

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO MÔN HỌC
ĐỒ HỌA MÁY TÍNH
CS105.N22.KHCL
ĐỀ TÀI: THIẾT KẾ SÂN CHƠI ẢO

GIẢNG VIÊN HƯỚNG DẪN: MAI TIẾN DŨNG

SINH VIÊN THỰC HIỆN: NGUYỄN VŨ DƯƠNG – 20520465

TP. HỒ CHÍ MINH, 05/2023

LỜI CẢM ƠN

Ngày nay công nghệ ngày càng phát triển, các hệ thống máy tính cũng ngày càng trở nên mạnh mẽ. Cùng với đó với sự bùng nổ của thực tế ảo và thực tế ảo tăng cường, nó đã giúp chúng ta có thể có đi khắp nơi trên thế giới và tương tác với các địa điểm đó chỉ qua một kính.

Là sinh viên ngành KHMT, em cũng muốn tìm hiểu về công nghệ này, và muốn tạo ra một thứ gì đó liên quan đến nó. Môn học này cung cấp cho em sơ lược về cách thức hoạt động của các mô hình 3D trong máy tính do đó em mong muốn mình tạo ra một cái gì đó hấp dẫn và thử sức với bản thân mình trong một lĩnh vực mà nói thật lòng thì em không giỏi lắm.

Em xin gửi lời cảm ơn chân thành đến Thầy Mai Tiến Dũng đã tận tình giảng dạy, hướng dẫn chúng em trong suốt thời gian học vừa qua và các bạn học đã góp ý và giúp đỡ nhóm trong quá trình thực hiện đồ án này.

Do kiến thức và thời gian thực hiện hạn chế, đồ án của em vẫn còn nhiều thiếu sót. Em rất mong nhận được góp ý của thầy và các bạn để đồ án của nhóm được hoàn thiện.

TP. Hồ Chí Minh, tháng 05 năm 2023.

MỤC LỤC

I)	Giới thiệu.....	1
1)	Khái niệm về đồ họa máy tính	1
2)	Lịch sử phát triển của đồ họa máy tính	1
3)	Kỹ Thuật Trong Đồ Họa Máy Tính	2
3.1)	Đồ họa raster	2
3.2)	Đồ họa vector.....	2
3.3)	Đồ họa 3D	3
4)	Thực tế ảo.....	4
5)	Thực tế ảo tăng cường	5
6)	Lý do chọn đề tài	6
II)	Các công cụ hỗ trợ.....	7
1)	Webpack.....	7
2)	WebGL	8
3)	GreenSock (GSAP).....	9
4)	CSS	10
5)	HTML5.....	11
6)	Blender	12
7)	ThreeJS	14
8)	Draco	14
III)	Thiết kế hệ thống.....	15
1)	Load các file lên website	15
1.1)	Load model sân chơi.....	15

1.2) Load các file khác cho website.....	16
2) Thiết kế màn hình loading.....	16
3) Các đối tượng trong cảnh 3D	17
3.1) Góc nhìn.....	17
3.2) Ánh sáng	17
4) Post Processing	18
6) Hiệu ứng camera cinematic.....	19
7) Vị trí đặc biệt trong mô hình.....	19
8) Chức năng điều khiển	21
9) Các chức năng khác	21
IV) KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	22
1) Hạn chế.....	22
2) Hướng phát triển.....	22
V) TÀI LIỆU THAM KHẢO	23

DANH MỤC HÌNH

Hình 1.1: Ví dụ về đồ họa raster	2
Hình 1.2: Ví dụ về đồ họa vector	3
Hình 1.3: Ví dụ về đồ họa 3D	3
Hình 1.4: Ứng dụng của thực tế ảo trong đời sống	5
Hình 1.5: Thực tế ảo tăng cường trong hện hò.....	6
Hình 2.1: Hình ảnh mô phỏng phòng với blender	13
Hình 2.2: Vẽ hình 2D trong chế độ 3D với blender.....	13
Hình 3.1: Model khi kết hợp cubeTextureLoader	15
Hình 3.2: Model gốc	16
Hình 3.3: Hình ảnh loading của website	17
Hình 3.4: Các interest point khi hiện	20
Hình 3.5: Các interest point khi ẩn	21

DANH MỤC CHỮ VIẾT TẮT

I) Giới thiệu

1) Khái niệm về đồ họa máy tính

- Đồ họa máy tính (computer graphics) là một lĩnh vực của khoa học máy tính nghiên cứu về cơ sở toán học, các thuật toán cũng như các kỹ thuật để cho phép tạo, hiển thị và điều khiển hình ảnh trên màn hình máy tính. Đồ họa máy tính có liên quan ít nhiều đến một số lĩnh vực như đại số, hình học giải tích, hình học họa hình, quang học,... và kỹ thuật máy tính, đặc biệt là chế tạo phần cứng (các loại màn hình, các thiết bị xuất, nhập, các vi mạch đồ họa,...).
- Theo nghĩa rộng hơn, đồ họa máy tính là phương pháp và công nghệ dùng trong việc chuyển đổi qua lại giữa dữ liệu và hình ảnh trên màn hình bằng máy tính. Đồ họa máy tính hay kỹ thuật đồ họa máy tính còn được hiểu dưới dạng phương pháp và kỹ thuật tạo hình ảnh từ các mô hình toán học mô tả các đối tượng hay dữ liệu lấy được từ các đối tượng trong thực tế.

2) Lịch sử phát triển của đồ họa máy tính

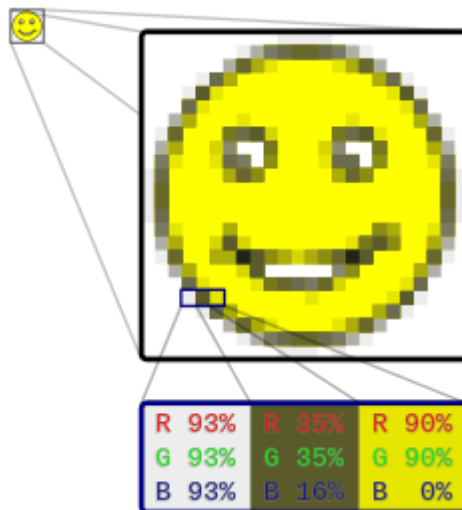
- Lịch sử của đồ họa máy tính vào thập niên 1960 còn được đánh dấu bởi dự án SketchPad được phát triển tại Học viện Công nghệ Massachusetts (MIT) bởi Ivan Sutherland. Các thành tựu thu được đã được báo cáo tại hội nghị Fall Joint Computer và đây cũng chính là sự kiện lần đầu tiên người ta có thể tạo mới, hiển thị và thay đổi được dữ liệu hình ảnh trực tiếp trên màn hình máy tính trong thời gian thực. Hệ thống Sketchpad này được dùng để thiết kế hệ thống mạch điện và bao gồm những thành phần sau:
 1. CRT màn hình
 2. Bút sáng và một bàn phím bao gồm các phím chức năng
 3. Máy tính chứa chương trình xử lý các thông tin
- Với hệ thống này, người sử dụng có thể vẽ trực tiếp các sơ đồ mạch điện lên màn hình thông qua bút sáng, chương trình sẽ phân tích và tính toán các thông số cần thiết của mạch điện do người dùng vẽ nên.
- Kỹ thuật đồ họa được liên tục hoàn thiện vào thập niên 1970 với sự xuất hiện của các chuẩn đồ họa làm tăng cường khả năng giao tiếp và tái sử dụng của phần mềm cũng như các thư viện đồ họa.
- Sự phát triển vượt bậc của công nghệ vi điện tử và phần cứng máy tính vào thập niên 1980 làm xuất hiện hàng loạt các vi mạch hỗ trợ cho việc truy xuất đồ họa đi cùng với sự giảm giá đáng kể của máy tính cá nhân làm đồ họa ngày càng đi sâu vào cuộc sống thực tế.

