

KHOA CÔNG NGHỆ THÔNG TIN &TRUYỀN THÔNG

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

**NIÊN LUẬN CƠ SỞ NGÀNH KTPM**

**MÔ PHỎNG NHIỆT ĐỘ**

**TRONG KHÔNG GIAN BA CHIỀU**

**TRÊN NỀN TẢNG WEB**

­

**Sinh viên thực hiện:**

Nguyễn Quốc Toàn

MSSV: B1704780

Lớp: DI1796A1

Khóa: 43

Trường đại học Cần Thơ

**Cán bộ hướng dẫn**

TS. Huỳnh Quang Nghi

HỌC KỲ 1, 2020 - 2021

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM

*(Học kỳ : 01, Năm học 2020-2021)*

**TÊN ĐỀ TÀI: MÔ PHỎNG NHIỆT ĐỘ TRONG KHÔNG GIAN BA CHIỀU TRÊN NỀN TẢNG WEB**

**GIÁO VIÊN HƯỚNG DẪN**:

|  |  |  |
| --- | --- | --- |
| STT | HỌ VÀ TÊN | MSCB |
| 1 | Huỳnh Quang Nghi |  |

**SINH VIÊN THỰC HIỆN:**

|  |  |  |  |
| --- | --- | --- | --- |
| HỌ VÀ TÊN | MSSV | THƯỞNG  (Tối đa 1,0 điểm) | ĐIỂM  (thang điểm 10) |
| Nguyễn Quốc Toàn | B1704780 |  |  |

|  |  |
| --- | --- |
| **I. HÌNH THỨC** |  |
| **Bìa** |  |
| * Đầy đủ các thông tin * Đúng định dạng |  |
| **Bố cục** |  |
| * Trang đánh giá kết quả thực hiện niên luận 1 * Mục lục: cấu trúc chương, mục và tiểu mục * Phụ lục (nếu có) * Tài liệu tham khảo |  |
| **II. NỘI DUNG** |  |
| **Giới thiệu** |  |
| * Mô tả bài toán * Mục tiêu cần đạt, hướng giải quyết |  |
| **Lý thuyết** |  |
| * Các khái niệm sử dụng trong chương trình * Kết quả vận dụng lý thuyết trong đề tài |  |
| **Ứng dụng** |  |
| * Phân tích yêu cầu, xây dựng các cấu trúc dữ liệu * Sơ đồ chức năng, lưu đồ giải thuật giải quyết vấn đề * Giới thiệu sử dụng chương trình |  |
| **Kết luận** |  |
| * Nhận xét kết quả đạt được * Hạn chế * Hướng phát triển |  |
| **III. CHƯƠNG TRÌNH DEMO** |  |
| **Giao diện thân thiện với người dùng** |  |
| **Hướng dẫn sử dụng** |  |
| **Kết quả thực hiện đúng với kết quả của phần ứng dụng** |  |
| * Kết quả đúng * Cách thức thực hiện hợp lý * Chức năng bổ sung, sáng tạo |  |

*Cần Thơ, ngày … tháng … năm 2020*

GIÁO VIÊN CHẤM

Huỳnh Quang Nghi

MỤC LỤC

[I. GIỚI THIỆU 4](#_Toc58925197)

[1. Mô tả bài toán 4](#_Toc58925198)

[1.1. Không gian ba chiều 4](#_Toc58925199)

[1.2. Ứng dụng mô phỏng nhiệt độ trong không gian ba chiều 4](#_Toc58925200)

[2. Mục tiêu cần đạt được 5](#_Toc58925201)

[3. Hướng giải quyết và kế hoạch thực hiện 5](#_Toc58925202)

[3.1. Hướng giải quyết 5](#_Toc58925203)

[3.2. Kế hoạch thực hiện 5](#_Toc58925204)

[II. CƠ SỞ LÝ THUYẾT 6](#_Toc58925205)

[1. Các khái niệm sử dụng trong đề tài 6](#_Toc58925206)

[1.1. Sơ lược về ReactJS 6](#_Toc58925207)

[1.2. Giới thiệu thư viện ThreeJS 6](#_Toc58925208)

[1.3. Giới thiệu công cụ OrbitControls.js 7](#_Toc58925209)

[1.4. Hệ màu HSL 8](#_Toc58925210)

[1.5. Hàm ánh xạ mã màu 8](#_Toc58925211)

[1.6. Góc Euler 9](#_Toc58925212)

[1.7. Ý tưởng xây dựng khối hộp trong không gian ba chiều 10](#_Toc58925213)

[1.8. Giải thuật duyệt mảng 10](#_Toc58925214)

[2. Kết quả vận dụng lý thuyết vào đề tài 11](#_Toc58925215)

[III. ỨNG DỤNG 11](#_Toc58925216)

