

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Báo cáo đồ án cuối kỳ

Chủ đề: Thiết kế sân chơi ảo

Môn học: Đồ họa máy tính

Giảng viên: Mai Tiến Dũng

Sinh viên thực hiện: Nguyễn Vũ Dương - 20520465

THUYẾT TRÌNH

Mục lục

- Lý do chọn đề tài
- Giới thiệu các công cụ
- Cấu trúc của đồ án
- Demo chạy đồ án

“

I. Lý do chọn đồ án

```
background: url(assets/...);
background-size: 100vw 100vh;
}
.box{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 400px;
  padding: 40px;
  background: rgba(0, 0, 0, 0.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0, 0, 0, 0.5);
  border-radius: 10px;
}
.box h2{
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box h3{
  margin: 0 0 10px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box .inputBox{
  margin: 10px 0;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
}
```




Lý do

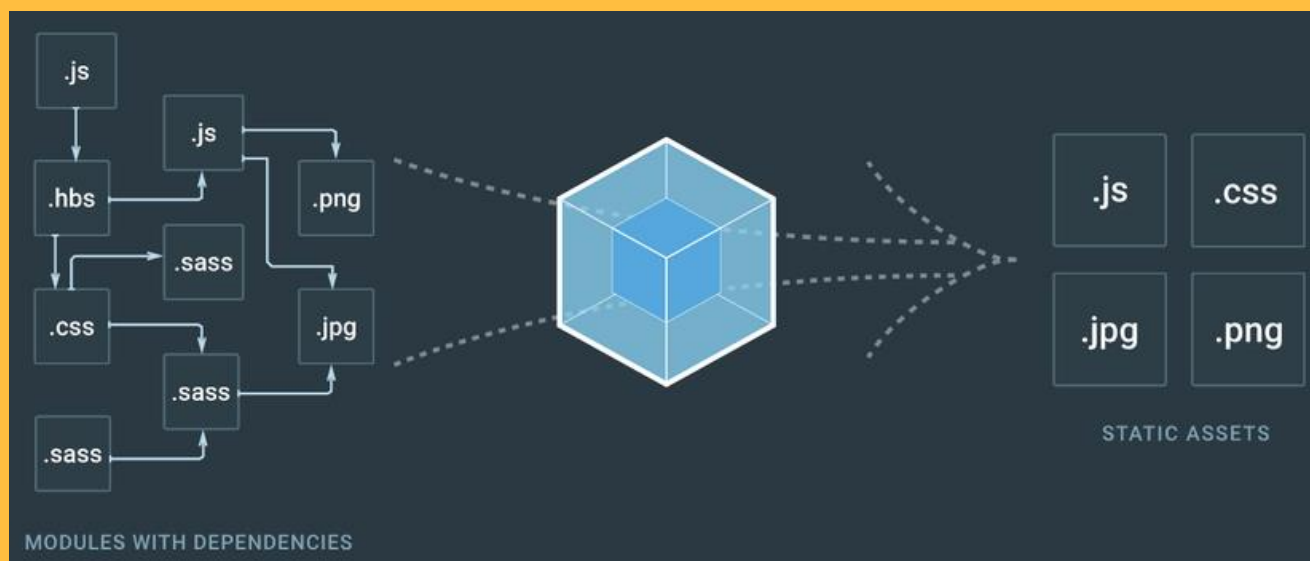
- **Lý do 1:** Muốn thỏa mãn trí tưởng tượng của bản thân và qua đồ án này muốn khám phá được giới hạn của bản thân.
- **Lý do 2:** Em nảy ra ý tưởng này là khi em đi bộ xung quanh chỗ khu vực em sinh sống , em chợt nhớ về sân chơi hồi xưa (hiện nay là một quán ăn) nơi chứa đựng bao nhiêu kỉ niệm tuổi thơ của em. Để có thể lưu giữ hình dung nó một cách cụ thể em đã chọn chủ đề này
- **Lý do 3:** Em mong muốn người dùng ghé thăm trang web của em cảm thấy yên bình và thoải mái sau một ngày làm việc học tập căng thẳng và gợi nhớ những kỷ niệm tươi đẹp

