TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🕮🙜-----

**BÁO CÁO MÔN PHÁT TRIỂN MÃ NGUỒN MỞ**

*Tên đề tài:*

**TRỰC QUAN KHOA HỌC**

|  |  |  |
| --- | --- | --- |
|  | Người hướng dẫn: ThS. **TRẦN QUỐC TRƯỜNG** | |
|  | Sinh viên thực hiện: | **Nhóm 2** |
|  |  | **BÙI PHONG PHÚ**  **NGUYỄN HÒA NINH ĐAN**  **MAI NHẬT HÀO**  **NGÔ THÁI DƯƠNG** |

TP HỒ CHÍ MINH – NĂM 2023

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🕮🙜-----

**BÁO CÁO MÔN PHÁT TRIỂN MÃ NGUỒN MỞ**

*Tên đề tài:*

**TRỰC QUAN KHOA HỌC**

|  |  |  |
| --- | --- | --- |
|  | Người hướng dẫn: ThS.**TRẦN QUỐC TRƯỜNG** | |
|  | Sinh viên thực hiện: | **Nhóm 2** |
|  |  | **BÙI PHONG PHÚ**  **NGUYỄN HÒA NINH ĐAN**  **MAI NHẬT HÀO**  **NGÔ THÁI DƯƠNG** |
|  |  |  |

TP HỒ CHÍ MINH – NĂM 2023

MỤC LỤC

[CHƯƠNG 1. GIỚI THIỆU 7](#_Toc152085635)

[1.1. Giới thiệu cơ bản về giao diện người dùng đồ họa 7](#_Toc152085636)

[1.1.1. Định nghĩa và Ý nghĩa của GUI 8](#_Toc152085637)

[1.1.2. Tầm quan trọng của GUI trong thiết kế ứng dụng 9](#_Toc152085638)

[1.1.3. Sự phát triển của giao diện người dùng 10](#_Toc152085639)

[1.2. Giới thiệu về OpenGL 11](#_Toc152085640)

[1.2.1. Định nghĩa của OpenGL 11](#_Toc152085641)

[1.2.2. Cách OpenGL hỗ trợ đồ họa 11](#_Toc152085642)

[1.2.3. Demo 13](#_Toc152085643)

[1.3. Giới thiệu về OGRE 15](#_Toc152085644)

[1.3.1. Ưu điểm và tính năng của OGNE 15](#_Toc152085646)

[1.3.2. Demo 16](#_Toc152085647)

[CHƯƠNG 2. Công cụ và ứng dụng trực quan hóa 18](#_Toc152085648)

[2.1. GNUplot 18](#_Toc152085649)

[2.1.3. Tính năng 18](#_Toc152085650)

[2.1.4. Một số ảnh minh họa về GNUplot 19](#_Toc152085651)

[2.2. Grace/XMGR 20](#_Toc152085652)

[2.2.1. Tính năng 20](#_Toc152085653)

[2.2.2. Một số ảnh minh họa về Grace/XMGR 20](#_Toc152085654)

[2.3. Xfig 22](#_Toc152085655)

[2.3.3. Tính năng 22](#_Toc152085656)

[2.3.4. Một số hình minh họa về Xfig 22](#_Toc152085657)

[2.4. FreeType 23](#_Toc152085658)

[2.4.1. Tính năng 24](#_Toc152085659)

[2.4.2. Một số hình minh họa FreeType 24](#_Toc152085660)

[2.5. Anti-Grain-Geimetry (AGG) 24](#_Toc152085661)

[2.5.1. Tính năng 25](#_Toc152085662)

[2.5.2. Một số ảnh minh họa AGG 26](#_Toc152085663)

[CHƯƠNG 3. Công nghệ hiện đại và thư viện 27](#_Toc152085664)

[3.1. GD (Graphics drawing) 27](#_Toc152085665)

[3.1.1. GD là gì 27](#_Toc152085666)

[3.1.2. Cách sử dụng thư viện GD 27](#_Toc152085667)

[3.2. Geomview 31](#_Toc152085668)

[3.2.3. Geomview là gì? 31](#_Toc152085669)

[3.2.4. Cài đặt geomview 32](#_Toc152085670)

[3.3. Hippodraw 34](#_Toc152085671)

[3.3.1. Hippodraw là gì ? 34](#_Toc152085672)

[3.3.2. Giao diện 35](#_Toc152085673)

[3.4. Ggobi 35](#_Toc152085674)

[3.4.1. Ggobi là gì ? 35](#_Toc152085675)

[3.4.2. Cài đặt ggobi 36](#_Toc152085676)

[CHƯƠNG 4. Ứng dụng thực tế và kết luận 39](#_Toc152085677)

[4.1. Paraview 39](#_Toc152085678)

[4.1.1. Các tính năng quan trọng của Paraview 39](#_Toc152085679)

[4.2. OpenDX (Open Data Explorer) 43](#_Toc152085680)

[4.2.1. Một số tính năng độc đáo 43](#_Toc152085681)

[4.2.2. Nhập dữ liệu trong OpenDX 46](#_Toc152085682)

[4.2.3. Quy trình chung trong OpenDX 46](#_Toc152085683)

[4.3. Tổng kết về các công cụ và ứng dụng trực quan hóa 49](#_Toc152085684)

[4.3.1. Paraview và VTK 49](#_Toc152085685)

[4.3.2. OpenDX 49](#_Toc152085686)

[4.3.3. Công cụ và ứng dụng khác 49](#_Toc152085687)

[4.4. Chia sẻ nhận định và đề xuất hướng phát triển trong tương lại 49](#_Toc152085688)

[4.4.1. Nhận định 49](#_Toc152085689)

[4.4.2. Đề xuất 50](#_Toc152085690)

[TÀI LIỆU 51](#_Toc152085691)

# MỤC LỤC CÁC HÌNH VẼ

[Hình 1 Giao diện windows 11 5](#_Toc152085820)

[Hình 2 Giao diện của 1 phần mền 6](#_Toc152085821)

[Hình 3 Giao diện của những năm 2000 6](#_Toc152085822)

[Hình 4 Giao diện trên Pc và Laptop 7](#_Toc152085823)

[Hình 5 Windows 11 8](#_Toc152085824)

[Hình 6 Windows XP 8](#_Toc152085825)

[Hình 7 Logo Open GL 9](#_Toc152085826)

[Hình 8 Minh họa OpenGL 1 9](#_Toc152085827)

[Hình 9 Minh họa OpenGL 2 10](#_Toc152085828)

[Hình 10 Minh họa các hoạt động của OpenGL 10](#_Toc152085829)

[Hình 11 Quy trình đường ống đồ họa 11](#_Toc152085830)

[Hình 12 Kiến trúc ONGE 13](#_Toc152085831)

[Hình 13 Minh họa ONGE 1 14](#_Toc152085832)

[Hình 14 Minh họa ONGE 2 14](#_Toc152085833)

[Hình 15 Logo Gnuplot 16](#_Toc152085834)

[Hình 16 Minh họa Gnuplot 1 17](#_Toc152085835)

[Hình 17 Minh họa GNUplot 2 17](#_Toc152085836)

[Hình 18 Minh họa GNUplot 3 17](#_Toc152085837)

[Hình 19 Logo Grace 18](#_Toc152085838)

[Hình 20 Minh họa Grace 1 19](#_Toc152085839)

[Hình 21 Minh họa Grace 2 19](#_Toc152085840)

[Hình 22 Minh họa Grace 3 19](#_Toc152085841)

[Hình 23 Logo Xfig 20](#_Toc152085842)

[Hình 24 Minh họa Xfix 1 21](#_Toc152085843)

[Hình 25 Minh họa Xfig 2 21](#_Toc152085844)

[Hình 26 Minh họa Xfig 3 21](#_Toc152085845)

[Hình 27 Logo FreeType 22](#_Toc152085846)

[Hình 28 Minh họa FreeType 1 22](#_Toc152085847)

[Hình 29 Minh họa FreeType 2 22](#_Toc152085848)

[Hình 30 Logo AGG 23](#_Toc152085849)

[Hình 31 Minh họa AGG 1 24](#_Toc152085850)

[Hình 32 Minh họa AGG 2 24](#_Toc152085851)

[Hình 33 Minh họa AGG 3 24](#_Toc152085852)

[Hình 34 Main panel 31](#_Toc152085853)

[Hình 35 Tool panel 31](#_Toc152085854)

[Hình 36 Camera 32](#_Toc152085855)

[Hình 37 Minh họa hippodraw 33](#_Toc152085856)

[Hình 38 Giao diện ggobi 34](#_Toc152085857)

[Hình 39 Mở một fiile dữ liệu 35](#_Toc152085858)

[Hình 40 Giao diện kết quả 35](#_Toc152085859)

[Hình 41 Giao diện kết quả 36](#_Toc152085860)

[Hình 42 Logo Paraview 37](#_Toc152085861)

[Hình 43 Giao diện chính 37](#_Toc152085862)

[Hình 44 Minh họa Paraview 1 40](#_Toc152085863)

[Hình 45 Minh họa Paraview 2 40](#_Toc152085864)

