

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



SKY GARDEN

BÁO CÁO BÀI TẬP LỚN MÔN: LẬP TRÌNH ĐỒ HỌA

NGÀNH KHOA HỌC MÁY TÍNH

Lớp: DH18CS81

Nhóm sinh viên thực hiện: Phạm Chí Năng
Bùi Phương Tùng
Nguyễn Nguyệt Phương Bảo
Trương Thiên Phong

TP. HỒ CHÍ MINH, 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



SKY GARDEN

BÁO CÁO BÀI TẬP LỚN MÔN: LẬP TRÌNH ĐỒ HỌA
NGÀNH KHOA HỌC MÁY TÍNH

Lớp: DH18CS81

Nhóm sinh viên thực hiện: Phạm Chí Năng - 1851010076
Bùi Phương Tùng - 1851010151
Nguyễn Nguyệt Phương Bảo - 185101013
Trương Thiên Phong -1851010098

Giảng viên hướng dẫn: ThS. VÕ THỊ HỒNG TUYẾT

TP. HỒ CHÍ MINH, 2021

LỜI CẢM ƠN

Đầu tiên, cho phép chúng em xin gửi lời cảm ơn chân thành đến với Trường Đại học Mở Thành phố Hồ Chí Minh, Khoa Công Nghệ Thông Tin đã và đang tạo điều kiện đưa vào chương trình đào tạo bộ môn Lập Trình Đồ Họa. Chúng em xin gửi lời cảm ơn đặc biệt chân thành đến với Giảng viên bộ môn lần này – Cô Võ Thị Hồng Tuyết đã tạo điều kiện và hướng dẫn chúng em trong suốt quá trình thực hiện.

Do tình hình dịch bệnh nên việc được làm việc trực tiếp với nhau khó khăn, dẫn đến kết quả của quá trình làm việc không được tốt cho lắm, thông qua làm việc online mà chúng em có thêm nhiều kinh nghiệm để được phát triển cũng như tự tìm hiểu các vấn đề liên quan đến bài tập lần này.

Chúng em xin chân thành cảm ơn. Kính chúc quý Thầy (Cô) sẽ có thật nhiều sức khỏe và luôn thành công trong cuộc sống.

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

TÓM TẮT BÁO CÁO

Với bài tập lần này, chúng em sẽ giới thiệu cho quý Thầy (Cô) về một tựa game mà hầu như các bạn trẻ thế hệ trước đều biết, đó là tựa game “Khu vườn trên mây” – “Sky Garden”. Bài báo cáo sẽ giúp cho mọi người hiểu được những vấn đề liên quan xung quanh đến đồ họa nói chung và việc sử dụng OpenGL trong C++ nói riêng.

Ngôn ngữ được sử dụng ở đây đa phần là C/C++. Đối với việc lập trình game, hay các ứng dụng liên quan thì C++ có vẻ hỗ trợ tốt hơn và được ưu tiên sử dụng nhiều hơn. Bên cạnh đó, C++ còn được ứng dụng nhiều trong các ứng dụng có giao diện người dùng, trình duyệt web, ứng dụng tính toán và đồ họa...

Mục tiêu của môn học lần này:

- Nắm vững các kiến thức về xử lý đồ họa ba chiều.
- Hiểu rõ vai trò và cấu trúc của OpenGL.
- Hiểu và áp dụng được giao diện lập trình đồ họa OpenGL.

Bài báo cáo Đồ Án Ngành sẽ bao gồm 4 chương:

- Chương 1: Giới thiệu.
- Chương 2: Sky Garden.
- Chương 3: Kết luận.

MỤC LỤC

Lời cảm ơn.....	II
Nhận xét giáo viên.....	III
Tóm tắt báo cáo	IV
Danh mục từ viết tắt	VII
Danh mục hình vẽ.....	VIII
Danh mục bảng.....	XI
Lời mở đầu.....	XII
CHƯƠNG 1. GIỚI THIỆU.....	1
1.1. Giới thiệu đề tài.....	1
1.2. Mục tiêu đề tài.....	1
1.3. Giới hạn đề tài.....	1
1.4. Bố cục	2
CHƯƠNG 2. SKY GARDEN	3
2.1. Các kiến thức cơ bản đi kèm.....	3
2.1.1. Đồ họa 3D	3
2.1.2. Các hiệu ứng 3D.....	3
2.1.3. Các hệ tọa độ	6
2.1.4. Các phép chiếu	8
2.1.5. OpenGL.....	9
2.1.6. Các hàm callback thường hay được sử dụng	11
2.1.7. Cấu trúc lệnh chung để vẽ đối tượng	14
2.1.8. Màu sắc	17
2.1.9. Các phép biến đổi tọa độ	17
2.1.10. Cách khử đối tượng khuất.....	20

2.2.	Phương hướng giải quyết.....	21
2.2.1.	Ý tưởng được đưa ra	21
2.2.2.	Giải quyết vấn đề và ý tưởng	22
2.3.	Kết quả thực hiện demo	41
CHƯƠNG 3.	KẾT LUẬN	43
3.1.	Kết quả đạt được	43
3.2.	Ưu điểm.....	43
3.3.	Nhược điểm.....	43
3.4.	Hướng phát triển trong tương lai	44
	Tài liệu tham khảo	44

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Nghĩa của từ viết tắt
OpenGL	Open Graphics Library
PS	Photoshop
AI	Adobe Illustrator
3D	3-Dimension
GLUT/Glut	openGL Utility Toolkit

DANH MỤC HÌNH VẼ

Hình 2.1. Đối tượng 3D được biểu diễn dưới dạng 3 chiều [3]	3
Hình 2.2 Phép phối cảnh và phương pháp khử mặt khuất [3]	4
Hình 2.3 Màu sắc và tô bóng [3]	4
Hình 2.4 Ánh sáng và đổ bóng [3]	4
Hình 2.5 Dán ảnh [3]	5
Hình 2.6 Sương mù với hiệu ứng của không khí [3]	5
Hình 2.7 Khử răng cưa [3]	6
Hình 2.8 Hệ tọa độ 2D Cartesian [3]	6
Hình 2.9 Viewport có kích thước gấp 2 lần vùng cắt [3]	7
Hình 2.10 Hệ tọa độ 3D Cartesian [3]	8
Hình 2.11 Phép chiếu từ 3D sang 2D [3]	8
Hình 2.12 Phép chiếu vuông góc	9
Hình 2.13. Biểu tượng của OpenGL	9
Hình 2.14. Sơ đồ tương tác giữa OpenGL và các thành phần trong ứng dụng	10
Hình 2.15 Cấu trúc sử dụng GLUT	11
Hình 2.16 Sơ đồ hàm hiển thị	12
Hình 2.17 Hàm xử lý sự kiện window thay đổi kích thước	13
Hình 2.18. Kết quả của việc vẽ các điểm tạo thành Spring-Shaped Path [3]	14
Hình 2.19 Ví dụ về vẽ các đường thẳng trong GL_LINES [3]	15
Hình 2.20 Ví dụ về vẽ đường thẳng trong GL_LINE_STRIP [3]	15
Hình 2.21 Ví dụ về vẽ đường thẳng trong GL_LINE_LOOP [3]	16
Hình 2.22 Một số độ rộng của đường được vẽ trong OpenGL [3]	16
Hình 2.23 Một số kiểu đường được vẽ trong OpenGL [3]	17
Hình 2.24 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi quan sát	18
Hình 2.25 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi mô hình	18
Hình 2.26 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép chiếu ...	19
Hình 2.27 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi viewport	20

Hình 2.28 Ảnh minh họa về mặt trước và mặt sau của mộ đa giác [3]	20
Hình 2.29 Các chức năng chính trong game	22
Hình 2.30 Kích thước màn hình game so với thực tế.....	23
Hình 2.31 Màn hình chính trong game.....	23
Hình 2.32 Ví dụ về hiển thị tên các màn hình trong game 1	24
Hình 2.33 Ví dụ về hiển thị tên các màn hình trong game 2	24
Hình 2.34 Sơ đồ chức năng tại màn hình NewGame	24
Hình 2.35 Màn hình chính của NewGame	25
Hình 2.36 Xử lý intro	25
Hình 2.37 Quản lý các đối tượng trong “pot.h”	26
Hình 2.38 Vị trí xuất hiện của mũi tên, dùng để đặt chậu	27
Hình 2.39 Quá trình đặt chậu	27
Hình 2.40 Sau khi click vào mũi tên lần 1	28
Hình 2.41 Chọn chậu và show chậu lên vị trí chọn chậu	28
Hình 2.42 Sau khi đặt đầy đủ các loại chậu	29
Hình 2.43 Quản lý các đối tượng trong “flower.h”	30
Hình 2.44 Mũi tên xuất hiện để cho người dùng click vào chọn hạt giống	30
Hình 2.45 Quá trình chọn hoa	31
Hình 2.46 Chọn hoa và vị trí cần đặt hoa	31
Hình 2.47 Sau khi chọn hạt giống	31
Hình 2.48 Thời gian phát triển của một loại hoa.....	32
Hình 2.49 Giai đoạn phát triển của 3 loại hoa.....	32
Hình 2.50 Giao diện tưới nước	33
Hình 2.51 Quá trình di chuyển giữa màn hình Gameplay và Shop.....	33
Hình 2.52 Giao diện Shop	34
Hình 2.53 Các sự kiện có trong SHOP.....	35
Hình 2.54 Số lượng mặc định về mỗi loại chậu, hoa	35
Hình 2.55 Giao diện Mission.....	36
Hình 2.56 Quá trình di chuyển giữa màn hình Gameplay và Shop.....	36
Hình 2.57 Giao diện khi mở tầng thứ 2	37
Hình 2.58 Giao diện khi mở tầng thứ 3	37

Hình 2.59 Quá trình di chuyển giữa màn hình Gameplay và BAG	38
Hình 2.60 Giao diện BAG	38
Hình 2.61 Các sự kiện trong BAG	39
Hình 2.62 Quá trình xem giúp đỡ.....	40
Hình 2.63 Màn hình Help	40
Hình 2.64 Quá trình thoát trò chơi	41
Hình 2.65 Màn hình exit.....	41

DANH MỤC BẢNG

Bảng 1 Bảng giá trị màu trong OpenGL	17
Bảng 2 Bảng phân công công việc	42
Bảng 3 Bảng liệt kê chức năng và giao diện	43

LỜI MỞ ĐẦU

[1] Đồ họa máy tính 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 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...).

Thuộc nhóm học phần chuyên ngành hướng Đồ họa máy tính, môn học cung cấp các kiến thức cơ bản về quy trình xử lý đồ họa trên máy tính và các giải thuật cơ bản trong đồ họa hai chiều và ba chiều [2].

Trong bài báo cáo lần này, nhóm chúng tôi sẽ giới thiệu đến mọi người về Game “Sky Garden” – một game được phát triển dựa trên tựa game nổi tiếng của ZingPlay là “khu vườn trên mây”. Game được xây dựng thông qua phần mềm VisualStudio 2010, và ngôn ngữ được viết ở đây là C++.

Với bố cục rõ ràng, hình ảnh đa dạng và phong phú. Dù bạn là người chưa từng tiếp cận đến vấn đề này, nhưng sau khi đọc qua bài báo cáo này bạn có thể sẽ hiểu đầu đó được cách mà chúng tôi thực hiện. Trong quá trình biên soạn, mặc dù đã có cố gắng hết sức nhưng vẫn khó tránh khỏi những sai sót, rất mong quý Thầy(Cô) và quý bạn đọc có được những đóng góp chân thành để chúng tôi có thể phát triển thêm về sau này.

TRÂN TRỌNG CẢM ƠN

CHƯƠNG 1. GIỚI THIỆU

1.1. Giới thiệu đề tài

“ Khu vườn trên mây ” - có cốt truyện chính từ những câu chuyện cổ tích nổi tiếng của “Jack và Cây đậu thần” và đã được các thành viên trong Firebat Game Studio (thuộc công ty VNG) nghiên cứu và thực hiện, để tạo nên một sân chơi thực sự hấp dẫn và đa dạng cho người chơi Việt Nam.

Tựa game này có lối chơi khá đơn giản, với những thao tác cơ bản như trồng cây, tưới nước, bắt sâu,... người chơi sẽ trở thành một người làm vườn thực thụ thông qua việc chơi game này. Có thể xem đây là một trong những tựa game giả lập người dùng từ những thời đầu phát triển.

Từ những lý do đó, kết hợp với những kiến thức đã và đang học đối với bộ môn “Lập trình đồ họa”. Chúng tôi đã phát triển và tạo ra một tựa game cũng tương tự như “Khu vườn trên mây”, tựa game có tên “Sky Garden”. Tại đây cũng có những thao tác cơ bản như trồng cây, tưới nước, bắt sâu, mua bán... Nhưng việc sử dụng OpenGL để phát triển thì cũng là điều khó khăn mà chúng tôi phải đối mặt.

1.2. Mục tiêu đề tài

Mục tiêu đề tài khá đơn giản, được thông qua những tiêu chí sau:

- Nắm vững các kiến thức về xử lý đồ họa ba chiều.
- Hoàn thành các chức năng cơ bản: đặt chậu, trồng cây, tưới nước...
- Hiểu rõ vai trò và cấu trúc của OpenGL.
- Hiểu và áp dụng được giao diện lập trình đồ họa OpenGL.

1.3. Giới hạn đề tài

Bên cạnh đó, việc giới hạn đề tài cũng là một phần trở ngại trong việc phát triển trò chơi:

- Sử dụng C++ và OpenGL để phát triển.
- Các hình ảnh sử dụng trong game không thuộc về các nguồn internet, thay vào đó là các hình ảnh tự tay vẽ thông qua các phần mềm có sẵn như PS, AI.

1.4. Bố cục

Bố cục của bài báo cáo gồm 3 chương:

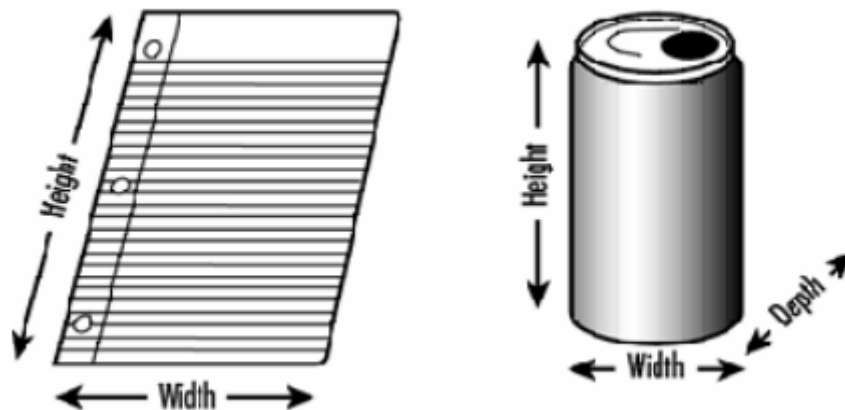
- Chương 1: Giới thiệu – Tại chương này, sẽ ghi lại ngắn gọn nhất và giới thiệu cho người đọc hiểu được những gì được giới thiệu trong bài báo cáo.
- Chương 2: Sky Garden – Chương này, người đọc sẽ đi sâu vào các vấn đề và các cách giải quyết bài toán một cách chi tiết nhất.
- Chương 3: Kết luận – Đây cũng là chương mà cho chúng ta biết được kết quả, cũng như là những ưu nhược điểm trong quá trình phát triển game

CHƯƠNG 2. SKY GARDEN

2.1. Các kiến thức cơ bản đi kèm.

2.1.1. Đồ họa 3D

3D là từ viết tắt của 3-Dimension (3 chiều) – Vậy đồ họa 3D chính là kỹ thuật dựng hình ảnh lên không gian 3 chiều, thể hiện các hiệu ứng đổ bóng vật lý bằng các phần mềm đồ họa vi tính như: After Effect, 3DS Max, Autodesk Maya... Kỹ thuật đồ họa 3D đòi hỏi người thiết kế cần phải có tư duy không gian và am hiểu hình học để tái hiện sinh động các đối tượng vật thể.



