**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**----------**

**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC: ĐỒ HỌA MÁY TÍNH**

**Năm học: 2020 – 2021**

**Lớp: CS105.L22.KHCL**

**Giảng viên hướng dẫn: Cáp Phạm Đình Thăng**

**Nhóm thực hiện:**

**Huỳnh Lữ Anh Khoa - 18520916**

**Hà Thúc Đăng Khoa - 18520914**

**MỤC LỤC**

[**THÔNG TIN NHÓM** 2](#_Toc77085507)

[**Chương I: Mô tả đồ án.** 3](#_Toc77085508)

[**Chương II: Chức năng.** 4](#_Toc77085509)

[1. Chức năng đã thực hiện được. 4](#_Toc77085510)

[a. Object: 4](#_Toc77085511)

[2. Chức năng chưa thực hiện được. 5](#_Toc77085512)

[**Chương III: Cài đặt.** 6](#_Toc77085513)

[1. Hướng dẫn cài đặt. 6](#_Toc77085514)

[2. Hướng dẫn sử dụng. 6](#_Toc77085515)

[**Chương IV: Kết luận.** 7](#_Toc77085516)

[1. Kết quả đạt được. 7](#_Toc77085517)

[2. Hướng phát triển. 7](#_Toc77085518)

[**Chương V: Demo** 7](#_Toc77085519)

# **THÔNG TIN NHÓM**

Nhóm gồm 2 thành viên (Mã lớp: CS105.L22.KHCL)

* Hà Thúc Đăng Khoa. MSSV: 18520914
* Huỳnh Lữ Anh Khoa. MSSV: 18520916

# **Chương I: Mô tả đồ án.**

Đồ án được thực hiện bằng thư viện Three.js, là thư viện của Javascript được sử dụng để tạo và hiển thị các đồ họa máy tính 3D trên trình duyệt web thông qua WebGL.

WebGL (Web Graphics Library) sẽ build hình ảnh 3D dựa trên code ThreeJS để vẽ các điểm, đường và tam giác. WebGL là công cụ tạo 3D graphics ở low-level cho web, dựa trên OpenG. ThreeJS giúp ta tạo nên các hình ảnh 3D trên browser chỉ bằng JS mà không cần phải tạo platform hay application nào.

Các chương trình được xây dựng bằng ThreeJS sẽ có các thành phần chính như sau:

* Scene: Component này sẽ chứa mọi thứ để hiển thị, là nơi mà các Object xuất hiện.
* Camera: tương tự như camera, nó được sử dụng để xem Scene.
* Canvas: Phần tử HTML Canvas, tương tự như một bức màn chiếu, hay bức tranh trống để ThreeJS có thể vẽ lên.
* Renderer: sẽ render từ Camera và Scene và trả về hình ảnh hiển thị trên Canvas.

Bên cạnh đó, nhóm còn tìm hiểu và sử dụng thêm các thư viện hỗ trợ như OrbitControls.js dùng điều khiển, thay đổi góc nhìn bằng thao tác trên chuột; TransformControls.js dùng để kiểm soát các thay đổi về vị trí, scale, rotation của object; Dat.GUI dùng để thêm giao diện cho tác vụ thay đổi các chức năng mà không cần thay đổi code…

Đồ án nhóm hướng đến các chức năng cơ bản, những tính năng cần có cho một chương trình dùng để hiển thị đồ họa máy tính bằng Three.js

# **Chương II: Chức năng.**

1. Chức năng đã thực hiện được.
2. Object: Vật thể trong ThreeJS thường được cấu tạo từ 2 yếu tố: Hình dạng (Geometry) và Vật liệu (Material). Hai yếu tố này được kết hợp lại bởi hàm Hiển thị tương ứng cho Hình dạng và Vật liệu đó.

* Hiển thị được các loại object:
  + “Cube”: Hình lập phương.

Sử dụng geometry: THREE.BoxGeometry

* + “Sphere”: Hình cầu.

Sử dụng geometry: THREE.SphereGeometry

* + “Cone”: Hình nón.

Sử dụng geometry: THREE.ConeGeometry

* + “Cylinder”: Hình trụ.

Sử dụng geometry: THREE.CylinderGeometry

* + “Torus”: Hình bánh xe.

Sử dụng geometry: THREE.TorusGeometry

* + “Knot”

Sử dụng geometry: THREE.TorusKnotGeometry

* + “Tetrahedron”

Sử dụng geometry: THREE.TetrahedronGeometry

* + “Dodecahedron”

Sử dụng geometry: THREE.DodecahedronGeometry

* Thay đổi được surface của object thông qua Display type với 3 loại:
  + Solid

Sử dụng material: THREE.MeshStandardMaterial

Sử dụng Mesh: THREE.Mesh

* + Point.

Sử dụng material: THREE.PointsMaterial

Sử dụng Points: THREE.Points

* + Line.

