

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**



**PHÁT TRIỂN ỨNG DỤNG ĐỒ HOẠ**  
**MÁY TÍNH TRONG LĨNH VỰC Y TẾ**

**ĐỒ ÁN II**

**Chuyên ngành: TOÁN TIN**

**Giảng viên hướng dẫn: TS. Vương Mai Phương**

**Sinh viên thực hiện: Phạm Thu Trang**

**Mã số sinh viên: 20195931**

**HÀ NỘI – 2023**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

### 1. Mục tiêu và nội dung của đề án

(a)

(b)

(c)

### 2. Kết quả đạt được

(a)

(b)

(c)

### 3. Ý thức làm việc của sinh viên:

(a)

(b)

(c)

*Hà Nội, ngày 23 tháng 2 năm 2023*

Giảng viên hướng dẫn

# Lời cảm ơn

Em xin trân trọng cảm ơn TS. Vương Mai Phương đã giúp em hoàn thiện đồ án này.

*Hà Nội, tháng 2 năm 2023*

Tác giả đồ án

**Phạm Thu Trang**

# Tóm tắt nội dung Đề án

1. .
2. .
3. .

*Hà Nội, tháng 2 năm 2023*

Tác giả đề án

**Phạm Thu Trang**

# Mục lục

<b>Danh sách bảng</b>	<b>1</b>
<b>Danh sách hình vẽ</b>	<b>1</b>
<b>Chương 1 Giới thiệu về WebGL</b>	<b>4</b>
1.1 WebGL . . . . .	4
1.2 Cách hoạt động của WebGL . . . . .	5
1.3 Giới thiệu về HTML5 canvas . . . . .	5
1.4 Một số thành phần cơ bản của WebGL . . . . .	6
1.4.1 Hệ tọa độ . . . . .	6
1.4.2 Các nền tảng . . . . .	6
1.4.3 Lưới (Mesh) . . . . .	6
1.5 Graphics Pipeline . . . . .	6
<b>Chương 2 WebGL với Three.js</b>	<b>7</b>
2.1 Three.js . . . . .	7
2.2 Cấu trúc ứng dụng Three.js . . . . .	7
2.2.1 Các thành phần cần thiết . . . . .	7
2.2.2 Mô hình cấu trúc . . . . .	8
<b>Chương 3 Mô hình 3D hoặc cách tạo video</b>	<b>9</b>

<b>Chương 4 Xây dựng chương trình và kết quả</b>	<b>10</b>
4.1 Giới thiệu mô hình 3D . . . . .	10
4.2 Sử dụng texture cho mô hình 3D . . . . .	10
4.3 Tạo các chú thích trên mô hình 3D . . . . .	10
4.4 Thực hiện tạo video giới thiệu mô hình 3D . . . . .	10
<b>Tài liệu tham khảo</b>	<b>12</b>

# Danh sách bảng

# Danh sách hình vẽ

1.1	Vẽ lại . . . . .	4
1.2	Pipeline . . . . .	6
2.1	Cấu trúc ứng dụng Three.js . . . . .	8



# Mở đầu

Tên đề tài: "**Phát triển ứng dụng đồ họa máy tính trong lĩnh vực y tế**".  
(tạm thời)

Đề tài được xây dựng dựa trên :

- Ngôn ngữ lập trình JavaScript, HTML5, CSS.
- Thư viện lập trình 3D sử dụng WebGL là Three.js.

Báo cáo gồm bốn chương:

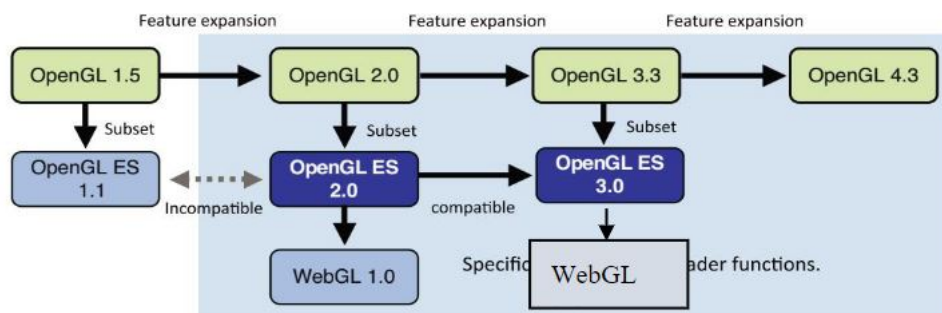
1. **Chương 1: Giới thiệu về WebGL.** Chương này giới thiệu cơ bản về WebGL.
2. **Chương 2: WebGL với Three.js.** Chương này giới thiệu cách tạo mô hình 3D với Three.js thông qua WebGL
3. **Chương 3: Mô hình 3D hoặc tạo video Cập nhật**
4. **Chương 4: Xây dựng chương trình và kết quả.** Chương này trình bày các bước xây dựng chương trình và kết quả thu được.

