

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
ĐỒ HỌA MÁY TÍNH

ĐỀ TÀI :

Chuyển đổi tọa độ 2D từ màn hình máy tính sang tọa độ thực

HỌ TÊN SINH VIÊN	MÃ SINH VIÊN	NHÓM HP
Trần Đức Trí	102210096	21Nh15
Lê Anh Tuấn	102210097	21Nh15
Đặng Hoàng Nguyên	102210094	21Nh15
Phạm Nguyễn Anh Phát	102210302	21Nh15

CBHD : Nguyễn Tấn Khôi

Đà Nẵng, ngày 7 tháng 5 năm 2023

Contents

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	2
1.1. CƠ SỞ LÝ THUYẾT	2
1.2. PHÁT BIỂU BÀI TOÁN	2
1.3. KẾT CHƯƠNG	2
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	3
2.1. PHÁT BIỂU BÀI TOÁN	3
2.2. PHÂN TÍCH HIỆN TRẠNG.....	3
2.3. PHÂN TÍCH CHỨC NĂNG.....	3
2.3.1. Đối tượng sử dụng	3
2.3.2. Chuyển đổi tọa độ thực sang tọa độ trên màn hình máy tính	4
2.3.3. Công nghệ sử dụng	5
2.4. TỔ CHỨC CHƯƠNG TRÌNH.....	5
2.4.1. Tổ chức thư mục	5
2.4.2. Index.html	6
2.4.3. Package.json	6
2.5. KẾT CHƯƠNG	6
CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	7
3.1. MÔ HÌNH TRIỂN KHAI	7
3.1.1. Mô hình triển khai	7
3.1.2. Các công cụ sử dụng.....	7
3.1.3. Cấu hình hệ thống.....	8
3.2. KẾT QUẢ THỰC NGHIỆM.....	8
3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ	10
3.4. KẾT CHƯƠNG	11
1. KẾT QUẢ ĐẠT ĐƯỢC	12
2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN	12

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
JS	Ngôn ngữ Javascript
NPM	Node Package Manager
API	Application Programming Interface
3D	3 dimensions
2D	2 dimensions

MỞ ĐẦU

1. Tổng quan về đề tài

- Đề tài dựa trên kiến thức từ môn Đồ Họa Máy Tính, áp dụng module “projection-3d-2d” để thực hiện biến đổi tọa độ pixel hình ảnh 2D khuôn viên trường đại học Bách Khoa sang tọa độ thực.

2. Mục đích và ý nghĩa của đề tài

2.1. Mục đích

- Mục đích của project này là giúp người dùng có thể hiểu rõ hơn về cách thức hoạt động của phép chiếu trong đồ họa 3D. Nó cũng cho phép người dùng thực hiện thử nghiệm và tìm hiểu cách hoạt động của phép chiếu phối cảnh trên các vật thể 3D đơn giản như hình cầu, hình trụ và hộp.

2.2. Ý nghĩa

- Ý nghĩa của project này là cung cấp cho người dùng một công cụ hữu ích để nghiên cứu và tìm hiểu về phép chiếu trong đồ họa 3D. Nó cũng có thể được sử dụng để giảng dạy hoặc giới thiệu về đồ họa 3D và các kỹ thuật liên quan đến phép chiếu. Ngoài ra, project còn mang tính ứng dụng cao khi cho phép người dùng chuyển đổi giữa vật thể 3D và vật thể 2D, giúp họ dễ dàng kiểm tra và sửa chữa các lỗi trong quá trình phát triển ứng dụng đồ họa 3D.

3. Bố cục của đồ án

Đồ 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

Chương này trình bày các nội dung cơ sở lý thuyết chính liên quan đến nội dung của đồ án. Nội dung cơ sở lý thuyết này sẽ được sử dụng trong phân phân tích và triển khai chương trình. Phần phát biểu bài toán sẽ mô tả nội dung và các vấn đề đặt ra. Ngoài ra cũng liệt kê và đánh giá các giải pháp đã có.