Hình 2.1. Đối tượng 3D được biểu diễn dưới dạng 3 chiều [3]

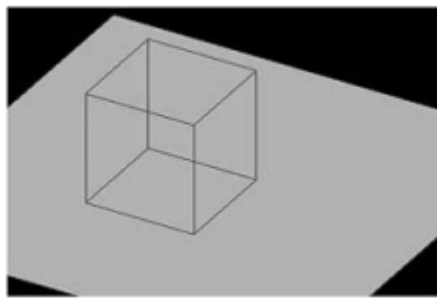
Công thức liên hệ giữa đồ họa 2D và 3D:

$$2D + \text{Perspective (Phép phối cảnh)} = 3D. \quad (1)$$

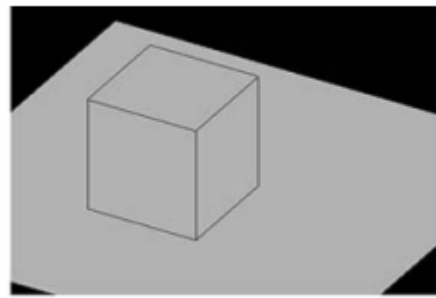
2.1.2. Các hiệu ứng 3D

Một số hiệu ứng 3D như:

- Phép phối cảnh và phương pháp khử mặt khuất.



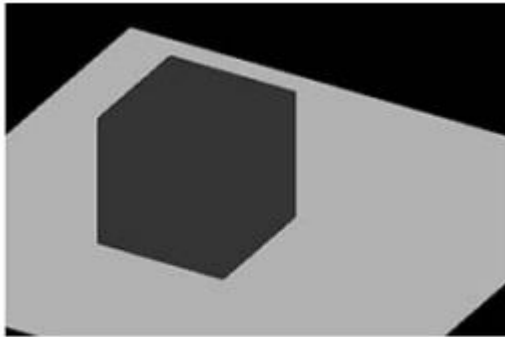
Chưa khử mặt khuất



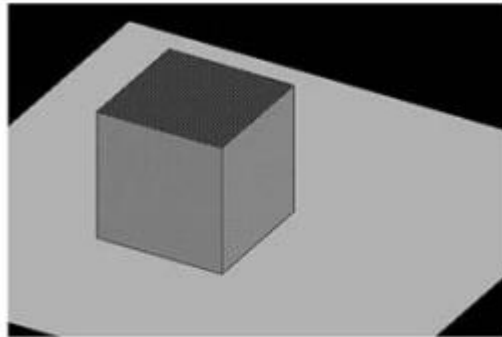
Sau khi khử mặt khuất

Hình 2.2 Phép phối cảnh và phương pháp khử mặt khuất [3]

- Màu sắc và tô bóng.



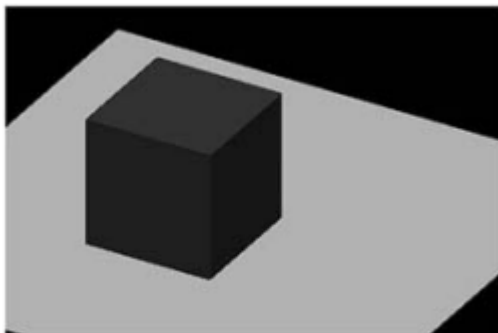
Các mặt tô cùng một màu



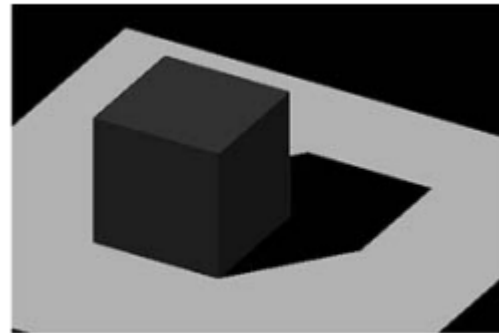
Các mặt tô màu khác nhau

Hình 2.3 Màu sắc và tô bóng [3]

- Ánh sáng và đổ bóng.



Ánh sáng chiếu đến hình hộp



Bóng của hình hộp in trên mặt bàn

Hình 2.4 Ánh sáng và đổ bóng [3]

- Dán ảnh.



Hình 2.5 Dán ảnh [3]

- Sương mù.

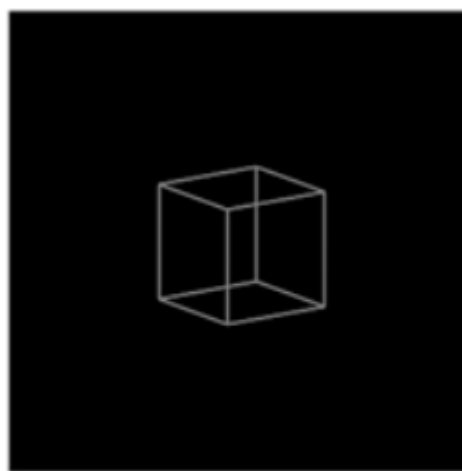


Hình 2.6 Sương mù với hiệu ứng của không khí [3]

- Pha màu và trong suốt.
- Khử răng cưa.



Chưa khử rỗng cửa



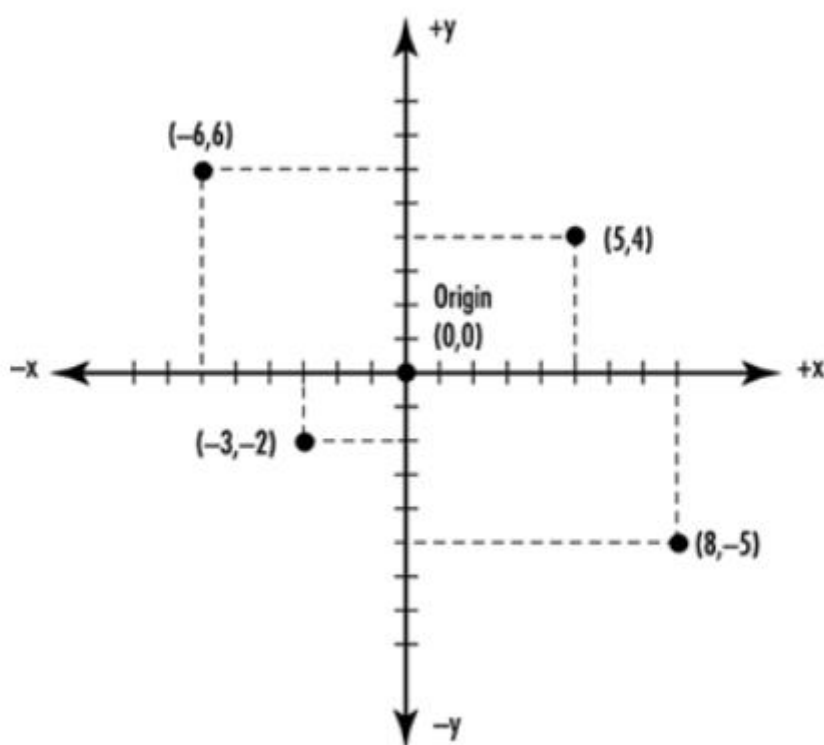
Sau khi khử rỗng cửa

Hình 2.7 Khử rỗng cửa [3]

2.1.3. Các hệ tọa độ

2.1.3.1. Khái niệm và hệ tọa độ 2D Cartesian

Được thể hiện được một đối tượng trên màn hình, trải qua nhiều phép biến đổi liên quan đến các hệ trục tọa độ.

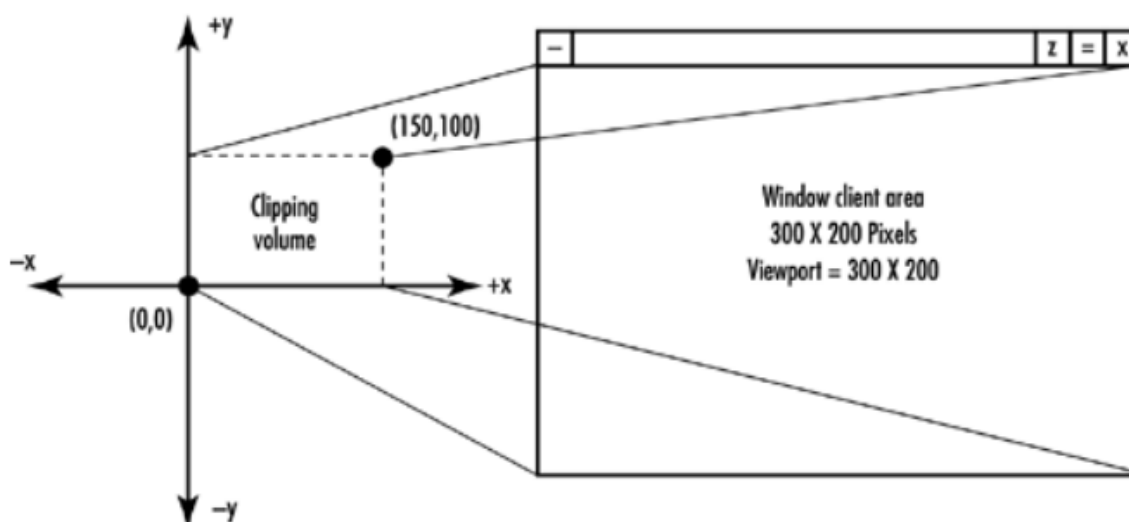


Hình 2.8 Hệ tọa độ 2D Cartesian [3]

2.1.3.2. Viewport và vertex

Viewport được ánh xạ từ hệ tọa độ vẽ vào hệ tọa độ window. Kích thước của vùng cắt khó phù hợp với kích thước của cửa sổ trên màn hình.

Cần xác định vùng cắt trong hệ tọa độ logic, sau đó chuyển đổi từ hệ tọa độ logic vào tọa độ cửa sổ. Phép biến đổi này được gọi là phép xác định viewport.

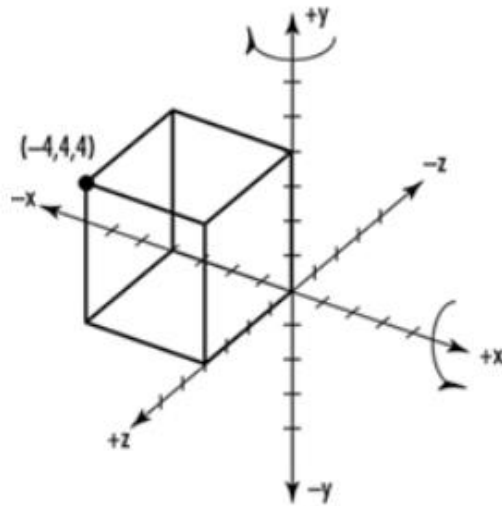


Hình 2.9 Viewport có kích thước gấp 2 lần vùng cắt [3]

Vertex được hiểu là một vị trí có mặt trong không gian. Hiểu một cách nôm na, các đối tượng 3D được xây dựng từ các thành phần nguyên tố. Các thành phần nguyên tố lại được xác định thông qua các vertex.

2.1.3.3. Hệ tọa độ 3D Cartesian

Tương tự như hệ tọa độ 2D Cartesian ta có được hệ tọa độ 3D Cartesian như sau:

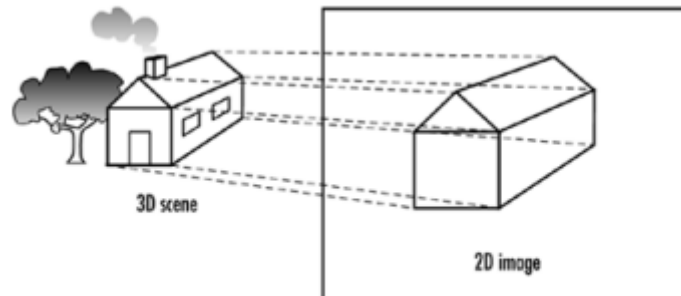


Hình 2.10 Hệ tọa độ 3D Cartesian [3]

2.1.4. Các phép chiếu

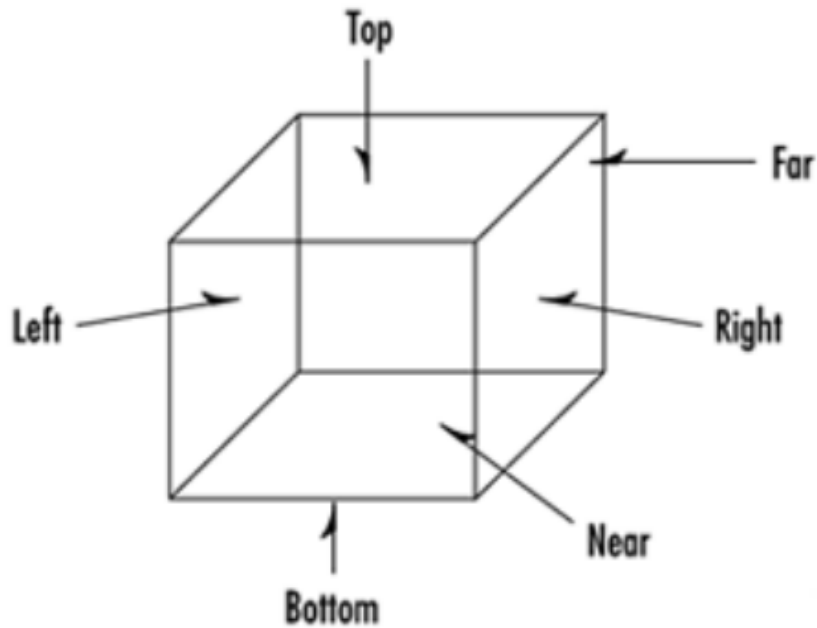
Phép biến đổi từ không gian 3 chiều sang không gian 2 chiều gọi là phép chiếu. Bên cạnh các hệ tọa độ 2D và 3D, ta còn có các phép chiếu như sau:

- Chiếu từ 3D sang 2D



Hình 2.11 Phép chiếu từ 3D sang 2D [3]

- Viewing volume : còn được gọi là không gian cắt. Khi thực hiện các phép chiếu, ta phải xác định các viewing volume. Chỉ những đối tượng nằm trong viewing volume mới được vẽ các màn hình [3].
- Phép chiếu vuông góc: là tổng hợp các phép chiếu có phương vuông góc với vật thể.



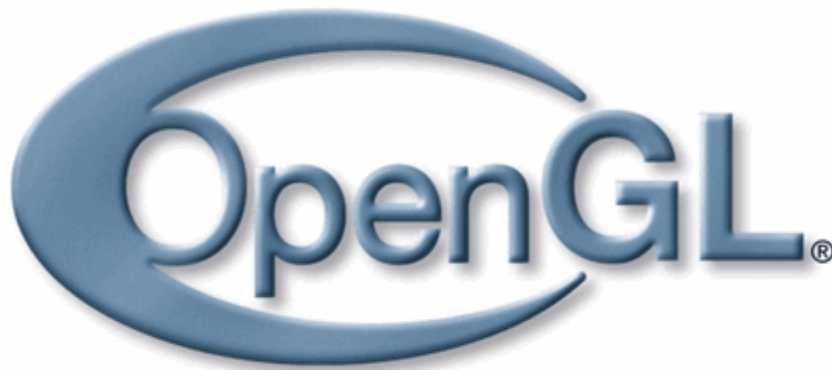
Hình 2.12 Phép chiếu vuông góc

- Phép chiếu phối cảnh.
- Frustum.

2.1.5. OpenGL

2.1.5.1. Tìm hiểu về OpenGL

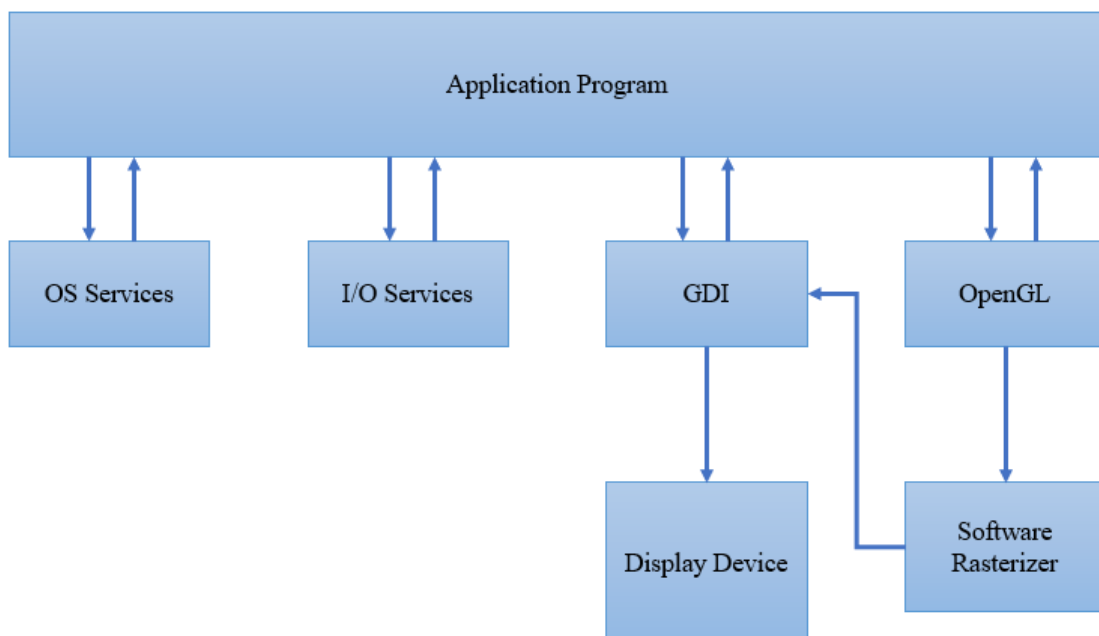
OpenGL (Open Graphics Library) là một API đa nền tảng, đa ngôn ngữ cho render đồ họa vector 2D và 3D. API thường được sử dụng để tương tác với bộ xử lý đồ họa (GPU), nhằm đạt được tốc độ render phần cứng.