## Chương 1

# Giới thiệu về WebGL

### 1.1 WebGL

WebGL (Thư viện đồ họa web) là API rasterization dựa trên OpenGL ES - thư viện đồ họa 2D và 3D trên hệ thống nhúng: điện thoại, đồ điện tử, xe cơ giới,... được phát triển bởi Khronos Group, Inc. để hiển thị đồ họa 2D và 3D trong trình duyệt web. WebGL cung cấp các chức năng cơ bản tương tự OpenGL ES và hoạt động tốt trên các phân cứng đồ họa 3D hiện đại.



**Figure 1.4** Relationship among OpenGL, OpenGL ES 1.1/2.0/3.0, and WebGL

Hình 1.1: Vẽ lại

WebGL sử dụng khả năng tăng tốc phần cứng cho phép các nhà phát triển có khả năng xây dựng đồ họa tương tác hiển thị thời gian thực hiệu suất cao trong trình duyệt web, chẳng hạn như trò chơi, mô phỏng, trực quan hóa dữ liệu, video

hoạt hình, mô hình 3D, hoạt ảnh phân tử, v.v.

Hiệu quả của WebGL có lẽ là một trong những ưu điểm chính vì nó cho phép các ứng dụng web có yêu cầu cao khai thác toàn bộ khả năng của card đồ họa. Hơn nữa, WebGL không có plugin và tương thích với tất cả các trình duyệt chính, vì vậy người dùng không cần lo lắng về việc tải xuống hoặc cài đặt bất kỳ ứng dụng bên ngoài nào để chạy ứng dụng WebGL.

## 1.2 Cách hoạt động của WebGL

Một chương trình WebGL bao gồm các mã được viết bằng OpenGL Shading Language (GLSL), JavaScript và HTML5. GLSL được sử dụng để viết các trình đồ bóng và các chương trình đặc biệt được thực thi trên phần cứng đồ họa tính toán các thuộc tính như đỉnh, kết cấu, màu sắc, tia chớp, v.v. trong quá trình hiển thị đối tượng hoặc cảnh. GLSL cung cấp cho các nhà phát triển quyền kiểm soát quan trọng đối với đường ống đồ họa. JavaScript và HTML được sử dụng chủ yếu làm ngôn ngữ ràng buộc và để cung cấp ngữ cảnh hiển thị. Một chương trình WebGL thường được tạo bởi nhiều lần vẽ trên phần tử HTML canvas được thực hiện bởi đơn vị xử lý đồ họa (GPU) thông qua một quy trình được gọi là quy trình kết xuất. Để tạo đồ họa có độ trung thực cao, sử dụng WebGL có thể khá phức tạp vì nó gần như là chương trình cấp thấp.

Tuy nhiên, có nhiều thư viện sẵn có giúp trừu tượng hóa một số khó khăn của WebGL khiến chúng ít dài dòng hơn, chẳng hạn như twgl, Three.js, PhiloGL và J3D.

## 1.3 Giới thiệu về HTML5 canvas

Canvas là 1 phần tử của HTML5 sẽ được sử dụng làm trình kết xuất đồ họa cho ThreeJS. ... (cập nhật)

## 1.4 Một số thành phần cơ bản của WebGL

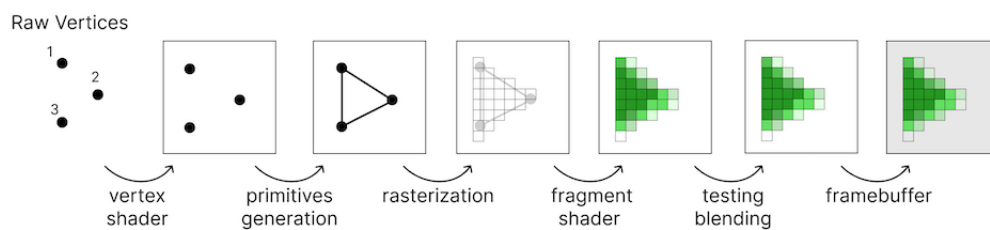
### 1.4.1 Hệ tọa độ

### 1.4.2 Các nền tảng

### 1.4.3 Lưới (Mesh)

## 1.5 Graphics Pipeline

Graphics Pipeline bao gồm một chuỗi các thao tác được yêu cầu mỗi khi một đối tượng (cảnh 2D/3D) được hiển thị trên màn hình.



Hình 1.2: Pipeline

Trong đó

1. **Vertices**

2. **Vertex Shader**

3. ...