[1. Phân tích yêu cầu bài toán, xây dựng các cấu trúc dữ liệu cần thiết. 11](#_Toc58925217)

[1.1. Dữ liệu đầu vào 11](#_Toc58925218)

[1.2. Yêu cầu bài toán 12](#_Toc58925219)

[2. Lưu đồ giải thuật để giải quyết vấn đề 13](#_Toc58925220)

[3. Giới thiệu chương trình 14](#_Toc58925221)

[IV. KẾT LUẬN 15](#_Toc58925222)

[1. Kết quả đạt được 15](#_Toc58925223)

[2. Hạn chế, nguyên nhân 15](#_Toc58925224)

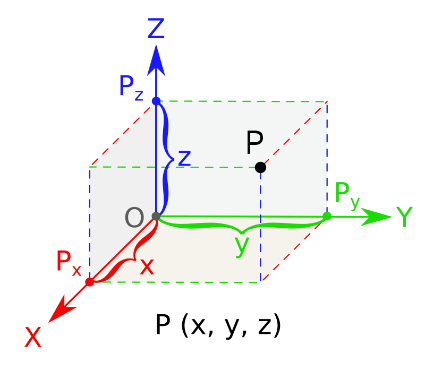
[3. Hướng phát triển của đề tài 15](#_Toc58925225)

[Tài liệu tham khảo 15](#_Toc58925226)

# GIỚI THIỆU

## 1. Mô tả bài toán

### 1.1. Không gian ba chiều



Hình 1: Minh họa không gian ba chiều

Không gian ba chiều là một mô hình hình học có ba thông số, trong đó bao gồm tất cả các vật chất được chúng ta biết đến. Ba chiều được nhắc đến ở đây thường là chiều dài (x), chiều rộng (y), chiều cao (z). Ba hướng bất kì nào cũng có thể được chọn, miễn là chúng không nằm trong cùng một mặt phẳng.

### 1.2. Ứng dụng mô phỏng nhiệt độ trong không gian ba chiều

Hình 2: Demo ứng dụng đang hoạt động

Là một thành phần trên trang web, biểu thị sự chênh lệch nhiệt độ trong một khu vực. Ứng với các tọa độ có nhiệt độ cao, màu sắc tại điểm đó sẽ là màu nóng. Ứng với các tọa độ có nhiệt độ thấp, màu sắc tại điểm đó sẽ là màu lạnh.

Ngoài ra, để xem được thông tin bên trong khối, người dùng còn có thể dùng chức năng cắt lớp để có thể “ẩn” đi một số phần tử không cần thiết. Khi đó, người dùng có thể quan sát được nội dung bên trong.

## 2. Mục tiêu cần đạt được

Về phía người dùng có thể tương tác với biểu đồ nhiệt, thực hiện các hành động như xoay, cắt lớp.

Về phía ứng dụng xác định mã màu cho các đơn vị trên mặt phẳng bằng hàm tính màu, biểu thị được màu sắc chính xác dự trên dữ liệu nhận vào, hoạt động trơ tru, tiêu thụ ít tài nguyên, không phát sinh lỗi đột ngột khi đang sử dụng.

## 3. Hướng giải quyết và kế hoạch thực hiện

### 3.1. Hướng giải quyết

Duyệt mảng giá trị

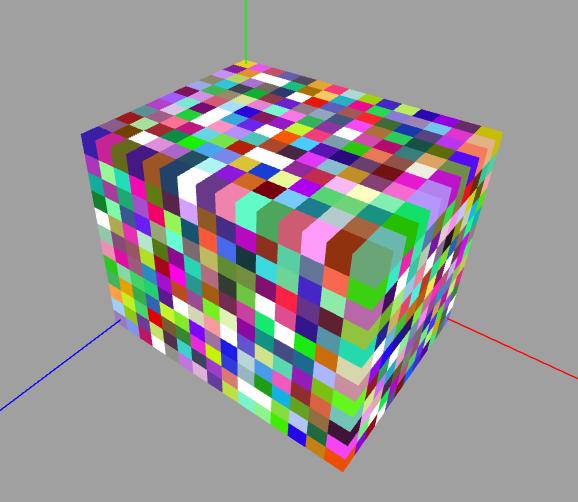
Ánh xạ từng phần tử trong mảng thành độ màu

Dựng khối hộp dựa trên sáu mặt phẳng

Tô các màu đã tính vào các mặt phẳng tương ứng.

### 3.2. Kế hoạch thực hiện

* Tìm hiểu thư viện ReactJS.
* Tìm hiểu thư viện ThreeJS.
* Dựng các phiên bản thử nghiệm:
  + Phiên bản 1: vẽ một khối hộp.
  + Phiên bản 2: vẽ nhiều khối hộp.
  + Phiên bản 3: cắt lớp theo một trục tọa độ.