Hình 2.13. Biểu tượng của OpenGL

Một số đặc điểm của OpenGL nên nhớ:

- OpenGL là API, không phải là một ngôn ngữ lập trình.
- Ngôn ngữ lập trình gốc làm việc với OpenGL là C.
- Bao gồm các thư viện động (DLL: Dynamic Linking Library) nằm trong thư mục hệ thống của Windows.
- OpenGL được thiết kế độc lập hệ điều hành. Code sử dụng OpenGL có thể chuyển đổi qua lại giữa các hệ điều hành.
- Vì lý do độc lập platform, OpenGL không cung cấp các hàm quản lý cửa sổ và màn hình cũng như các hàm xử lý sự kiện chuột và bàn phím.

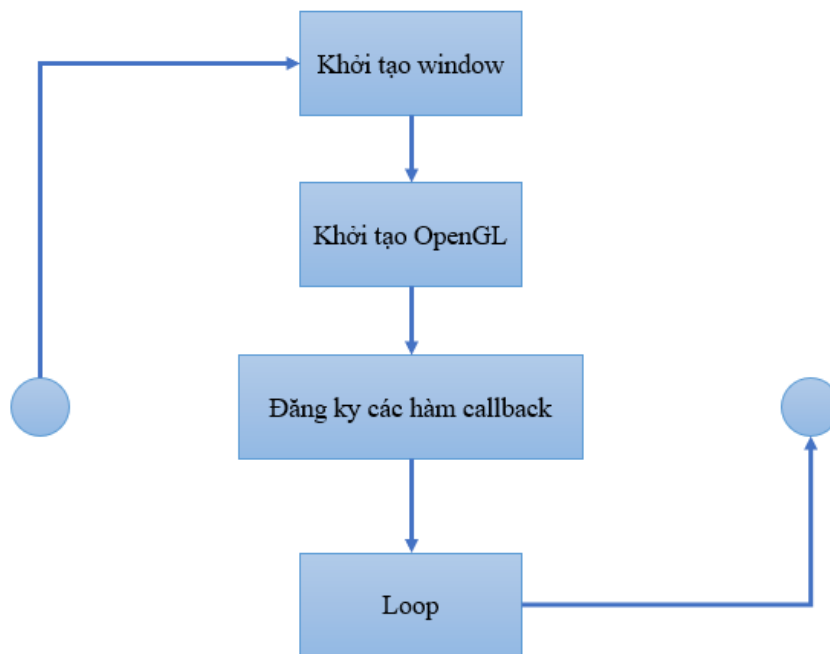


Hình 2.14. Sơ đồ tương tác giữa OpenGL và các thành phần trong ứng dụng

2.1.5.2. Tìm hiểu về Glut

GLUT (openGL Utility Toolkit) là một thư viện nhằm giúp chúng ta có thể nhanh chóng tạo ra các cửa sổ cũng như quản lý các sự kiện từ bàn phím, chuột mà không phải quan tâm quá nhiều tới các hàm trực tiếp của OpenGL vì GLUT sẽ gọi các hàm này thay cho chúng ta.

Cấu trúc chương trình sử dụng GLUT



Hình 2.15 Cấu trúc sử dụng GLUT

2.1.6. Các hàm callback thường hay được sử dụng

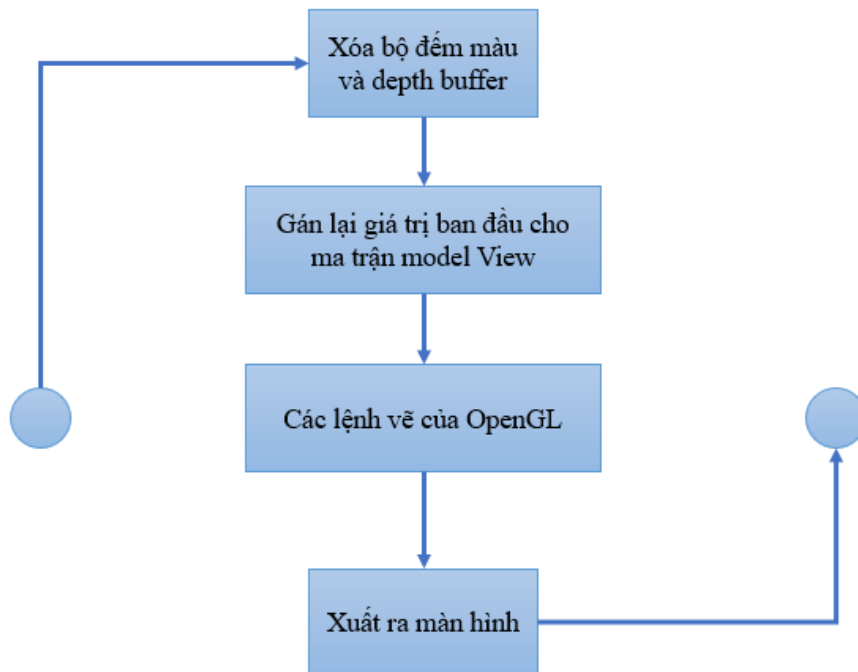
2.1.6.1. Hàm hiển thị.

Đăng ký hàm hiển thị:

```
void glutDisplayFunc (void (*func) (void));
```

Và hàm hiển thị:

```
void RenderScene(void) ;
```

Hình 2.16 Sơ đồ hàm hiển thị

2.1.6.2. Xóa bộ đệm màu ,depth buffer và gán lại giá trị ban đầu cho ma trận modelview

Để xóa bộ đệm màu và depth buffer, sử dụng:

```
glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
```

Để gán lại giá trị ban đầu cho ma trận modelview, sử dụng:

```
glMatrixMode(GL_MODELVIEW);

glLoadIdentity();
```

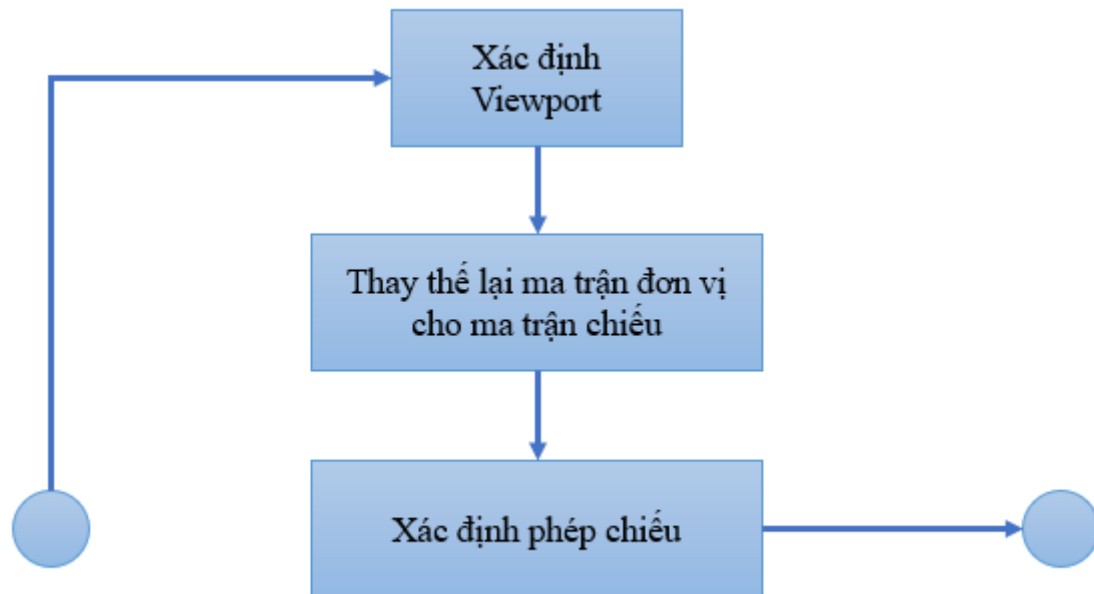
2.1.6.3. Hàm xử lý sự kiện window thay đổi kích thước.

Hàm đăng ký:

```
void glutReshapeFunc(void (*func)(int width, int height));
```

Hàm xử lý sự kiện window thay đổi kích thước:

```
void ChangeSize(GLsizei w, GLsizei h);
```



Hình 2.17 Hàm xử lý sự kiện window thay đổi kích thước

2.1.6.4. Hàm xử lý các sự kiện bàn phím.

Hàm đăng ký:

```
void glutKeyboardFunc(void (*func)(unsigned char key,
int x, int y));
```

Hàm xử lý các sự kiện bàn phím:

```
void keyboard(unsigned char key, int x, int y);
```

2.1.6.5. Hàm xử lý các sự kiện chuột.

Hàm đăng ký:

```
void glutMouseFunc(void (*func)(int button, int state,
int x, int y));
```

2.1.6.6. Hàm định thời.

Hàm đăng ký:

```
void glutTimerFunc(unsigned int msecs, void
(*func)(int value), value);
```

Hàm callback:

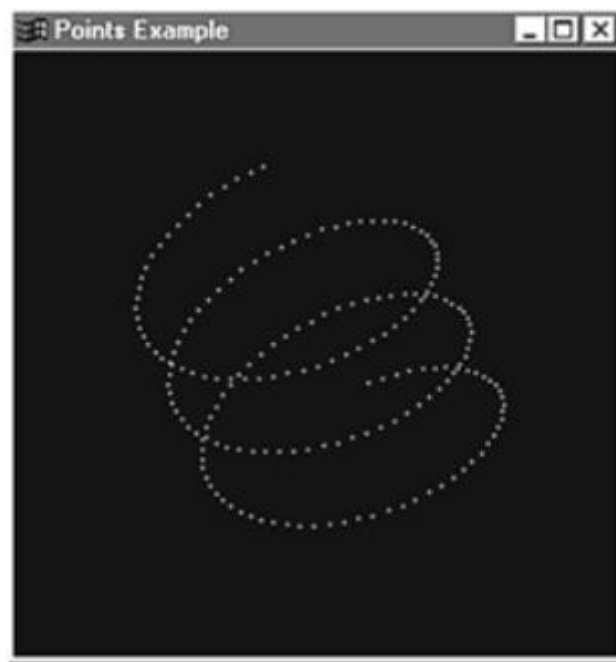
```
void TimerFunction(int value);
```

2.1.7. Cấu trúc lệnh chung để vẽ đối tượng

2.1.7.1. Vẽ điểm

Với việc vẽ điểm, bao gồm:

- Vẽ một điểm.
- Vẽ nhiều điểm.
- Vẽ các điểm tạo thành Spring-Shaped Path.

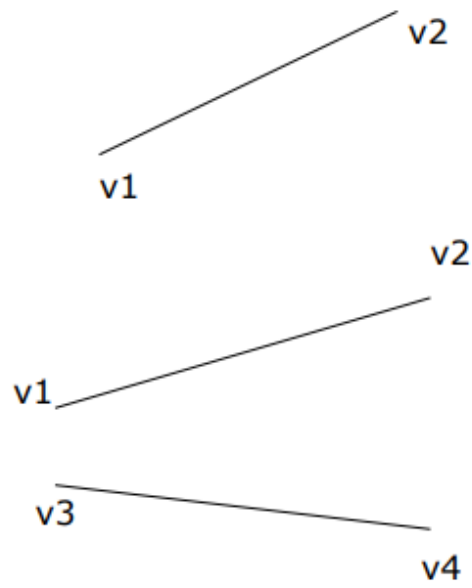


Hình 2.18. Kết quả của việc vẽ các điểm tạo thành Spring-Shaped Path [3]

2.1.7.2. Vẽ đường

Có các kiểu đường thường được sử dụng trong OpenGL, như:

- GL_LINES



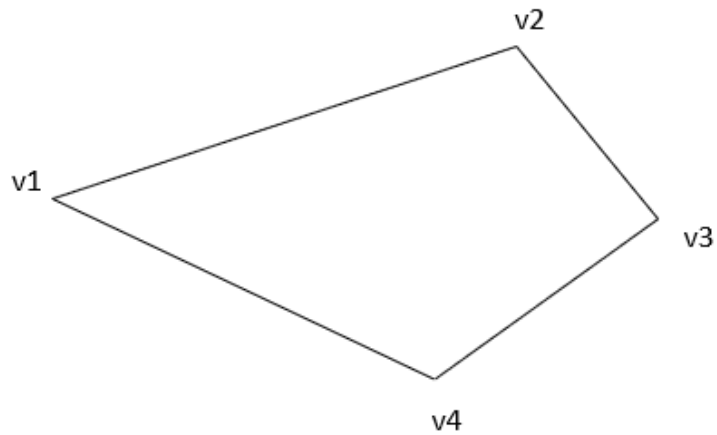
Hình 2.19 Ví dụ về vẽ các đường thẳng trong *GL_LINES* [3]

- *GL_LINE_STRIP*



Hình 2.20 Ví dụ về vẽ đường thẳng trong *GL_LINE_STRIP* [3]

- *GL_LINE_LOOP*

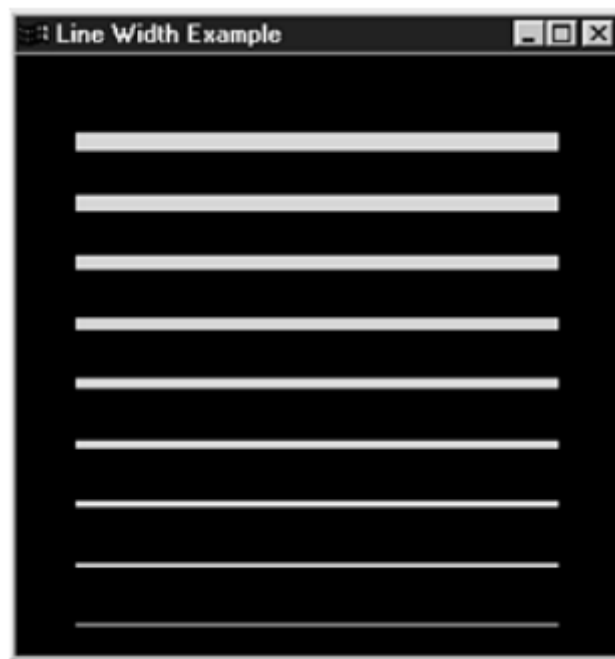


Hình 2.21 Ví dụ về vẽ đường thẳng trong GL_LINE_LOOP [3]

2.1.7.3. Các đặc tính của đường

Đối với các loại đường trong OpenGL, chúng thường có đặc tính như sau:

- Độ rộng của đường: mặc định thường là 1.0f.



Hình 2.22 Một số độ rộng của đường được vẽ trong OpenGL [3]

- Kiểu đường



Hình 2.23 Một số kiểu đường được vẽ trong OpenGL [3]

2.1.8. Màu sắc

Màu sắc được sử dụng trong OpenGL thuộc loại chuẩn màu RGBA.