## Chương 2

# WebGL với Three.js

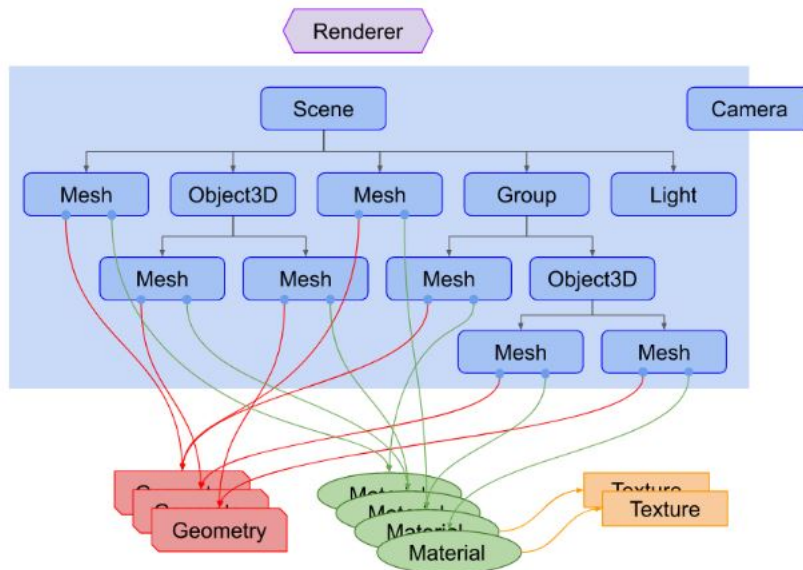
### 2.1 Three.js

ThreeJS là thư viện JS sử dụng WebGL để vẽ 3D. Tức là WebGL sẽ build hình ảnh 3D dựa theo code ThreeJS của bạn để vẽ các điểm, đường và tam giác. ThreeJS giúp chúng ta tạo nên các hình ảnh 3D trên browser chỉ bằng JS mà không cần phải tại platform, application nào để người dùng có thể trải nghiệm hình ảnh 3D.

### 2.2 Cấu trúc ứng dụng Three.js

#### 2.2.1 Các thành phần cần thiết

- Một đối tượng Scene: Component này sẽ chứa mọi thứ, giống như là 'vũ trụ thu nhỏ' vậy, là nơi mà các 3D object tồn tại
- Một camera: Giống như là camera trong thế giới thực, bạn sẽ sử dụng nó để xem 'scene' (kiểu máy chiếu ấy)
- Một đối tượng WebGLRenderer: 1 thứ 'máy móc' với input là camera và scene và output là vẽ những hình ảnh.



Hình 2.1: Cấu trúc ứng dụng Three.js

### 2.2.2 Mô hình cấu trúc

Hiểu đơn giản như này: camera của điện thoại chính là camera, bạn chụp ảnh thì chỉ có thể chụp được trong tầm nhìn của camera (xa thì bị mờ), vì vậy camera chính là thứ quyết định bạn có thể xem được cảnh - scene gì. Lúc này điện thoại đóng vai trò là renderer lấy ra các cảnh - scene được camera ghi lại ... (Giải thích lại)

## Chương 3

# Mô hình 3D hoặc cách tạo video

## Chương 4

# Xây dựng chương trình và kết quả

4.1 Giới thiệu mô hình 3D

4.2 Sử dụng texture cho mô hình 3D

4.3 Tạo các chú thích trên mô hình 3D

4.4 Thực hiện tạo video giới thiệu mô hình 3D



# Kết luận

**Đồ án đã đạt được mục tiêu đề ra**

1. ...
2. ...

**Kết quả của đồ án**

1. ...
2. ...

**Hướng phát triển của đồ án trong tương lai**

1. Xây dựng thêm chức năng đánh giá đồ án.
2. Hỗ trợ giám sát tiến độ đồ án của sinh viên.
3. Cải thiện giao diện đẹp mắt và thân thiện với người dùng hơn.

# Tài liệu tham khảo

## Tiếng Việt

- [1] Quy định về việc tổ chức, thực hiện các học phần Đồ án và Thực tập, ban hành kèm theo Quyết định số 07/QĐ – ĐHBK – TUD và TH, ngày 07 tháng 01 năm 2016 của Viện trưởng Viện Toán ứng dụng và Tin học.

## Tiếng Anh

- [2] D.Gale & L.S.Shapley, "College Admissions and the Stability of Marriage", *American Mathematical Monthly* 69, 9-14, 1962.
- [3] Harry Mairson, *The Stable Marriage Problem*, The Brandeis Review 12, 1992.
- [4] Gusfield, D.; Irving, R. W. (1989), *The Stable Marriage Problem: Structure and Algorithms*, MIT Press. p. 54.
- [5] Iwama, Kazuo; Miyazaki, Shuichi (2008), "A Survey of the Stable Marriage Problem and Its Variants", *International Conference on Informatics Education and Research for Knowledge-Circulating Society (ICKS 2008)*. IEEE. pp. 131–136.
- [6] Pittel, B, *On likely solutions of a stable marriage problem*, The Annals of Applied Probability, 1992.

- [7] Knuth, D.E, *Stable Marriage and Its Relation to Other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms*. CRM *Proceedings and Lecture Notes*, American Mathematical Society, 1996.