Hình 3: Minh hoạ phiên bản 3 bằng tập dữ liệu ngẫu nhiên

* Cải tiến giải thuật duyệt, loại bỏ các thao tác thừa, thêm ràng buộc cho dữ liệu.
* Xây dựng hàm ánh xạ mã màu.
* Thêm tính năng tô màu theo dự liệu đầu vào.
* Nâng cấp tính năng cắt lớp.
* Điều chỉnh cú pháp, tái cấu trúc mã nguồn.

# CƠ SỞ LÝ THUYẾT

## 1. Các khái niệm sử dụng trong đề tài

### 1.1. Sơ lược về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web hấp dẫn, nhanh và hiệu quả với mã hóa tối thiểu. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản. Sức mạnh của nó xuất phát từ việc tập trung vào các thành phần riêng lẻ. Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một nhà phát triển/lập trình viên có thể phá vỡ giao diện người dùng phức tạp thành các thành phần đơn giản hơn.

Ứng dụng xây dựng trên nền tảng ReactJS sử dụng cú pháp JavaScript ES6, được biên dịch thông qua BabelJS giúp cho mã nguồn dễ đọc và dễ dàng phát trển hơn. Với sự hỗ trợ của ES6 và V8 Engine có thể nói rằng JavaScript hiện tại không thua gì các ngôn ngữ lập trình hướng đối tượng khác.

Các ứng dụng sử dụng ReactJS nổi tiếng có thể kể đến như: Facebook, Instagram, Zalo web, Netflix,...

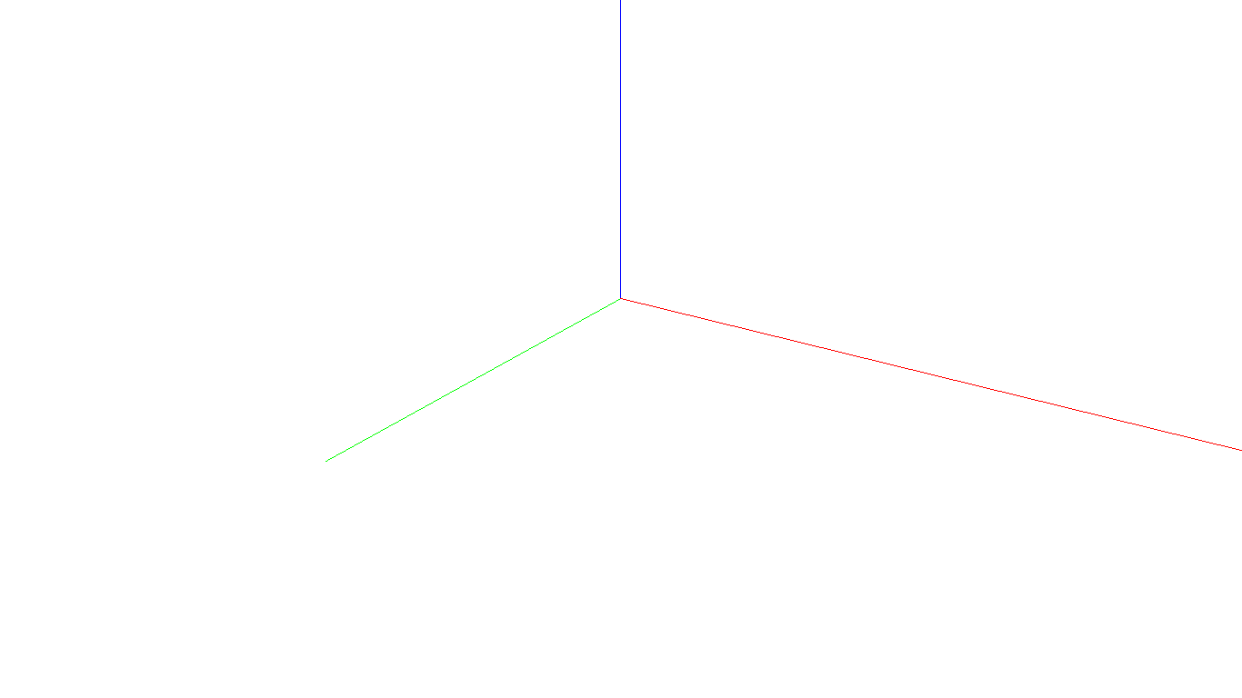
### 1.2. Giới thiệu thư viện ThreeJS

Three.js là một thư viện JavaScript 3D dễ sử dụng, gọn nhẹ. Nó cung cấp nhiều trình kết xuất như canvas, WebGL, v.v.