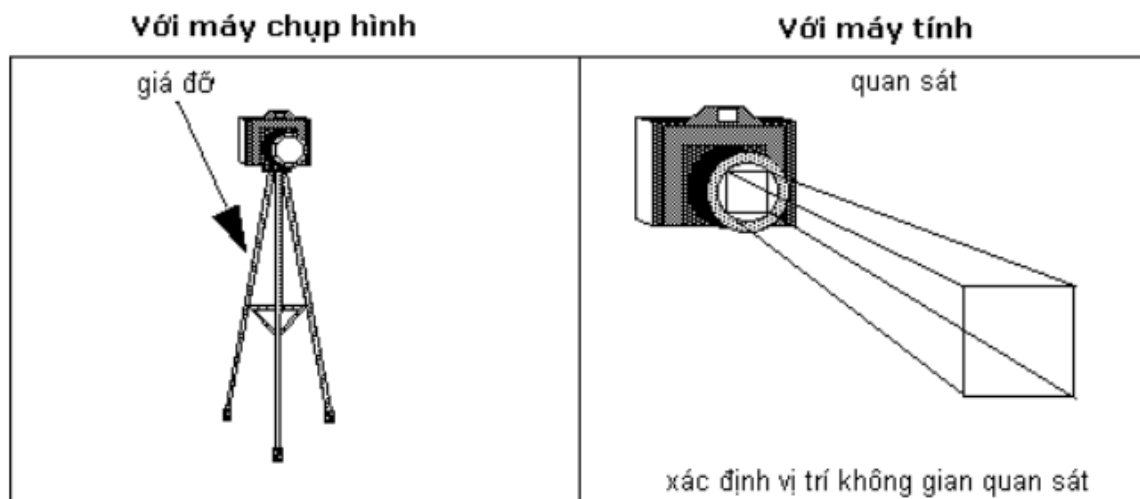
Code	Màu
<code>glColor3f(0.0, 0.0, 0.0)</code>	Black
<code>glColor3f(1.0, 0.0, 0.0)</code>	Red
<code>glColor3f(0.0, 1.0, 0.0)</code>	Green
<code>glColor3f(1.0, 1.0, 0.0)</code>	Yellow
<code>glColor3f(0.0, 0.0, 1.0)</code>	Blue
<code>glColor3f(1.0, 0.0, 1.0)</code>	Magenta
<code>glColor3f(0.0, 1.0, 1.0)</code>	Cyan
<code>glColor3f(1.0, 1.0, 1.0)</code>	White

Bảng 1 Bảng giá trị màu trong OpenGL

2.1.9. Các phép biến đổi tọa độ

2.1.9.1. Phép biến đổi quan sát

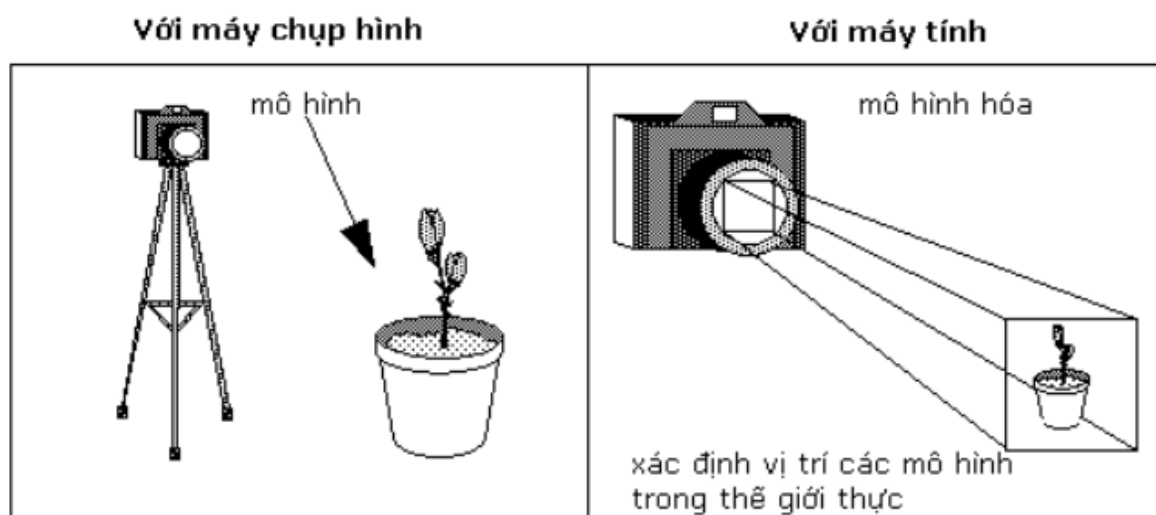
Phép biến đổi quan sát được sử dụng để xác định vị trí người quan sát hay camera.



Hình 2.24 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi quan sát

2.1.9.2. Phép biến đổi mô hình

Phép biến đổi mô hình được sử dụng để di chuyển các đối tượng xung quanh trong khung cảnh.



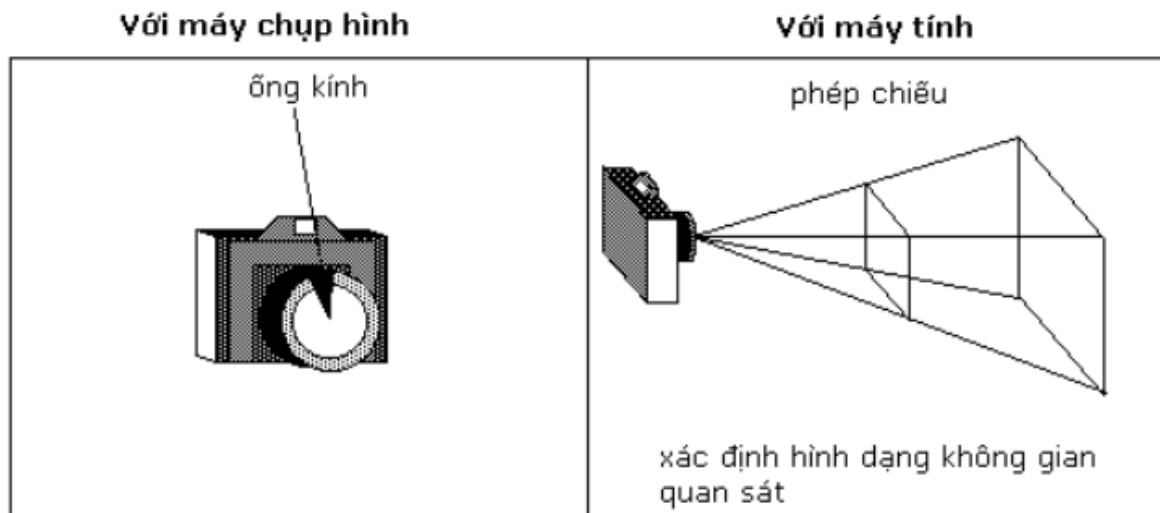
Hình 2.25 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi mô hình

2.1.9.3. Phép biến đổi quan sát – mô hình

Phép biến đổi quan sát – mô hình được sử dụng khi kết hợp giữa 2 phép biến đổi quan sát và biến đổi mô hình.

2.1.9.4. Phép chiếu

Phép chiếu được sử dụng để cắt và xác định kích thước của viewing volume.



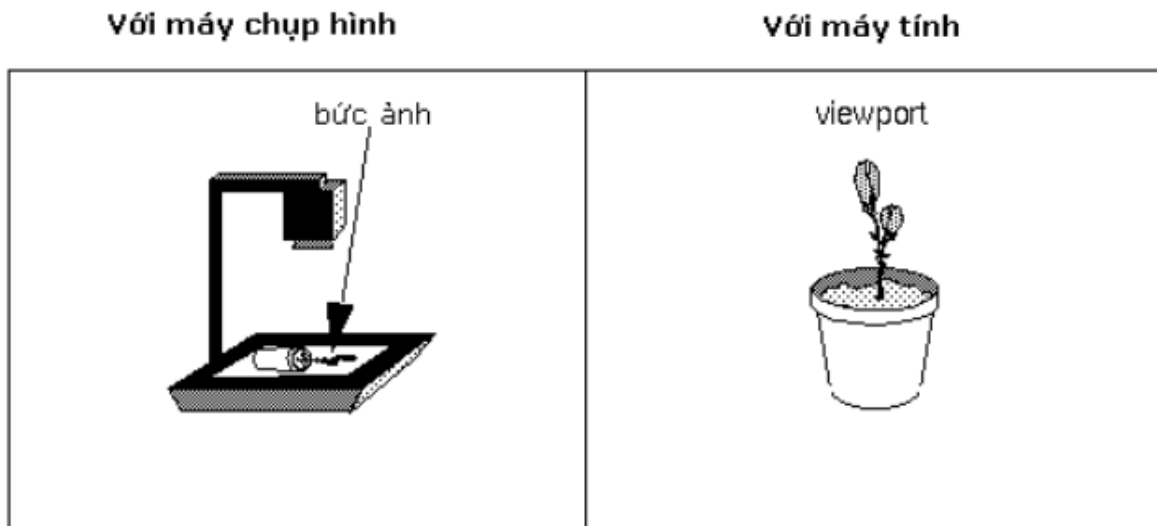
Hình 2.26 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép chiếu

Trong phép chiếu, có 2 loại:

- Phép chiếu vuông góc.
- Phép chiếu phối cảnh.

2.1.9.5. Phép biến đổi viewport

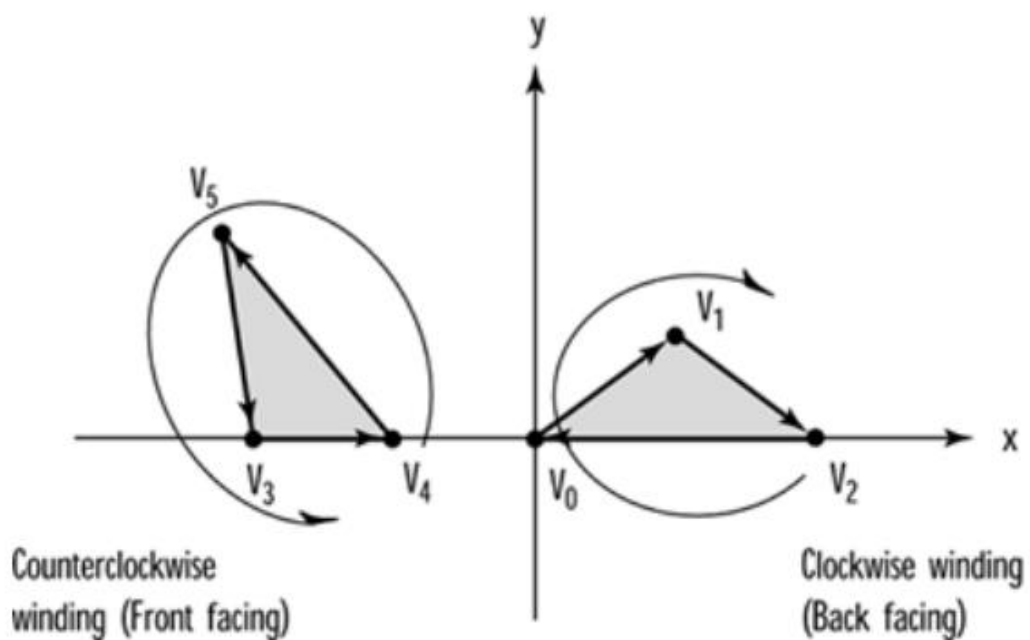
Phép biến đổi viewport được dùng để biến đổi kết quả cuối cùng lên window.



Hình 2.27 So sánh giữa chụp ảnh và vẽ trong không gian ba chiều với phép biến đổi viewport

2.1.10. Cách khử đối tượng khuất

2.1.10.1. Khử mặt sau



Hình 2.28 Ảnh minh họa về mặt trước và mặt sau của một đa giác [3]

Khử mặt sau có mục đích giúp cho việc hiển thị hình ảnh cũng như vật thể sẽ trở nên nhanh chóng hơn.

2.1.10.2. Một số phương pháp khử khuất

Một số phương pháp được dùng để khử khuất:

- Sử dụng phương pháp sắp xếp (sorting) và cố kết (coherence) để cải thiện sự thực hiện [3].
- Sắp xếp: so sánh độ sâu sau này, sắp xếp thứ tự các đường, mặt, và các đối tượng trong ảnh dựa vào khoảng cách từ chúng đến mặt phẳng quan sát [3].
- Phương pháp cố kết: tạo sự cân đối trong ảnh. Một đường quét riêng lẻ chứa các giá trị về độ sáng của các pixel, và các mẫu đường quét (scan-line patterns) thường thay đổi ít từ đường này đến đường kế tiếp [3].

Ngoài những phương pháp được kể trên, bên cạnh đó việc khử khuất còn có thể dựa trên phương trình mặt hoặc khử khuất với z-buffer

2.2. Phương hướng giải quyết

2.2.1. Ý tưởng được đưa ra

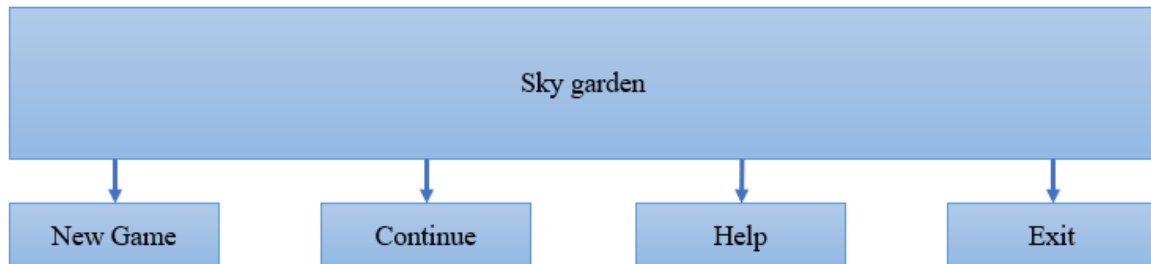
Trò chơi sẽ phát triển dựa theo game “Khu vườn trên mây”. Bao gồm những tính năng được đưa ra như sau:

- Newgame: Để giải quyết vấn đề chơi game mới
- Continue: Tiếp tục màn chơi vừa rồi
- Có danh sách nhiệm vụ và thực hiện theo danh sách.
- Tương tác được với game : đặt chậu, trồng hoa, tưới hoa, bắt sâu thu hoạch và buôn bán...

Để giải quyết các vấn đề trên, nhóm đã thực hiện những công việc như sau :

- Đối với mỗi hình ảnh được đưa vào, sẽ được đăng ký tọa độ cho từng đối tượng (background, button, chậu, hoa...)
- Giải quyết các vấn đề liên quan đến sử dụng chuột như :
 - o Chọn vị trí đặt chậu
 - o Chọn vị trí đặt hoa.
 - o Sử dụng chuột để thu hoạch.
 - o Sử dụng chuột để tưới nước.
 - o Sử dụng chuột để mua hàng trong Shop.
- Dùng class để quản lý các đối tượng như:

- Chậu cây (pot).
- Hoa (flower).
- Cửa hàng (store).
- Load ảnh (loadimage).
- Tạo hiệu ứng tưới cây (Rain_Cloud).