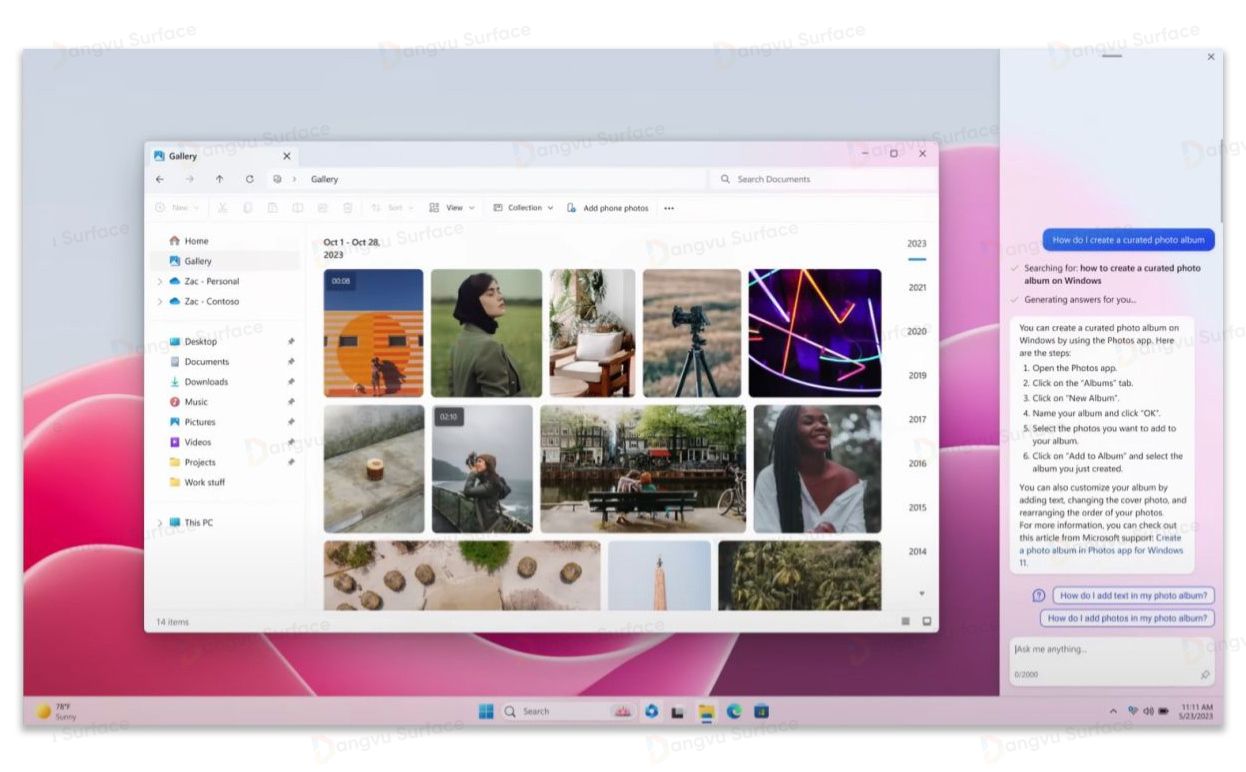
[Hình 46 Minh họa Paraview 3 41](#_Toc152085865)

[Hình 47 Logo OpenDX 41](#_Toc152085866)

# GIỚI THIỆU

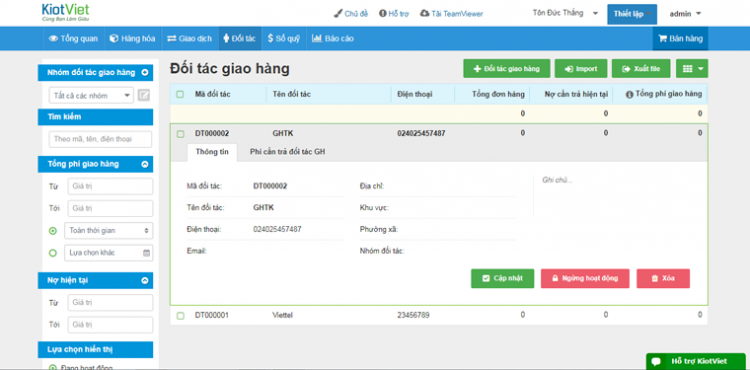
## Giới thiệu cơ bản về giao diện người dùng đồ họa

Trong cuộc sống hàng ngày, hình ảnh không chỉ đơn giản là một phần của trải nghiệm, mà nó đóng vai trò quan trọng trong cách chúng ta hiểu và tương tác với thế giới xung quanh. Hình ảnh không chỉ làm cho thông tin trở nên trực quan hóa mà còn kích thích giác quan, tạo ra một kết nối sâu sắc hơn với người dùng. Vì hình ảnh giúp ta dễ liên tưởng đến sự vật sự việc mà ta nghĩ đến. Nói đúng hơn thì chúng ta sống mà không thể thiếu hình ảnh. Hình ảnh như 1 ngôn ngữ giúp ta hiểu được nhiều vấn đề



Hình 1 Giao diện windows 11

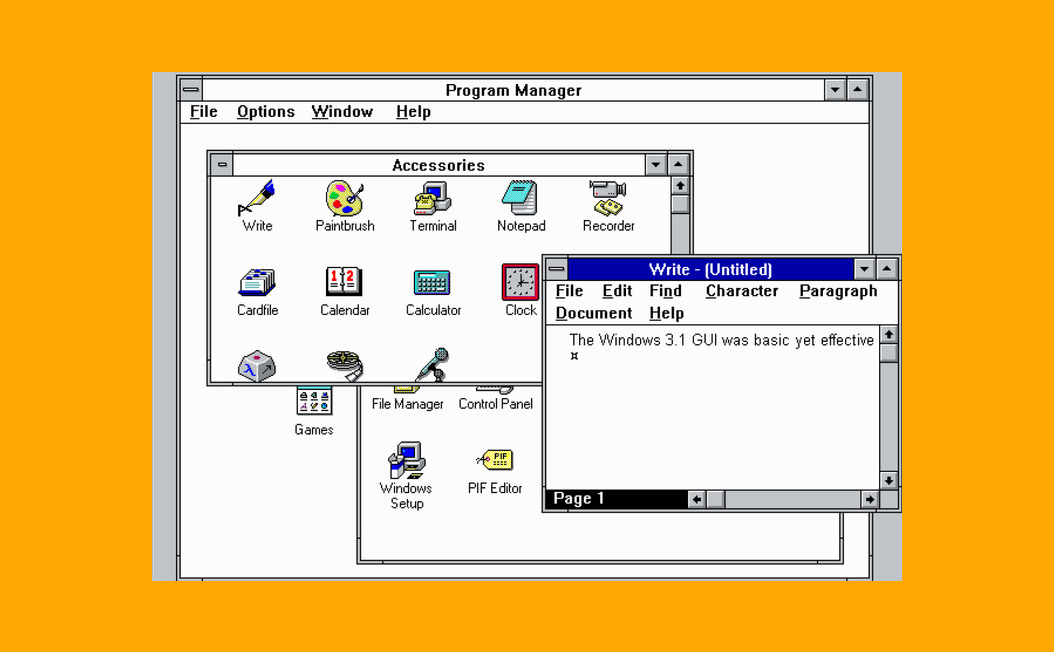
Chúng ta tiếp xúc với hình ảnh mỗi ngày, từ các logo trên điện thoại di động cho đến giao diện người dùng của các ứng dụng và trang web mà chúng ta truy cập. Hình ảnh chính là cầu nối, tạo ra trải nghiệm tương tác mạnh mẽ và thú vị cho người dùng.



Hình 2 Giao diện của 1 phần mền

### Định nghĩa và Ý nghĩa của GUI

Giao Diện Người Dùng Đồ Họa (GUI) là một khía cạnh quan trọng của trải nghiệm người dùng hiện đại. GUI không chỉ là một phần mềm đơn thuần, mà là ngôn ngữ của trải nghiệm người dùng. Nó tạo ra sự tương tác thông qua hình ảnh và đồ họa thay vì văn bản khó hiểu.



Hình 3 Giao diện của những năm 2000

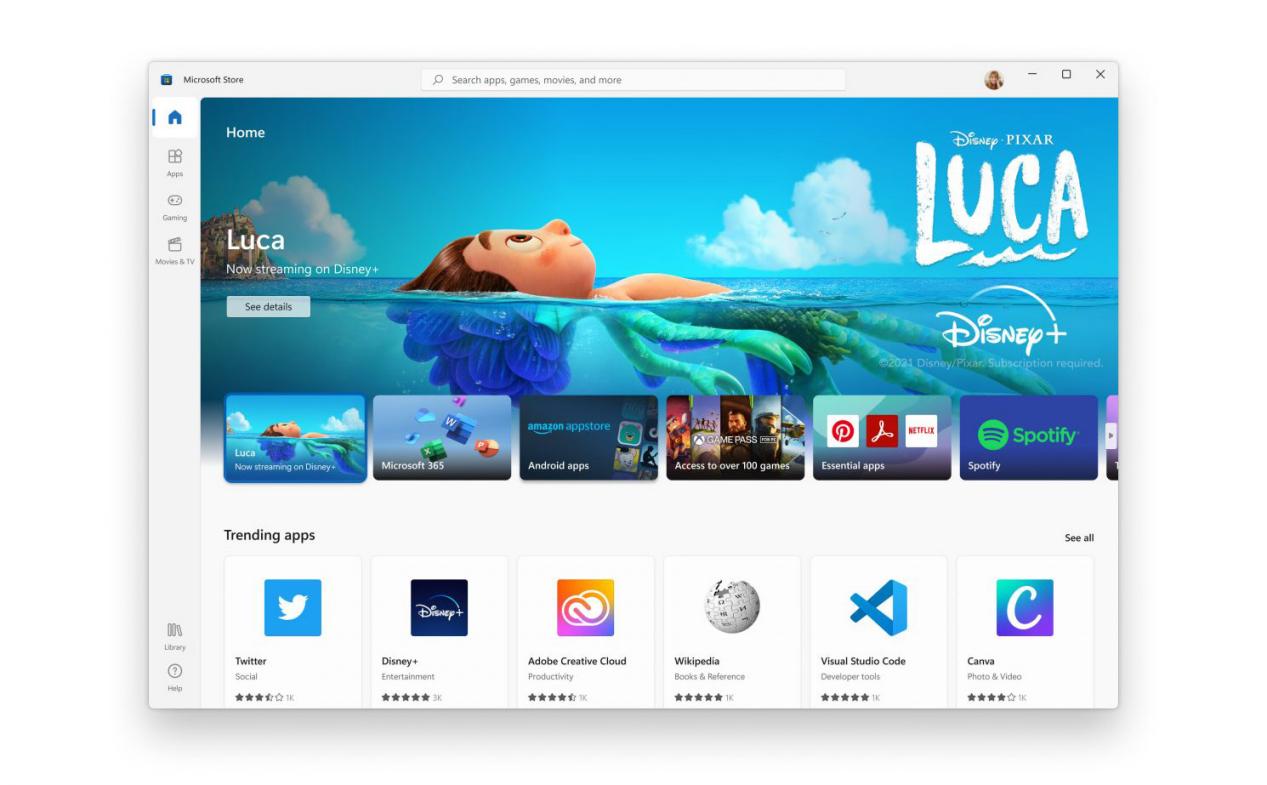
Tầm quan trọng của GUI nằm ở khả năng trực quan hóa thông tin một cách hiệu quả, giúp người dùng dễ dàng hiểu và tương tác với ứng dụng. Điều này không chỉ tạo ra sự thuận tiện mà còn tăng cường trải nghiệm người dùng.