3) Kỹ Thuật Trong Đồ Họa Máy Tính

- Có rất nhiều kỹ thuật hiển thị hình ảnh được áp dụng trong ngành đồ họa máy tính, mỗi kỹ thuật lại có tuổi đời và những ưu, nhược điểm khác nhau. Dưới đây là 3 kỹ thuật hiển thị hình ảnh của đồ họa máy tính phổ biến nhất:

3.1) Đồ họa raster

- Thiết kế đồ họa raster (đồ họa hoặc hình ảnh Bitmap) là một trong những kỹ thuật hiển thị hình ảnh lâu đời và phổ biến nhất với nền tảng kỹ thuật lấy từ công nghệ màn hình tivi đã tồn tại rất lâu trước khi máy tính điện tử ra đời. Với kỹ thuật này, tất cả các hình ảnh đều được làm nên từ các ô vuông có màu nhỏ li ti được gọi là pixel (phần tử ảnh).
- Tùy thuộc vào độ phân giải, một hình ảnh có thể chứa hàng nghìn đến hàng triệu pixel, giống như một bức tường được xây lên từ nhiều viên gạch vậy. Ưu điểm lớn nhất của kỹ thuật raster là các hình ảnh raster có thể hiển thị các chi tiết rõ ràng, đẹp với màu sắc đa dạng, hài hòa. Tuy nhiên, hình ảnh raster có thể sẽ bị “vỡ” hoặc mờ nếu phóng to hoặc bị nén quá nhiều. Dung lượng các file ảnh raster cũng khá lớn nếu có độ phân giải cao.



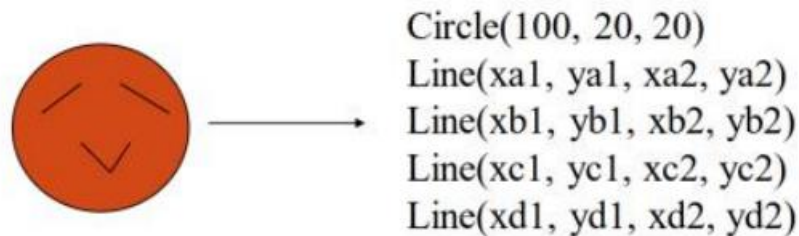
Hình 1.1: Ví dụ về đồ họa raster

3.2) Đồ họa vector

- Đồ họa vector là kỹ thuật tạo dựng hình ảnh bằng các đường kẻ quy định bởi các công thức toán học lần đầu tiên được sử dụng cho màn hình máy tính trong những năm 60 và 70 của thế kỷ 20. Tuy không phổ biến bằng

kỹ thuật đồ họa raster và đã từng có một thời gian gần như bị “xóa sổ” bởi raster, đồ họa vector đang được ưa chuộng trở lại.

- Nhờ vào những ưu điểm như các hình ảnh vector đơn giản, dễ dàng phóng to mà không bị giảm chất lượng, có dung lượng nhỏ hơn so với raster, dễ chỉnh sửa, rất thích hợp với việc thiết kế các loại đồ họa ít màu sắc, đơn giản như logo, icon hay biểu tượng.



Hình 1.2: Ví dụ về đồ họa vector

3.3) Đồ họa 3D

- Đồ họa 3D là kỹ thuật đồ họa đang được tập trung phát triển nhất trong thời điểm hiện tại, với sự quan tâm và tiềm năng của các ứng dụng như không gian ảo hay hình chiếu ba chiều. Nhà thiết kế đồ họa phải thực hiện rất nhiều bước khác nhau và áp dụng nhiều kỹ thuật tạo dựng hình ảnh phức tạp để có được một đối tượng hình ảnh 3D đúng nghĩa.
- Trước hết, khung cơ bản (wire-frame) của vật thể phải được dựng trong một phần mềm đồ họa máy tính, sau đó các phần của vật thể sẽ được thêm vào và nối với nhau (rigged) để tạo sự liên kết chân thực, đặc biệt là với các vật thể có khả năng chuyển động. Sau đó vật thể phải được render.
- Đây là bước mất thời gian và phức tạp nhất vì một vật thể 3D có nhiều bề mặt khác nhau với các chất liệu khác nhau, độ trong suốt hay mờ đục, màu sắc, mức độ bắt sáng hay phản sáng khác nhau. Để tạo được một vật thể 3D thật nhất có thể, nhiều kỹ thuật toán học được áp dụng để tính toán và mô phỏng cách ánh sáng chiếu và phản chiếu vào các loại mặt phẳng khác nhau như ray tracing hay radiosity,...



