Nghiên cứu, phân tích, triển khai và đánh giá hiệu suất của ứng dụng, bao gồm các yêu cầu xây dựng bộ dữ liệu của mỗi SV; thay thế các hình ảnh trong ví dụ thành các hình ảnh của Trường ĐHBK/ TP Đà Nẵng,... do SV tạo ra; bổ sung chức năng lựa chọn các bộ dữ liệu để hiển thị theo kịch bản 1, 2, 3...; g) Bổ sung chức năng cho phép di chuyển đến nhiều địa điểm trong một khu vực.

1.3. KẾT CHƯƠng

Chương này đã trình bày các khái niệm cơ sở về WebGL, Three.js, mô hình 3D và các kỹ thuật tối ưu hóa hiệu suất liên quan. Ngoài ra, bài toán triển khai mô hình 3D phức tạp trên nền web cũng đã được phát biểu rõ ràng. Những nội dung này sẽ là cơ sở lý thuyết quan trọng cho việc phân tích, thiết kế và triển khai hệ thống trong các chương tiếp theo.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. PHÁT BIỂU BÀI TOÁN

Bài toán đặt ra là xây dựng một hệ thống cho phép triển khai mô hình 3D trên nền tảng local web, sử dụng WebGL và Three.js dựa trên một mẫu example `webgl_animation_skinning_ik` của thư viện Threejs

2.2. PHÂN TÍCH HIỆN TRẠNG

Hiện trạng đề cập đến việc thực hiện các ví dụ đã được cung cấp trên trang chủ của ThreeJS, bao gồm:

- Triển khai chạy trên local Web Server: Sử dụng Node.js hoặc một máy chủ web local khác để chạy các ví dụ.
- Phân tích các chức năng của hệ thống: Đánh giá chức năng hiển thị mô hình 3D, tương tác với người dùng.
- Phân tích các module của hệ thống: Nắm vững cấu trúc và chức năng của các module trong ví dụ được triển khai, bao gồm WebGL, ThreeJS, và các thư viện hỗ trợ khác.
- Xây dựng bộ dữ liệu của mỗi SV: Chuẩn bị dữ liệu ảnh và các tài nguyên khác để thay thế vào ví dụ.
- Thay thế các hình ảnh trong ví dụ: Tạo ra hình ảnh mới phù hợp với mục đích của đề tài.
- Bổ sung chức năng lựa chọn bộ dữ liệu: Tạo chức năng cho phép người dùng lựa chọn bộ dữ liệu hiển thị.
- Bổ sung chức năng di chuyển đến nhiều địa điểm: Mở rộng tính năng để người dùng có thể di chuyển giữa các vị trí trên mô hình 3D.

2.3. PHÂN TÍCH CHỨC NĂNG

2.3.1. Đối tượng sử dụng

- Sinh viên năm nhất đại học bách khoa muốn tìm hiểu về trường mình
- Cựu sinh viên muốn xem lại kỷ niệm về trường
- Các thầy cô hoặc sinh viên khác trong hoặc ngoài trường
- Những người cần biết thông tin về các khu vực trong trường

2.3.2. Các chức năng chính

- Thay thế những file .png, .jpg,... trong ví dụ threejs bằng hình ảnh do sinh viên tạo ra, khi chạy server các hình ảnh của ví dụ threejs cũng phải thay tương tự
- Chức năng lựa chọn bộ dữ liệu để hiển thị theo kịch bản 1, 2, 3, ...
- Cho phép di chuyển nhiều địa điểm xung quanh trong 1 khu vực

2.4. CƠ SỞ DỮ LIỆU

- Cơ sở dữ liệu ảnh của các khu vực trường Đại học Bách Khoa - Đại học Đà Nẵng được lưu dưới dạng file .png và trong folder textures của project

2.5. TỔ CHỨC CHƯƠNG TRÌNH

2.5.1. Tổ chức thư mục

Name	Date modified	Type	Size
build	5/12/2024 7:48 AM	File folder	
jsm	5/12/2024 7:48 AM	File folder	
models	5/12/2024 7:48 AM	File folder	
textures	5/12/2024 7:48 AM	File folder	
fiveserver	5/8/2024 12:31 AM	JavaScript File	13 KB
index	5/12/2024 8:05 AM	CocCoc HTML Do...	4 KB
school	5/12/2024 6:10 AM	JavaScript File	5 KB
webgl_animation_skinning_ik	5/12/2024 6:43 AM	JavaScript File	9 KB