Hình 4 Giao diện trên Pc và Laptop

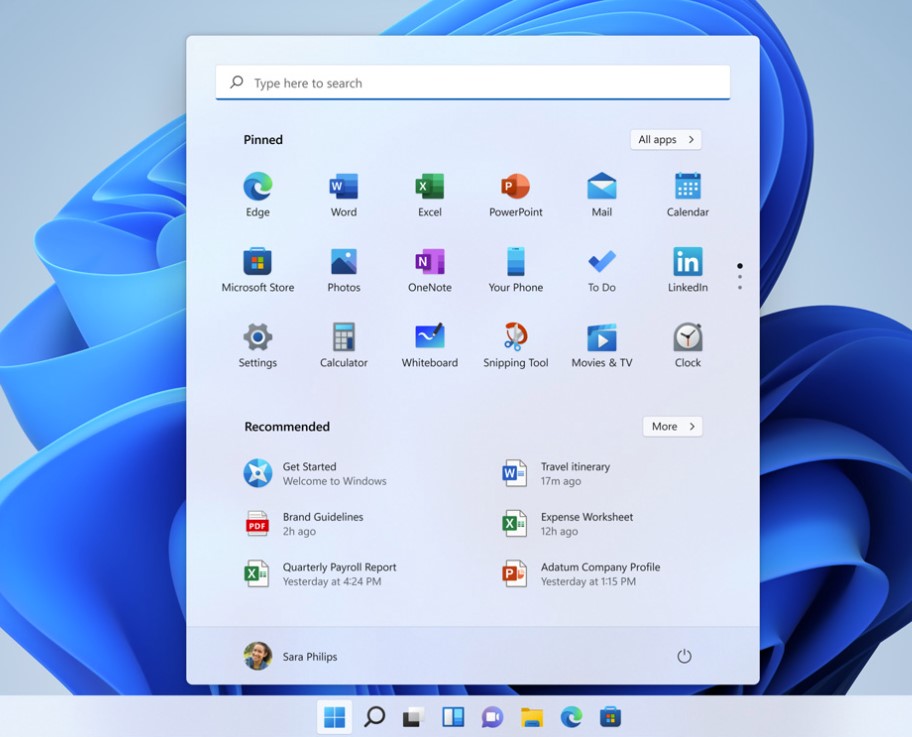
### Tầm quan trọng của GUI trong thiết kế ứng dụng

Trong quá trình thiết kế ứng dụng, GUI chơi một vai trò quyết định. Không chỉ giảm độ phức tạp của ứng dụng, GUI còn tạo ra giao diện thân thiện, dễ sử dụng, và tăng tính tương tác. Những lợi ích này bao gồm giảm thời gian học và sử dụng ứng dụng, tăng cường khả năng tương tác và tạo ra một trải nghiệm người dùng thú vị.