“

II. Các công cụ

```
background: url(assets/...);
background-size: 100vw 100vh;
}
.box{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 400px;
  padding: 40px;
  background: rgba(0, 0, 0, 0.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0, 0, 0, 0.5);
  border-radius: 10px;
}
.box h2{
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box h3{
  margin: 0 0 10px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box .inputBox{
  margin: 10px 0;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
}
```


1. Webpack



Webpack được biết đến là một công cụ phần mềm được sử dụng để quản lý các module JavaScript. Webpack sẽ đóng gói tất cả các mã nguồn của chương trình cũng như CSS, font, hình ảnh,... khi nó hoạt động. Assets chính là tên để gọi những thứ được đóng gói này và chúng sẽ được Webpack đóng gói thành 1 file hoặc một vài file.

Ưu Điểm	Nhược Điểm
Có khả năng xử lý asset tĩnh, đặc biệt là CSS	Có ít tài liệu hướng dẫn sử dụng.
Dễ dàng thực hiện chia tách mã nguồn	Việc thiết lập cấu hình cho Webpack gặp phải nhiều khó khăn với những cú pháp khó hiểu.
Kiểm soát quá trình xử lý asset.	Sử dụng ngôn ngữ được ít người biết đến và không được giải thích một cách kỹ lưỡng.
Giúp quá trình triển khai sản phẩm được ổn định thông qua việc giảm nguy cơ deploy code mà lại thiếu file ảnh và đăng nhầm file CSS cũ.	Webpack chỉ có một nhà phát triển do đó đôi lúc sẽ không kịp theo đuổi sự phát triển của công nghệ và những tài liệu được ghi chép cũng không đầy đủ.
Giúp tiết kiệm thời gian khi bạn sở hữu một cấu hình chuẩn nhất là với những người mới sử dụng hệ thống.	