Hình 1: Tổ chức thư mục

2.5.2. Tập tin index.html

- Chứa mã html chính của dự án

2.5.3. Tập tin school.js

- Chứa mã javascript dùng để load mô hình 3D các khu vực của trường đại học

2.5.4. Tập tin webgl_animation_skinning_ik

- Chứa mã javascript dùng để load mô hình ví dụ 3D của threejs (đề 1)

2.6. Deploy ứng dụng

- Ứng dụng được deploy trên nền tảng github pages

2.7. KẾT CHƯƠNG

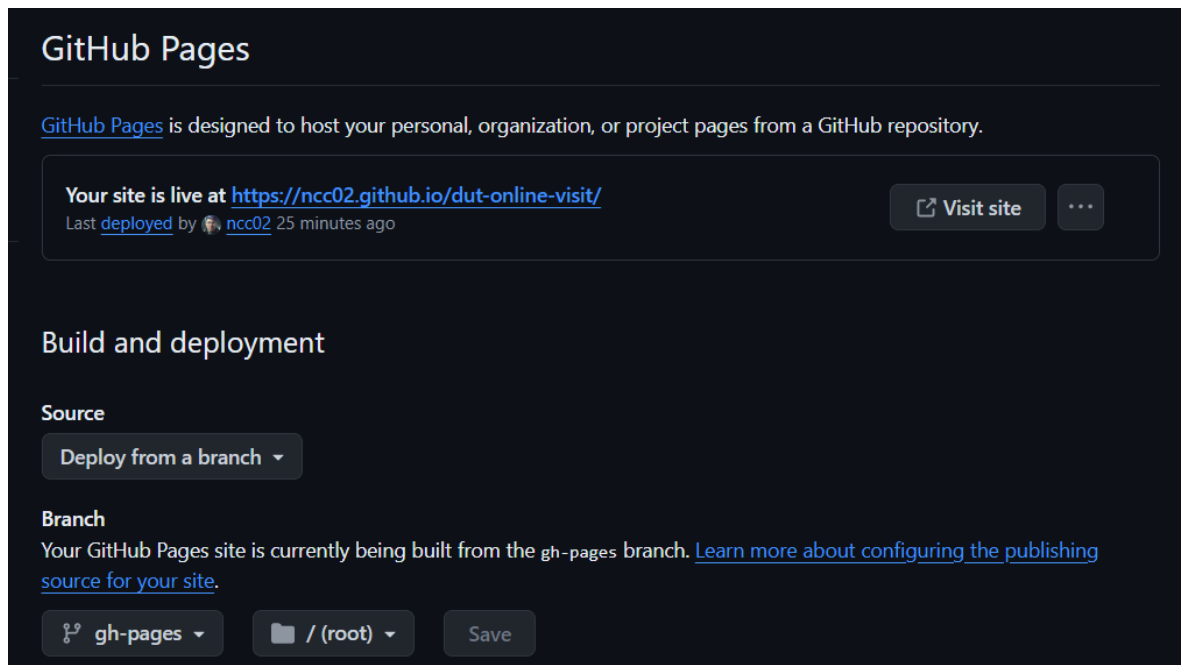
Chương này đã trình bày chi tiết về việc phân tích và thiết kế hệ thống cho bài toán triển khai mô hình 3D dựa trên nền tảng WebGL và Three.js trên local web. Những nội dung này sẽ là cơ sở quan trọng cho việc triển khai chi tiết hệ thống trong các chương tiếp theo. Các yêu cầu và thiết kế được đưa ra sẽ giúp đảm bảo hệ thống được xây dựng đáp ứng đúng mục tiêu đề ra và có khả năng mở rộng trong tương lai.

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. MÔ HÌNH TRIỂN KHAI

3.1.1. Mô hình triển khai

- Triển khai trên local web dựa trên nền tảng WebGL và Threejs, sử dụng Bootstrap để code giao diện frontend
- Triển khai deploy trên nền tảng web bằng github page, link tới ứng dụng <https://ncc02.github.io/dut-online-visit/>



Hình 1.1: Deploy ứng dụng trên github pages

3.1.2. Các công cụ sử dụng

- App “Go Street View Photo Sphere” trên di động để chụp ảnh 360 của các khu vực trường