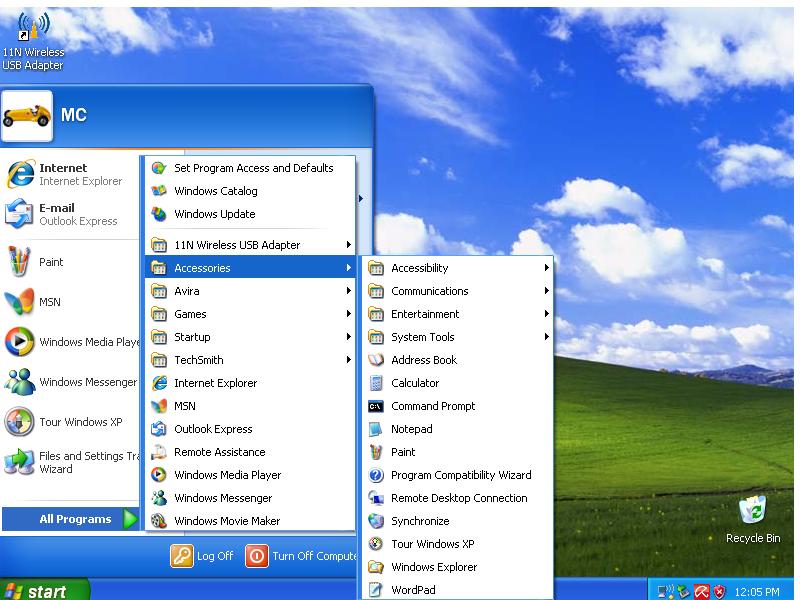


### Sự phát triển của giao diện người dùng

* So sánh Windows XP và Windows 11



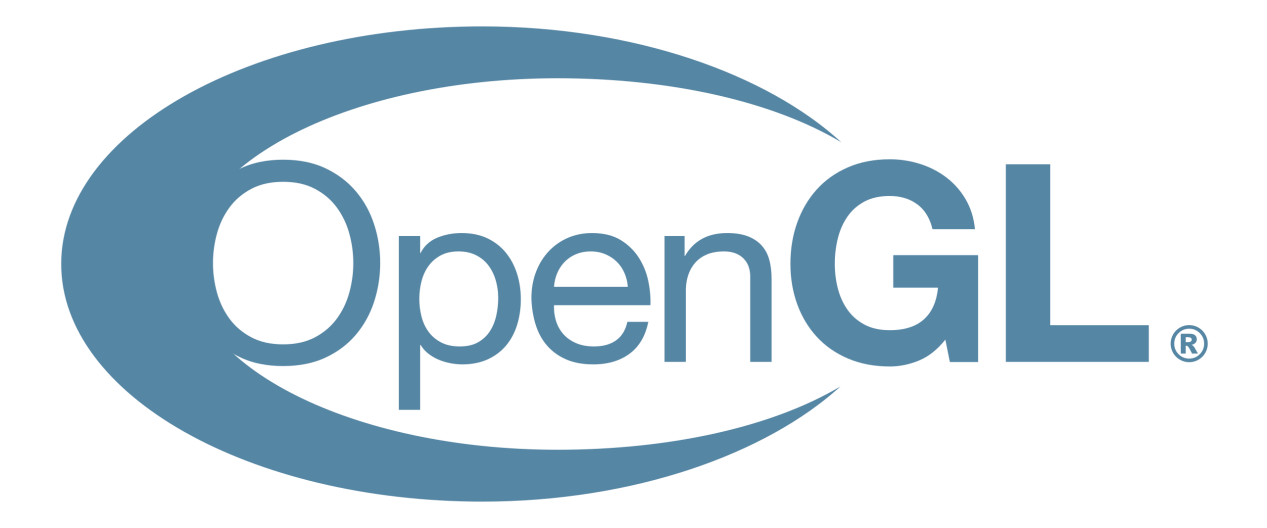
Hình 5 Windows 11



Hình 6 Windows XP

## Giới thiệu về OpenGL

### Định nghĩa của OpenGL

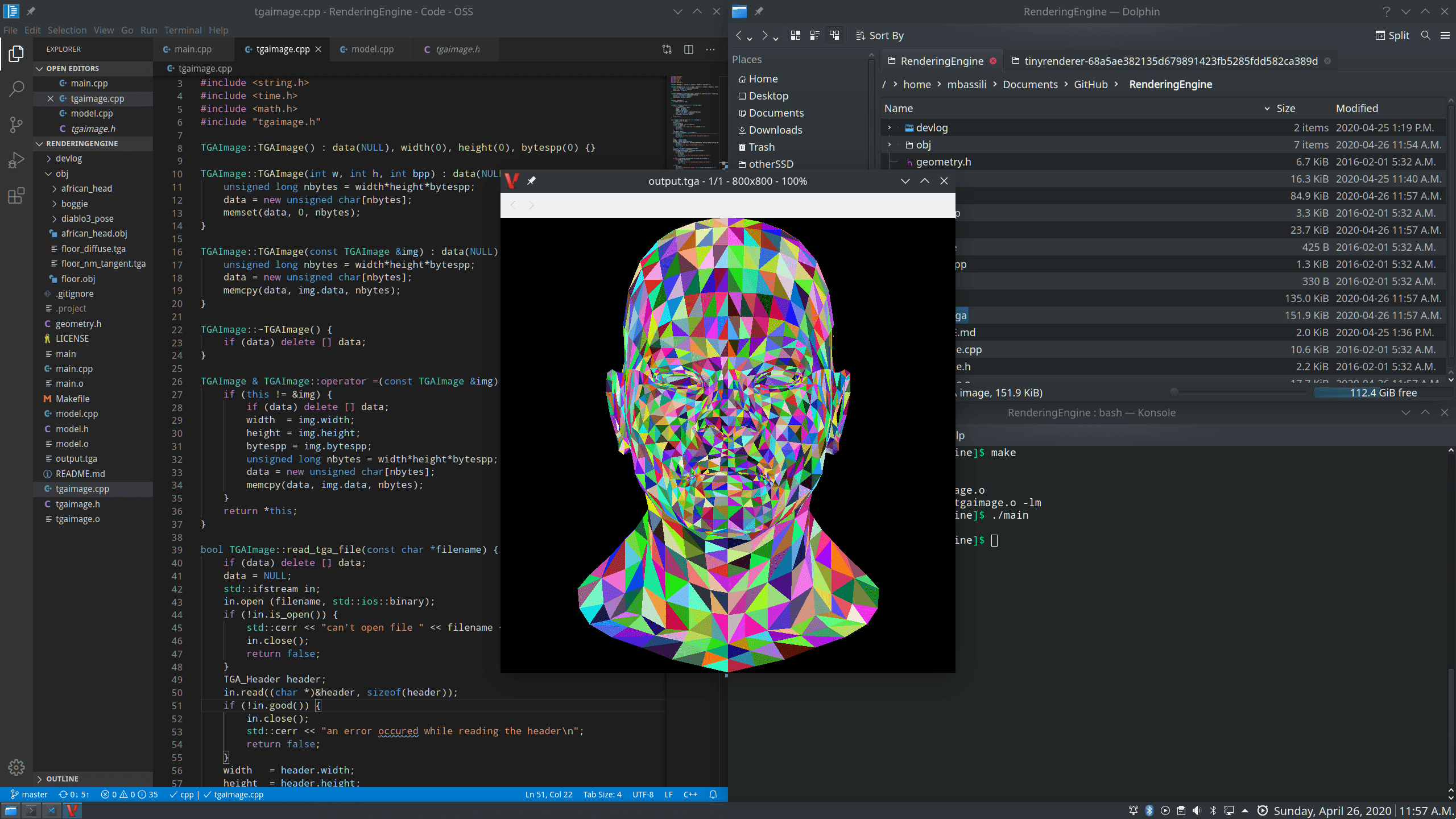


Hình 7 Logo Open GL

OpenGL, là một tiêu chuẩn đồ họa 3D, đóng vai trò quan trọng trong việc tạo ra đồ họa chất lượng cao trên nền tảng đa dạng. Nó là một giao diện lập trình ứng dụng (API) mạnh mẽ, cho phép nhà phát triển tận dụng sức mạnh của GPU (đơn vị xử lý đồ họa) để tối ưu hóa hiệu suất và tạo ra hình ảnh 3D đẹp mắt.

### Cách OpenGL hỗ trợ đồ họa

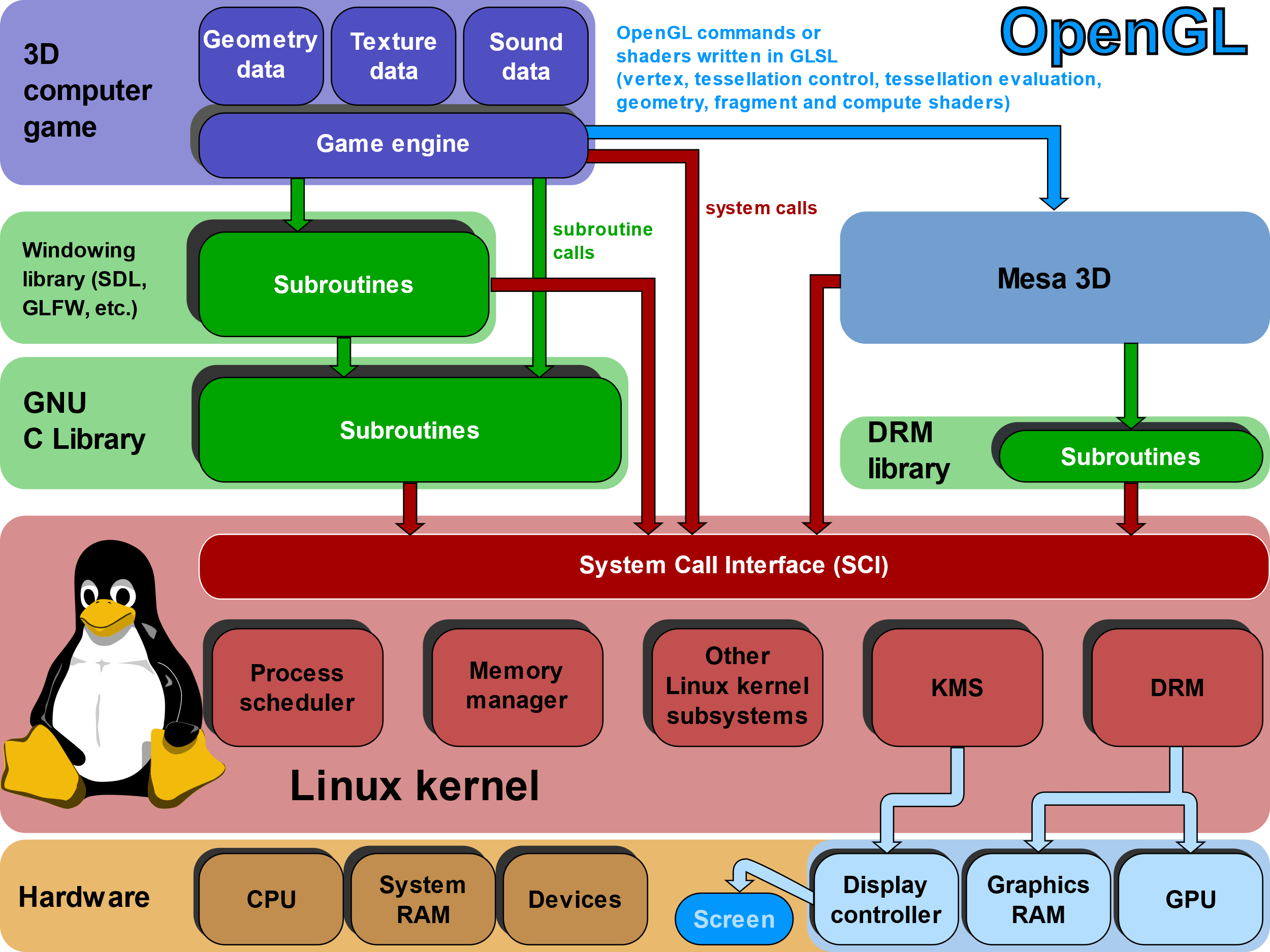
OpenGL sử dụng GPU để thực hiện các phép toán đồ họa, từ việc xử lý dữ liệu đến kết xuất hình ảnh trên màn hình. Quy trình đường ống đồ họa, một chuỗi các giai đoạn và pha xử lý, giúp tối ưu hóa quá trình hiển thị đồ họa 3D và tăng cường hiệu suất.