1.1. CƠ SỞ LÝ THUYẾT

- Project về phép chiếu 3D sang 2D sử dụng công nghệ **projection-3d-2d** được xây dựng dựa trên các cơ sở lý thuyết về đại số tuyến tính và hình học trong không gian 3 chiều (3D).
- Perspective Projection là một phép chiếu phi tuyến trong đó các đối tượng ba chiều được chiếu trên một *mặt phẳng hình ảnh 2D*.
- Perspective Projection, trong khi đó, tạo ra các hình ảnh có chiều sâu và giúp ta có cảm giác như đang nhìn vào một cảnh 3D thực sự. Trong phép chiếu Perspective Projection, các đối tượng được giảm kích thước theo tỉ lệ đến mặt phẳng chiếu theo quy luật "giảm dần theo khoảng cách". Tức là, các đối tượng nằm gần mặt phẳng chiếu sẽ lớn hơn các đối tượng nằm xa hơn. Phép chiếu này được sử dụng phổ biến trong các ứng dụng thực tế như trong các trò chơi 3D, đồ họa 3D, trò chơi điện tử, các ứng dụng thực tế ảo và thiết kế kiến trúc để hiển thị các đối tượng ba chiều trên màn hình hai chiều

1.2. PHÁT BIỂU BÀI TOÁN

- Bài toán của project trên là chuyển đổi từ ảnh 2D sang vật thể Tọa độ 2D thông qua phép chiếu phối cảnh.

1.3. KẾT CHƯƠNG

Chương này trình bày cơ sở lý thuyết về phép chiếu phối cảnh(Perspective Projection) 3D sang 2D.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương này trình bày về phân tích và triển khai hệ thống, bao gồm phân tích hiện trạng, phân tích yêu cầu của bài toán.

2.1. PHÁT BIỂU BÀI TOÁN

- Bài toán được giới thiệu trong trang npmjs của package "projection-3d-2d" là việc chuyển đổi từ tọa độ 3D của một đối tượng trong không gian sang tọa độ 2D trên một mặt phẳng, phù hợp để hiển thị trên các màn hình hiển thị 2D.

- Để giải quyết bài toán này, package "projection-3d-2d" sử dụng phương pháp chiếu (projection) 3D đến mặt phẳng 2D bằng các phép biến đổi hình học, bao gồm di chuyển (translation), quay (rotation) và thu phóng (scaling).

- Input : Góc tọa độ theo pixel, theo độ dài thực của vật trên ảnh. Con trỏ chuột tại một vị trí bất kì trên ảnh.
- Output: Đầu ra là tọa độ thực của một điểm bên cạnh con trỏ chuột.

2.2. PHÂN TÍCH HIỆN TRẠNG

- Hiện tại, việc thực hiện phép chiếu 3D sang 2D trong lập trình đồ họa là một nhu cầu cơ bản của nhiều ứng dụng, đặc biệt là trong lĩnh vực trực quan hóa dữ liệu và đồ họa 3D. Tuy nhiên, để thực hiện phép chiếu này đòi hỏi kiến thức về đại số tuyến tính và đồ họa máy tính, và yêu cầu các lập trình viên phải có kinh nghiệm và năng lực về lập trình đồ họa.

- Hiện tại, có nhiều thư viện đồ họa cung cấp các hàm để thực hiện phép chiếu 3D sang 2D, như OpenGL, DirectX, Three.js, Babylon.js, hay Processing. Tuy nhiên, đa phần các thư viện này đều có cấu trúc phức tạp và đòi hỏi người sử dụng phải có kiến thức sâu về đồ họa máy tính. Ngoài ra, việc tìm hiểu và sử dụng các thư viện này cũng đòi hỏi thời gian và công sức đáng kể.