WebGL (Thư viện đồ họa Web) là một API JavaScript mang đồ họa 3D được tăng tốc phần cứng cho trình duyệt mà không cần sự trợ giúp của các tiện ích bổ sung nào khác. WebGL chạy trên GPU (Bộ xử lý đồ họa) của card đồ họa, điều đó có nghĩa là CPU của bạn (Bộ xử lý trung tâm) có thể tập trung vào các tác vụ khác. Từ quan điểm của người dùng, các ứng dụng chỉ đơn giản là chạy nhanh hơn nhiều. Thư viện Three.js giúp làm việc với WebGL dễ dàng hơn.

Các đối tượng được sử dụng trong đề tài:

* Scene: Đối tượng màn ảnh, nơi hiện thị mọi thứ từ không gian ba chiều..
* Color: Đối tượng màu sắc, hộ trợ hầu hết các hệ màu hiện tại như RGB, ARGB, HSL, ...
* PerspectiveCamera: Đối tượng máy quay, là cặp mắt nhìn vào bên trong thế giới ba chiều và vẽ lên màn ảnh.
* WebGLRenderer: Đối tượng hiển thị trả về một DOM Element để chèn vào trình duyệt.
* AxesHelper: Vẽ ba trục tọa độ X, Y, Z tương ứng với màu đỏ, xanh lá, xanh dương.



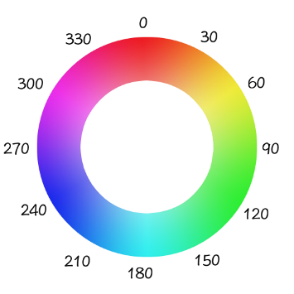
Hình 4: Minh họa đối tượng AxesHelper được vẽ trên màn hình

* Vector3: Đối tượng Vector, chứa tọa độ ứng với một điểm trong không gian hoặc một vector tính từ gốc tọa độ O(0, 0, 0) tới điểm đó.
* ArrowHelper: Công cụ hỗ trợ vẽ mũi tên, giúp vẽ hình mũi tên trong không gian ba chiều.
* LineBasicMaterial: Đối tượng vật chất hình đường thẳng, giúp tạo khung một đường thẳng trong không gian ba chiều.
* BufferGeometry: Đối tượng chứa các vật thể (không có thứ tự).
* Line: Đối tượng đường thẳng.
* Texture: Đối tượng tài nguyên (hình ảnh, font, âm thanh,… )
* SpriteMaterial: Đối tượng vật chất đặc biệt, luôn hướng về phía camera.
* Sprite: Đối tượng luôn hướng về phía camera.
* PlaneGeometry: Đối tượng hình thể dạng mặt phẳng.
* MeshBasicMaterial: Đối tượng vật chất cơ bản.
* Mesh: Một đối tượng vật thể hoàn chỉnh, có đầy đủ khung, vật chất, màu sắc và vị trí.
* Object3D: Đối tượng vật thể trong không gian ba chiều.

### 1.3. Giới thiệu công cụ OrbitControls.js

OrbitControls.js là một công cụ được đính kèm trong ThreeJS cung cấp cho giao diện web một khung nhìn có thể kéo thả, thu phóng và xoay tự do. Từ đó, giúp người dùng dễ dàng quan sát các vật thể được vẽ trong một khu vực nhất định. Mặc định, nếu không được cấu hình, công cụ sẽ giúp ta quan sát tại vị trí (0, 0, 0) và góc quay là 360 độ theo chiều ngang, 90 độ ứng với miền nửa trên của hệ trục tọa độ.

### 1.4. Hệ màu HSL



Hình 5: Minh họa bảng màu HSL

Không gian màu HSL, còn gọi là không gian màu HSV hay HSB, là một không gian màu dựa trên ba số liệu:

* **H**: (Hue) vùng màu hay còn gọi là góc độ màu.
* **S**: (Saturation) độ bão hòa màu,
* **L** (hay V, B): (Lightness, Value hay Bright) độ sáng.

### 1.5. Hàm ánh xạ mã màu

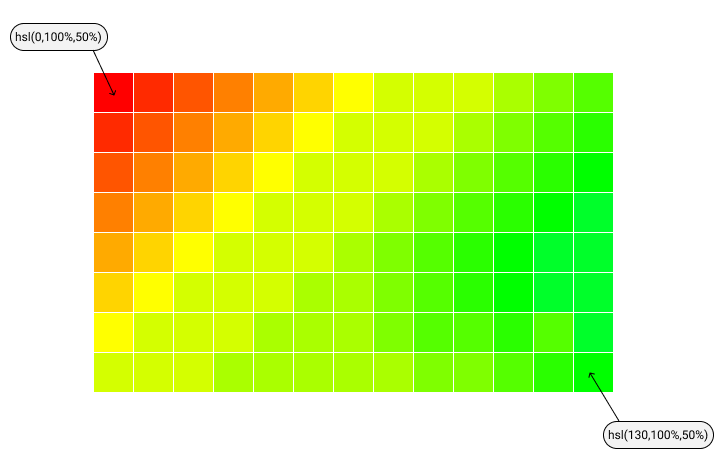
Ứng dụng sử dụng các màu trong hệ màu HSL có sắc độ từ 0 độ cho đến 240 độ để mô tả nhiệt độ. Nhiệt độ thấp nhất tương ứng với 240 độ (màu xanh), nhiệt độ cao nhất tương ứng 0 độ (màu đỏ).