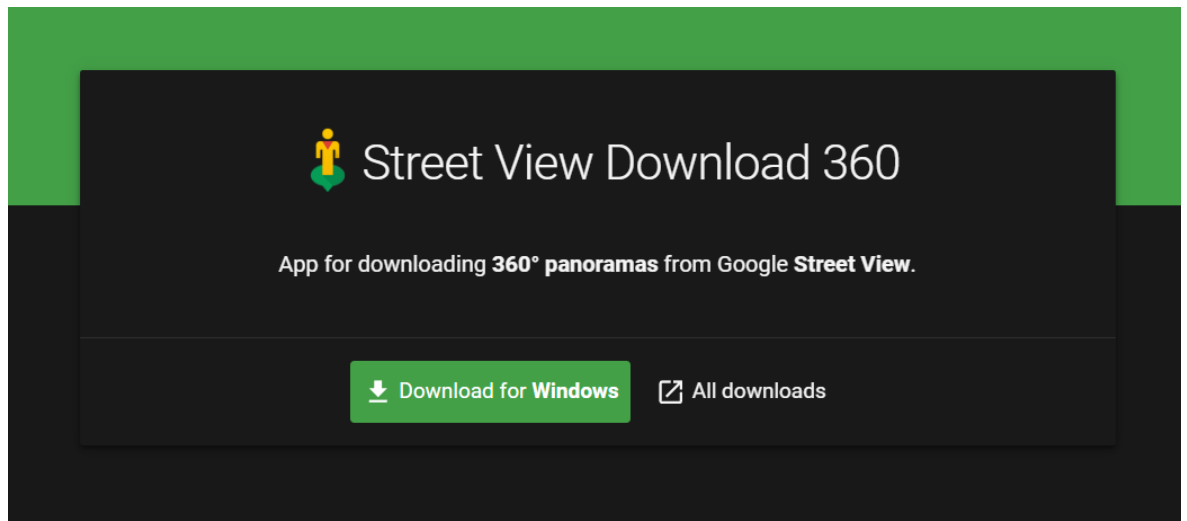


Hình 1.2: App Go Street View Photo Sphere trên CH-Play



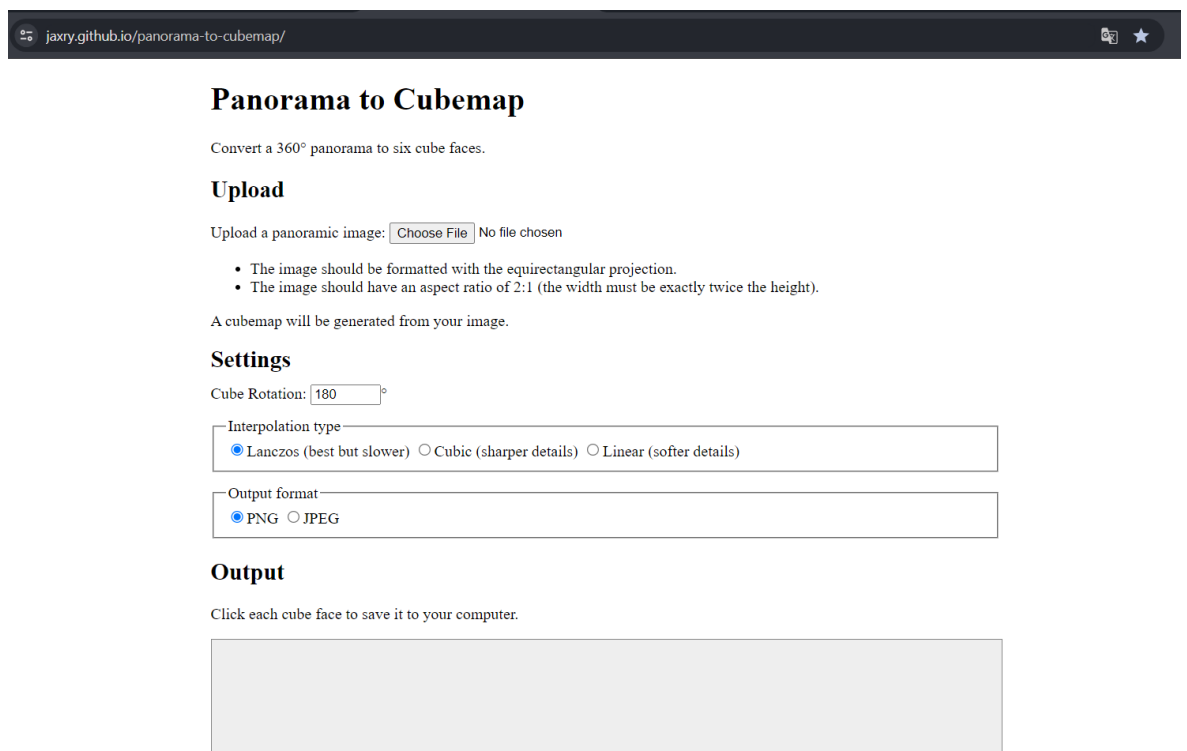
Hình 1.3: Ảnh 360 độ được chụp từ ứng dụng