Hình 2.29 Các chức năng chính trong game

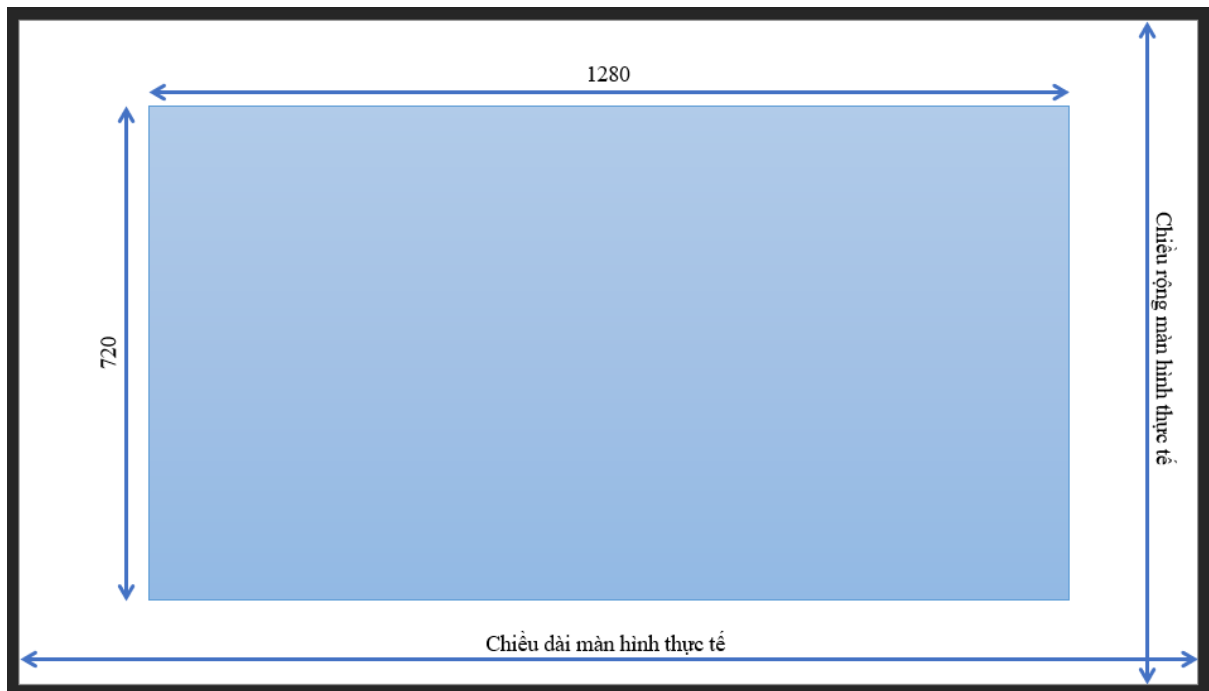
2.2.2. Giải quyết vấn đề và ý tưởng

2.2.2.1. Tạo màn hình game

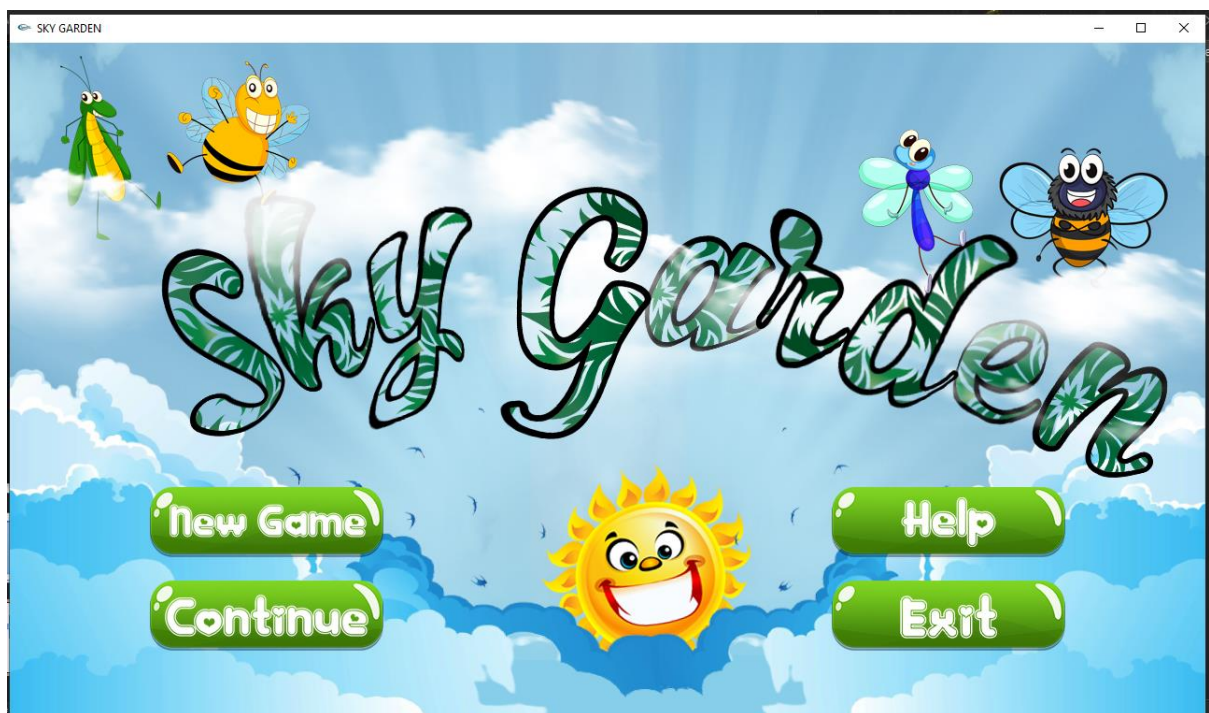
Game được phát triển có kích thước màn hình mặc định là: 1280×720. Khi bắt đầu thực thi chương trình, màn hình sẽ được hiển thị ở chính giữa màn hình (với các màn hình có kích thước khác nhau).

Ý tưởng được đặt ra là:

- Tạo 2 biến chứa chiều dài và chiều rộng.
- Lấy chiều dài và rộng của màn hình hiện tại lần lượt trừ cho 2 biến trên để lấy 2 tọa độ đầu tiên tải màn hình game.



Hình 2.30 Kích thước màn hình game so với thực tế

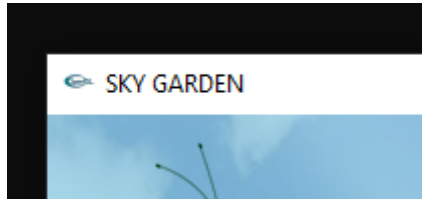


Hình 2.31 Màn hình chính trong game

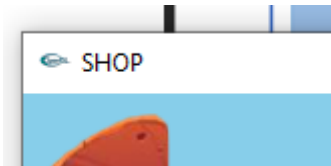
2.2.2.2. Đặt tên cho mỗi màn hình

Tại mỗi màn hình game khác nhau, thì màn hình game sẽ có tên khác nhau, chúng được tạo bởi câu lệnh:

```
glutCreateWindow("Tên cần đặt cho màn hình");
```

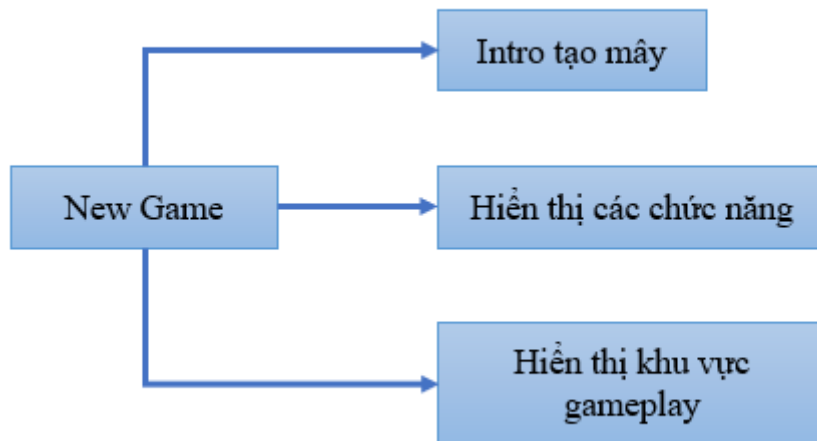


Hình 2.32 Ví dụ về hiển thị tên các màn hình trong game 1



Hình 2.33 Ví dụ về hiển thị tên các màn hình trong game 2

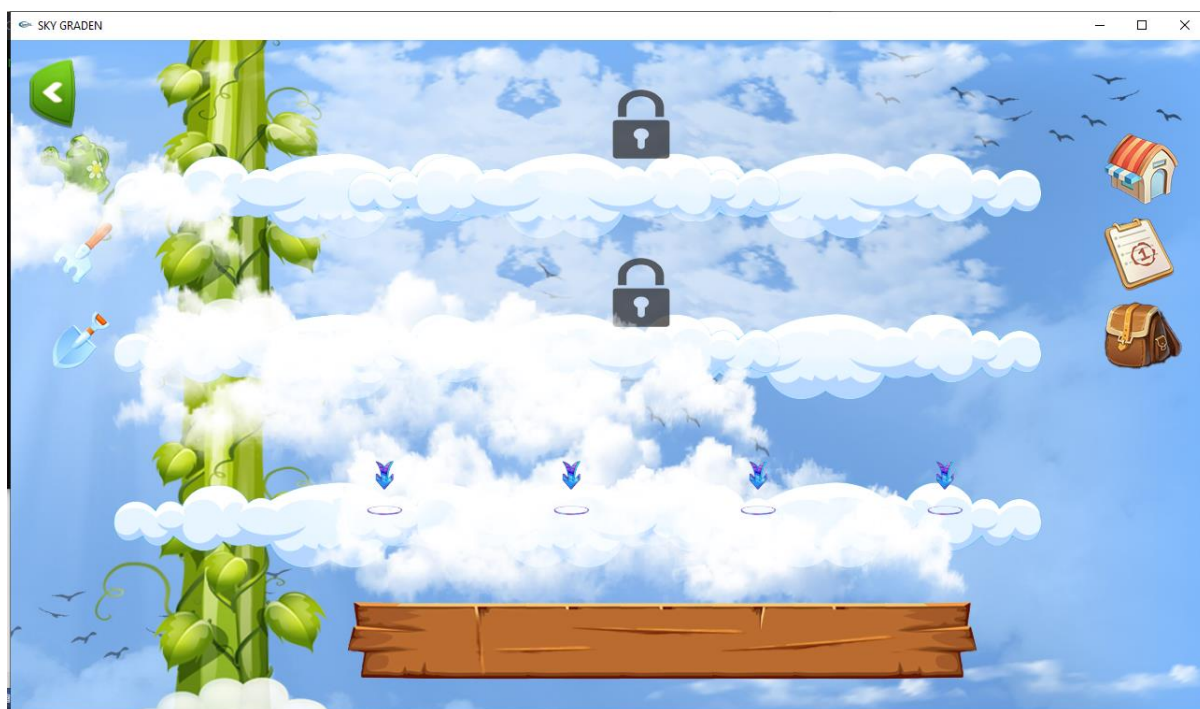
2.2.2.3. Tạo mới trò chơi (New Game)



Hình 2.34 Sơ đồ chức năng tại màn hình NewGame

Ý tưởng đặt ra:

- Click vào button NewGame tại màn hình hình chính sẽ chuyển sang màn hình gameplay.
- Sau đó hiển thị intro game (mây và mưa).
- Hiển thị các button chức năng ở 2 bên.
- Khu vực gameplay sẽ có các tầng mây để đặt chậu và trồng hoa.

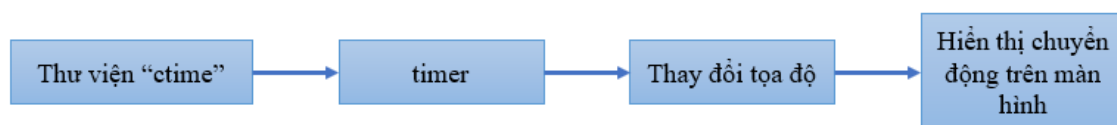


Hình 2.35 Màn hình chính của NewGame

Giải quyết vấn đề: Đầu tiên, tạo các class để quản lý các đối tượng cần thiết: pot.h, store.h, flower.h, Rain_Cloud.h, loadimage.h.

Tại các button bên phải, bao gồm: shop, mission, bag. Khi click chuột trái vào chúng sẽ chuyển sang các screen tương ứng.

Xử lý intro

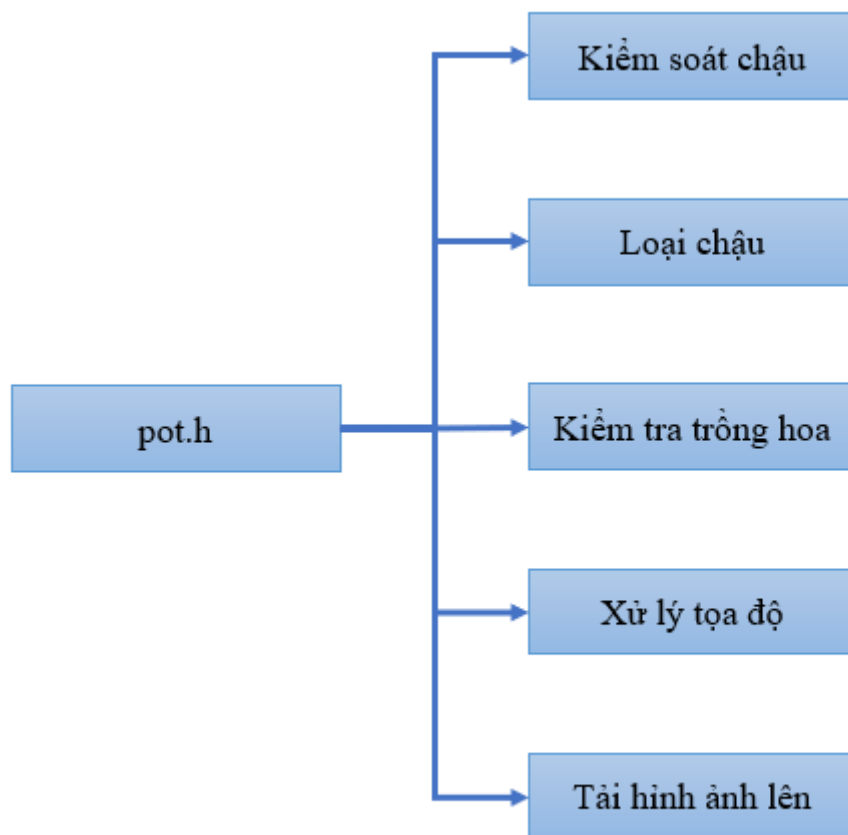


Hình 2.36 Xử lý intro

Sử dụng Rain_Cloud.h để quản lý việc khởi tạo tọa độ và load hình ảnh mây, mưa. Thông qua thư viện "ctime" tạo các timer để lặp lại chuyển động của mây và mưa bằng hình thức cộng dần tọa độ của hình ảnh sau mỗi lần timer thay đổi.

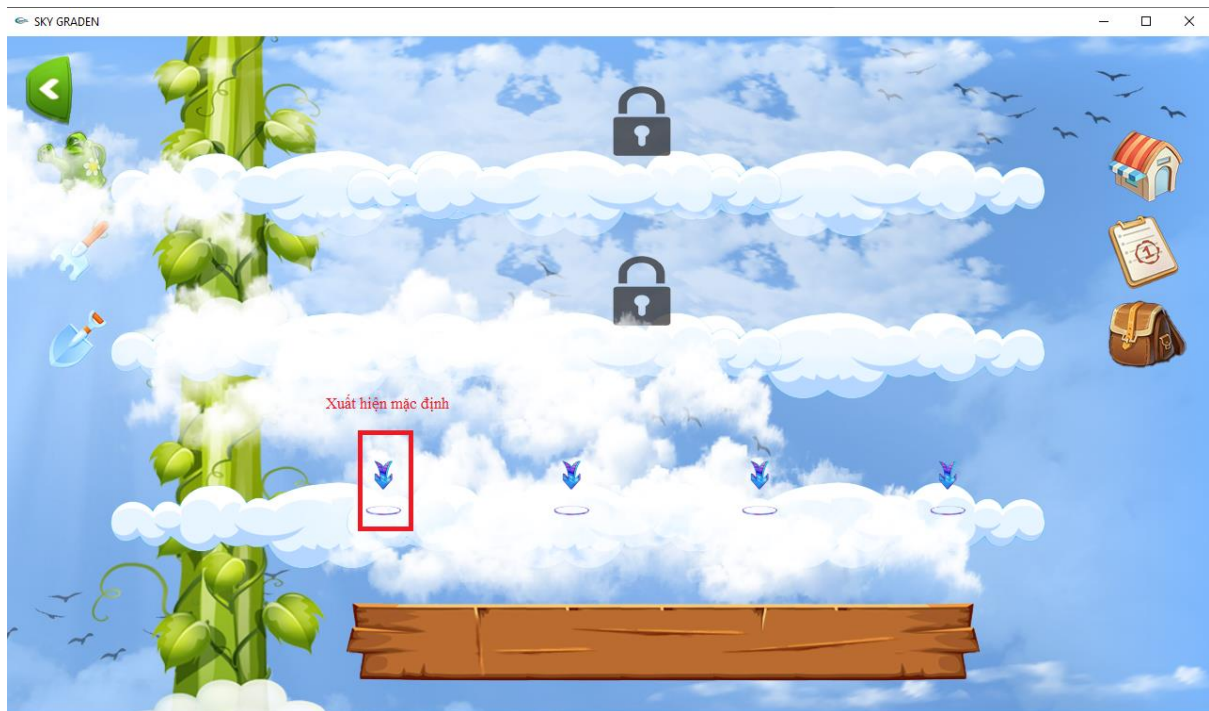
Xử lý đặt chậu:

Quản lý các đối tượng của chậu trong class “pot.h” như kiểm soát chọn chậu, loại chậu, kiểm tra trồng hoa, xử lý tọa độ và tải hình ảnh lên.