Chia khoảng màu đó vào khoảng giữa giá trị nhỏ nhất và giá trị lớn nhất của mảng dữ liệu, ta có được sắc độ cần thiết để tô vào ô màu có giá trị tương ứng.

Hàm ánh xạ nói trên được xác định bởi công thức:

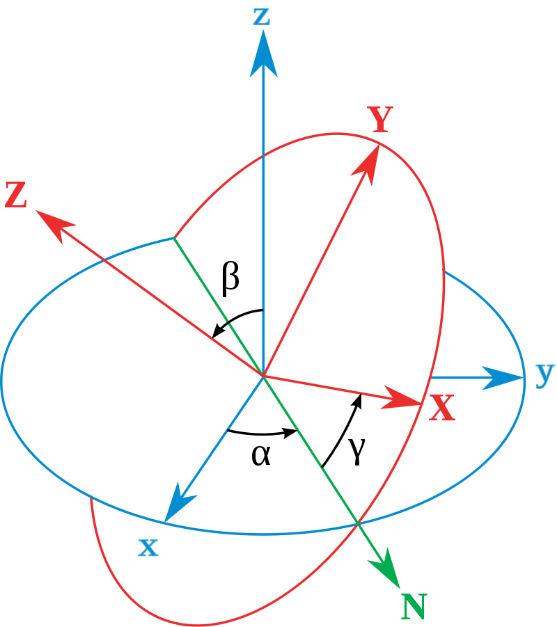


Với giá trị **S** và **L** lần lượt là 50% và 100% không đổi, sử dụng hàm ánh xạ để tính **H**, ta có mã màu HSL tại một tọa độ xác định là **hsl(H, 100%, 50%)**



Hình 6: Minh họa màu của một mặt khối hộp sau khi tô màu

### 1.6. Góc Euler

 **Góc Euler** là ba góc được giới thiệu bởi Leonhard Euler để miêu tả định hướng của một vật thể rắn. Để miêu tả như một định hướng trong không gian ba chiều Euclide theo ba tham số được yêu cầu. Chúng có thể được đưa ra bằng nhiều cách và Góc Euler là một trong số chúng.

Hình 7: Mô tả góc Euler trong không gian

Leonhard Euler là người cố gắng biểu diễn phương hướng đầu tiên. Ông đã tưởng tượng ra 3 khung tham chiếu có thể quay lần lượt vòng quanh nhau và nhận ra rằng bằng cách sử dụng một khung tham chiếu cố định và biểu diễn ba vòng quay ông có thể dùng bất kỳ khung tham chiếu trong không gian (sử dụng 2 vòng quay trên một trục dọc cố định và những vòng quay khác trên 2 trục khác). Giá trị của 3 vòng quay được gọi là Góc Euler.

### 1.7. Ý tưởng xây dựng khối hộp trong không gian ba chiều

Ý tưởng ban đầu là xây dựng hàm tạo từng khối vuông nhỏ, sau đó tịnh tiến chúng đến tọa độ tương ứng với index của mảng dữ liệu và tô màu dựa trên dữ liệu đầu vào. Tuy giải pháp này đơn giản nhưng lại gặp một vấn đề đó là khi dữ liệu đầu vào quá lớn sẽ làm cho ứng dụng bị chậm, gây khó khăn khi sử dụng và chiếm nhiều tài nguyên của thiết bị.

Qua xem xét, tôi nhận thấy rằng giải pháp ban đầu ngoài vẽ những khối

ban đầu ra, ứng dụng còn vẽ những khối bên trong ngay cả khi những khối đó bị che khuất và không thể nào quan sát được. Vì vậy, việc thay đổi hướng tiếp cận đề tài trong trường hợp này là cần thiết. Thay vì vẽ tất cả các khối ta chỉ cần vẽ sáu mặt phẳng sau đó tịnh tiến và quay chúng chúng về đúng vị trí tương ứng của chúng trên hệ trục tọa độ.

Trong phạm vi đề tài, chúng tôi sử dụng góc Euler để quay các mặt phẳng. Để tăng hiệu suất ứng dụng, thay vì tô màu cả hai mặt của mặt phẳng, ứng dụng chỉ tô màu một mặt giúp làm giảm tiêu tốn bộ nhớ khi vận hành.