- App Street View Download 360 để tải ảnh 360 độ từ google map 3D

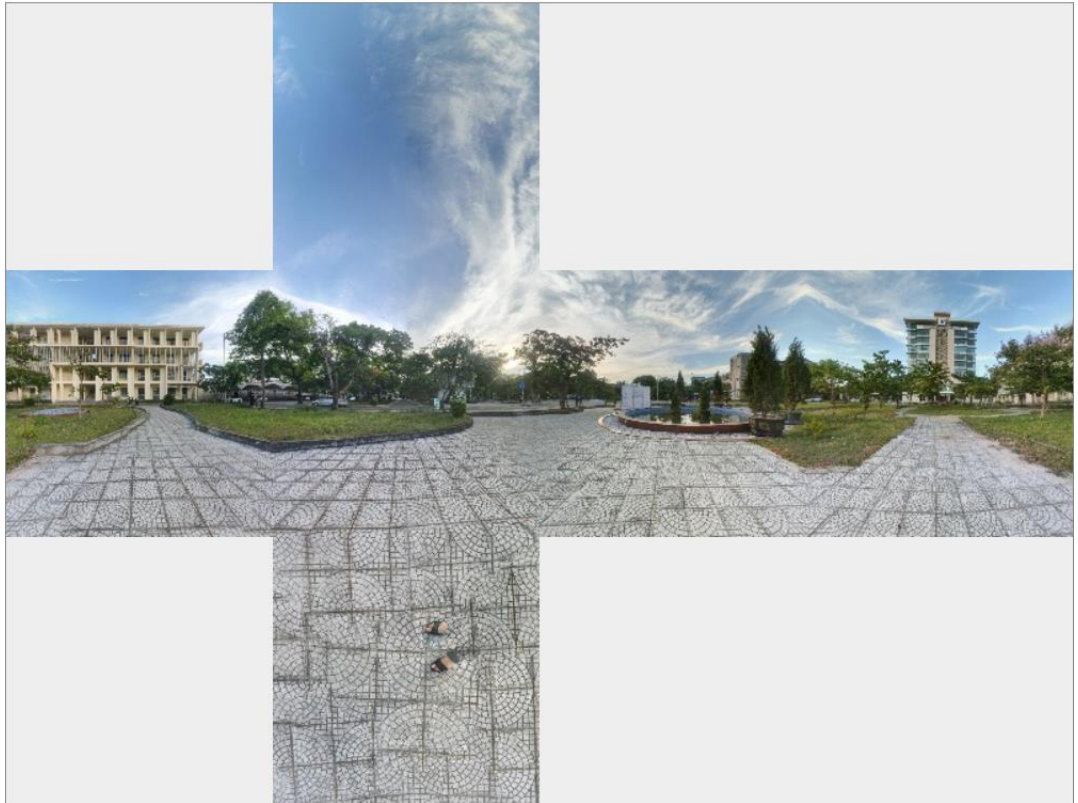


Hình 1.4: App Street View Download 360 tải về cho PC

- Web panoramas to images cube để chuyển đổi ảnh 360 độ

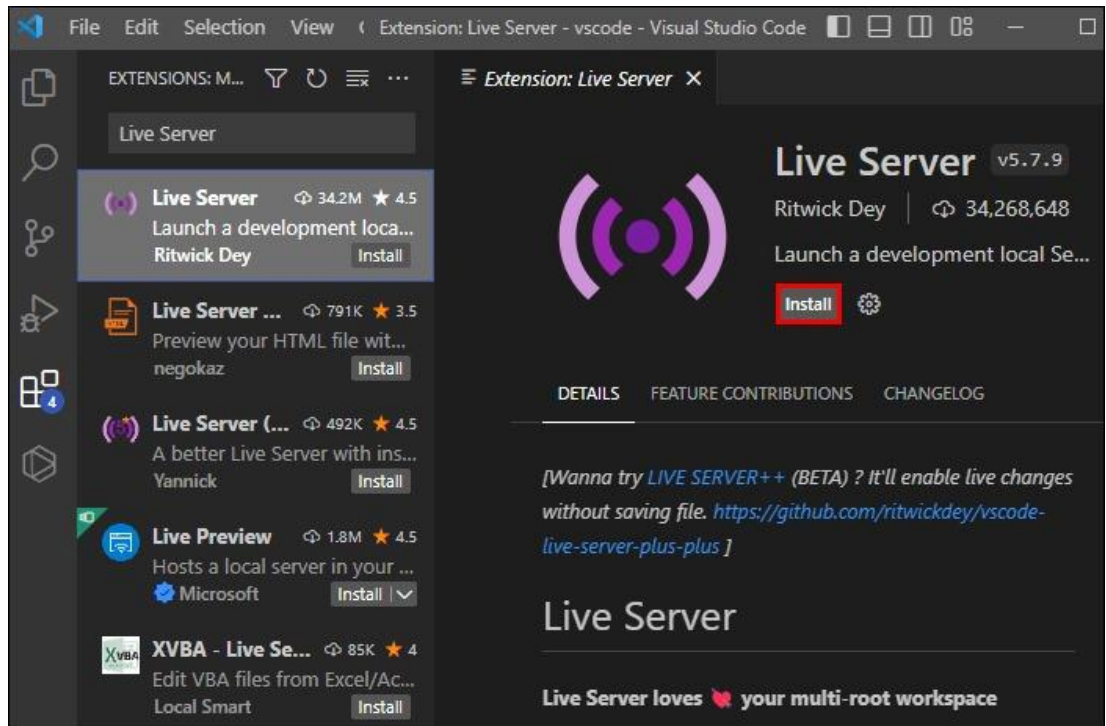


Hình 1.5: Web chuyển đổi ảnh 360 độ panorama to cube



Hình 1.6: Kết quả web “Panorama to Cubemap” trả về

- VSCode dùng để code và extension Liveserve để code và khởi chạy serve