- Trong bối cảnh đó, project Projection-3d-2d cung cấp một giải pháp đơn giản và dễ dàng để thực hiện phép chiếu 3D sang 2D trong lập trình đồ họa. Với những người mới bắt đầu học lập trình đồ họa hoặc muốn thực hiện phép chiếu một cách nhanh chóng và hiệu quả, thư viện này là một lựa chọn tốt. Nó cũng giúp cho việc học tập và nghiên cứu đồ họa máy tính trở nên dễ dàng và thú vị hơn.

2.3. PHÂN TÍCH CHỨC NĂNG

2.3.1. Đối tượng sử dụng

- Đối tượng sử dụng có thể là các nhà phát triển ứng dụng đồ họa, các nhà nghiên cứu khoa học, giảng viên và sinh viên các trường đại học chuyên

ngành đồ họa, hoặc bất kỳ ai có nhu cầu trong việc hiển thị và xử lý dữ liệu 3D.

2.3.2. Chuyển đổi tọa độ thực sang tọa độ trên màn hình máy tính

- `project()`: Chức năng này cho phép thực hiện phép chiếu 3D sang 2D trên một đối tượng 3D. Đối tượng 3D được truyền vào dưới dạng một mảng các tọa độ 3D. Kết quả trả về là một mảng các tọa độ 2D tương ứng với các tọa độ 3D đã cho.

- `projectX()`, `projectY()`, `projectZ()`: Các chức năng này thực hiện phép chiếu 3D sang trục x, y hoặc z tương ứng. Đối tượng 3D được truyền vào dưới dạng một mảng các tọa độ 3D. Kết quả trả về là một mảng các tọa độ 2D tương ứng với các tọa độ 3D đã cho trên trục x, y hoặc z.

- `orthogonalProjection()`: Chức năng này thực hiện phép chiếu Orthographic Projection trên một đối tượng 3D. Đối tượng 3D được truyền vào dưới dạng một mảng các tọa độ 3D. Kết quả trả về là một mảng các tọa độ 2D tương ứng với các tọa độ 3D đã cho.

- `perspectiveProjection()`: Chức năng này thực hiện phép chiếu Perspective Projection trên một đối tượng 3D. Đối tượng 3D được truyền vào dưới dạng một mảng các tọa độ 3D. Kết quả trả về là một mảng các tọa độ 2D tương ứng với các tọa độ 3D đã cho.

- Hàm `translate()` để di chuyển vật thể : Hàm này có một tham số đầu vào là một vector ba chiều, biểu thị khoảng cách mà vật thể sẽ di chuyển theo các trục x, y, z. Hàm này thực hiện việc tính toán ma trận dịch chuyển tương ứng và nhân nó với ma trận chiếu để tính toán ra ma trận chiếu mới.

- Hàm `rotateX()`, `rotateY()` và `rotateZ()` để xoay vật thể quanh các trục x, y, z. Mỗi hàm này có một tham số đầu vào là một góc quay theo độ. Hàm này thực hiện việc tính toán ma trận xoay tương ứng và nhân nó với ma trận chiếu để tính toán ra ma trận chiếu mới.

- Hàm `setScale()` để thay đổi tỉ lệ co giãn của vật thể. Hàm này có một tham số đầu vào là một vector ba chiều, biểu thị tỉ lệ co giãn theo các trục x, y, z. Hàm này thực hiện việc tính toán ma trận tỉ lệ tương ứng và nhân nó với ma trận chiếu để tính toán ra ma trận chiếu mới.

- Hàm `setCameraPosition()` để thiết lập vị trí camera. Hàm này có ba tham số đầu vào lần lượt là vị trí của camera theo các trục x, y, z. Hàm này thực hiện

việc tính toán ma trận view tương ứng và sử dụng nó để cập nhật ma trận chiếu.

- Hàm `setCameraTarget()` để thiết lập mục tiêu của camera. Hàm này có ba tham số đầu vào lần lượt là vị trí của mục tiêu theo các trục x, y, z. Hàm này thực hiện việc tính toán ma trận view tương ứng và sử dụng nó để cập nhật ma trận chiếu.