Hình 1.3: Ví dụ về đồ họa 3D

4) Thực tế ảo

- Thực tế ảo hay còn gọi là thực tại ảo (tiếng Anh là virtual reality, viết tắt là VR) là thuật ngữ miêu tả một môi trường được giả lập (ảo hóa) được tạo ra bởi con người nhờ vào các phần mềm chuyên dụng, và được điều khiển bởi một thiết bị thông minh.
- Ngoài việc tạo ra không gian ảo, công nghệ thực tế ảo VR còn có thể tương tác thực tế với người dùng qua cử chỉ và nhiều giác quan khác nhau như: Thính giác, khứu giác và xúc giác.
- Ứng dụng của thực tế ảo:
 - **Trong Quảng Cáo Và Marketing:** Việc sử dụng VR trong tiếp thị kỹ thuật số giúp nâng cao trải nghiệm mua sắm của khách hàng bằng cách tăng tương tác trực tiếp, chạm vào và dùng thử sản phẩm mà họ muốn mua thông qua các hiệu ứng hình ảnh đa chiều do VR tạo ra.
 - **Trong Lĩnh Vực Game**
 - **Trong Ngành Du Lịch Ảo:** Chỉ bằng cách đeo kính VR, bạn có thể trải nghiệm đi đến điểm du lịch thực tế mà không cần quá tốn kém nhiều chi phí. Trong thế giới ảo, bạn sẽ như lạc vào những viện bảo tàng, di tích văn hóa, hòa mình vào thế giới tự nhiên, chiêm ngưỡng những danh lam thắng cảnh tuyệt đẹp của thế giới cũng như tại Việt Nam.
 - **Khi Mua Sắm Tại Nhà:** Chính kính VR đã giúp khách hàng tránh việc mất nhiều thời gian để dùng thử. Bạn chỉ cần đeo kính và thử sản phẩm trên thế giới ảo, nếu sản phẩm nào phù hợp bạn có thể quyết định mua ngay lập tức. Ngoài ra, bạn luôn có thể thử quần áo mà không cần phải đến trực tiếp cửa hàng.
 - **Trong Huấn Luyện Quân Sự**
 - **Trong Đào Tạo Bác Sĩ:** Ứng dụng mô phỏng trong môi trường thực tế ảo mang tên HumanSim giúp đào tạo các chuyên gia y tế bằng cách đưa họ vào các tình huống hiểm nghèo, như phải mô phỏng lại một trường hợp bệnh nhân bị chấn thương nặng và họ phải xử lý thành công vấn đề
 - **Trong Thiết Kế Xe Hơi:** Nhiều nhà sản xuất ô tô như Ford từ lâu đã sử dụng các phòng thí nghiệm tái hiện bằng thực tế ảo để cho phép các nhân viên trải nghiệm một chiếc xe trước khi nó được chế tạo. Từ đây, họ có thể xem chi tiết các bộ phận của từng chiếc xe và thậm chí cho phép người dùng đánh giá chúng và sau đó nhận phản hồi trước khi mô hình được chế tạo.
 - **Trong Ngành Giáo Dục:** Giáo dục cũng là một ứng dụng vô cùng tiềm năng khi ứng dụng công nghệ thực tế ảo, nó cho phép

các học sinh được tham quan các công trình kiến trúc cổ đại trong giờ lịch sử, tham gia các bài tập giải phẫu trong lớp sinh học hoặc được truyền cảm hứng từ những hiện tượng thiên nhiên kỳ thú mà sách vở có thể không bao giờ mô tả được.



Hình 1.4: Ứng dụng của thực tế ảo trong đời sống

5) Thực tế ảo tăng cường

- Thực tế tăng cường (tiếng Anh: augmented reality, viết tắt là AR) là góc nhìn trực tiếp hay gián tiếp về môi trường vật lý, thực tế nơi mà các yếu tố được "tăng cường" bởi thông tin nhận thức do máy tính tạo ra, lý tưởng trên nhiều phương thức cảm quan bao gồm thị giác, thính giác, xúc giác, khứu giác, và khứu giác. Các phủ thông tin cảm giác có thể được xây dựng (tức là phụ đề các môi trường tự nhiên) hoặc phá hoại (tức là mặt nạ của môi trường tự nhiên) và là không gian đã đăng ký với thế giới vật chất như vậy mà nó được coi như là một nhập vai khía cạnh của môi trường. Theo cách này, thực tế tăng cường làm thay đổi một hiện tại của nhận thức về một thế giới thực môi trường, trong khi thực tế ảo thay thế giới thực môi trường với một mô phỏng
- Thực tế gia tăng được sử dụng để tăng cường môi trường tự nhiên hoặc các tình huống và cung cấp trải nghiệm làm giàu cảm tính. Với sự trợ giúp của các công nghệ AR tiên tiến (ví dụ như bổ sung tầm nhìn và nhận diện đối tượng máy tính) thông tin về thế giới thực xung quanh của người sử dụng sẽ trở nên tương tác và thao túng kỹ thuật số. Thông tin về môi trường và các đối tượng của nó được che phủ trên thế giới thực.

- Ứng dụng của thực tế ảo tăng cường:
 - Trong dịch vụ hẹn hò
 - Trong các ứng dụng thể thao
 - Trong tiếp thị
 - Trong mua sắm
 - Trong y học
 - Trong xây dựng cảnh quan
 - Trong bản đồ và du lịch
 - Trong các hội nghị trực tuyến
 - Trong giáo dục



Hình 1.5: Thực tế ảo tăng cường trong hẹn hò

6) Lý do chọn đề tài

- Có thể thấy các ứng dụng thực tế ảo ngày nay đều có nguồn gốc từ việc ứng dụng của đồ họa máy tính trong đó. Chúng ta có thể chỉ cần một cái kính ta có thể đi khắp nơi khám phá các địa điểm trên khắp thế giới này. Vì lý do này, em cũng muốn mô phỏng một thứ gì đó mà hiện tại em khó có thể nhìn thấy nó hằng ngày
- Em cũng là một người có tính tò mò và muốn khám phá và học hỏi thêm nhiều thứ qua đề án này thế nên em sẽ làm một đề án em coi là lớn và làm hết tất cả khả năng em có thể làm được.
- Em chọn mô phỏng sân chơi vì trong một dịp đi dạo xung quanh chỗ nơi em sinh sống, em chợt gợi nhớ lại những kỷ niệm tươi đẹp của mình thời ấu thơ với những người bạn trong khu vui chơi đó (hiện tại nơi này là một quán ăn). Để luôn nhắc nhở rằng để là mình của hiện tại thì mình không được quên những thứ đã giúp mình phát triển và xây dựng một thời thơ ấu tuyệt vời thì em đã mô phỏng lại để giúp cho em dễ dàng nhắc nhở mình hơn

- Vì làm đến sâu chơi nên em mong muốn ai đó tham quan đến website sẽ cảm thấy thật sự thoải mái và yên bình bởi điệu nhạc của trang web và phong cảnh em thiết kế sau một ngày học tập làm việc căng thẳng.