Hình 1.7: Tiện ích LiveServer trong VSCode

- Trình duyệt web trên PC hoặc di động dùng để debug code và demo sản phẩm

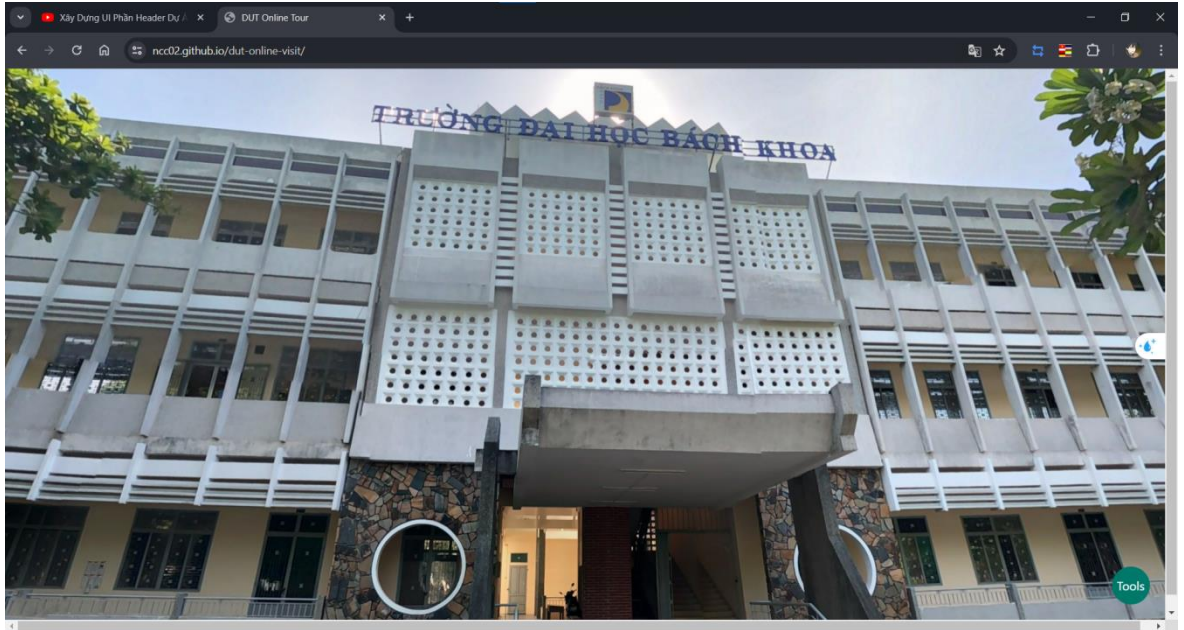
3.1.3. Cấu hình hệ thống

Cấu hình đề hệ thống đề xuất để chạy chương trình:

	Cấu hình đề nghị
CPU	AMD hoặc Intel \geq 2ghz
GPU	NVIDIA GeForce GTX 1050Ti hoặc cao hơn
RAM	8GB
Dung lượng ổ đĩa trống	Trống 1GB
Hệ điều hành	64-bit Windows 10 trở lên