Hình 2.37 Quản lý các đối tượng trong “pot.h”

Mặc định mỗi tầng mây sẽ có được 4 vị trí đặt chậu. Bên cạnh đó, khi thực hiện sang màn NewGame thì sẽ mặc định xuất hiện vị trí của các mũi tên trên tầng mây, để người dùng có thể click vào chọn đặt chậu.



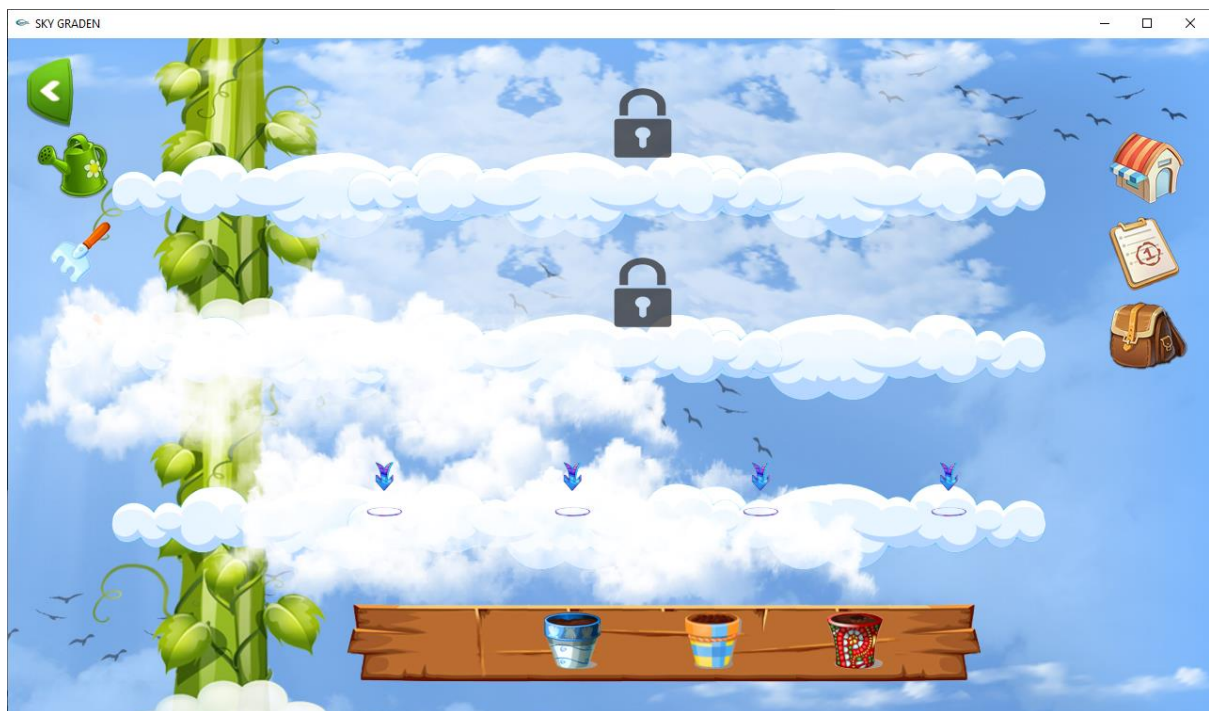
Hình 2.38 Vị trí xuất hiện của mũi tên, dùng để đặt chậu

Tiếp theo tạo mảng để kiểm soát số chậu được phép đặt trong game. Dùng hàm xử lý chuột và các đối tượng của chậu để xử lý quá trình đặt chậu thông qua hàm choosePots.



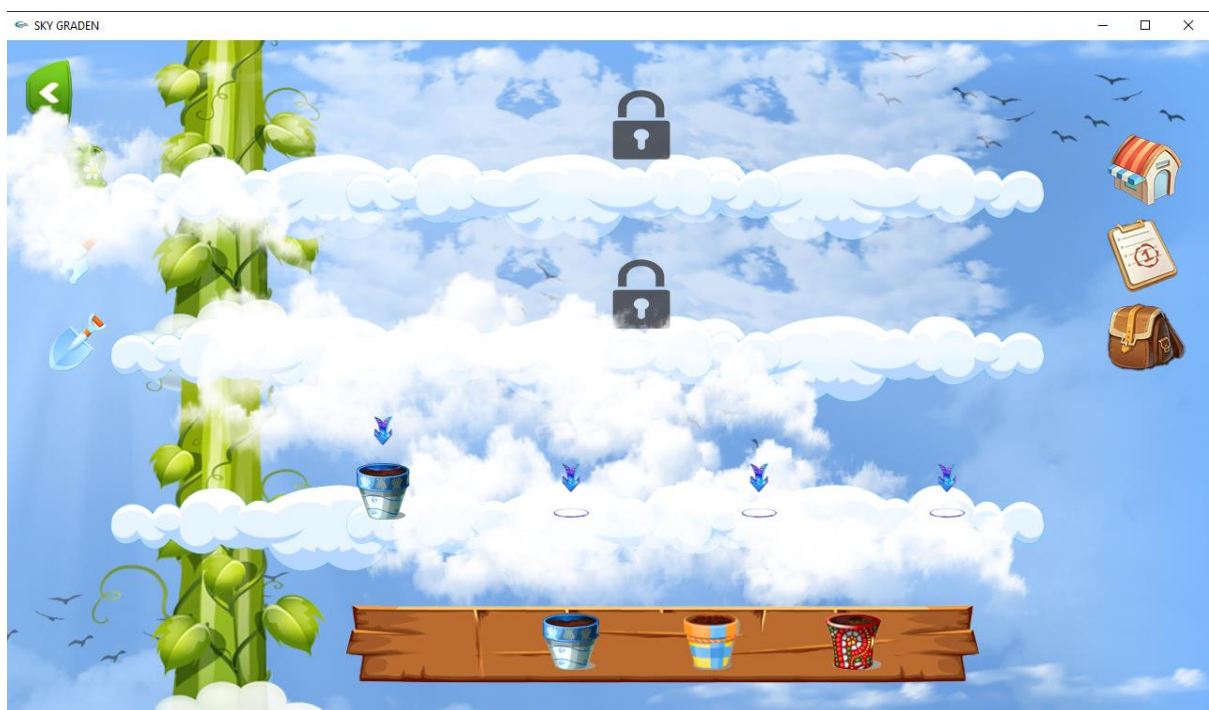
Hình 2.39 Quá trình đặt chậu

Sau khi click vào mũi tên bất kỳ trên tầng mây thứ nhất, sau đó trên thanh “container” sẽ show ra danh sách 3 loại chậu cho việc lựa chọn.

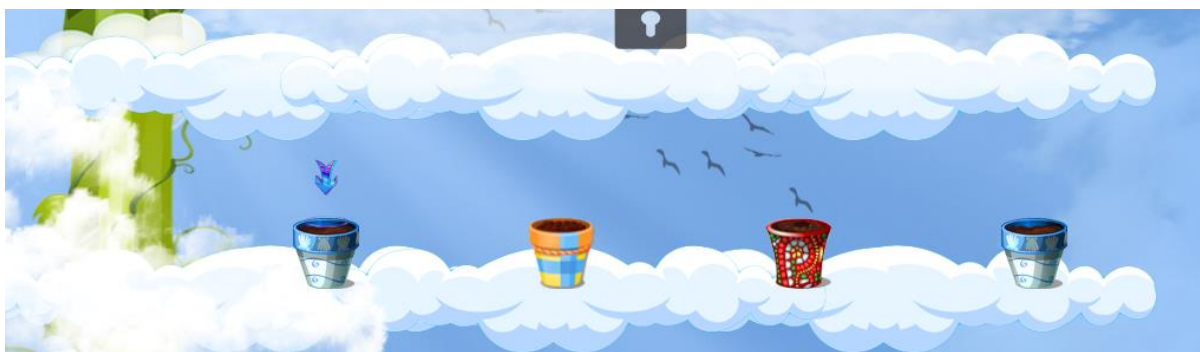


Hình 2.40 Sau khi click vào mũi tên lần 1

Sau khi người dùng chọn loại chậu, tại vị trí chọn lúc trước sẽ cho hiển thị hình ảnh.



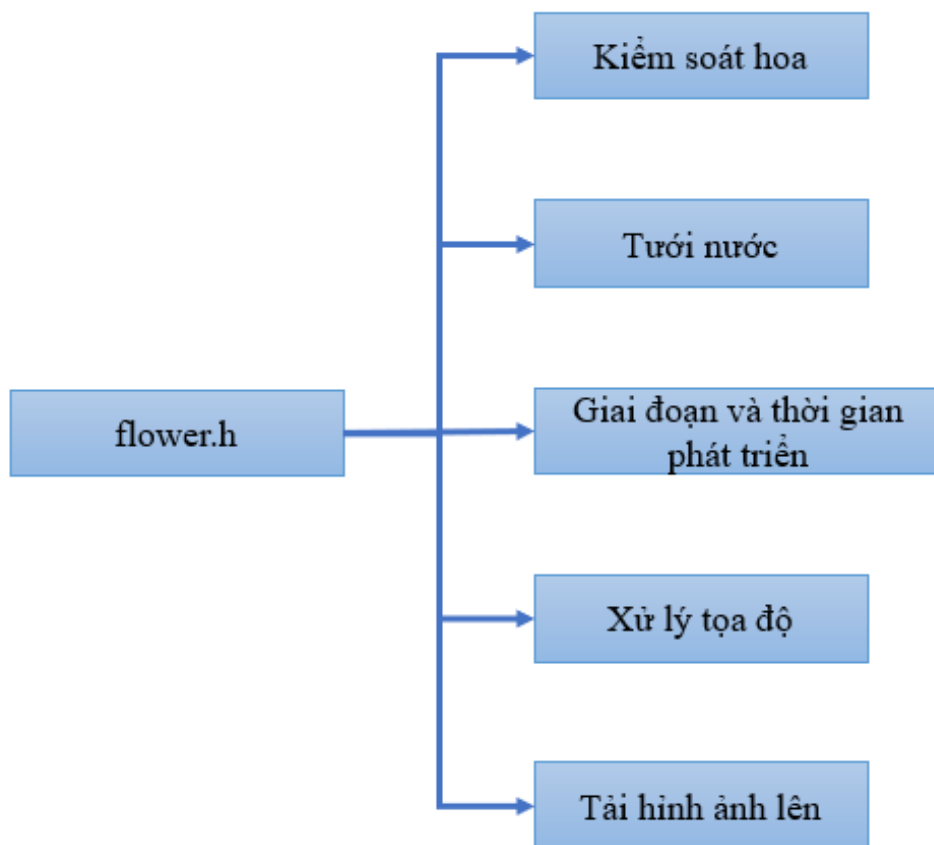
Hình 2.41 Chọn chậu và show chậu lên vị trí chọn chậu



Hình 2.42 Sau khi đặt đầy đủ các loại chậu

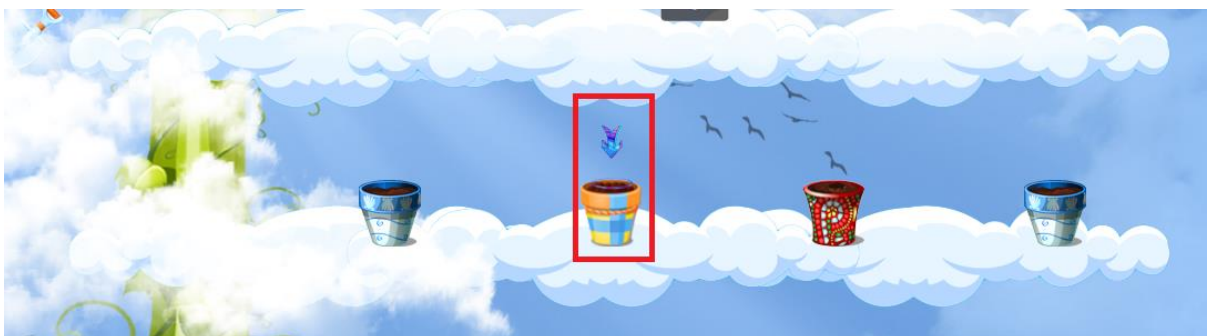
Xử lý đặt hoa:

Quản lý các đối tượng của hoa trong class “flower.h” như kiểm soát loại hoa, tưới nước, giai đoạn và thời gian phát triển, xử lý tọa độ và tải hình ảnh lên.



Hình 2.43 Quản lý các đối tượng trong “flower.h”

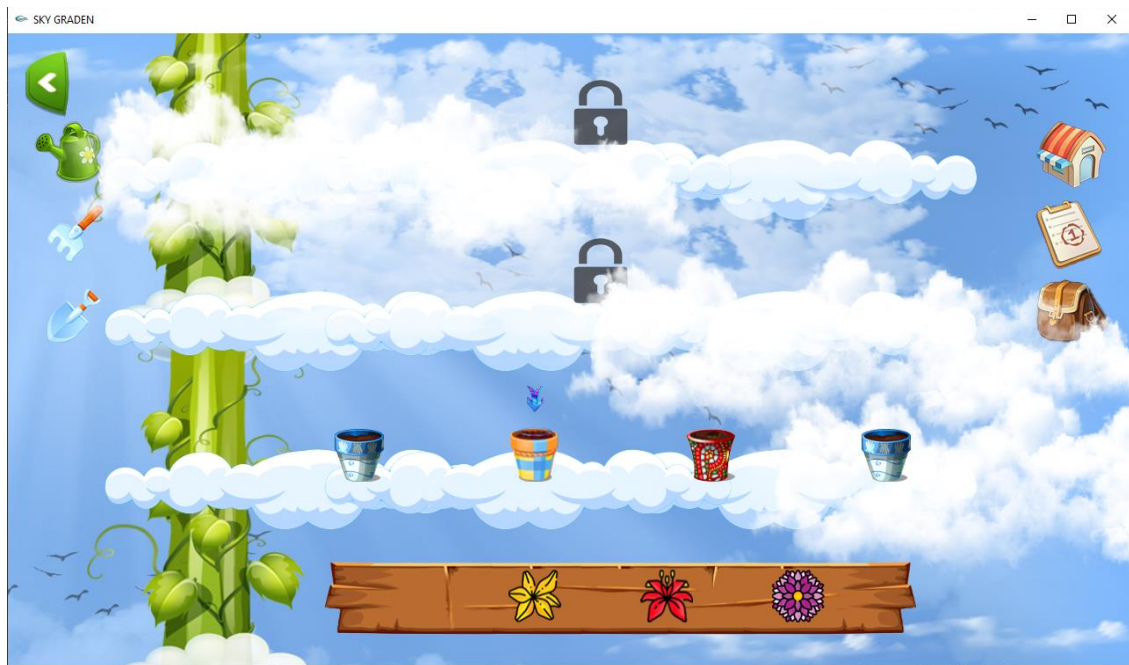
Sau khi đặt chậu thành công, tại vị trí đặt của chậu sẽ xuất hiện 1 mũi tên để nhắc nhở người chơi đừng quên trồng hoa.



Hình 2.44 Mũi tên xuất hiện để cho người dùng click vào chọn hạt giống



Hình 2.45 Quá trình chọn hoa



Hình 2.46 Chọn hoa và vị trí cần đặt hoa

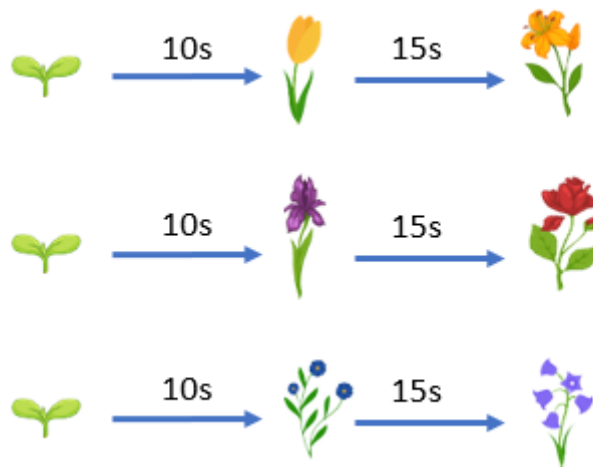


Hình 2.47 Sau khi chọn hạt giống

Sau khi chọn hạt giống muốn trồng, tại vị trí chậu sẽ hiển thị mầm của hạt giống, đối với các hạt giống khác nhau sẽ có từng giai đoạn phát triển khác nhau.