II) Các công cụ hỗ trợ

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.
- Mặc dù đóng gói rất nhiều dữ liệu nhưng chúng được đóng gói một cách rất cẩn thận, bài bản và ngăn nắp, nó được sắp xếp với cấu trúc tương tự như viết mã code. Những dữ liệu này được lập trình sẵn xem cái nào chạy trước, cái nào chạy sau và phần nào sẽ phụ thuộc vào nhau.
- Hiện nay với sự phát triển như vũ bão của Javascript đặc biệt là các đoạn mã code được viết sẵn như ReactJS, AngularJS,... giúp giảm thiểu code và các CSS giúp nâng cao trải nghiệm người dùng khiến cho bộ nhớ và các module JS trong ứng dụng không thể kiểm soát và gây tình trạng phình website. Lúc này, vấn đề xảy ra là sắp xếp các mã nguồn như thế nào để dễ quản lý và truy xuất dữ liệu. Chính vì vậy, việc sử dụng Webpack nhằm giúp cải thiện tình trạng bừa bãi và quá tải là một sự lựa chọn đúng đắn. Ngoài ra, Webpack còn có thể thực hiện được nhiều hoạt động khác nhau như:
 - Giúp đóng gói resources.
 - Theo dõi những thay đổi và chạy lại các task.
 - Có thể thực hiện biên dịch Babel chuyển sang ES5, cho phép người dùng sử dụng các tính năng JavaScript mới nhất mà không phải lo về hỗ trợ trình duyệt.
 - Có thể dịch từ CoffeeScript sang JavaScript.
 - Có thể chuyển đổi ảnh inline sang data URIs.
 - Cho phép người dùng sử dụng require() cho các file CSS.
 - Có thể chạy một web server dùng cho development.
 - Có thể xử lý hot module replacement.
 - Có thể tách các file output thành nhiều file khác nhau để tránh tải file js lớn trong lần đầu tải trang.
 - Có thể thực hiện tree shaking.

- **Ưu và nhược điểm của Webpack:**

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 nhậm 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 hay Web Graphics Library là một thư viện đồ họa cho phép vẽ, hiển thị hay tương tác với các đồ họa 3D được dẫn xuất từ OpenGL ES (thư viện đồ họa 2D và 3D trên hệ thống nhúng) dành cho trang web. Với các chức năng cơ bản tương tự OpenGL ES, WebGL có khả năng hoạt động tốt trên các phần cứng của đồ họa 3D hiện đại.

- **Ưu và nhược điểm của WebGL:**

Ưu điểm	Nhược điểm
Dễ dàng thiết lập và chạy, chỉ cần một text editor và trình duyệt.	Nó chậm hơn so với người tiền nhiệm của nó (OpenGL)
WebGL mã nguồn mở.	Lỗi hỏng bảo mật có thể xảy ra
WebGL không cần thiết phải biên dịch để chạy.	Một số trình duyệt thì plugin của WebGL có thể không tương thích

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.	
--	--

3) GreenSock (GSAP)

- 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.
- Thư viện này có thể cập nhật các giá trị một cách nhanh chóng và độ chính xác cao, vượt trội hơn khoảng 25 lần so với các phương pháp animation khác. GSAP là một công cụ hữu ích để tạo ra các animation và tăng tính tương tác của website hay ứng dụng web. Nó rất đáng tin cậy và hiệu quả, ngay cả Google cũng đề xuất nó cho JavaScript animations.
- Cơ chế hoạt động của GSAP:
 - Bước 1: Nhận vào giá trị ban đầu và giá trị kết thúc của một thuộc tính.
 - Bước 2: Tính toán ra các giá trị trung gian cần thiết để chuyển từ giá trị ban đầu đến giá trị kết thúc. GSAP sử dụng các hàm tween để làm việc này.
 - Bước 3: Áp dụng các giá trị trung gian tính toán được vào thuộc tính của đối tượng cần điều khiển.
 - Bước 4: Thực hiện các hiệu ứng, hoạt hình, thay đổi vị trí, kích thước, màu sắc, độ trong suốt,... đối với đối tượng được điều khiển.
- ➔ GSAP cho phép tạo ra các hiệu ứng hoạt hình động, mượt mà và linh hoạt trên trang web.
- Ưu điểm của GSAP:
 - Hiệu suất cao: GSAP được thiết kế để hoạt động với hiệu suất cao, giúp cuộn trang mượt mà và giảm thời gian tải trang.

- Dễ sử dụng: GSAP có các hàm đơn giản và dễ dàng để sử dụng, cho phép người dùng tạo ra được các hiệu ứng động đẹp mắt một cách dễ dàng.
- Tính tương thích cao: GSAP tương thích với nhiều trình duyệt phổ biến như Google Chrome, Firefox, Safari, Opera, Microsoft Edge, vv.
- Các tính năng mạnh mẽ: GSAP cung cấp nhiều tính năng mạnh mẽ như TweenMax, TimelineMax, EasePack, và CSSPlugin để giúp người dùng tạo ra các hiệu ứng động phức tạp.
- Cộng đồng hỗ trợ: GSAP có một cộng đồng lớn và tích cực hỗ trợ trong việc giải đáp thắc mắc và chia sẻ kiến thức về sử dụng thư viện này.
- Các công cụ chính của GSAP:
 - TweenLite: 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: 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: 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: 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.
- CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.
- Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều

kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

- Mỗi tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.
- Bố cục CSS thường chủ yếu dựa vào hình hộp và mỗi hộp đều chiếm những khoảng trống trên trang của bạn với các thuộc tính như:
 - o Padding: Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
 - o Border: Là đường viền nằm ngay bên ngoài phần đệm.
 - o Margin: Là khoảng cách xung quanh bên ngoài của phần tử.
- Ưu điểm của CSS:
 - o CSS giúp người dùng nhiều styles trên một trang web HTML nên khả năng điều chỉnh trang của bạn trở nên vô hạn.
 - o 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
 - o 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

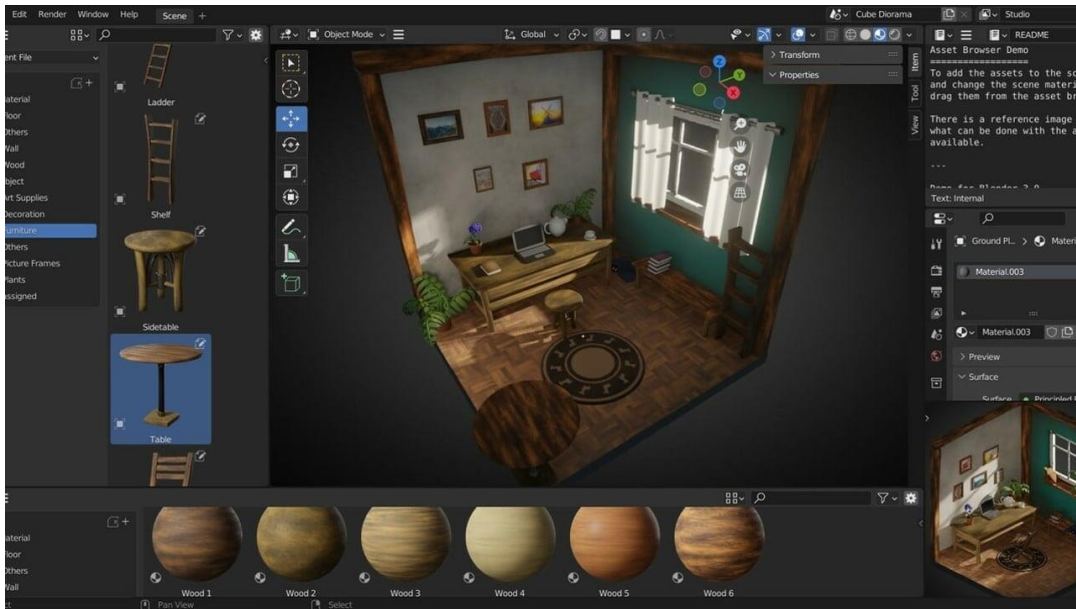
- HTML (Hypertext Markup Language) là ngôn ngữ được sử dụng rộng rãi trong thiết kế trang web. Hypertext là cách mà các trang web kết nối được với nhau, do đó mà các link trên trang cũng được gọi với cái tên này. HTML là ngôn ngữ đánh dấu (Markup Language) một tài liệu bằng các thẻ (tag) để nói với trình duyệt web cấu trúc hiển thị ra màn hình.
- HTML5 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 và nhược điểm của HTML5:

Ưu Điểm	Nhược điểm
HTML5 đa nền tảng và responsive..	Nó yêu cầu các trình duyệt hiện đại để sử dụng
Nó hỗ trợ âm thanh và video.	
Có nhiều layout elements hơn cho nội dung	Có nhiều chuẩn video khác nhau cho HTML5.
Nó hoạt động tốt với tính nhất quán tuyệt vời	
Cung cấp lợi ích tối ưu hóa công cụ tìm kiếm.	

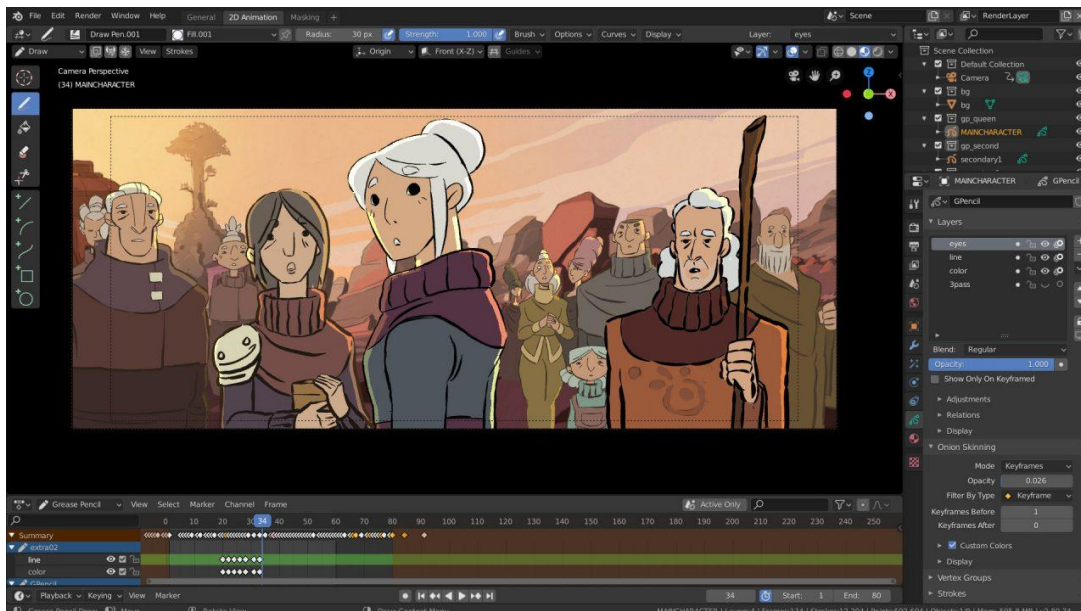
6) Blender

- Blender là một bộ phần mềm tạo 3D mã nguồn mở miễn phí hỗ trợ khá nhiều mọi khía cạnh của phát triển 3D. Với nền tảng vững chắc về khả năng tạo mô hình, còn có kết cấu mạnh mẽ, thiết bị, hoạt ảnh, ánh sáng và một loạt các công cụ khác để tạo 3D hoàn chỉnh. Phần mềm này rất tuyệt vời cho dù bạn chỉ muốn xử lý các mô hình tĩnh hay tham gia vào thế giới hoạt hình.
- Mặc dù miễn phí, Blender vẫn có thể truy cập và có giá trị đối với nhiều người dùng, từ những người mới bắt đầu sở thích đến những nhà làm phim hoạt hình chuyên nghiệp. Ngay cả NASA cũng sử dụng nó cho nhiều mô hình công cộng của mình! Vì nó liên tục được tinh chỉnh bởi những người dùng nâng cao, nó có thể trình bày một chút đường cong học tập cho những người hoàn toàn không chuyên.
- Ưu điểm của blender:
 - o Hỗ trợ nhiều loại file đa dạng và khác nhau.
 - o Siêu nhẹ và siêu nhanh khi Blender nhắc đến tốc độ load, ở đĩa chỉ tốn 250Mb.
 - o Miễn phí hoàn toàn phần mềm và nếu như biết code người sử dụng có thể chỉnh sửa tùy ý.
 - o 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.
- Tính năng của blender:
 - o Các công cụ trực quan, điêu khắc 3D: Tính năng điêu khắc sẽ giúp bạn tạo hình sản phẩm có độ chính xác và chi tiết cao, tương tự như sản phẩm của các nghệ sĩ điêu khắc. Bên cạnh đó, có thể chuyển đổi giữa các tính năng tạo hình và điêu khắc chỉ với một lệnh nhanh.
 - o Tạo hiệu ứng kỹ xảo chuyên nghiệp: Phần mềm còn giúp người dùng tạo hiệu ứng kỹ xảo một cách chuyên nghiệp. Các thao tác thực hiện vô cùng đơn giản, bạn sẽ nhập cảnh thô, theo dõi cảnh, tạo lớp phủ, theo dõi sự chuyển động hiệu ứng kỹ xảo 3D, sau đó thì chỉnh sửa để hình ảnh được chân thực hơn.
 - o Chỉnh sửa các đoạn video, vlog: Blender không chỉ là phần mềm thiết kế thông thường, phần mềm còn đem lại cho người dùng tính năng chỉnh sửa video, vlog vô cùng mạnh mẽ. Với tính năng này, người dùng có thể chỉnh sửa, cắt ghép và biên tập video một cách nhanh chóng.
 - o Kết cấu và mở gói UV: Tính năng mở gói UV sẽ cho phép bạn làm phẳng bề mặt của mô hình, và sau đó thì bạn có thể vẽ theo kết cấu của riêng mình. Tính năng còn hỗ trợ bạn điều chỉnh độ mờ, độ khuếch tán, phản xạ ánh sáng hoặc ánh sáng nền của sản phẩm.