2.3.3. Công nghệ sử dụng

- Công nghệ sử dụng trong package "projection-3d-2d" là WebGL, một công nghệ đồ họa trên web cho phép xử lý đồ họa 3D trong trình duyệt mà không cần phải sử dụng các plugin bổ sung.

- Package này cung cấp một API đơn giản để thực hiện phép chiếu 3D sang 2D, và được sử dụng bằng cách tạo một đối tượng `Projection` từ module. Đối tượng này chứa các phương thức để thực hiện phép chiếu và cung cấp các tham số cần thiết để điều chỉnh phép chiếu.

- OpenGL: Là một công nghệ đồ họa 3D phổ biến được sử dụng trong nhiều ứng dụng đồ họa và trò chơi. OpenGL hỗ trợ nhiều tính năng đồ họa 3D, bao gồm phép chiếu 3D sang 2D. OpenGL được sử dụng chủ yếu trong các ứng dụng desktop, nhưng cũng có thể được sử dụng trong các ứng dụng web thông qua WebGL.

- Three.js: Là một thư viện JavaScript đồ họa 3D phổ biến được sử dụng để tạo các ứng dụng đồ họa và trò chơi trên web. Three.js hỗ trợ nhiều tính năng đồ họa 3D, bao gồm phép chiếu 3D sang 2D. Thư viện này cung cấp các lớp đối tượng đơn giản để tạo và kiểm soát các đối tượng 3D, và cho phép tùy chỉnh các tham số của phép chiếu 3D sang 2D.

- Babylon.js: Là một thư viện JavaScript đồ họa 3D khác được sử dụng để tạo các ứng dụng đồ họa và trò chơi trên web. Babylon.js hỗ trợ nhiều tính năng đồ họa 3D, bao gồm phép chiếu 3D sang 2D. Thư viện này cung cấp một API đơn giản để tạo và kiểm soát các đối tượng 3D, và cho phép tùy chỉnh các tham số của phép chiếu 3D sang 2D.

2.4. TỔ CHỨC CHƯƠNG TRÌNH

2.4.1. Tổ chức thư mục

- Thư mục `src`: Chứa các file mã nguồn JavaScript để triển khai phép chiếu 2D sang 2D.

- Thư mục `dist`: Chứa các file đầu ra sau khi build project, bao gồm các file JavaScript và CSS để sử dụng trong ứng dụng.

2.4.2. Index.html

/ projection-3d-2d /		
dist/	folder	80.1 kB
test/	folder	2.57 kB
LICENSE	text/plain	1.06 kB
README.md	text/markdown	6.94 kB
index.js	application/javascript	7.02 kB
main.d.ts	application/typescript	1.11 kB
package.json	application/json	1.22 kB
webpack.config.js	application/javascript	473 B

- Là file HTML chính của ứng dụng, chứa các thành phần giao diện để hiển thị phép chiếu 3D sang 2D.

2.4.3. Package.json

- Là file cấu hình cho project, chứa các thông tin về project, các dependencies và scripts để chạy và build project.

2.5. KẾT CHƯƠng

- Tổng quan về phân tích hệ thống của dự án Projection-3D-2D, chúng ta đã được thấy rõ cấu trúc tổng quan của dự án, từ đó hiểu được các chức năng và tính năng của từng thành phần trong dự án.
- Dự án Projection-3D-2D có cấu trúc rõ ràng và dễ hiểu, với hai phần chính là "src" và "test". Phần "src" chứa tất cả các module và các chức năng chính của dự án, bao gồm phần chuyển đổi tọa độ, phép chiếu và các thuật toán toán học khác để tính toán các giá trị cần thiết.
- Phần "test" cung cấp các ví dụ minh họa về cách sử dụng các chức năng của dự án, giúp người dùng hiểu rõ hơn về cách sử dụng các tính năng của dự án.
- Từ phân tích hệ thống trên, chúng ta có thể thấy rằng dự án Projection-3D-2D là một công cụ hữu ích trong việc chuyển đổi các tọa độ và tính toán các giá trị cần thiết cho việc vẽ đồ họa 2D từ các dữ liệu 3D. Dự án có tính linh hoạt và dễ sử dụng, giúp giảm thiểu thời gian và công sức trong việc tính toán các giá trị phức tạp.

CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Chương này trình bày về cách triển khai hệ thống, và đưa ra các đánh giá độ chính xác dựa trên kết quả thực thi...

3.1. MÔ HÌNH TRIỂN KHAI

3.1.1. Mô hình triển khai

- Mô hình triển khai:

- Các thuật toán chiếu: Module này cung cấp một tập hợp các thuật toán chiếu khác nhau để chuyển đổi tọa độ từ không gian 3 chiều sang không gian 2 chiều. Các thuật toán này bao gồm Perspective Projection, Orthographic Projection, và các phương pháp cắt góc (clipping).
- Mô hình tọa độ: Module này sử dụng mô hình tọa độ phổ biến trong đồ họa máy tính, gồm tọa độ XYZ trong không gian 3 chiều và tọa độ xy trong không gian 2 chiều.
- Thư viện toán học: Module này sử dụng các thư viện toán học phổ biến như Vector.js để thực hiện các tính toán liên quan đến chuyển đổi tọa độ.
- API: Module này cung cấp một API đơn giản cho người dùng để sử dụng các chức năng của nó. API này bao gồm các hàm chuyển đổi tọa độ 3D sang tọa độ 2D, và ngược lại.

3.1.2. Các công cụ sử dụng

- Công cụ sử dụng:

- Node.js: Để sử dụng module này, bạn cần có Node.js cài đặt trên máy tính của mình. Node.js là một nền tảng phát triển ứng dụng được xây dựng trên Chrome's JavaScript runtime, cho phép bạn chạy JavaScript ngoài trình duyệt.
- NPM: "projection-3d-2d" module có sẵn trên npm (Node Package Manager), cho phép bạn cài đặt module này và các dependencies của nó trong dự án của mình.
- Một trình biên dịch mã JavaScript: Bạn cần một trình biên dịch mã JavaScript để viết và chạy code của mình. Các trình biên dịch phổ biến bao gồm Visual Studio Code, Atom, Sublime Text, và Notepad++.
- Thư viện toán học: Module này sử dụng các thư viện toán học phổ biến như Vector.js để thực hiện các tính toán liên quan đến chuyển đổi tọa độ. Vì vậy,

bạn cần phải có kiến thức cơ bản về toán học và sử dụng các thư viện này để sử dụng module này một cách hiệu quả.