Giả sử mảng dữ liệu DATA có kích thước X\*Y\*Z. Theo thứ tự ưu tiên tọa độ từ x, y đến z. Đặt các mặt của khối hộp lần lượt là A, B, C, D, E, F. Góc quay tương ứng của các mặt phẳng là:

* A, B: E(3π/2, 0, 0)
* C, D: E(3π/2, 3π/2, 0)
* E, F: E(π, 0, 0)

Tương tự, kích thước của các mặt phẳng tương ứng là:

* A, B: X\*Z
* C, D: Y\*Z
* E, F: X\*Y

Cuối cùng, căn chỉnh tọa độ của các mặt phẳng theo kích thước của dữ liệu để có được khối hộp hoàn chỉnh.

### 1.8. Giải thuật duyệt mảng

Giả sử mảng dữ liệu DATA có kích thước M\*N\*O trong hệ trục tọa độ Oxyz.

Ở các phiên bản 1, 2 và 3, ứng dụng sử dụng giải thuật tham lam để duyệt mảng giá trị, mỗi mặt phẳng duyệt một lần, một khối hộp có sáu mặt phẳng nên sẽ duyệt 6\*N\*M\*O lần! Vì vậy, điều này làm cho ứng dụng trở nên chậm chạp và cồng kềnh. Để khắc phục, thay vì duyệt toàn bộ mảng, ứng dụng chỉ duyệt qua các phần tử cần thiết. Từ đó, giảm số lần duyệt xuống còn 2\*M\*N+2\*N\*O+2\*M\*O.

Theo thứ tự ưu tiên tọa độ từ x, y đến z. Đặt các mặt của khối hộp lần lượt là A, B, C, D, E, F. Sử dụng hai vòng lặp **for** để duyệt cho mỗi mặt. Vòng lặp **for** bên ngoài có biến chạy là **i**, vòng lặp for bên trong có biến chạy là **j**. Ứng với từng mặt phẳng, ta sẽ lấy ra được giá trị tương ứng cần sử dụng như sau:

* A(i, j) = DATA[j][0][i]
* B(i, j) = DATA[j][N-1][i]
* C(i, j) = DATA[0][j][i]
* D(i, j) = DATA[M-1][j][i]
* E(i, j) = DATA[j][i][0]
* F(i, j) = DATA[j][i][O-1]

## 2. Kết quả vận dụng lý thuyết vào đề tài

Vận dụng thành công giải thuật duyệt mảng để tô màu cho khối hộp.

Vận dụng ThreeJS để tạo ra biểu đồ ba chiều trên nền tảng web.

Vận dụng ReactJS để đóng gói mã nguồn thành một thành phần web để có thể tái sử dụng.

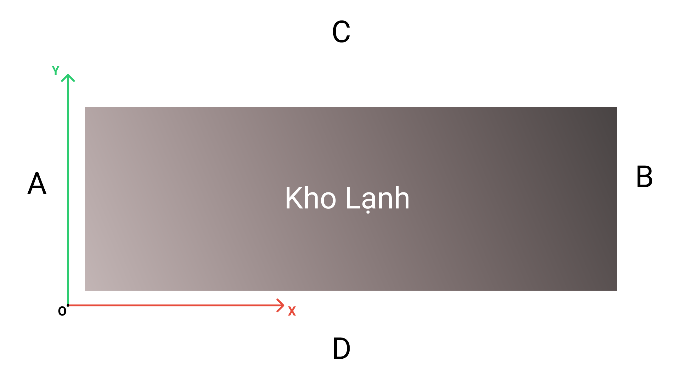
# ỨNG DỤNG

## 1. Phân tích yêu cầu bài toán, xây dựng các cấu trúc dữ liệu cần thiết.

### 1.1. Dữ liệu đầu vào

Dữ liệu cấu hình kho lạnh. Đây là một đối tượng có ba thuộc tính:

* size: là đối tượng quy định kích thước của kho lạnh, gồm bốn thuộc tính:
  + x: số khối tối đa theo chiều rộng của kho lạnh.
  + y: số khối tối đa theo chiều dài của kho lạnh.
  + z: số khối tối đa theo chiều cao của kho lạnh.
  + tilesize: kích thước của một đơn vị thể hiện trên mô hình, đơn vị là điểm ảnh (px).
* door: là đối tượng cấu hình cửa, gồm hai thuộc tính:
  + show: quyết định có hiển thị cửa trên mô hình hay không. Nếu “show = true” thì hiển thị cửa trên mô hình, ngược lại “show = false” thì không hiển thị.
  + direction: mô tả vị trí cửa của kho lạnh theo có 4 hướng: A, B, C, D. Các hướng này được định vị theo trục tọa độ như sau:



* axis-labels: mô tả các nhãn được gắn trên ba trục tọa độ.
  + - axis-x: mô tả các nhãn trên trục x.
    - show: xác định xem có hiển thị các nhãn trên trục x hay không.
    - list: mảng các số nguyên chứa tọa độ của các nhãn trên trục x.
    - axis-y: mô tả các nhãn trên trục y.
    - show: xác định xem có hiển thị các nhãn trên trục y hay không.
    - list: mảng các số nguyên chứa tọa độ của các nhãn trên trục y.
    - axis-z: mô tả các nhãn trên trục z.
* show: xác định xem có hiển thị các nhãn trên trục z hay không.
* list: mảng các số nguyên chứa tọa độ của các nhãn trên trục z.