- Grease Pencil: công cụ hỗ trợ vẽ 3D bằng bút 2D
- Các hiệu ứng mô phỏng vật lý
- Mô phỏng khói, áp vân bề mặt, hạt
- Công cụ render của phần mềm vô cùng mạnh mẽ sẽ hỗ trợ bạn xuất hình ảnh, video ở mọi độ phân giải
- Tập lệnh tích hợp sẽ cho phép người dùng định hình chương trình theo sở thích



Hình 2.1: Hình ảnh mô phỏng phòng với blender



Hình 2.2: Vẽ hình 2D trong chế độ 3D với blender

7) ThreeJS

- Three.js là thư viện JavaScript đa trình duyệt và giao diện lập trình ứng dụng (API) được sử dụng để tạo và hiển thị đồ họa máy tính 3D hoạt hình trong trình duyệt web bằng WebGL.
- 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.
- Three.js cho phép tạo hoạt ảnh 3D được tăng tốc bởi đơn vị xử lý đồ họa (GPU) bằng cách sử dụng ngôn ngữ JavaScript như một phần của trang web mà không cần phụ thuộc vào các plugin trình duyệt độc quyền. Điều này có thể thực hiện được nhờ sự ra đời của WebGL, một API đồ họa cấp thấp được tạo riêng cho web.
- Ưu điểm của ThreeJS:
 - o Dễ học
 - o Có nhiều ví dụ
 - o Cộng đồng hỗ trợ lớn, nhiều plugin từ bên thứ ba
 - o Tài liệu tham khảo dễ hiểu
 - o Hiệu suất tốt
 - o Hỗ trợ hầu hết cho tất cả các định dạng của model 3D
 - o Có trình soạn thảo riêng
- Nhược điểm của ThreeJS:
 - o Không có quy trình hiển thị bị trì hoãn, chỉ có chuyển tiếp. Điều này làm cho rất nhiều kỹ thuật kết xuất hiện đại không thể/không khả thi để triển khai với ThreeJS
 - o Nó công phải game engine. Nếu mà người dùng các tính năng ngoài render thì người dùng sẽ không tìm thấy nhiều ở thư viện này.
 - o Thư viện này hướng đối tượng là người mới -> các tính năng nâng cao sẽ bị ẩn đi.
- Có rất nhiều trang web bán hàng hiện nay cũng sử dụng ThreeJS để giúp cho các người bán hàng hiển thị các mặt hàng của họ cho người mua và giúp người mua có cái nhìn cụ thể hơn để đưa ra quyết định mua hay không.

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

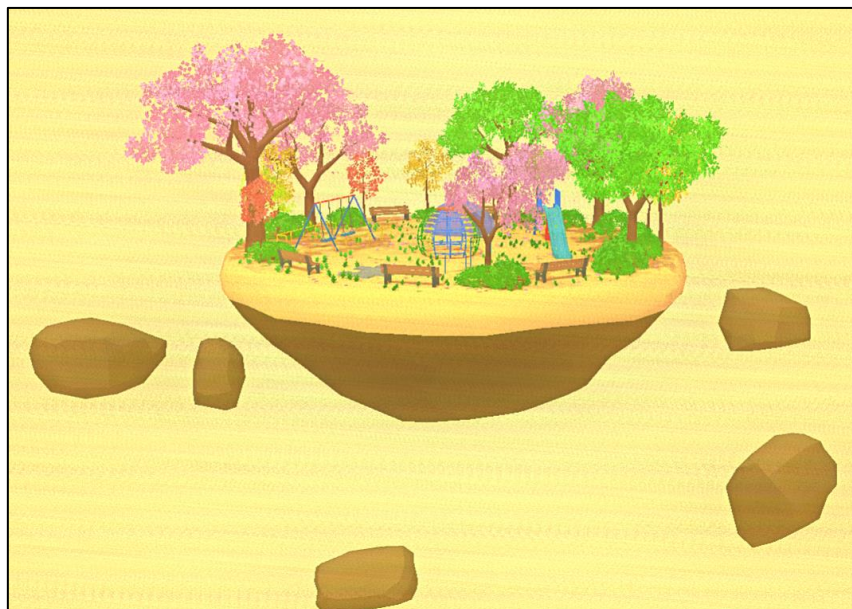
III) Thiết kế hệ thống

- Giới thiệu tổng quát về các công việc em làm để tạo ra đồ án.

1) Load các file lên website

1.1) Load model sân chơi

- Model sân chơi này do em dùng blender để vẽ và thiết kế nó sao cho nó đẹp nhất có thể và dùng thư viện load model 3D để load nó vào.
- Thế nhưng trong blender thì model em nhìn nó rất là đẹp thế nhưng khi đưa thành định dạng .glb upload lên thì màu sắc và kết cấu của model nó không còn được như trong blender.
- Để giải quyết vấn đề đó và cải thiện chất lượng của sân chơi thì trong ThreeJs cung cấp hàm **cubeTextureLoader**, em dùng hàm này để set cái environment map của **scene** trong ThreeJS cho tất cả các vật liệu (material) trong scene.
- Bằng việc sử dụng Draco khi load model lên website nó đã giúp cho website em nhẹ hơn và nhanh hơn đáng kể.



Hình 3.1: Model khi kết hợp cubeTextureLoader



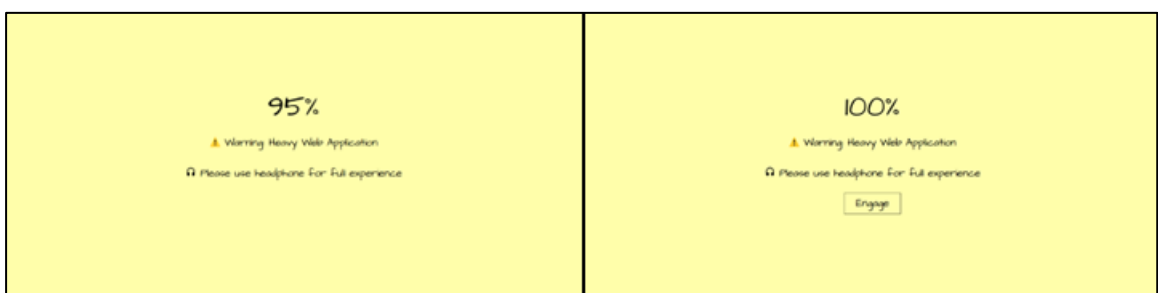
Hình 3.2: Model gốc

1.2) Load các file khác cho website

- Load background của scene(trong threeJS) để giúp cho nó nhìn giống mặt trời chiếu sáng để cho khi dùng các hiệu ứng ánh sáng trông giống thật hơn.
- Load các file âm thanh để tạo ra các âm thanh giúp cho người dùng cảm thấy thư thái và bớt nhàm chán.