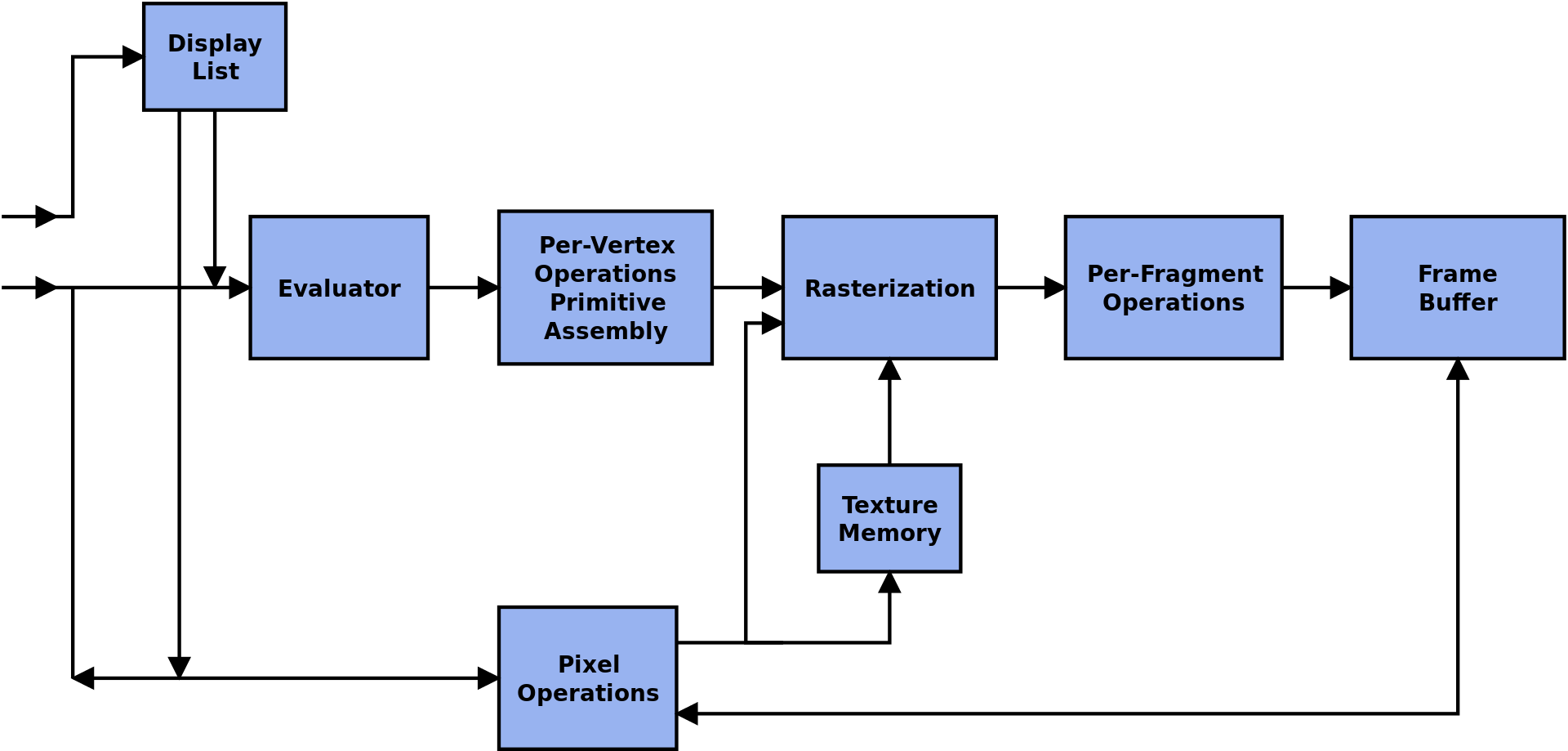
Hình 8 Minh họa OpenGL 1



Hình 9 Minh họa OpenGL 2



Hình 10 Minh họa các hoạt động của OpenGL



Hình 11 Quy trình đường ống đồ họa

### Demo

1. Kiểm tra cập nhật

A close up of a black and white text

Description automatically generated

1. Cài đặt thư viện



1. Tạo file cpp



1. Viết code khởi tạo chương trình

A screenshot of a computer program

Description automatically generated

1. Biên dịch chương trình

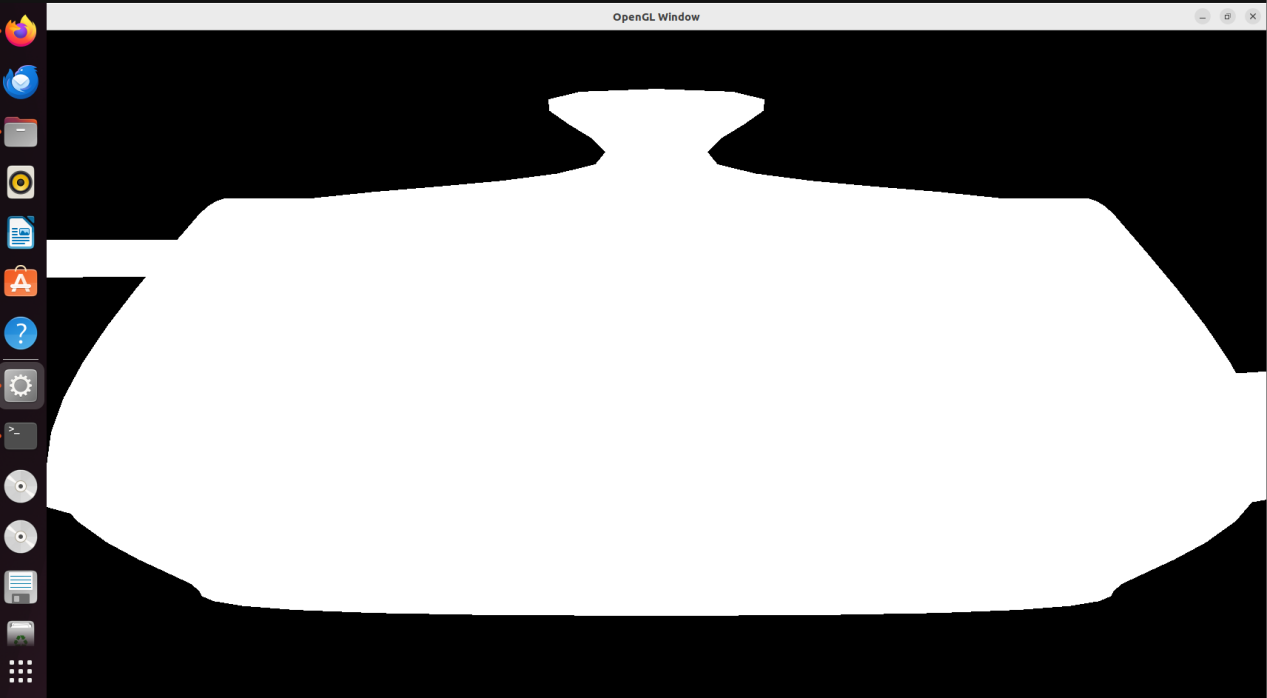
A screenshot of a computer

Description automatically generated

1. Chạy chương trình



1. Kết quả



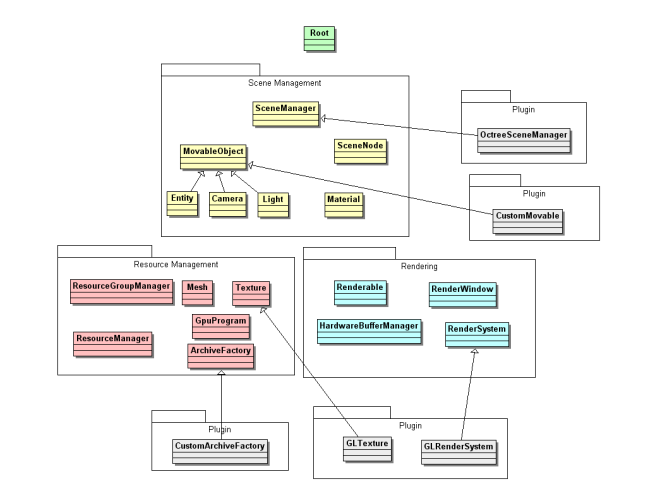
## Giới thiệu về OGRE

OGRE là một khung làm việc kết xuất đồ họa 3D mã nguồn mở, xây dựng trên nền tảng OpenGL. Kiến trúc của OGRE tập trung vào việc tạo ra hình ảnh đồ họa 3D đẹp mắt với sự đa dạng và linh hoạt.

## Ogre3d-logo

### Ưu điểm và tính năng của OGNE

OGRE không chỉ mang lại ưu điểm với khả năng kết xuất cảnh 3D, hỗ trợ ánh sáng và chế độ xem camera đa dạng, mà còn cung cấp tính linh hoạt cho nhà phát triển. Tuy nhiên, cũng cần xem xét các hạn chế và nhược điểm của OGRE trong quá trình phát triển để có quyết định đúng đắn.



Hình 12 Kiến trúc ONGE



Hình 13 Minh họa ONGE 1



Hình 14 Minh họa ONGE 2

### Demo

1. Kiểm tra cập nhật



1. Cài đặt thư viện



1. Tạo file main.cpp



1. Viết chương trình

A screenshot of a computer program

Description automatically generated

1. Biên dịch chương trình



1. Chạy chương trình



# Công cụ và ứng dụng trực quan hóa

## GNUplot

Một công cụ mã nguồn mở có thể tạo ra các biểu đồ hai và ba chiều của dữ liệu và hàm toán học. Tên của nó có GNU nhưng không liên quan đến dự án GNU, mặc dù nó có giấy phép mã nguồn mở riêng của nó

Giấy phép này cho phép cấp phép thương mại cho các tệp nhị phân; cung cấp quyền cấp phép bằng sáng chế không thu phí hiện đại; và khả năng cho các tác phẩm liên kết sử dụng các giấy phép khác, bao gồm cả giấy phép thương mại.

• GNUplot được phát hành lần đầu vào năm 1986.

• Những người tạo ra GNUplot bao gồm Thomas Williams, Colin Kelley và nhiều người khác.

• GNUplot không phải là một phần của Dự án GNU

• . Về giấy phép, GNUplot sử dụng giấy phép Eclipse Public License.

A black background with a black square

Description automatically generated with medium confidence

Hình 15 Logo Gnuplot

### Tính năng

• Các định dạng đầu ra: GNUplot có thể hiển thị trực tiếp trên cửa sổ X-window, nhưng cũng có thể tạo ra các tệp ảnh cứng ở nhiều định dạng như PNG, JPEG, SVG và EPS1.

• Các phần mềm toán học: Nhiều phần mềm toán học (như Octave, Maxima và SAGE) sử dụng GNUplot cho nhu cầu vẽ biểu đồ của chúng1.

• Hệ thống trợ giúp: GNUplot có một hệ thống trợ giúp rộng lớn được tích hợp trong nó, có thể được truy cập bằng cách gõ gnuplot> help

• Lệnh plot: Lệnh plot của GNUplot là lệnh mạnh nhất, và có thể được sử dụng để vẽ nội dung của các tệp dữ liệu và hàm toán học. Định dạng đầu ra có thể được thay đổi bằng cách sử dụng lệnh set term. Một trong những đầu cuối như vậy là đầu cuối ASCII, được hiển thị dưới đây. GNUplot cũng có thể vẽ các hình ảnh trong xfig.

### Một số ảnh minh họa về GNUplot

Hình 16 Minh họa Gnuplot 1

Hình 17 Minh họa GNUplot 2

Hình 18 Minh họa GNUplot 3

## Grace/XMGR

Grace là một công cụ vẽ biểu đồ 2D WYSIWYG cho hệ thống X-Window và được triển khai bằng Motif1

• Grace được phát hành lần đầu vào năm 1991.

• Những người tạo ra Grace bao gồm Paul Turner ở Portland, Oregon3, người tiếp tục phát triển cho đến phiên bản 4.00.

• Năm 1996, việc phát triển được tiếp tục bởi nhóm phát triển ACE/gr, do Evgeny Stambulchik dẫn dắt tại Viện Khoa học Weizmann, Israel.

• Về giấy phép, Grace sử dụng giấy phép GNU General Public License (GPL)

Giấy phép Công cộng GNU (GPL) là một giấy phép phần mềm miễn phí, copyleft, nhằm đảm bảo quyền tự do chia sẻ và thay đổi tất cả các phiên bản của một chương trình, đảm bảo rằng nó vẫn là phần mềm tự do cho tất cả người dùng

Hình 19 Logo Grace

### Tính năng

• Định dạng đầu ra: Bao gồm đầu ra sang định dạng EPS, PDF và SVG.