Dữ liệu nhiệt độ là một đối tượng có ba thuộc tính như sau:

* values: là một mảng ba chiều chứa thông tin nhiệt độ của kho lạnh. Mảng có dạng A[x1][y1][z1]. Nhiệt độ ở một điểm P(x,y,z) sẽ tương ứng phần tử A[x][y][z] của mảng. Ví dụ: nhiệt độ ở vị trí P(1,2,5) = A[1][2][5].
* min: nhiệt độ thấp nhất trong kho lạnh.
* max: nhiệt độ cao nhất trong kho lạnh.

Dữ liệu cấu hình lát cắt

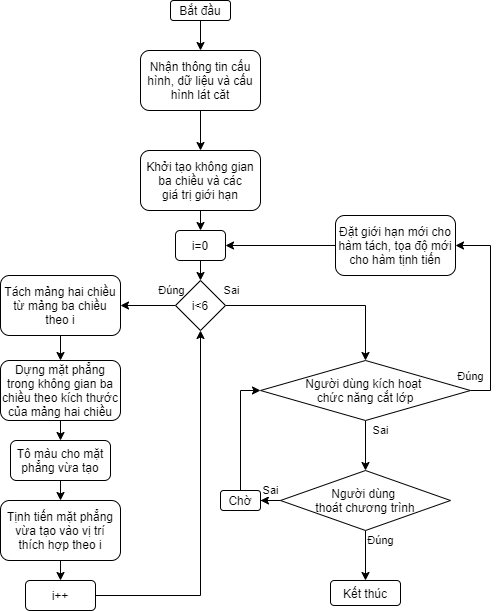
* Dữ liệu cấu hình lát cắt là một đối tượng chứa hai điểm tạo thành cạnh chéo dài nhất của hình hộp chữ nhật:
* origin: điểm bắt đầu.
* destination: điểm kết thúc.

### 1.2. Yêu cầu bài toán

Kết quả đầu ra mong muốn

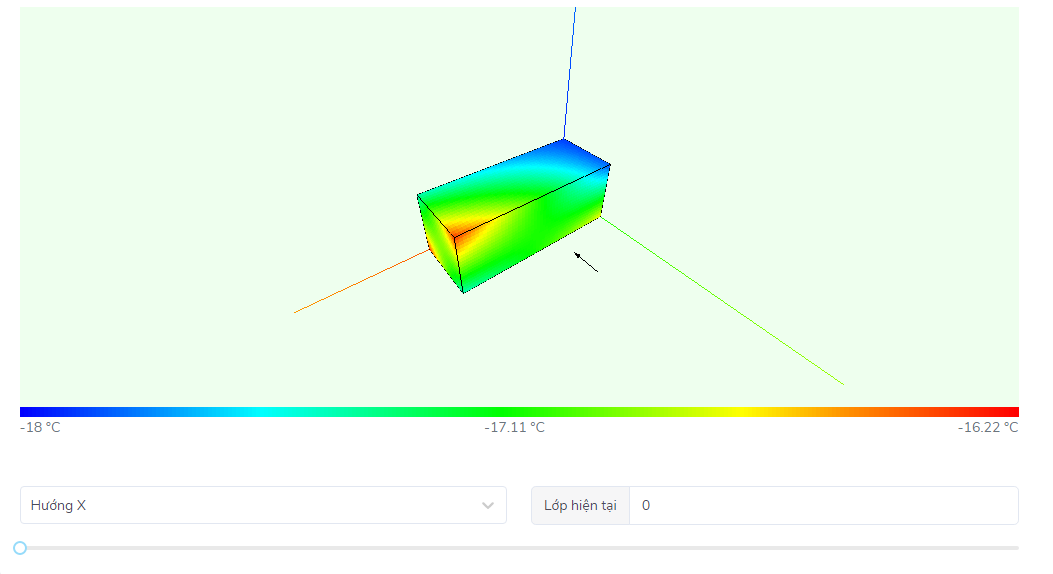
* Hiển thị khối hình hộp với các ô màu tương ứng với dữ liệu nhiệt độ đã cung cấp trước đó.
* Có thể xoay 360 độ, không có ô nào bị che khuất.
* Có thể cắt lớp khối kho lạnh theo ba trục x, y và z bằng giao diện đồ họa.
* Có thể hiển thị các nhãn đã cài đặt trên ba trục tọa độ, vị trí chính xác, nội dung rõ ràng.