2. WebGL

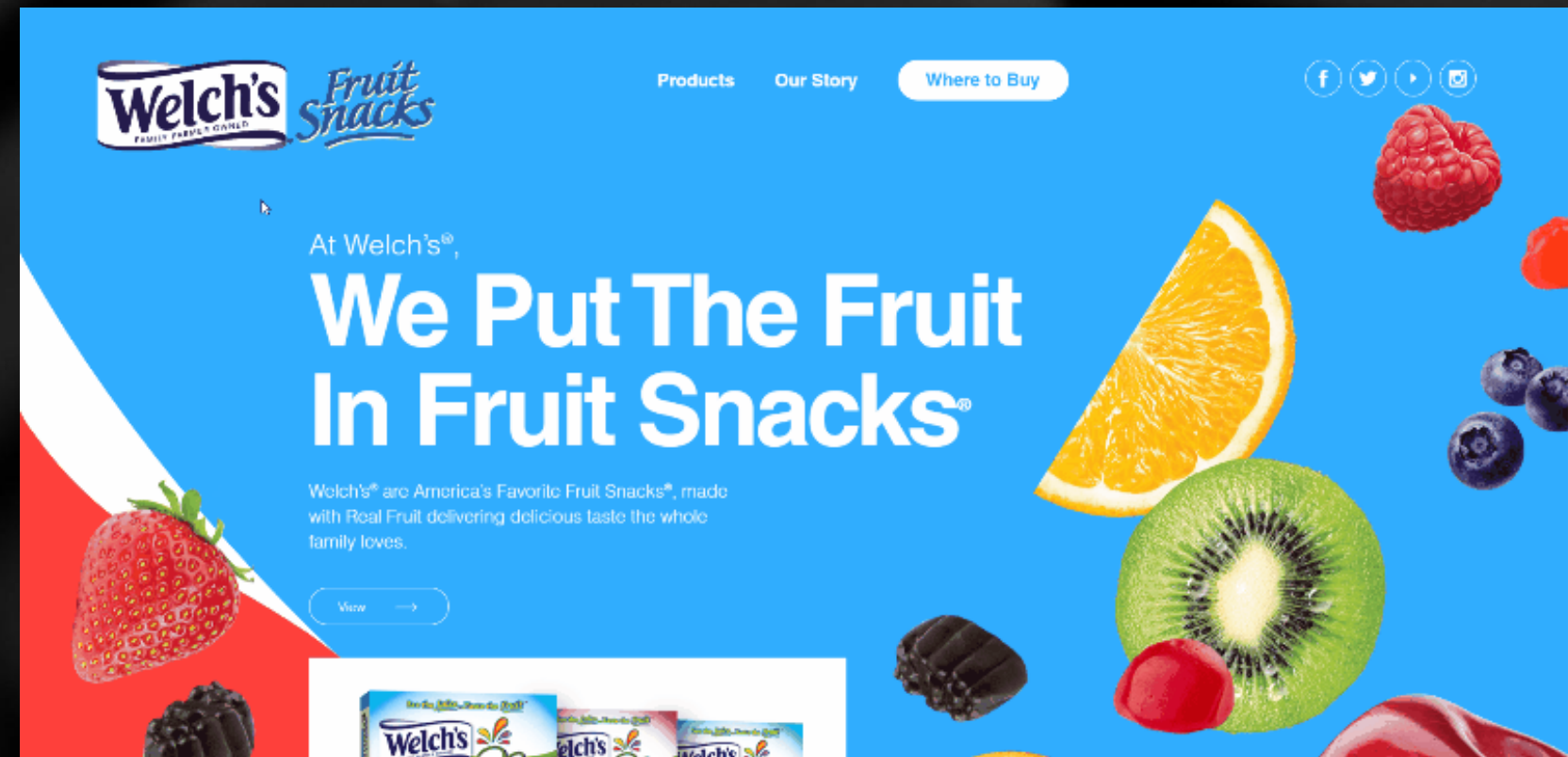
- **WebGL** (Web Graphics Library) là một thư viện đồ họa dành cho web, nó được dẫn xuất từ OpenGL ES (thư viện đồ họa 2D và 3D trên hệ thống nhúng: điện thoại, đồ điện tử, xe cơ giới)
- **Ưu điểm của WebGL:**
 - + WebGL mã nguồn mở.
 - + WebGL không cần thiết phải biên dịch để chạy.
 - + Ứng dụng WebGL được viết bằng javascript nên các ứng dụng này có thể tương tác trực tiếp với các phần tử HTML, ngoài ra ta có thể sử dụng thêm các thư viện javascript và các công nghệ HTML để hỗ trợ cho ứng dụng.
 - + Dễ dàng thiết lập và chạy, chỉ cần một text editor và trình duyệt.



3. GreenSock

GreenSock là một nền tảng giúp tạo hiệu ứng web mạnh mẽ cho phép tạo animations gần như với mọi thuộc tính của phần tử DOM, giá trị css, đối tượng canvas và nhiều hơn thế

Cơ chế của GreenSock (GSAP) là nhận vào giá trị ban đầu và kết thúc của một thuộc tính rồi animate nó 60 lần mỗi giây. Chính vì vậy nó mới có tên là "GreenSock Property Manipulator" (GSPM) nhưng vì cái tên đó không được đẹp cho lắm nên đã được gọi với một cái tên mới là GSAP.



3. GreenSock

GSAP có 4 công cụ chính

TweenLite	<ul style="list-style-type: none">- Là thành phần core của GSAP, dùng để tạo các hiệu ứng cho các thuộc tính với giá trị number ví dụ như: width, height. Cùng với CSS plugin bạn có thể tạo hiệu ứng cho bất kỳ CSS property ví dụ như fontSize, backgroundColor.- TweenLite thì tốt cho các hiệu ứng đơn giản với một vài elements
TimelineLite	<ul style="list-style-type: none">- Bao gồm nhiều hiệu ứng hoặc dòng thời gian. Pause, reverse, restart, speed up, slow down, seek time, add labels...- Thường dùng để tạo hiệu ứng theo trình tự thời gian.
TimelineMax	<ul style="list-style-type: none">- Được thêm vào các function như repeat, yoyo, tweening to previous or next label, custom callback functions...- TimelineMax dùng để tạo các hiệu ứng nâng cao theo trình tự thời gian.
TweenMax	<ul style="list-style-type: none">- Bao gồm tất cả các tính năng trên và được bổ sung các function như repeat, yoyo, delay and stagger individual tweens or timelines và nhiều hơn nữa. Nó cũng đi kèm với nhiều plugin phổ biến.- TweenMax dùng là công cụ tạo hiệu ứng mạnh mẽ nhất.