• Tính linh hoạt của đồ thị: Số lượng đồ thị không giới hạn, số lượng đường cong trên đồ thị không giới hạn, đánh dấu màu và chú thích văn bản1.

• Khớp đường cong: Khớp nhỏ nhất bình phương tuyến tính và phi tuyến, tính toán sai số, và hạn chế khu vực.

• Phân tích: Bao gồm FFT, tích phân và vi phân, biểu đồ, spline và tích phân.

• Định dạng dữ liệu: Đọc các tệp netCDF.

• Lập trình: Thư viện lập trình được tích hợp, các hàm toán học, và các hàm do người dùng định nghĩa thông qua các mô-đun có thể tải.

### Một số ảnh minh họa về Grace/XMGR

Hình 20 Minh họa Grace 1

Hình 21 Minh họa Grace 2

Hình 22 Minh họa Grace 3

## Xfig

là một trình chỉnh sửa đồ họa vector mã nguồn mở chạy dưới hệ thống X-window . Xfig được sử dụng để vẽ các hình ảnh sẵn sàng cho máy ảnh.

• Giấy phép: Xfig được phát hành theo Giấy phép Phần mềm Tự do GNU.

• Tác giả: Xfig được tạo ra bởi Supoj Sutanthavibul vào tháng 5 năm 19853.

• Năm phát hành: Phiên bản đầu tiên của Xfig được phát hành vào năm 1985.

Thông tin phần mềm: Xfig là một trình chỉnh sửa đồ họa vector mã nguồn mở.

Giấy phép Công cộng GNU (GPL) là một giấy phép phần mềm miễn phí, copyleft, nhằm đảm bảo quyền tự do chia sẻ và thay đổi tất cả các phiên bản của một chương trình, đảm bảo rằng nó vẫn là phần mềm tự do cho tất cả người dùng

Hình 23 Logo Xfig

### Tính năng

• Xfig hỗ trợ vẽ các hình ảnh sẵn sàng cho máy ảnh,

• vẽ các sơ đồ kỹ thuật và các minh họa kỹ thuật khác chứa hộp, đường, hình tròn, hình elip, đường cong spline và văn bản.

• Nó cũng cho phép xuất hình ảnh thành các định dạng tệp khác nhau như JPEG, EPS và LATEX.

• Giống như hầu hết các công cụ UNIX khác, xfig, lưu nội dung vẽ của nó dưới dạng tệp chỉ chứa văn bản (được gọi là tệp ‘.fig’). Xfig cũng có thể nhập các hình

### Một số hình minh họa về Xfig

Hình 24 Minh họa Xfix 1

Hình 25 Minh họa Xfig 2

Hình 26 Minh họa Xfig 3

## FreeType

là một thư viện phần mềm mã nguồn mở thực hiện chức năng tạo hình phông chữ. FreeType đơn giản hóa quy trình này bằng cách cung cấp một giao diện dễ dàng và đồng nhất để truy cập vào nội dung của các tệp mô tả phông chữ1. Một dự án liên quan khác là Pango, đây là một công cụ hiển thị văn bản đa ngôn ngữ mã nguồn mở.

* Giấy phép: FreeType sử dụng giấy phép GNU General Public License (GPL).
* Tác giả: FreeType được phát triển bởi một nhóm các nhà phát triển do David Turner, Robert Wilhelm và Werner Lemberg lãnh đạo.
* Năm phát hành: Phiên bản đầu tiên của FreeType được phát hành vào năm 19963.

Hình 27 Logo FreeType

### Tính năng

FreeType cung cấp các chức năng như tạo hình phông chữ, hỗ trợ nhiều định dạng phông chữ khác nhau, và cung cấp một API dễ sử dụng để truy cập và xử lý các tệp phông chữ4. Nó cũng hỗ trợ chức năng tạo hình phông chữ chống răng cưa4. Một ví dụ về việc sử dụng FreeType để hiển thị các ký tự phông chữ dưới dạng các dải pixel được hiển thị trong danh sách

### Một số hình minh họa FreeType

Hình 28 Minh họa FreeType 1

Hình 29 Minh họa FreeType 2

## Anti-Grain-Geimetry (AGG)

là một thư viện đồ họa mã nguồn mở được viết bằng C++. AGG cung cấp một công cụ hiển thị tạo ra hình ảnh pixel chính xác trong bộ nhớ từ dữ liệu vector.

Ngoài ra, AGG còn hỗ trợ chống răng cưa, chất lượng cao, hiệu suất cao và ổn định số học (quan trọng cho đầu vào suy biến). AGG hỗ trợ vẽ các đa giác tùy ý với các kiểu đường khác nhau, và kiểu đường. Nó sử dụng thư viện General Polygon Clipper để thực hiện các phép toán Boolean trên đa giác.

• Tác giả: Maxim Shemanarev

• Giấy phép: AGG được phát hành dưới các điều khoản và điều kiện của giấy phép GNU GPL. Tuy nhiên, phiên bản 2.4 của AGG vẫn có sẵn dưới giấy phép BSD 3-clause và gần như giống hệt với phiên bản 2.5.

• Năm phát hành: Phiên bản ổn định 2.5.0 được phát hành vào ngày 1 tháng 10 năm 2006.

Giấy phép BSD 3-Clause, còn được gọi là “New BSD License” hoặc “Modified BSD License”, là một giấy phép phần mềm tự do linh hoạt, cho phép phần mềm được sử dụng cho bất kỳ mục đích nào, được sửa đổi và được phân phối lại dưới dạng đã sửa đổi hoặc không sửa đổi, với ít yêu cầu tối thiểu

Hình 30 Logo AGG

### Tính năng

• Chống răng cưa (Anti-Aliasing): giúp tạo ra hình ảnh mượt mà, không bị hiện tượng răng cưa.

• Độ chính xác Subpixel: cho phép vẽ với độ chính xác cao hơn nhiều so với độ phân giải pixel.

• Chất lượng cao nhất có thể: AGG tập trung vào việc tạo ra hình ảnh 2D chất lượng cao.

• Hiệu suất cao: AGG được thiết kế để hoạt động nhanh và hiệu quả.

• Độc lập với nền tảng và tương thích: AGG có thể hoạt động trên nhiều hệ điều hành và môi trường lập trình khác nhau.

• Linh hoạt và mở rộng: AGG cho phép bạn thay thế hoặc thêm vào các phần của thư viện nếu cần.

• Thiết kế nhẹ nhàng: AGG không phụ thuộc vào thư viện ngoại vi và không sử dụng STL của C++ trong việc triển khai thuật toán cơ bản.

• Độ tin cậy và ổn định (bao gồm cả ổn định số học): AGG được thiết kế để hoạt động ổn định và không gây ra lỗi.

### Một số ảnh minh họa AGG

Hình 31 Minh họa AGG 1

Hình 32 Minh họa AGG 2

Hình 33 Minh họa AGG 3

# Công nghệ hiện đại và thư viện

## GD (Graphics drawing)

### GD là gì

GD là một thư viện mã nguồn mở để các lập trình viên tạo hình ảnh động. GD được viết bằng C và "trình bao bọc" có sẵn cho Perl, PHP, Ruby và nhiều liên kết khác.GD thường được sử dụng để tạo biểu đồ, đồ họa, hình thu nhỏ và hầu hết mọi thứ khác một cách nhanh chóng. Nó có trọng lượng nhẹ và phù hợp với các mục đích sử dụng như phát triển web, nhúng hoặc bất kỳ mục đích sử dụng nào khác mà bạn có thể cần.Nó hỗ trợ độ trong suốt, pha trộn, chuyển đổi hình ảnh và các bộ lọc khác nhau. Thiết kế của nó cho phép bổ sung các tính năng tùy chỉnh một cách rất thân thiện. Thư viện ban đầu được phát triển bởi Thomas Boutell và hiện được Pierre Joye và nhiều người đóng góp có giá trị duy trì, dưới sự bảo trợ của PHP.net. GD ban đầu là viết tắt của "GIF Draw". Tuy nhiên, kể từ khi giấy phép Unisys bị thu hồi, nó đã được viết tắt một cách không chính thức là "Graphics Draw". GD được sử dụng rộng rãi với PHP, trong đó phiên bản sửa đổi hỗ trợ các tính năng bổ sung được bao gồm theo mặc định kể từ PHP 4.3 và là một tùy chọn trước đó. Kể từ PHP 5.3, phiên bản hệ thống của GD cũng có thể được sử dụng để có được các tính năng bổ sung mà trước đây chỉ có ở phiên bản đi kèm của GD.

* Các ngôn ngữ được hỗ trợ:

C, PHP, Perl, Python, OCaml, Tcl, Lua, Pascal, GNU Octave, Rexx, Ruby and Go.

* Trình biên dịch:

Ngôn ngữ lập trình gốc của nó là ANSI C, nhưng nó có giao diện cho nhiều ngôn ngữ lập trình khác,

* License:

BSD-2-Clause

PHP-License

### Cách sử dụng thư viện GD

1. Cập nhật danh sách các gói phần mềm:



1. Cài đặt thư viện libgd:



1. Cấu hình php.ini



1. Dùng lệnh Ctrl + w và tìm cụm “;extension=gd” bỏ dấu ; ở đằng trước.

A screenshot of a computer

Description automatically generated

1. Tạo 1 file php ở đường dẫn thư mục /var/www của apache2

ở đây tạo thư mục demo và có file index.php bên trong



1. Dưới đây là đoạn code demo đơn giản tạo ra một hình tròn:

A screenshot of a computer

Description automatically generated

1. Kết quả:

A screenshot of a computer

Description automatically generated

* **Vẽ barchart**

A computer screen shot of a computer code

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a program code

Description automatically generated

**Kết quả**

A screenshot of a computer

Description automatically generated

## Geomview

### Geomview là gì?

Geomview là một chương trình đồ họa 3D được sử dụng để hiển thị và tương tác với các đối tượng hình học trong không gian ba chiều. Nó thường được sử dụng trong ngữ cảnh của các ứng dụng và nghiên cứu về hình học và toán học.