Sử dụng material: THREE.WireframeGeometry

Sử dung Line: THREE.LineSegments

* Thực hiện được các loại animation trên object:
  + Reset: Đứng yên.
  + Rotation: Xoay vòng tại chỗ.
  + Orbiting: Xoay vòng tròn xung quanh.
* Thực hiện được các loại Tranform trên object:
  + Translate: Thay đổi vị trị của object bằng cách kéo object theo hướng các mũi tên hiển thị.
  + Rotate: Xoay vòng object theo hướng mũi tên hiển thị
  + Scale: Phóng to thu nhỏ object theo hướng mũi tên hiển thị.
* Điều chỉnh được màu của object bằng bảng hệ màu hexa.

1. Camera position: Điều chỉnh, thay đổi vị trí góc nhìn của camera bằng UI mà không cần thay đổi code.

* Điều chỉnh được camera zoom in, zoom out trên giao diện Dat.GUI.
* Thay đổi được Field of view của camera trên giao diện Dat.GUI.
* Xoay xung quanh object bằng cách kéo chuột, thay đổi độ gần xa object bằng con lăn chuột giữa.

1. Light and shadow: Thay đổi vị trí ánh sáng, cường độ sáng, chức năng gọi bóng lên vật thể của nguồn sáng.

* Ánh sáng: sử dụng THREE.PointLight.
* Thay đổi cường độ sáng của Pointlight với Intensity trong khoảng từ 0-10.
* Thay đổi chức năng rọi bóng của light (castShadow) bằng checkbox.
* Thay đổi vị trí của PointLight với các giá trị x, y, z thay đổi trong khoảng từ -20 đển 20.

1. Chức năng chưa thực hiện được.

* Chưa cài đặt được object dạng Teapot.
* Chưa cài đặt được chọn texture theo hình có sẵn cho object.

# **Chương III: Cài đặt.**

1. Hướng dẫn cài đặt.

* LƯU Ý: Do các thành viên trong nhóm đều sử dụng Window để code nên phần hướng dẫn cài đặt sẽ dành cho máy chạy Window.
* Cài đặt python với version 3.x trên máy. ([Download Python | Python.org](https://www.python.org/downloads/))
* Chạy local server bằng Python 3.x:
  + Mở Command Prompt tại folder CS105.L22.KHCL\_Final-project-main sau khi đã unzip file (có thể tải file về từ github: [jonathanha52/CS105.L22.KHCL\_Final-project: Repository for final project of UIT Computer Graphic course.](https://github.com/jonathanha52/CS105.L22.KHCL_Final-project) )
  + Để mở local server, gõ command: python -m http.server
  + Mở browser trên máy (Chrome, Edge…) gõ vào URL: local:8000
  + Xong.
* Lỗi có thể gặp:
  + Failed to load module script:….

Nếu gặp lỗi dạng này, mở code của file index.html. Bỏ chế độ comment của dòng thứ 25 và xóa bỏ (hoặc thêm vào chế độ comment) ở dòng thứ 26.

* + Can’t reach this page.

Kiểm tra trong cmd xem port đang nhận là bao nhiêu rồi thay đổi port trên URL.

1. Hướng dẫn sử dụng.

* Sau khi vào được thành công, việc đầu tiên cần chọn để sử dụng các chức năng khác là chọn Display type của object trong phần Mesh setting.
* Từ đây, bạn có thể chọn và thay đổi các tính năng của object, camera, light bằng giao diện Dat.GUI tùy ý.

# **Chương IV: Kết luận.**

1. Kết quả đạt được.

* Sau thời gian thực hiện và chạy thử, chương trình mà nhóm đã làm được những đặc trưng, những tính năng tương đối đầy đủ. Đáp ứng được các yêu cầu căn bản của chương trình hiện thị đồ họa cần có.
* Các thành viên trong nhóm học được căn bản cách hoạt động, cách thiết kế các đồ họa 3D đơn giản Three.js trên nền WebGL.

1. Hướng phát triển.

Hướng phát triển cho những version sau của chương trình:

* Có thêm các hình dạng mới hơn, đa dạng hơn cho object.
* Có thêm được nhiều loại surface của vật thể hơn, đồng thời cho phép người dùng họn texture cho vật thể.
* Có thêm nhiều lựa chọn cho loại light hơn trong chương trình.
* Có thêm các animation bắt mắt, độc đáo hơn.
* Sửa chữa các lỗi mà version hiện tại chưa sửa được.
* Có thể thêm nhiều object vào scene trong cùng lúc

# **Chương V: Demo**

* Link video demo: <https://drive.google.com/file/d/1Q3KeWnIRtxuAFnLrNkwj_eavNm9tM1zs/view?usp=sharing>
* Link Github:

https://github.com/jonathanha52/CS105.L22.KHCL\_Final-project