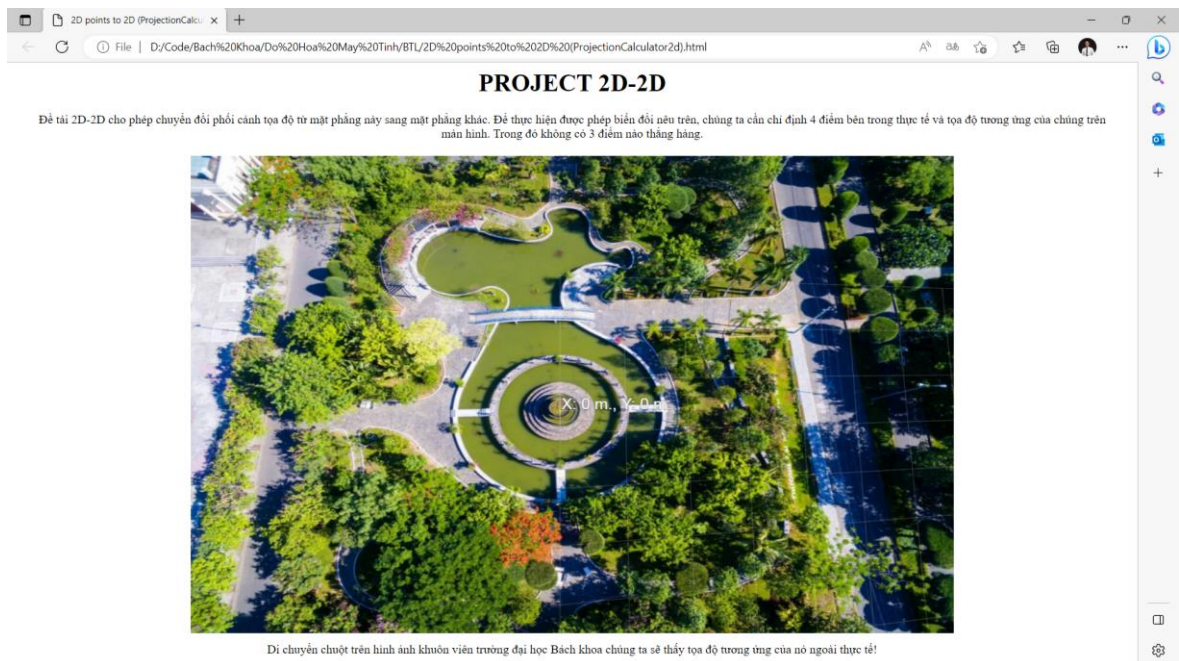
3.1.3. Cấu hình hệ thống

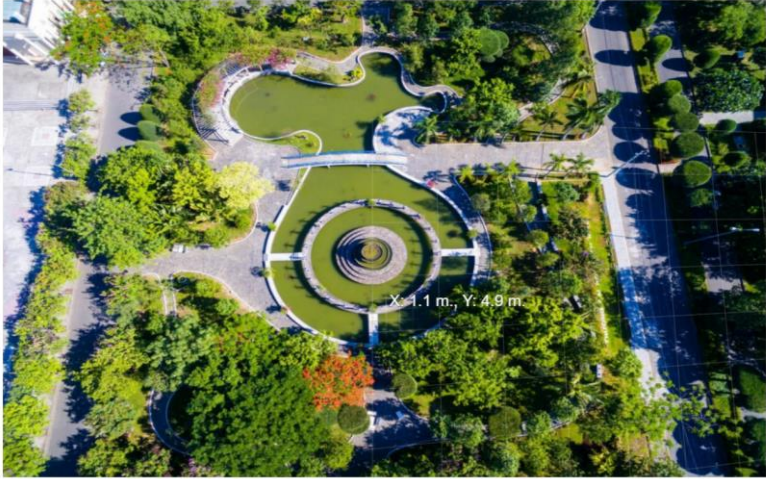
- Cấu hình hệ thống yêu cầu:

- Hệ điều hành: "projection-3d-2d" module hỗ trợ trên nhiều hệ điều hành như Windows, macOS, và Linux.
- Node.js: Để sử dụng module này, bạn cần cài đặt Node.js phiên bản 12 trở lên trên máy tính của mình. Node.js là một nền tảng phát triển ứng dụng được xây dựng trên Chrome's JavaScript runtime, cho phép bạn chạy JavaScript ngoài trình duyệt.