4. CSS

- CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web.
- Ưu điểm khi sử dụng CSS:
 - + CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.
 - + Nhờ CSS mà mã nguồn của trang Web sẽ được tổ chức gọn gàng hơn, trật tự hơn, nội dung trang web sẽ được tách bạch hơn trong việc định dạng hiển thị
 - + CSS tạo ra nhiều kiểu dáng nên có thể được áp dụng với nhiều trang web, từ đó giảm tránh việc lặp lại các định dạng của các trang web giống nhau.



5. HTML5

- HTML5 là một ngôn ngữ cấu trúc và trình bày nội dung cho World Wide Web. Đây là phiên bản thứ 5 của ngôn ngữ HTML, được giới thiệu bởi World Wide Web Consortium (W3C). HTML5 vẫn sẽ giữ lại những đặc điểm cơ bản của HTML4 và bổ sung thêm các đặc tả nổi trội của XHTML, DOM cấp 2, đặc biệt là JavaScript.
- Ưu điểm khi sử dụng HTML5:
 - + HTML5 đa nền tảng và responsive..
 - + Nó hỗ trợ âm thanh và video.
 - + Có nhiều layout elements hơn cho nội dung của bạn.
 - + Nó hoạt động tốt với tính nhất quán tuyệt vời.
 - + Cung cấp lợi ích tối ưu hóa công cụ tìm kiếm.
- Nhược điểm :
 - + Nó yêu cầu các trình duyệt hiện đại để sử dụng
 - + Có nhiều chuẩn video khác nhau cho HTML5.