Geomview được viết bằng ngôn ngữ lập trình C và sử dụng Toolkit Geomview (libgeomview) để xây dựng giao diện đồ họa.

Được phát triển và nghiên cứu tại The Geometry Center – là một trung tâm nghiên cứu và giáo dục toán học tại đại học Minesota

Geomview được xây dựng trên Thư viện đồ họa hướng đối tượng (OOGL). Cảnh được hiển thị và các thuộc tính của các đối tượng trong đó có thể được xử lý bằng ngôn ngữ lệnh đồ họa

License:

[GNU Lesser General Public License (GPL)](http://www.gnu.org/copyleft/lesser.html)

### Cài đặt geomview

1. Cập nhật danh sách các gói phần mềm:



1. Cài đặt geomview:



1. Mở geomview:
   * Mặc định:



* + Khởi tạo mẫu có sẵn:



**Giao diện**:

A screenshot of a computer

Description automatically generated

Bảng bên trái là bảng điều khiển chính của Geomview; nó được gọi là Main panel. Bảng ở giữa là bảng Công cụ và dùng để chọn các loại chuyển động khác nhau. Cửa sổ bên phải là cửa sổ camera và trong đó bạn thấy một khối tứ diện lớn và một khối mười hai mặt bị che khuất một phần bởi khối tứ diện.

A screenshot of a computer

Description automatically generated

Hình 34 Main panel

A screenshot of a computer

Description automatically generated

Hình 35 Tool panel

A screenshot of a computer

Description automatically generated

Hình 36 Camera

## Hippodraw

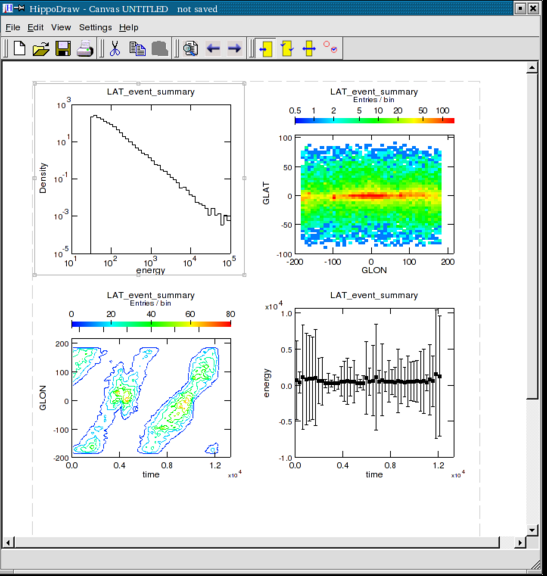
### Hippodraw là gì ?

HippoDraw là gói phân tích dữ liệu thống kê hướng đối tượng được viết bằng C++, với sự tương tác của người dùng thông qua GUI dựa trên Qt và giao diện có thể viết bằng Python.

Nó được phát triển bởi Paul Kunz tại SLAC, chủ yếu để phân tích và trình bày dữ liệu vật lý hạt và vật lý thiên văn, nhưng có thể được sử dụng tốt như nhau trong các lĩnh vực khác nơi việc xử lý dữ liệu là quan trọng.

License: GNU GENERAL PUBLIC LICENSE

### Giao diện



Hình 37 Minh họa hippodraw

## Ggobi

### Ggobi là gì ?

GGobi (pronounced "gee-gobi") là một phần mềm được phát triển để hiển thị và tương tác với dữ liệu thống kê và đồ thị đa biến. GGobi chủ yếu được sử dụng trong lĩnh vực thống kê và khám phá dữ liệu để giúp người sử dụng hiểu rõ hơn về mối quan hệ giữa các biến trong dữ liệu đa biến.

Lịch sử: GGobi là cái tên được đặt sau phiên bản XGobi. Nó được phát triển dưới sự bồi dưỡng của Bellcore. Bellcore là phòng thí nghiệm nghiên cứu của tất cả các công ty viễn thông địa phương vào cuối những năm 80 và đầu những năm 90. Để có thể phát hành phần mềm, nó phải được sự chấp thuận của các luật sư, những người cũng nhất quyết yêu cầu một cái tên.

Ban đầu XGobi được gọi là Xdataviewer, vì nó được phát triển từ một phần mềm cũ hơn có tên Dataviewer, được viết bằng Lisp bởi Andreas Buja, Catherine Hurley, John McDonald và Werner Stuetzle, khi còn học tại Đại học Washington, Seattle. Các luật sư phản đối cái tên này (Dataviewer quá chung chung và được quá nhiều người khác sử dụng) và khẳng định chúng tôi đã chọn một cái tên mới. Tất cả những cái tên hợp lý đều bị luật sư từ chối nên Debby và Andreas bắt đầu đưa ra những cái tên vô lý đó là tên của sa mạc(Gobi) vì mục đích chính của phần mềm giống với cát xoáy của sa mạc. Gobi là cái tên sa mạc mà tất cả đều ưa thích.

Các loại license

Common Public License

GNU Public License

BSD License

Mục đích của việc tạo rag GGobi là trực quan hóa dữ liệu thông qua việc tìm cách vẽ những bức tranh về dữ liệu nhiều chiều. Nó đã phát triển để xử lý các loại dữ liệu khác nhau, bao gồm các giá trị bị thiếu và dữ liệu mạng/biểu đồ.

### Cài đặt ggobi

Cài đặt ggobi:



A white rectangular object with a black border

Description automatically generated with medium confidence

Hình 38 Giao diện ggobi

A screenshot of a computer

Description automatically generated

Hình 39 Mở một fiile dữ liệu

A screenshot of a computer

Description automatically generated

Hình 40 Giao diện kết quả

ở mục display có thể lựa chọn New Scatterploy Matrix để hiển thị dạng ma trận:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 41 Giao diện kết quả

# Ứng dụng thực tế và kết luận

## Paraview

là một phần mềm mã nguồn mở hỗ trợ phân tích và hiển thị dữ liệu lớn thông qua việc sử dụng tài nguyên tính toán phân tán.

A black text on a white background

Description automatically generated

Hình 42 Logo Paraview

A screenshot of a computer

Description automatically generated

Hình 43 Giao diện chính

### Các tính năng quan trọng của Paraview

+ **Hỗ trợ cho dữ liệu quy mô tera:** ParaView có khả năng hỗ trợ xử lý và hiển thị dữ liệu có quy mô lớn, đến hàng tera.

+ **Hỗ trợ cho dữ liệu có cấu trúc:** Trong lĩnh vực trực quan hóa và phân tích khoa học, dữ liệu có cấu trúc đề cập đến dữ liệu hiện diện và thu thập trên các lưới (i) đều đặn chữ nhật, (ii) không đều đặn chữ nhật và (iii) chữ nhật uốn cong.

+ **Sự nhất quán:** Các hoạt động xử lý như bộ lọc có thể tạo ra các bộ dữ liệu, cho phép chuỗi các hoạt động xử lý.

+ **Trích xuất đường đồng mức:** Các bề mặt đồng mức và đường đồng đẳng có thể được trích xuất từ dữ liệu.

+ **Cắt ghép:** Một khu vực con có thể được cắt ghép. Việc cắt ghép có thể là một mặt phẳng (được chỉ định như một ngưỡng) hoặc một khối quan tâm (chỉ dành cho dữ liệu có cấu trúc).

+ **Tích hợp Python:**

. ParaView được triển khai bằng cách sử dụng khung Qt, do đó rất dễ di động.

. Nó sử dụng MPI và có thể chạy trên các nền tảng tính toán phân tán để phân tích các bộ dữ liệu có quy mô lớn.

. Nó được triển khai dưới dạng một phần mềm máy chủ-khách.

. Sử dụng Bộ công cụ Trực quan hóa (VTK), ParaView thực hiện một mô hình cấp độ chi tiết (LOD) để duy trì tốc độ khung hình cao khi hiển thị các mô hình lớn.

. Chúng ta có thể xây dựng một đối tượng 3D trong ParaView bằng cách thêm hình học nguyên từ menu "“Sources”".

. Một đối tượng được tạo bao gồm một hình trụ, một hình cầu và một hộp, các đối tượng có thể được cắt bằng một mặt cắt như được hiển thị trong hình sau:

A screenshot of a computer

Description automatically generated

Hình 44A screenshot of a computer

Description automatically generated Minh họa Paraview 1

A screenshot of a computer

Description automatically generated

Hình 45 Minh họa Paraview 2

Hình 46A screenshot of a computer

Description automatically generated Minh họa Paraview 3

## OpenDX (Open Data Explorer)

là một phần mềm trực quan hóa dữ liệu khoa học được viết bởi IBM và có sẵn dưới dạng mã nguồn mở, nó không chỉ được sử dụng để hiển thị dữ liệu mà còn để tạo ứng dụng tương tác dựa trên sự tương tác của người dùng. Người dùng có thể tương tác với hiển thị dữ liệu hiện tại bằng cách sử dụng nhiều công cụ tương tác có thể là trực tiếp (như quay góc nhìn hiện tại) hoặc gián tiếp (như áp dụng bộ lọc).

A close-up of a pair of sunglasses

Description automatically generated

Hình 47 Logo OpenDX

### Một số tính năng độc đáo

+ **Data prompter:** Giao diện người dùng để mô tả dữ liệu cần đưa vào OpenDX.

+ **Data model:** Các trường dữ liệu, đối tượng hình học và quy tắc mô tả dữ liệu.

+ **Data browser:** Giao diện người dùng để xem tệp dữ liệu.