2) Thiết kế màn hình loading

- Quá trình tải các tài nguyên cần thiết có thể sẽ nhanh hay chậm để không tạo ra sự nhàm chán thì mục đích của màn hình loading chính là cho người dùng biết được khi website sẵn sàng.
- Để theo dõi tiến trình tải các tài nguyên lên website hay chưa thì trong thư viện ThreeJS có một hàm hỗ trợ cho việc kiểm tra đó là LoadingManager.
- Sau khi quá trình load xong thì màn hình loading sẽ hiển thị nút “engage” sau khi người dùng nhấn nút đó thì sẽ đến giao diện chính của website.



Hình 3.3: Hình ảnh loading của website

3) Các đối tượng trong cảnh 3D

3.1) Góc nhìn

- Để mô phỏng góc nhìn thế giới thực thì sự lựa chọn của em chính là PerspectiveCamera. PerspectiveCamera sẽ mô phỏng hành động của một camera quay phim trong đời thực. Đối tượng càng xa Camera thì trông càng bé. Vị trí của camera và hướng của nó sẽ quyết định phần nào của khung cảnh được render trên màn hình
- Các thông số em truyền vào cho Perspective camera ở đây là do cảm tính (thấy vừa ý) và qua góc nhìn của em khi website hiển thị
 - o Field of view (FOV) - góc nhìn theo chiều dọc. Góc càng rộng thì chúng ta càng nhìn được nhiều hơn, các đối tượng càng nhỏ đi. Với mục setting này em cho thông số là 45
 - o aspect: tỷ lệ chiều ngang / chiều dọc của vùng mà chúng ta render đầu ra của cảnh. Tỷ lệ này quyết định sự khác nhau giữa FOV theo chiều ngang và FOV theo chiều dọc. Để hình ảnh không bị biến dạng, hãy để giá trị này bằng tỷ lệ chiều ngang / chiều dọc của canvas.
 - o near: mặt phẳng gần. Nếu đối tượng cách Camera nhỏ hơn near thì sẽ không nhìn thấy. Giá trị này em để 0.01
 - o far: mặt phẳng xa. Nếu đối tượng cách Camera lớn hơn far thì sẽ không nhìn thấy. Giá trị em cảm thấy phù hợp là 200
- Để giống thật hơn nữa thì em cũng sử dụng thêm hàm Frustum. Hàm này giúp xác định những gì bên trong vùng nhìn của camera -> Nhờ đó trong quá trình render – các đối tượng nằm ngoài phạm vi của camera sẽ không cần thiết phải render.

3.2) Ánh sáng

- Việc lựa chọn ánh sáng sao cho phù hợp với background cũng không hề được dễ dàng qua quá trình tìm hiểu về sự phối hợp ánh sáng ở diễn đàn của ThreeJS thì hầu như mọi người đều sử dụng 2 loại ánh sáng này kết hợp để phù hợp với sự chiếu sáng của ánh nắng mặt trời, đó là:
 - o Ambient Light là ánh sáng tổng thể bao phủ toàn bộ không gian để cung cấp mức chiếu sáng tối thiểu. Lớp ánh sáng này có thể đến từ ánh sáng tự nhiên qua cửa sổ lớn, mái lợp kính hay các nguồn chiếu sáng nhân tạo.
 - o Directional Light là nguồn sáng mà các tia sáng chiếu song song theo một chiều. Vì trong ThreeJs loại ánh sáng này có thể tạo ra

bóng của các loại đồ vật và nó có hướng tương tự với ánh sáng mặt trời kết hợp với dải màu phù hợp đã giúp em tạo ra ánh sáng mặt trời hợp với mình

4) Post Processing

- Quá trình hậu kỳ ở đây được hiểu trong đồ án là quá trình xử lý cho các hiệu ứng hậu kỳ hình ảnh nó giúp cho cải thiện tính thẩm mỹ hơn cho đồ án
- Bằng việc sử dụng các hàm post processing trong threejs giúp em tạo ra các hiệu ứng hậu kỳ đẹp mắt:
 - o Hàm RenderPass được sử dụng render cảnh chính (cảnh 3D).
 - o Hàm UnrealBloomPass được sử dụng để tạo ra hiệu ứng ánh sáng
 - o FilmPass được sử dụng để tạo ra hiệu ứng phim cho cảnh 3D
 - o EffectComposer để triển khai các hiệu ứng hậu xử lý ở trên. Lớp này quản lý một chuỗi các bước xử lý hậu kỳ để tạo ra kết quả trực quan cuối cùng. Các lượt xử lý hậu kỳ được thực hiện theo thứ tự bổ sung/chèn của chúng. Cuối cùng được tự động hiển thị trên màn hình.

5) Shader

- Để tạo ra các hiệu ứng shader cho đồ án em sử dụng Uniform, là một loại biến trong GLSL (ngôn ngữ shader sử dụng trong WebGL và Three.js) mà giá trị của nó được định nghĩa bên ngoài shader và được truyền vào shader từ chương trình JavaScript. Uniform có thể được sử dụng trong cả vertex shader và fragment shader và giá trị của chúng giống nhau cho tất cả các đỉnh và fragment được xử lý trong một lần gọi draw.
- Shader uniform trong đồ án em muốn áp dụng là cho các đối tượng trong đồ án là đảo nổi, bùm rậm, cỏ, lá (dùng cho mục đích làm hiệu ứng lá rụng)

Cách thức tạo ra shader

- Để tạo ra các hiệu ứng như thế đầu tiên thì ta cần phải khai báo đối tượng chứa các giá trị uniform cho shader. Khi tài nguyên của web đã được load xong thì hàm callback được gọi đến.
- Trong hàm callback, đoạn code khởi tạo một AnimationMixer cho đối tượng được tải và lấy danh sách các clip hoạt hình từ nó. Sau đó, nó sử dụng vòng lặp for để lặp qua các clip hoạt hình và sử dụng setTimeout để phát chúng với một khoảng thời gian trễ nhất định.
- Tiếp theo, tìm kiếm một số đối tượng con cụ thể trong cây đối tượng của đối tượng được tải và thực hiện một số thao tác trên chúng. Ví dụ:

nó thay đổi thuộc tính `onBeforeCompile` của vật liệu của các đối tượng `grassMaterial`, `bushMaterial`, `sakuraMaterial`, v.v. để thêm hiệu ứng động cho chúng bằng cách sử dụng hàm `vertexDisplacement`.

- Hàm `vertexDisplacement` nhận vào một shader, các thông số để điều chỉnh hiệu ứng và thực hiện thay thế mã shader để thêm uniform `uTime` và hai hàm `rotate` và `rotateMatrix`. Nó cũng thay thế mã trong phần `#include <begin_vertex>` để áp dụng hiệu ứng xoay cho các đỉnh của vật liệu.
- Cuối cùng, đối tượng được tải vào cảnh bằng cách sử dụng phương thức `add` của đối tượng `scene`.