3.2. KẾT QUẢ THỰC NGHIỆM

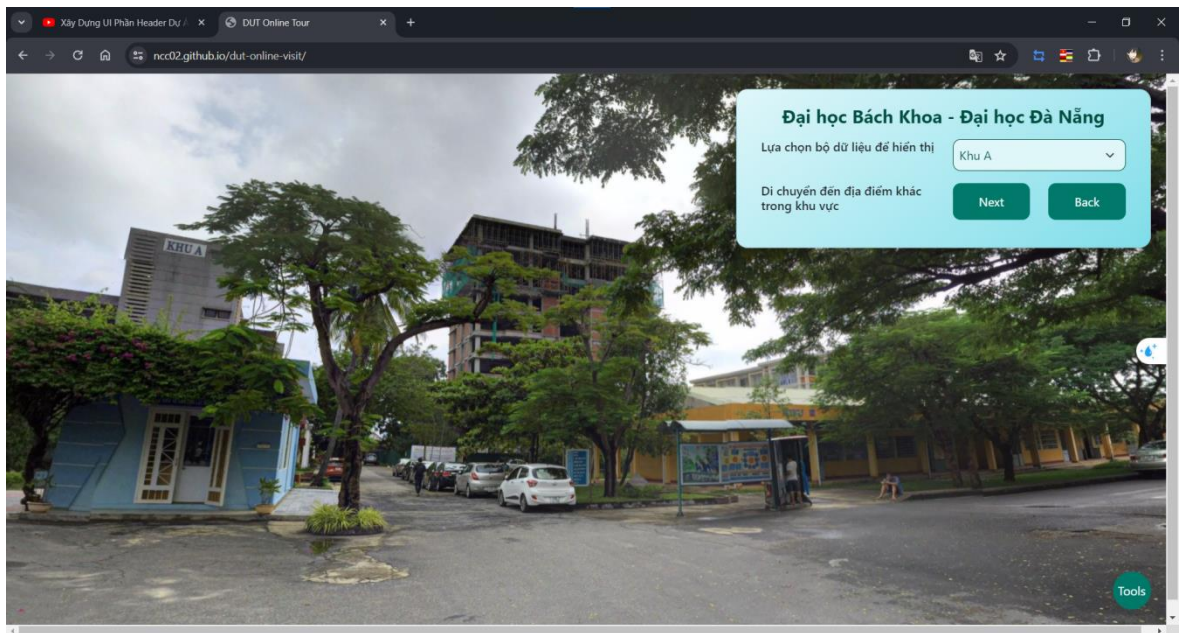
3.2.1. Kịch bản 1 – Khởi động hệ thống



Hình 2: Khởi động hệ thống

Khởi động ứng dụng bằng LiveServer trong VSCode, trình duyệt sẽ tự động mở tap của chương trình hoặc vào link <https://ncc02.github.io/dut-online-visit/> đã deploy trong ảnh.

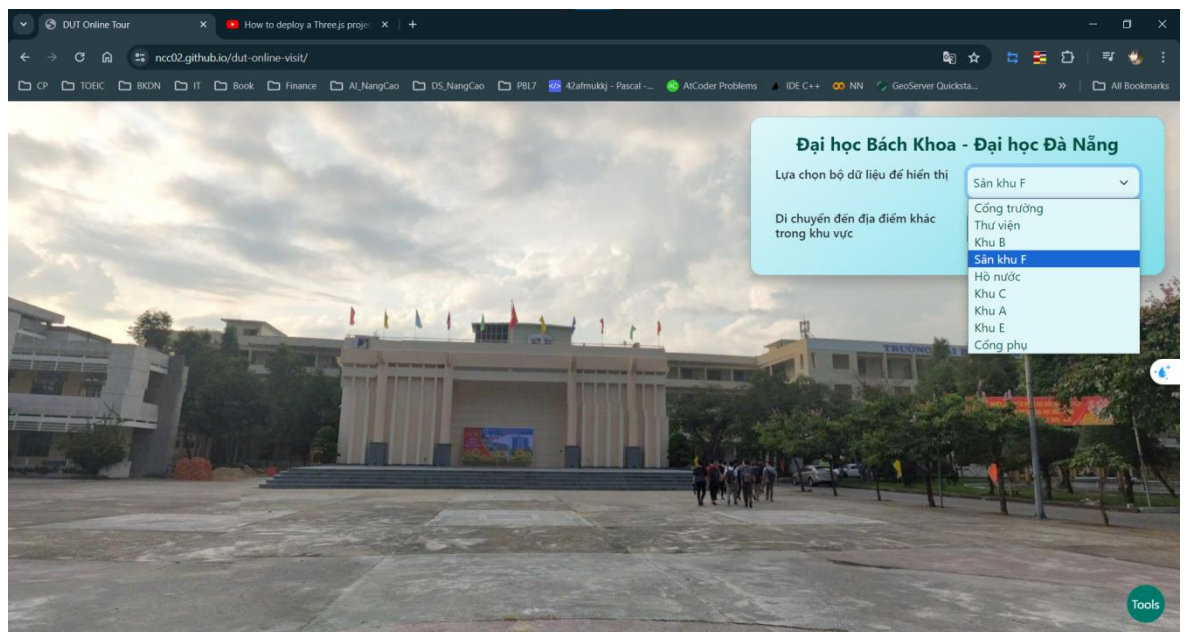
3.2.2. Kịch bản 2 – Chức năng hiển thị hoặc ẩn thanh công cụ