+ **Ngôn ngữ kịch bản.**

+ **Visual program editor:** Giao diện đồ họa để tạo và chỉnh sửa các mạng (visual programs).

+ **Modules:** Các khối tạo thành mạng chương trình hình ảnh. Mỗi module thực hiện một số hành động.

+ **Module builder.**

+ **Control panel.**

A screenshot of a computer program

Description automatically generated+ **Main window:** Cửa sổ hiển thị.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

* Như đã đề cập ở trên, khái niệm visual program hoặc network của OpenDX làm nó nổi bật hơn so với các phần mềm trực quan hóa dữ liệu khác.
* Bằng cách sử dụng mạng, người dùng có thể hiển thị cả dữ liệu quan sát và dữ liệu mô phỏng. Các module, nhiều trong số đó thực hiện các biến đổi và bộ lọc cơ bản, được gửi kèm trong OpenDX và có thể được người dùng sử dụng để tạo ra một mạng nhanh chóng.
* Hơn nữa, trình chỉnh sửa visual program có thể được sử dụng để tạo ra visual program một cách dễ dàng hơn.

### Nhập dữ liệu trong OpenDX

OpenDX có khả năng nhập dữ liệu người dùng theo nhiều cách khác nhau:

1. **Người nhập mảng chung:** Cho phép nhập dữ liệu từ mảng chung, hỗ trợ việc đưa vào OpenDX dữ liệu từ nhiều nguồn khác nhau.
2. **Nhập tệp NetCDF, CDF, HDF:** Hỗ trợ việc nhập dữ liệu từ các tệp định dạng NetCDF, CDF, và HDF, phổ biến trong khoa học dữ liệu và trực quan hóa.
3. **Nhập từ bảng tính OpenDX:** Cho phép nhập dữ liệu từ các bảng tính được tạo và duy trì trong OpenDX.
4. **Mô-đun Nhập OpenDX:** Cung cấp khả năng nhập dữ liệu từ các định dạng được hỗ trợ vào OpenDX thông qua một mô-đun nhập dữ liệu.
5. **Đọc hình ảnh để nhập tệp TIFF:** Hỗ trợ việc nhập dữ liệu từ tệp hình ảnh TIFF sử dụng chức năng ReadImage.
6. **Đọc định dạng native của OpenDX:** Cho phép đọc dữ liệu từ định dạng nguyên bản của OpenDX, giúp bảo toàn thông tin đặc trưng của dữ liệu.
7. **Sử dụng gis2dx:** Gis2dx là một công cụ được sử dụng để chuyển đổi dữ liệu từ hệ thống thông tin địa lý (GIS) sang định dạng OpenDX, giúp tích hợp dữ liệu địa lý vào quá trình trực quan hóa của OpenDX.

### Quy trình chung trong OpenDX

Các bước để trực quan hóa dữ liệu bằng OpenDX có thể được chia thành các bước sau:

1. **Thu thập dữ liệu:** Bao gồm việc xác định nguồn dữ liệu, cơ sở dữ liệu, hoặc các tệp dữ liệu mà bạn muốn khám phá và hiển thị.
2. **Nhập dữ liệu:** Sử dụng các công cụ và mô-đun nhập dữ liệu có sẵn để chuyển đổi dữ liệu từ định dạng nguồn sang định dạng mà OpenDX có thể xử lý.
3. **Xác định hình thức và khái niệm trực quan:** Xác định loại biểu đồ, biểu đồ, hay mô hình 3D mà bạn muốn tạo ra. Đồng thời, đặt ra những khái niệm trực quan để chú thích và giải thích dữ liệu.
4. **Xác định tương tác của người dùng:** Xác định các công cụ tương tác, điều khiển, hoặc bảng điều khiển mà họ có thể sử dụng để thay đổi góc nhìn, áp dụng bộ lọc, hay thực hiện các thao tác khác.
5. **Chuẩn bị đầu ra:** Có thể là việc lưu ảnh, tạo video, hay tích hợp trực tiếp vào các ứng dụng khác. Đảm bảo rằng đầu ra đáp ứng mục tiêu của bạn và giúp truyền đạt thông điệp bạn muốn chia sẻ từ dữ liệu.

**Để kết luận phần OpenDX bằng cách trình bày một ví dụ nhỏ về phân tích dữ liệu 2D:**

Giả sử chúng ta có một loạt dữ liệu 2D phụ thuộc vào vị trí và chúng ta muốn phát triển một môi trường tương tác để phân tích và hiển thị nó. Kích thước của dữ liệu là trung bình, ví dụ, 100x100, nhưng cũng có thể lớn. Chúng ta có thể sử dụng OpenDX để nhanh chóng xây dựng một chương trình hình ảnh sẽ thực hiện việc hiển thị tương tác. Dữ liệu ở trong một tệp văn bản với các giá trị số thực 100x100.

Chúng ta viết một công cụ nhập dữ liệu bằng cách sử dụng công cụ visual Data Import như sau:

A white screen with black text

Description automatically generated

Đây là công cụ nhập dữ liệu "rain.general". Tiếp theo, chúng ta tạo một chương trình hình ảnh bằng cách sử dụng trình chỉnh sửa chương trình hình ảnh đồ họa (VPE) như được thể hiện như hình:

A screenshot of a computer

Description automatically generatedDữ liệu chuyển từ mô-đun Nhập dữ liệu, đến AutoColor, đến Collector (kết hợp các điểm ảnh Isosurface) để tạo thành hình ảnh cuối cùng:

A screen shot of a computer

Description automatically generated

Như thấy trong hình, các điều khiển của mô-đun Isosurface được kiểm soát bằng các điều khiển đầu vào mà người dùng cuối có thể điều khiển. Hình ảnh được hiển thị có thể được quay, thu phóng và lưu vào một tệp trên đĩa.

## Tổng kết về các công cụ và ứng dụng trực quan hóa

### Paraview và VTK

**Ưu điểm:** Hỗ trợ xử lý dữ liệu terascale và dữ liệu có cấu trúc. Cung cấp tính nhất quán trong quá trình xử lý dữ liệu và trực quan.

**Ví dụ:** Đã thảo luận về các tính năng quan trọng như hỗ trợ dữ liệu lớn, cắt biên, và tích hợp Python.

### OpenDX

**Ưu điểm:** Sử dụng mô hình mạng để tạo và chỉnh sửa chương trình hình ảnh. Hỗ trợ nhiều tính năng độc đáo như Data prompter, Data browser, và Module builder.

**Ví dụ:** Đã thảo luận về quy trình nhập và trực quan hóa dữ liệu 2D thông qua OpenDX.

### Công cụ và ứng dụng khác

**OpenGL:** Đã thảo luận về tầm quan trọng của OpenGL trong trực quan hóa và các thư viện tiện ích như GLUT và GLUI.

**Ứng dụng có sẵn:** Đã đề cập đến một số ứng dụng có sẵn như ParaView, OpenDX, Geomview chuyên về trực quan hóa và kết xuất dữ liệu.

**Gói phân tích:** HippoDraw và GGobi được đề cập, chúng thực hiện phân tích thống kê.

**Công cụ tạo hình ảnh:** Đã giới thiệu các công cụ để tạo hình ảnh tĩnh và động từ dữ liệu như XFig, gnuplot, Inkscape và Graphviz dot.

* Tóm lại, các công cụ và ứng dụng trực quan hóa như GGobi, ParaView, OpenDX, cùng với các thư viện như OpenGL, đóng góp quan trọng vào quá trình hiểu và trình bày thông tin từ dữ liệu đa chiều và lớn. Các tính năng và ưu điểm của từng công cụ giúp người dùng nắm bắt thông tin một cách hiệu quả và sáng tạo.

## Chia sẻ nhận định và đề xuất hướng phát triển trong tương lại

### Nhận định

* Trực quan hóa dữ liệu đang ngày càng trở nên quan trọng và không thể thiếu trong nhiều lĩnh vực.
* Các công cụ như GGobi, ParaView và OpenDX đều mang lại những đóng góp quan trọng trong việc hiểu và trình bày thông tin từ dữ liệu phức tạp và lớn.
* Sự kết hợp giữa động, tương tác và khả năng xử lý lớn của các công cụ này đã mở ra nhiều cơ hội mới trong nghiên cứu, khoa học và quản lý dữ liệu.
* Tuy nhiên, với sự phát triển không ngừng của công nghệ và sự gia tăng về khối lượng dữ liệu, có một số thách thức và cơ hội cần được xem xét.

### Đề xuất

* **Tăng Cường Tính Di Động và Tính Thời Gian Thực:** Điều này sẽ hỗ trợ người dùng trong việc theo dõi và hiểu thông tin ngay lập tức khi dữ liệu thay đổi.
* **Tích Hợp Trí Tuệ Nhân Tạo và Học Máy:** Điều này sẽ giúp người dùng tiết kiệm thời gian và tăng hiệu suất trong việc đưa ra quyết định.
* **Mở Rộng Các Công Cụ Di Động:** Điều này sẽ giúp người dùng trực quan hóa dữ liệu mọi nơi và mọi lúc.
* **Tăng Cường An Toàn và Bảo Mật Dữ Liệu:** Đảm bảo rằng việc trực quan hóa dữ liệu không gây ra rủi ro về bảo mật và quyền riêng tư.
* **Hợp Nhất Các Công Cụ và Tiêu Chuẩn:** Điều này sẽ giúp tạo ra môi trường làm việc mạnh mẽ và hiệu quả hơn.
* **Phát Triển Cộng Đồng và Giáo Dục:** Điều này sẽ giúp mọi người hiểu rõ hơn về khả năng và tiềm năng của trực quan hóa dữ liệu.

# TÀI LIỆU

1. Sách HandbookofOpenSourceTools
2. Tài liệu trên trang chủ các tools
3. <https://sourceforge.net/>
4. https://github.com/