6) Hiệu ứng camera cinematic

- Hiệu ứng camera ở đây là em tự định nghĩa, có nghĩa là hiệu ứng khi chuyển đổi qua lại các vị trí trong sân chơi nhằm mục đích cho không nhàm chán và đẹp mắt được áp dụng trong chế độ “Play” của website.
- Bằng việc sử dụng GSAP (GreenSock Animation Platform) để tạo ra các hiệu ứng chuyển động cho camera.

Cách thức hoạt động:

- GSAP được sử dụng để tạo ra các hiệu ứng chuyển động cho vị trí của camera bằng cách sử dụng phương thức `fromTo` của đối tượng `gsap`. Phương thức này cho phép bạn chỉ định giá trị ban đầu và giá trị cuối cùng cho một thuộc tính của một đối tượng và thời gian chuyển động. Ví dụ: `gsap.fromTo(camera.position, {x: -50, y: 17, z: 82}, {x: -15, y: 8, z: 21, duration: 10})` sẽ tạo ra một hiệu ứng chuyển động cho vị trí của camera từ (-50, 17, 82) đến (-15, 8, 21) trong 10 giây.
- Bằng việc sử dụng phương thức `setTimeout` để lên lịch thực hiện các hành động tại các thời điểm nhất định. Phương thức `setTimeout` cho phép chỉ định một hàm callback và một khoảng thời gian trễ (tính bằng `milisecond`) và sẽ gọi hàm callback sau khi khoảng thời gian trễ đã trôi qua. Ví dụ: `setTimeout(() => {}, 1000)` sẽ lên lịch gọi hàm callback sau 1000 `milisecond` (tức là 1 giây).
- Phương thức `setTimeout` được sử dụng để lên lịch thực hiện các hành động như thay đổi vị trí của camera và điểm nhìn của nó (`point`), phát các hiệu ứng chuyển tiếp (`cinTween[1]` và `cinTween[2]`) và thêm các hiệu ứng chuyển động cho camera bằng cách sử dụng GSAP (`cinTween[0]`, `cinTween[3]`, v.v.).

7) Vị trí đặc biệt trong mô hình

- Dùng cho chế độ “explore” giúp cho người tham quan có thể dễ dàng đi đến những chỗ em coi là đẹp và có góc nhìn tuyệt vời. Ở đây thì em cung cấp 2 chức năng là ẩn các điểm đó hoặc hiện các điểm đó.

Cách thức hoạt động:

- Tạo mảng points chứa các đối tượng mô tả các điểm quan tâm. Mỗi đối tượng trong mảng này có hai thuộc tính: position và element. Thuộc tính position là một đối tượng THREE.Vector3 mô tả vị trí của điểm quan tâm trong không gian 3D. Thuộc tính element là một phần tử HTML được lấy từ trang bằng cách sử dụng phương thức querySelector.
- Hàm hideAllPOI sử dụng phương thức forEach của mảng để lặp qua tất cả các điểm quan tâm trong mảng points và gỡ bỏ class visible khỏi phần tử HTML của chúng bằng cách sử dụng phương thức classList.remove. Điều này sẽ ẩn tất cả các điểm quan tâm.
- Hàm showAllPOI cũng sử dụng phương thức forEach của mảng để lặp qua tất cả các điểm quan tâm trong mảng points và thêm class visible vào phần tử HTML của chúng bằng cách sử dụng phương thức classList.add. Điều này sẽ hiển thị tất cả các điểm quan tâm.



Hình 3.4: Các interest point khi hiện



Hình 3.5: Các interest point khi ẩn

8) Chức năng điều khiển

- Bằng việc sử dụng hàm OrbitControls trong thư viện threejs nó giúp cho 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 tiện lợi cho việc xem các cảnh vật xung quanh cái sân chơi của em

9) Các chức năng khác

- Ngoài các chức năng chính như ở trên , thì để cho đồ án hấp dẫn hơn thì em cũng đã thiết kế các chế độ khác nhau:
 - Play: Ở chế độ này người dùng sẽ được đi một vòng những điểm mà em cảm thấy đẹp và thú vị trong đồ án
 - Explore: Ở chế độ này người dùng sẽ tự do khám phá tất cả nơi trên sân chơi của em. Họ có thể điều chỉnh góc nhìn vị trí sao cho phù hợp với sở thích cá nhân.
- Thêm vào đó , em cũng thiết kế thêm các cho màn hình chính VD: About, Hướng dẫn , các nút chức năng để giúp cho website trở nên thật là hoàn chỉnh

IV) KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1) Hạn chế

- Vì đây là đồ án em làm 1 mình nên không thể hạn chế 1 số sai sót:
 - Do chỉ tập trung quá nhiều vào cảnh vật nên quên mất là thiếu bóng dáng của con người.
 - Các hiệu ứng animation của lá rơi thì còn thiếu bóng của nó khi ánh sáng mặt trời chiếu vào.

2) Hướng phát triển

- Sẽ trau dồi thêm kiến thức để có thể phát triển đồ án hơn nữa
- Phát triển thêm các chức năng mà còn thiếu ở phần hạn chế
- Và phát triển thêm các chức năng khác:
 - Có thể điều khiển để đi bộ trong khu vui chơi.
 - Tương tác với cảnh vật trong khu vui chơi.
 - Thời gian thay đổi (Có buổi sáng và buổi tối).
 - Thêm nhiều cảnh vật hơn.

V) TÀI LIỆU THAM KHẢO

- [1] Thư viện ThreeJS: <https://threejs.org/>
- [2] Khóa học threejs journey của Bruno Simon : <https://threejs-journey.com/>
- [3] Diễn đàn Threejs: <https://discourse.threejs.org/>
- [4] GSAP Animation Tutorial | Create Awesome Animations With Javascript:
<https://youtu.be/5RyrIPCs47A>
- [5] Three.js Shaders (GLSL) Crash Course For Absolute Beginners:
<https://youtu.be/oKbCaj1J6EI>
- [6] Blender 3.0 Basics Course:
<https://youtube.com/playlist?list=PL3GeP3YLZn5ixsnIOIx9tB4v6s-rsw48X>
- [7] Shader Tutorial Series:
<https://youtube.com/playlist?list=PL4neAtv21W0mIrTrkNO3xCyrxg4LKkrF7>
- [8] 3D model: <https://free3d.com/3d-models/obj>
- [9] HTML document: https://www.w3schools.com/jsref/dom_obj_document.asp
- [10] CSS document: <https://www.w3schools.com/css/default.asp>
- [11] WebGL tutorial: <https://www.tutorialspoint.com/webgl/index.htm>
- [12] Webpack document: <https://devdocs.io/webpack~5/>