Hình 3: Hiển thị hoặc ẩn thanh công cụ

Khi click vào button tools góc dưới bên phải, hệ thống sẽ tự động hiển thị dữ liệu thanh công cụ nếu chưa có ngược lại thì ẩn đi.

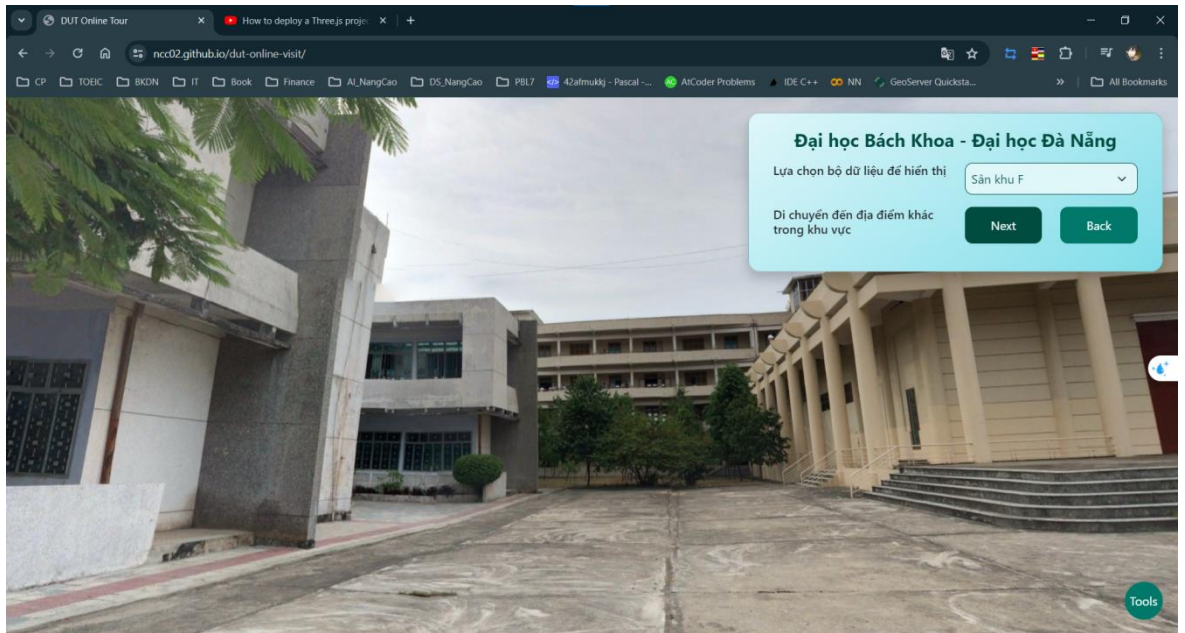
3.2.3. Kịch bản 3 – Lựa chọn dữ liệu để hiển thị theo kịch bản



Hình 4: Lựa chọn dữ liệu để hiển thị theo kịch bản

Khi click vào dropdown hệ thống cho phép người dùng lựa chọn bộ dữ liệu để hiển thị, dữ liệu 3D sẽ tự động thay đổi và load lại trên trang web.

3.2.4. Kịch bản 4 – Di chuyển đến nhiều địa điểm trong khu vực



Hình 5: Di chuyển đến nhiều địa điểm trong khu vực

Khi click vào button next người dùng sẽ di chuyển đến địa điểm tiếp theo trong khu vực của bộ dữ liệu hiện tại, hoặc click vào button back thì người dùng sẽ trở về địa điểm lúc trước.