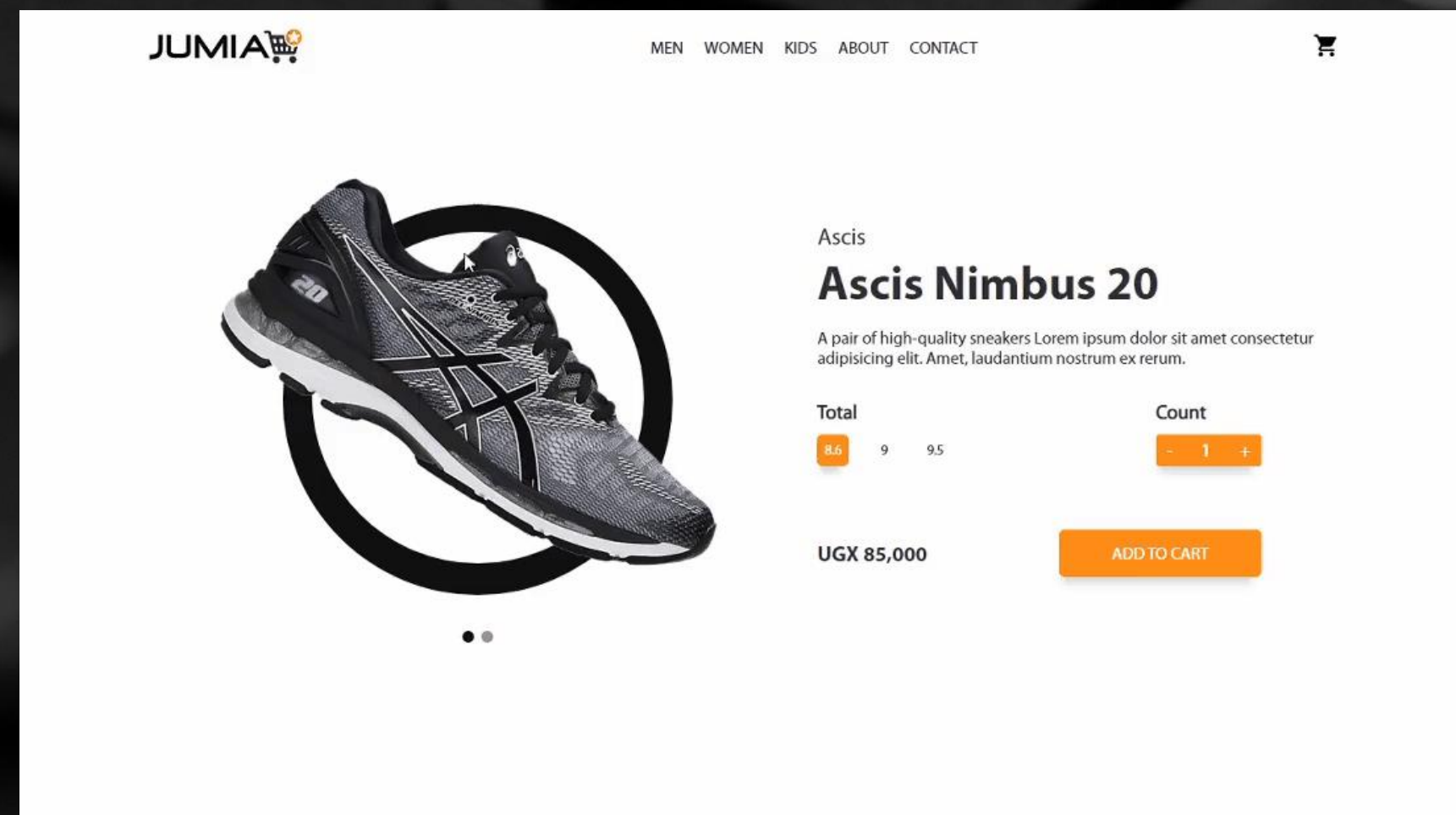
6. Blender

- **Blender** được biết đến là một trong các phần mềm phổ biến về thiết kế đồ họa 3D hiện nay. Để tạo ra các kỹ xảo hình ảnh, hiệu ứng hình ảnh, nhân vật 3D đều có thể sử dụng phần mềm này một cách đẳng cấp và hoàn hảo. Blender đặc biệt còn là một phần mềm không hề mất chi phí trong việc thiết kế 3D còn được kèm theo các tính năng vô cùng hiện đại.
- **Ưu điểm:**
 - + Hỗ trợ nhiều loại file đa dạng và khác nhau.
 - + Siêu nhẹ và siêu nhanh khi Blender nhắc đến tốc độ load, ở đĩa chỉ tốn 250Mb.
 - + Miễn phí hoàn toàn phần mềm và nếu như biết code người sử dụng có thể chỉnh sửa tùy ý.
 - + Có Viewport Realtime, hỗ trợ Python Scripting, hỗ trợ vật liệu Node Base, hỗ trợ công nghệ Path Tracing, các thể loại hiệu ứng đầy đủ.



💡 7. ThreeJS

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.



8. Draco

- **Draco** là thư viện dùng để nén và giải nén lưới (mesh) hình học 3D(tập hợp các đỉnh ,cạnh và mặt xác định hình dạng của một đa diện) và point clouds (là một tập hợp các điểm dữ liệu rời rạc trong không gian). Thư viện này nhằm mục đích cải thiện việc lưu trữ và truyền tải đồ họa 3D
- **Draco** được thiết kế và tạo ra để tăng về hiệu suất và tốc độ nén. Thư viện hỗ trợ các điểm nén, kết nối các thông tin ,tọa độ kết cấu , thông tin về màu sắc, quy tắc và bất kỳ thuộc tính chung nào khác liên quan đến hình học. Với Draco, các ứng dụng sử dụng đồ họa 3D có thể ít dung lượng hơn đáng kể mà không ảnh hưởng đến độ trung thực của hình ảnh -> Giúp người dùng có thể tải nhanh hơn và có thể hiển thị một cách nhanh chóng