3.2. KẾT QUẢ THỰC NGHIỆM

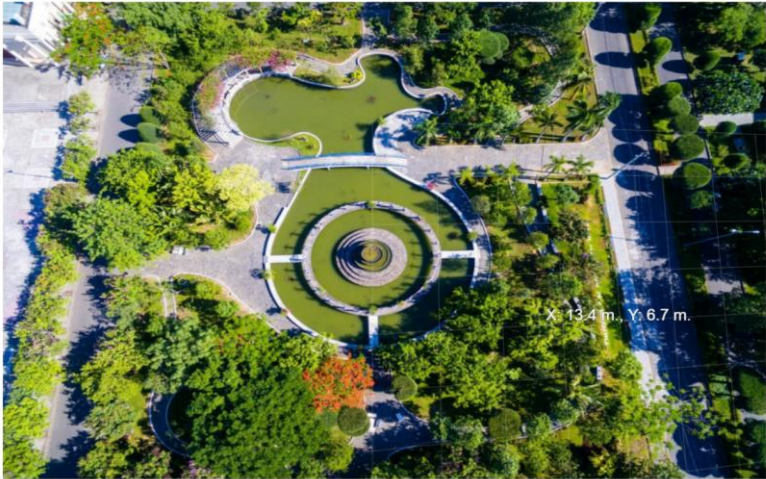
- Kết quả tọa độ thực trả về khi di chuyển chuột trên hình ảnh khuôn viên trường đại học Bách khoa:



2D points to 2D (ProjectionCalc...
+
File | D:/Code/Bach%20Khoa/Dor%20Hoa%20May%20Tinh/BTL/2D%20points%20to%202D%20(ProjectionCalculator2d).html
PROJECT 2D-2D
Đề tài 2D-2D cho phép chuyển đổi phối cảnh tọa độ từ mặt phẳng này sang mặt phẳng khác. Để thực hiện được phép biến đổi nêu trên, chúng ta cần chỉ định 4 điểm bên trong thực tế và tọa độ tương ứng của chúng trên màn hình. Trong đó không có 3 điểm nào thẳng hàng.


X: 4.1 m., Y: 4.9 m.

Di chuyển chuột trên hình ảnh khảm viên trường đại học Bách khoa chúng ta sẽ thấy tọa độ tương ứng của nó ngoài thực tế!

2D points to 2D (ProjectionCalc...
+
File | D:/Code/Bach%20Khoa/Dor%20Hoa%20May%20Tinh/BTL/2D%20points%20to%202D%20(ProjectionCalculator2d).html
PROJECT 2D-2D
Đề tài 2D-2D cho phép chuyển đổi phối cảnh tọa độ từ mặt phẳng này sang mặt phẳng khác. Để thực hiện được phép biến đổi nêu trên, chúng ta cần chỉ định 4 điểm bên trong thực tế và tọa độ tương ứng của chúng trên màn hình. Trong đó không có 3 điểm nào thẳng hàng.


X: 13.4 m., Y: 6.7 m.

Di chuyển chuột trên hình ảnh khảm viên trường đại học Bách khoa chúng ta sẽ thấy tọa độ tương ứng của nó ngoài thực tế!



3.3. NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

- Qua kết quả thực nghiệm, chúng em có những nhận xét như sau:

- "projection-3d-2d" là một module Node.js cho phép chuyển đổi tọa độ 3D sang tọa độ 2D và ngược lại bằng các phép chiếu perspective hoặc orthographic. Module cung cấp các hàm để tính toán khoảng cách giữa hai tọa độ 3D và chuyển đổi tọa độ 2D sang tọa độ 3D.
- Module này có điểm mạnh là cung cấp các hàm chuyển đổi tọa độ rất dễ sử dụng, người dùng chỉ cần cung cấp các tham số đầu vào cho các hàm để nhận được kết quả chuyển đổi. Ngoài ra, module này còn hỗ trợ cho các phép

chiều perspective và orthographic, giúp người dùng linh hoạt trong việc chọn lựa phương pháp chuyển đổi phù hợp với nhu cầu của mình.

- Tuy nhiên, module này có một số hạn chế như:

- Vẫn còn tương đối nhiều sai số trong việc xác định tọa độ thực và tọa độ tương ứng pixel trên màn hình máy tính.
- Chỉ hỗ trợ chuyển đổi tọa độ 3D sang tọa độ 2D và ngược lại, không hỗ trợ các chức năng phức tạp hơn như vẽ đối tượng 3D hoặc tạo ra các đối tượng 3D từ các tọa độ 2D.
- Không hỗ trợ các tính năng liên quan đến hiệu suất, do đó nếu áp dụng cho các ứng dụng có khối lượng dữ liệu lớn, module có thể không đáp ứng được yêu cầu về tốc độ xử lý.
- Không cập nhật thường xuyên, do đó module có thể không hỗ trợ các tính năng mới nhất hoặc sửa lỗi gần đây.