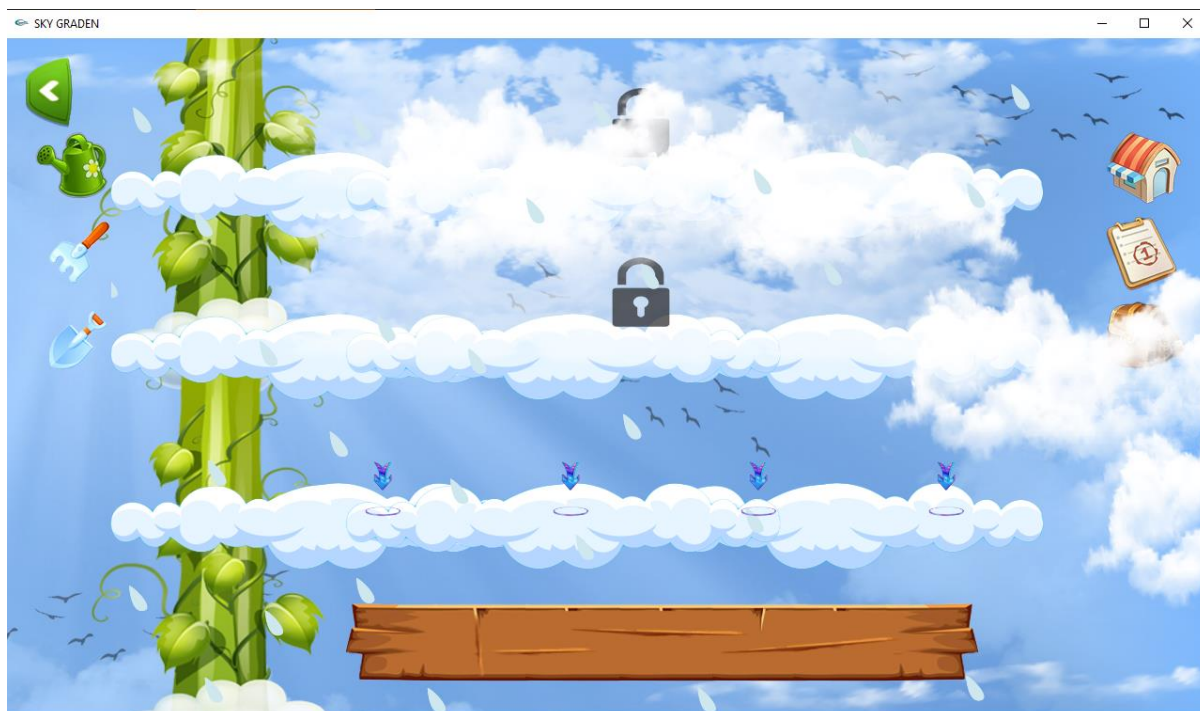
Hình 2.48 Thời gian phát triển của một loại hoa



Hình 2.49 Giai đoạn phát triển của 3 loại hoa

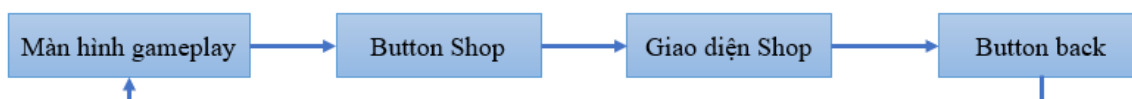
Xử lý tưới nước:

Về việc tưới nước, tại đây không sử dụng tưới nước cho từng cây mà thay vào đó người dùng sẽ được làm chủ thiên nhiên. Khi người dùng click vào button tưới nước, thì sẽ tạo 1 cơn mưa đi ngang qua và các cây sẽ được cung cấp nước. Đặc biệt, nếu các loại cây không được tưới nước vẫn sẽ phát triển bình thường. Việc xử lý cũng tương tự như việc tạo intro mây trôi. Nhưng thay vào đó khi click thì sẽ gọi mưa và biến tưới nước của cây sẽ được thay đổi.

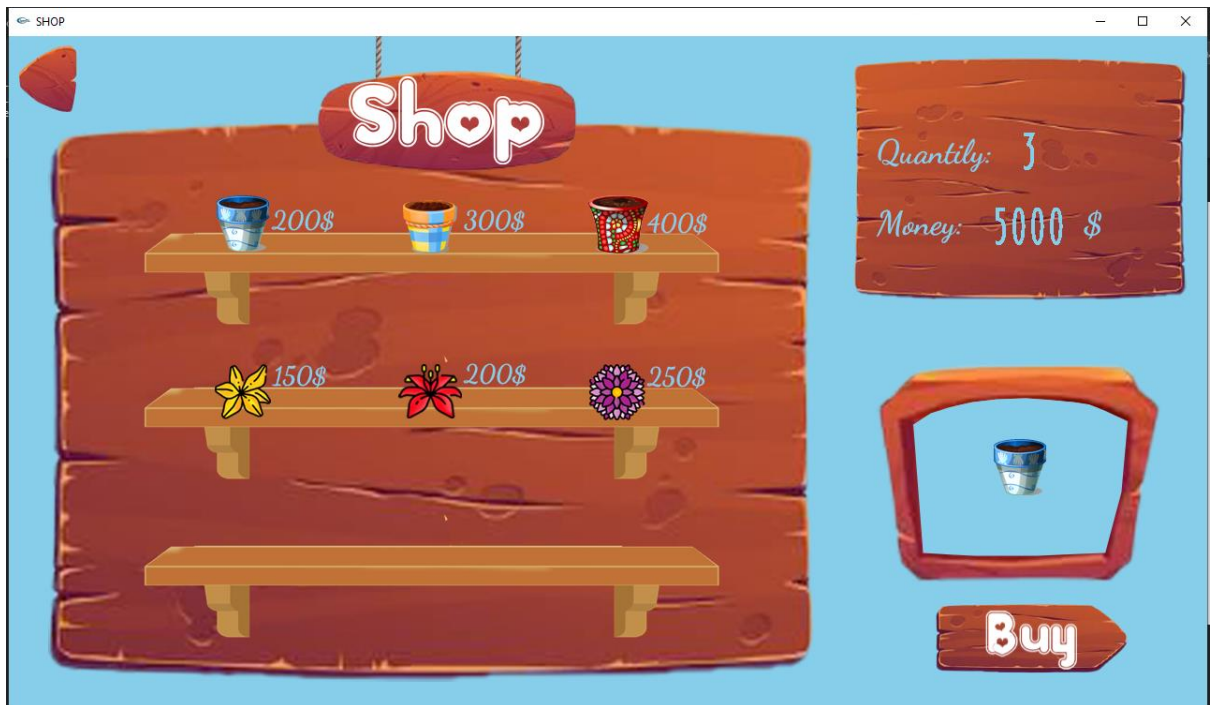


Hình 2.50 Giao diện tưới nước

Giao diện SHOP:



Hình 2.51 Quá trình di chuyển giữa màn hình Gameplay và Shop



Hình 2.52 Giao diện Shop

Tại đây sẽ hiển thị thông tin của các loại chậu, hoa. Người dùng có thể chọn bất kỳ vật phẩm nào muốn mua, sau đó click “Buy” để xác nhận mua. Giới hạn trong mỗi lần mua là 1 vật phẩm. Sau khi mua số lượng và giá tiền sẽ thay đổi.



Hình 2.53 Các sự kiện có trong SHOP

Mặc định đối với người đầu tiên chơi game, hệ thống sẽ cung cấp sẵn số lượng chậu, hoa và một số tiền để người dùng có thể trải nghiệm.



Hình 2.54 Số lượng mặc định về mỗi loại chậu, hoa

Với mỗi loại chậu và hoa, đều được cung cấp với số lượng là 3. Và số tiền được cung cấp là: 5000\$

Giao diện Mission:

Giao diện Mission là hộp thoại hiển thị các nhiệm vụ mà người dùng có thể xem và làm theo để có thể mở khóa những tính năng mới.

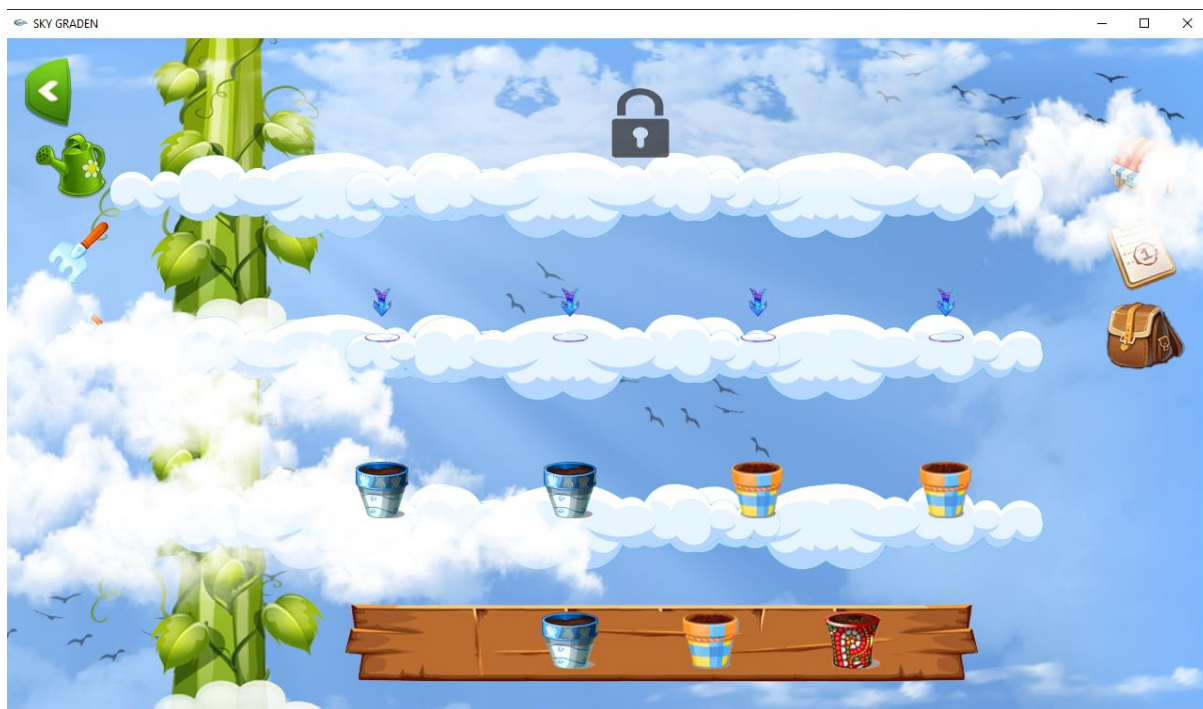


Hình 2.55 Giao diện Mission



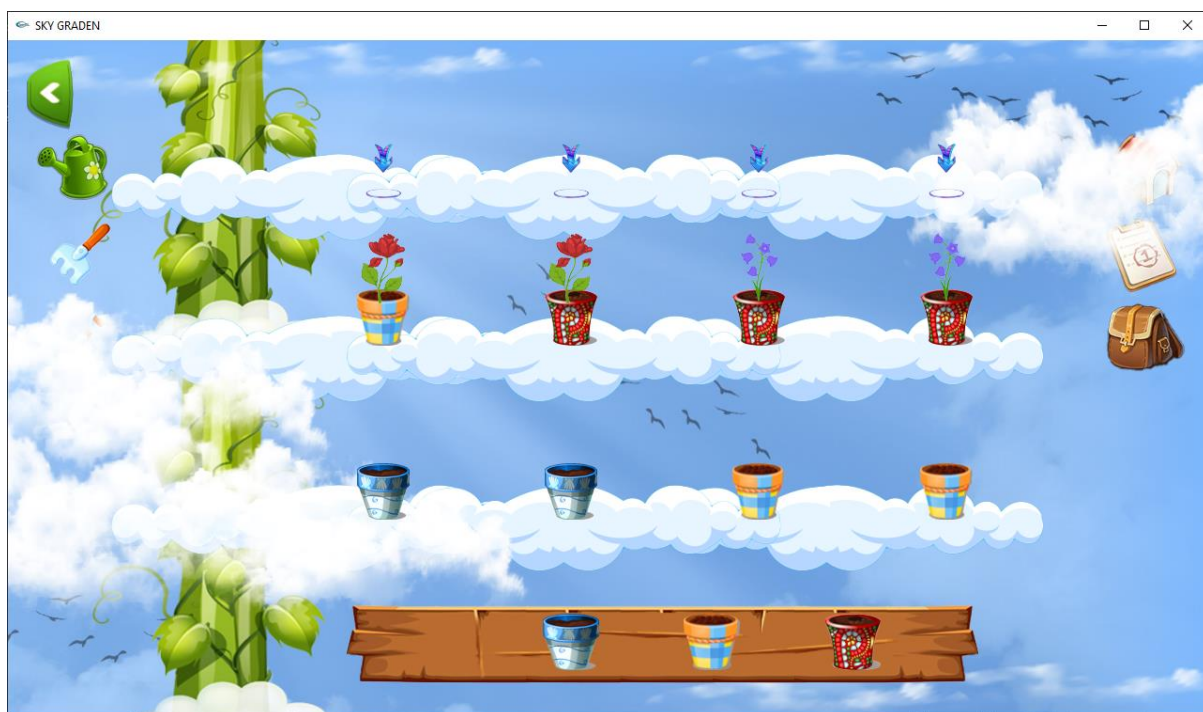
Hình 2.56 Quá trình di chuyển giữa màn hình Gameplay và Shop

Để mở tầng mây thứ 2: Người chơi cần phải thực hiện tưới cây đủ 3 lần.



Hình 2.57 Giao diện khi mở tầng thứ 2

Để mở tầng mây thứ 3: Người chơi cần phải thực hiện hoàn thành nhiệm vụ đầu tiên, sau đó là thực hiện thu hoạch đủ 4 lần.



Hình 2.58 Giao diện khi mở tầng thứ 3

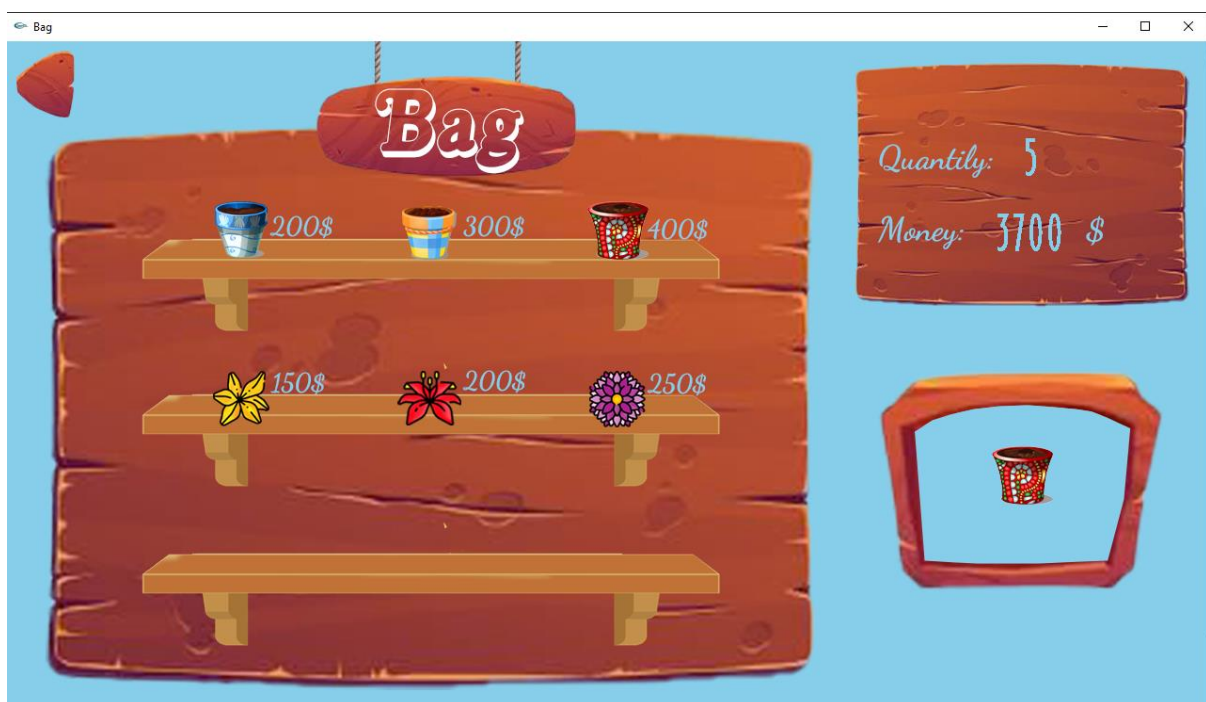
Tại đây, người dùng nếu thực hiện vừa đủ các nhiệm vụ trên tại duy nhất tầng 1 thì sẽ mở khóa một lượt 2 tầng còn lại.