III. Cấu trúc đồ án

```
background: url(assets/img/...);
background-size: 100vw 100vh;
}
.box{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 400px;
  padding: 40px;
  background: rgba(0, 0, 0, 0.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0, 0, 0, 0.5);
  border-radius: 10px;
}
.box h2{
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box h3{
  margin: 0 0 10px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box .inputBox{
  margin: 10px 0;
  padding: 0;
  color: #fff;
  text-align: center;
}
```


Cấu trúc đồ án (Trang web)

Gồm 7 phần cơ bản:

1. **Import các thư viện và module được giới thiệu ở trên**
2. **Màn hình loading:** Tạo màn hình loading nhằm mục đích kiểm tra xem các tài nguyên của trang web đã được import vào trang web hay chưa và nhằm mục đích giảm sự nhàm chán khi chờ đợi trang web load các tài nguyên. Để tạo màn hình loading thì ta sẽ sử dụng **THREE.LoadingManager** để theo dõi tiến trình tải các tài nguyên như mô hình 3D, vật liệu và âm thanh. Khi người dùng nhấn nút bắt đầu (engage) thì màn hình loading này sẽ ẩn đi và sau đó hiển thị giao diện chính của trang web.
3. **Các đối tượng trong cảnh 3D:** Tạo các đối tượng cần thiết như vị trí camera ,ánh sáng ,v,v. Đối với camera thì em sử dụng với góc nhìn **PerspectiveCamera** của thư viện **Threejs** (Vì nó mô phỏng hành động của một camera quay phim trong đời thực càng xa camera thì càng nhỏ và ngược lại) và đặt tại một vị trí cụ thể. Về ánh sáng thì em sử dụng 2 loại ánh sáng là **AmbientLight** (ánh sáng môi trường) và **DirectionalLight**(ánh sáng hướng) để tạo hiệu ứng ánh sáng cho cảnh 3D có trong trang web. Các material và mô hình 3D sẽ được tải lên bằng việc sử dụng GLTFLoader và DRACOLoader. Các material được sử dụng để khiến cho mô hình 3D rõ ràng hơn về hiệu ứng màu sắc và kết cấu

Cấu trúc đồ án (Trang web)

4. Hiệu ứng hậu kỳ: Sử dụng các hiệu ứng có sẵn trong thư viện **Threejs** để tạo ra các hiệu ứng hình ảnh cho cảnh 3D. Hiệu ứng đầu tiên em sử dụng là **EffectComposer** để kết hợp với các hiệu ứng hậu kỳ khác. **RenderPass** được sử dụng để render cảnh 3D. **UnrealBloomPass** được sử dụng để tạo ra hiệu ứng ánh sáng. **FilmPass** được sử dụng để tạo ra hiệu ứng phim cho cảnh 3D.

5. Chức năng điều khiển camera : Cho phép người dùng có thể xoay, di chuyển , thu phóng camera để xem các đối tượng trong cảnh 3D từ các góc độ khác nhau. Để thực hiện được thì em đã sử dụng **OrbitControls** trong thư viện threejs để hỗ trợ.

6. Tạo ra các hiệu ứng: Để cho đồ án nó giống thật hơn và không quá nhàm chán thì em đã kết hợp với thư viện **GSAP** để tạo ra các hiệu ứng VD: lá / hoa rụng xuống để chân thực hơn bằng việc sử dụng các phương thức có sẵn trong **GSAP** như **to()** , **from()** và **fromto()** để giúp có thể xoay , di chuyển , thay đổi kích thước của vật thể.

7. Tạo ra các event: Các sự kiện này bao gồm Play , Explore . Play thì sẽ đưa người dùng tham quan đến các địa điểm có những góc view được xác định trước đó. Chế độ explore khiến người dùng có thể tự do tham quan các địa điểm.

IV. DEMO

```
background: url(../img/...);
background-size: 100vw 100vh;
}
.box{
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  width: 400px;
  padding: 40px;
  background: rgba(0, 0, 0, 0.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0, 0, 0, 0.5);
  border-radius: 10px;
}
.box h2{
  margin: 0 0 30px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box h3{
  margin: 0 0 10px;
  padding: 0;
  color: #fff;
  text-align: center;
}
.box .inputBox{
```




**THANK
YOU**