3.4. KẾT CHƯƠng

- Trong chương này, chúng ta đã tìm hiểu về module "projection-3d-2d" trong Node.js. Chúng ta đã xem xét cách mà module cung cấp các hàm để chuyển đổi tọa độ giữa không gian 3 chiều và 2 chiều bằng các phép chiếu perspective hoặc orthographic. Chúng ta cũng đã đánh giá các ưu và nhược điểm của module này.

- Tổng quan, module "projection-3d-2d" là một công cụ hữu ích cho việc chuyển đổi tọa độ trong các ứng dụng Node.js. Nó cung cấp các hàm dễ sử dụng và hỗ trợ cho các phép chiếu perspective và orthographic. Tuy nhiên, module này có những hạn chế như chỉ hỗ trợ các chức năng đơn giản, không hỗ trợ hiệu suất và không được cập nhật thường xuyên.

- Đối với một ứng dụng đơn giản, module "projection-3d-2d" có thể đáp ứng được yêu cầu. Tuy nhiên, đối với các ứng dụng phức tạp hơn về mặt đồ họa và hiệu suất, chúng ta nên xem xét các công cụ khác phù hợp hơn.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

- Trong quá trình tìm hiểu và sử dụng module "projection-3d-2d" trong Node.js, chúng ta đã có được các kết quả như sau:

- Hiểu được cách sử dụng module "projection-3d-2d" để chuyển đổi tọa độ giữa không gian 3 chiều và 2 chiều bằng các phép chiếu perspective hoặc orthographic.
- Nắm được các hàm cơ bản của module để tính toán khoảng cách giữa hai tọa độ 3D và chuyển đổi tọa độ 2D sang tọa độ 3D.
- Đánh giá được các ưu và nhược điểm của module "projection-3d-2d" như hỗ trợ dễ sử dụng và linh hoạt, tuy nhiên không hỗ trợ các tính năng phức tạp hơn và không có tính năng liên quan đến hiệu suất.

- Dựa trên các kết quả này, chúng ta có thể áp dụng module "projection-3d-2d" vào các ứng dụng đơn giản để chuyển đổi tọa độ giữa không gian 3 chiều và 2 chiều một cách dễ dàng. Tuy nhiên, đối với các ứng dụng phức tạp hơn về mặt đồ họa và hiệu suất, chúng ta nên xem xét các công cụ khác phù hợp hơn.

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Một số hướng nghiên cứu và phát triển của đề tài như sau:

- Bổ sung và hoàn thiện một số chức năng của hệ thống ...
- Đánh giá hiệu năng trên các môi trường khác nhau ...
- Kiểm thử các chức năng của chương trình ...
- Cải thiện độ chính xác của hệ thống

TÀI LIỆU THAM KHẢO

- [1] https://www.npmjs.com/package/projection-3d-2d?fbclid=IwAR1yNSu7_yzKZlhufAQl7Sii1cibRN2-3l8HlCx9H1e862O7y4Img16XGGE
- [2] https://en.wikipedia.org/wiki/3D_projection?fbclid=IwAR0UBovHhKQr1VoKq8-FqNKUcqjQytUPG7NTvAhIuu8m8JN9zW5JldrC6Q#Perspective_projection

PHỤ LỤC

1. Nội dung mã nguồn 01: https://www.npmjs.com/package/projection-3d-2d?fbclid=IwAR1yNSu7_yzKZlhufAQI7Sii1cibRN23l8HlCx9H1e862O7y4Img16XGGE
2. Nội dung mã nguồn 02: https://github.com/tranductri2003/2D-2D-PROJECTION_DUT