Giao diện BAG:

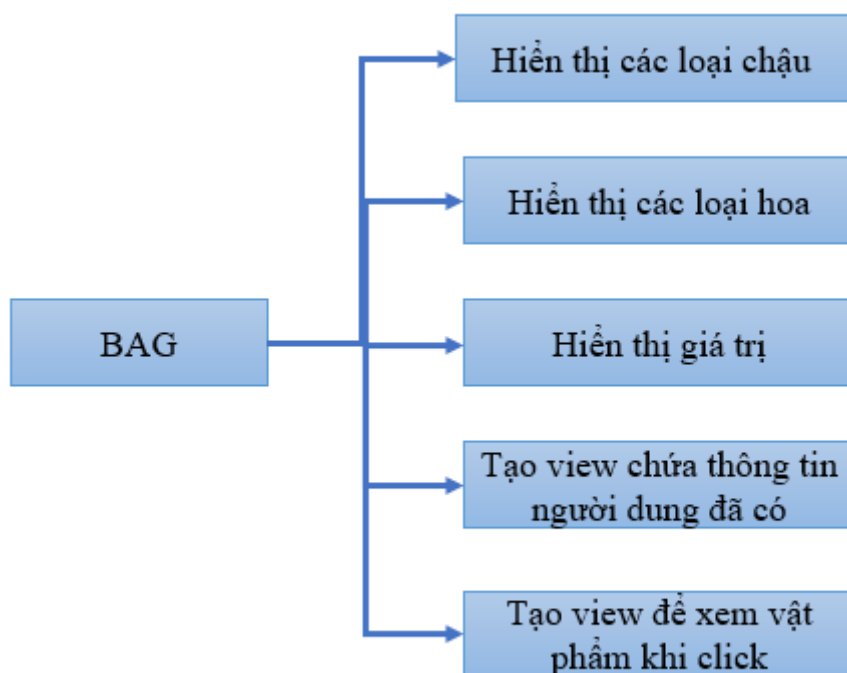


Hình 2.59 Quá trình di chuyển giữa màn hình Gameplay và BAG

Giao diện BAG tương tự như giao diện của SHOP, nhưng thay vào đó, tại đây chỉ có thể xem thông tin chứ không thực hiện mua như của SHOP



Hình 2.60 Giao diện BAG



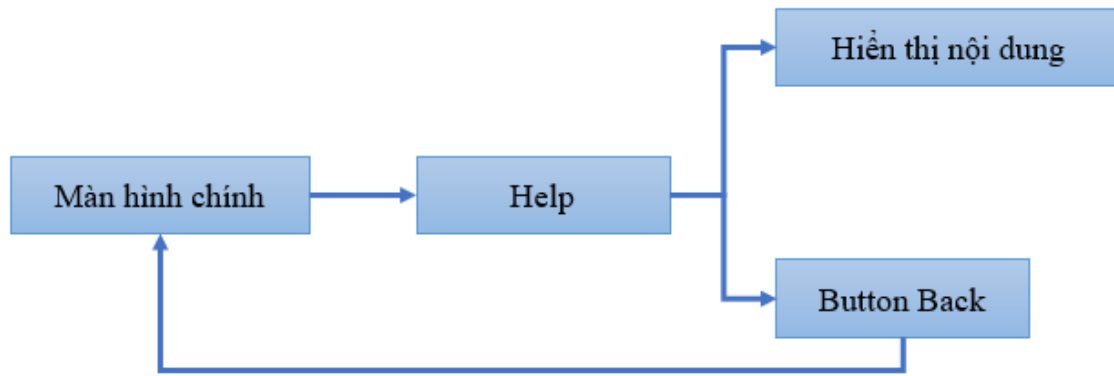
Hình 2.61 Các sự kiện trong BAG

2.2.2.4. Tiếp tục trò chơi (Continue)

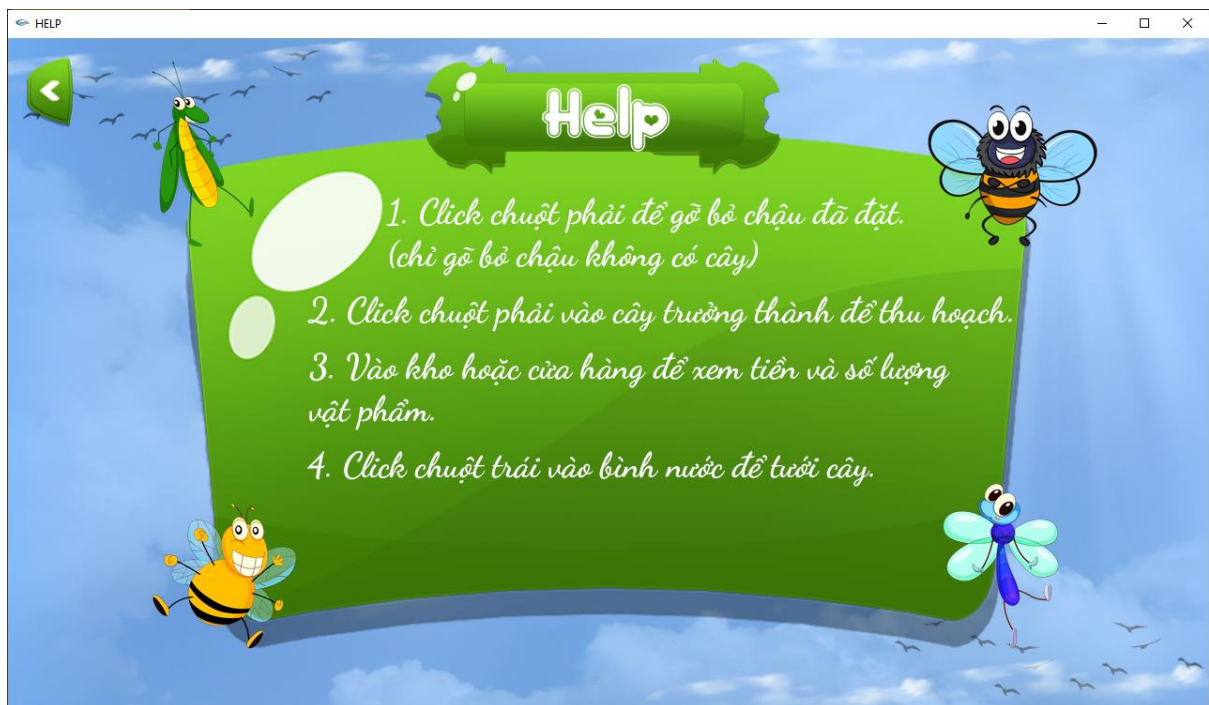
Xử lý việc bắt click chuột trên button Continue. Sau đó tải lại trò chơi và giữ các thông số cũ của người chơi.

2.2.2.5. Màn hình giúp đỡ (Help)

Hiển thị một màn hình với các thông tin giúp người chơi hiểu rõ hơn về trò chơi. Ở phần này sẽ sử dụng hình ảnh có sẵn văn bản được nhóm vẽ lên.



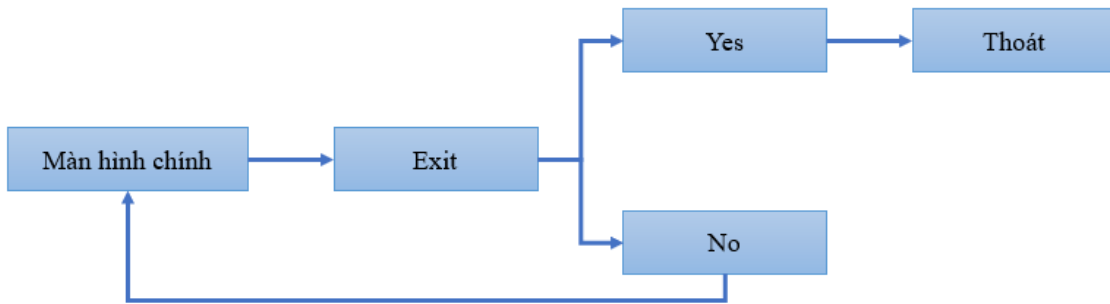
Hình 2.62 Quá trình xem giúp đỡ



Hình 2.63 Màn hình Help

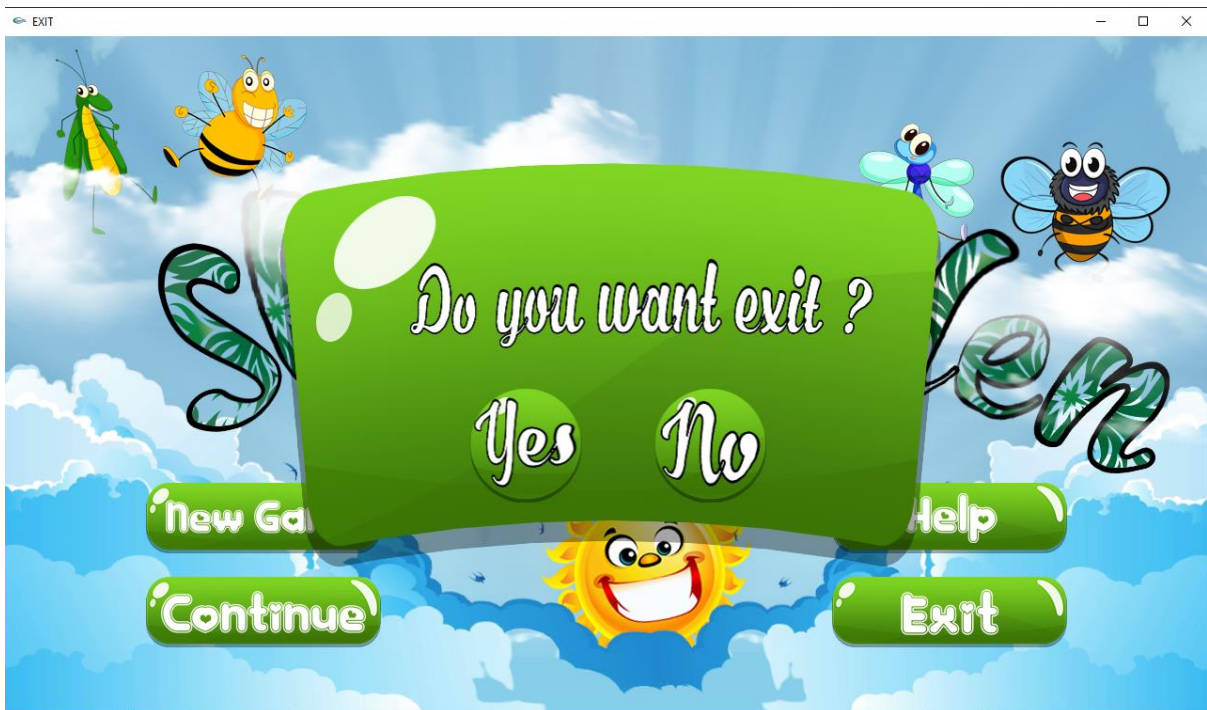
2.2.2.6. Thoát trò chơi (Exit)

Hiển thị ra màn hình với 2 lựa chọn Yes, No để người chơi lựa chọn việc rời khỏi trò chơi. Ở chức năng này chủ yếu xử lý việc bắt tọa độ click chuột để gọi các hàm tương ứng.



Hình 2.64 Quá trình thoát trò chơi

- Nếu chọn Yes, thực hiện lệnh exit và rời khỏi màn hình game.
- Nếu chọn No, thì quay lại màn hình chính của game.



Hình 2.65 Màn hình exit

Tại màn hình này, cần giải quyết lại vấn đề cho load lại các hình ảnh từ màn hình display, nếu không thì chỉ xuất hiện duy nhất mỗi vùng chọn yes/no.

2.3. Kết quả thực hiện demo

Link github đối với đề tài lần này:

<https://github.com/PhamChiNang151020/Sky-Garden.git>

Bảng phân công công việc:

STT	Họ và tên	Công việc
1	Phạm Chí Năng	Thiết kế hình ảnh (photoshop) Viết báo cáo (Word) Hỗ trợ làm Powerpoint với Tùng. Code giao diện bên ngoài đối với game.
2	Bùi Phương Tùng	Hỗ trợ Năng viết báo cáo (Word) Làm Powerpoint với Năng. Code chính.
3	Nguyễn Nguyệt Phương Bảo	Code chính các chức năng trong game (Đặt chậu, trồng hoa,...) Hỗ trợ viết Word và Power Point.
4	Trương Thiên Phong	Hỗ trợ mọi người code và fix bug

Bảng 2 Bảng phân công công việc

STT	Nhiệm vụ	Kết quả	Ghi chú
1	Sử dụng hình ảnh tự vẽ/ thiết kế	Pass	Sử dụng PS, AI
2	Giao diện NewGame	Pass	Chỉ được tạo lần đầu tiên khi chạy chương trình
3	Giao diện Continue	Pass	Chỉ sử dụng được khi đã NewGame
4	Giao diện Help	Pass	
5	Giao diện Exit	Pass	Yes để thoát, No để quay lại
6	Chức năng đặt chậu	Pass	
7	Chức năng trồng hoa	Pass	
8	Chức năng tưới cây	Pass	Tưới cho toàn bộ số lượng cây (mưa)
9	Chức năng bắt sâu	Fail	Chưa thực hiện
10	Giao diện Shop	Pass	Chứa thông tin về số lượng và giá trị

11	Chức năng Mua hàng từ shop	Pass	
12	Giao diện Mission	Pass	Hiển thị nhiệm vụ đặt ra
13	Chức năng trong Mission	Pass	Hoàn thành nhiệm vụ để mở tầng mây mới
14	Giao diện Bag (Túi)	Pass	Tương tự với Shop nhưng không thực hiện mua

Bảng 3 Bảng liệt kê chức năng và giao diện

CHƯƠNG 3. KẾT LUẬN

3.1. Kết quả đạt được

- Phát triển được một game từ nền tảng C++ thông qua sự giúp đỡ của của thư viện “OpenGL”.
- Hiểu được cách sử dụng hình ảnh tự thiết kế vào trong game.
- Giúp tăng khả năng sáng tạo cho đề tài.
- Nắm vững các kiến thức về xử lý đồ họa ba chiều.
- Hiểu rõ vai trò và cấu trúc của OpenGL.
- Hiểu và áp dụng được giao diện lập trình đồ họa OpenGL.

3.2. Ưu điểm

- Giao diện khá đẹp, bắt mắt, tạo hứng thú cho người chơi.
- Cho phép người chơi tương tác với một số chức năng tương tự so với game “Khu vườn trên mây”.
- Game dễ chơi, dễ hiểu.
- Giao diện Gameplay có mây trôi, tạo cảm giác cho người chơi như đang trồng cây trên mây.
- Thực hiện được việc mua hàng trong Shop.
- Thay vì sau khi thu hoạch thì phải bán lại. Nhưng trong game, sau khi thu hoạch thì sẽ được số lượng tiền tương ứng ngay.

3.3. Nhược điểm

- Chưa hiển thị được 3D
- Time trồng cây và phát triển từng loại cây chưa được phân biệt.
- Chưa phát triển được việc trang trí.

3.4. Hướng phát triển trong tương lai

- Hiện thị được 3D đối với chậu và hoa.
- Giải quyết vấn đề time trồng cây và tiền thưởng của mỗi loại hoa.
- Tạo thêm các thông báo mới khi mở tầng mây mới cũng như khi hoa đến giai đoạn thu hoạch.
- Trang trí cho gameplay tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] T. N. T. Anh, "Bài giảng lập trình đồ hoạ máy tính với OpenGL," [Online]. Available: <https://timoday.edu.vn/bai-giang-lap-trinh-do-hoa-may-tinh-voi-opengl/>. [Accessed 02 2022].
- [2] P. C. Năng, "Báo cáo BTL DUCK SHOOTING," 2020.
- [3] T. V. T. H. Tuyết, LẬP TRÌNH ĐỒ HOẠ - Tổng quan về đồ họa ba chiều.