## 2. Lưu đồ giải thuật để giải quyết vấn đề



## 3. Giới thiệu chương trình

Giao diện chính của chương trình, gồm có 3 vùng như hình dưới:

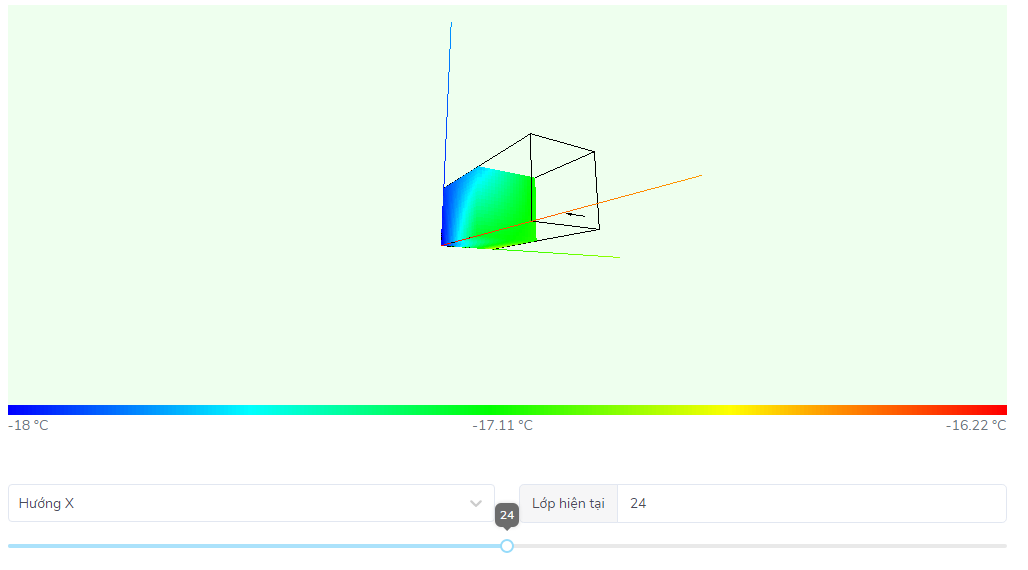


Vùng 3D

Thanh chia màu

Công cụ cắt lớp

* Vùng 3D: là một đối tượng canvas trên trang web, người dùng có thể tương tác bằng cách kéo thả chuột để xoay khối hộp, cuộn con lăn chuột để phóng to hoặc thu nhỏ và dùng các phím mũi tên trên bàn phím để dịch chuyển khung nhìn.
* Thanh chia màu: hiển thị nhiệt độ lớn nhất, nhỏ nhất trong phòng. Chọn một điểm trên thanh chia màu để biết được màu sắc đó ứng với nhiệt độ bao nhiêu.
* Công cụ cắt lớp: Gồm một menu cho phép chọn lớp cắt, một input cho phép nhập số để quyết định số lớp cắt và một thanh kéo cũng có tính năng là quyết định số lớp cắt.



Hình 8: Minh họa một biểu đồ đang bị cắt

# KẾT LUẬN

## 1. Kết quả đạt được

Ứng dụng được thiết kế và triển khai hoàn thành được yêu cầu đã đề ra ban đầu. Ngoài ra, ứng dụng còn được cái tiến về hiệu suất và tính năng, giúp tránh được một số lỗi phát sinh khi sử dụng.

Nội dung công việc của đề tài đã hoàn thành đầy đủ các yêu cầu đã đề ra cụ thể như sau:

* Giao diện biểu đồ ba chiều.
* Giao diện công cụ cắt lớp, chọn lớp cắt.
* Tính năng xoay tự do giúp người dùng có thể quan sát từ mọi hướng.
* Tính năng cắt lớp giúp người dùng xem được thông tin bị che khuất.

## 2. Hạn chế, nguyên nhân

Khi nhiệt độ thấp nhất bằng nhiệt độ cao nhất, biểu đồ không hoạt động. Nguyên nhân là do trong trường hợp này, hàm ánh xạ có thao thác chia cho không tạo ra giá trị vô cùng.

## 3. Hướng phát triển của đề tài

* Tiếp tục đa hóa hiệu suất, sử dụng ít tài nguyên hơn để hoạt động.
* Xử lý các trường hợp lỗi bất ngờ khiến chương trình đột ngột ngừng hoạt động.

# V. TÀI LIỆU THAM KHẢO

[1] **mrdoob**, “*threejs docs*”, <https://threejs.org/docs/>

[2] **gaearon**, “*readme.md*”,<https://github.com/facebook/react/>

[3] **Facebook Inc**, <https://reactjs.org/>

[4] **Wikipedia**, “Góc Euler”, <https://vi.wikipedia.org/wiki/G%C3%B3c_Euler>