3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

Qua kết quả thực nghiệm, tác giả có những nhận xét như sau:

- Vấn đề dữ liệu: Hiện tại dữ liệu chỉ được thu nhập bằng cách sinh viên tự chụp ảnh hoặc lấy từ google map nên còn khá ít dữ liệu cho dự án.
- Vấn đề hiệu năng: Đối với một số máy tính yếu trang web có thể bị chập khi load hoặc đổi bộ dữ liệu của khu vực trường đại học.
- Vấn đề giao diện: Giao diện dễ tiếp cận đối với người mới tuy nhiên không hiện đại và bắt mắt.

3.4. KẾT CHƯƠng

Chương này trình bày về kết quả triển khai hệ thống, cấu hình hệ thống và các thành phần chức năng. Kết quả được đánh giá thông qua các kịch bản thực nghiệm khác nhau nhằm thể hiện các ưu/nhược của giải pháp đề xuất.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong quá trình thực hiện đồ án, những kết quả chính đã đạt được bao gồm:

Về mặt lý thuyết, đồ án đã tìm hiểu và trình bày cơ sở lý thuyết về WebGL, Three.js và các kỹ thuật liên quan đến việc triển khai mô hình 3D trên nền web. Các khái niệm, nguyên lý hoạt động và cấu trúc của WebGL, Three.js đã được phân tích và giải thích chi tiết.

Về mặt thực tiễn ứng dụng, chúng tôi đã phát triển một hệ thống web dựa trên ThreeJS cho phép người dùng khám phá hình ảnh 360 độ của các địa điểm thực tế.

Kết quả đóng góp của đồ án được thể hiện như sau:

- Tạo website hiển thị hình ảnh 360 độ, giúp người dùng có những trải nghiệm ở các góc nhìn khác nhau trong hệ thống.
- Xây dựng một giao diện dễ sử dụng và thân thiện với người dùng, từ minimap đến giao diện camera 360 độ.
- Tạo ra một cơ sở dữ liệu hình ảnh đa dạng, bao gồm các địa điểm thực tế và cảnh quan tự nhiên.

Tuy nhiên, trong quá trình triển khai, vẫn còn một số hạn chế như sau:

- Lượng dữ liệu mô hình 3D cho các khu vực của trường Đại học Bách Khoa - Đà Nẵng chưa đầy đủ, chỉ mới thu thập được một số khu vực.
- Hiệu năng của hệ thống có thể bị ảnh hưởng khi chạy trên các máy tính cấu hình thấp, đặc biệt là lúc load hoặc chuyển đổi bộ dữ liệu mô hình 3D mới.
- Giao diện người dùng vẫn còn đơn giản, chưa thực sự hiện đại và bắt mắt.

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Dựa trên những kết quả đạt được và hạn chế hiện tại, một số hướng phát triển tiếp theo cho đồ án bao gồm:

- Thu thập và tích hợp thêm nhiều bộ dữ liệu mô hình 3D cho các khu vực khác trong khuôn viên trường Đại học Bách Khoa - Đà Nẵng để tăng tính phủ sóng và đa dạng của ứng dụng.
- Đánh giá và kiểm thử hiệu năng của hệ thống trên nhiều môi trường và cấu hình khác nhau để có cái nhìn toàn diện hơn về khả năng hoạt động của ứng dụng.
- Cải thiện và làm đẹp giao diện người dùng để tăng trải nghiệm, đồng thời nghiên cứu thêm các tính năng tương tác nâng cao khác để mang lại sự thân thiện và thú vị hơn cho người dùng.
- Xem xét và áp dụng các biện pháp bảo mật, xác thực và kiểm soát quyền truy cập nếu triển khai ứng dụng trên môi trường web công cộng để đảm bảo an toàn cho hệ thống.

Với hướng phát triển đề xuất, đồ án có thể tiếp tục được mở rộng, hoàn thiện và ứng dụng một cách linh hoạt hơn trong lĩnh vực triển khai mô hình 3D trên nền web, đáp ứng tốt hơn nhu cầu của người dùng.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Làm quen với môi trường 3D của ThreeJS: <https://viblo.asia/p/threejs-bai-1-lam-quen-voi-moi-truong-3d-cua-threejs-vyDZO7ROZwj>
- [2] Tìm hiểu về WebGL phần 1: thư viện đồ họa web: <https://viblo.asia/p/tim-hieu-ve-webgl-phan-1-thu-vien-do-hoa-web-PwRkgmyVGEd>

Tiếng Anh

- [3] WebGL: 2D and 3D graphics for the web: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

Internet

- [4] <https://threejs.org/>
- [5] <https://getbootstrap.com/>