**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Nguyễn Huy Tú**

**Nguyễn Duy Vũ**

**18120254**

**18120264**

**ĐỒ ÁN GIỮA KỲ**

**TRÒ CHƠI BÓNG BÀN**

**Môn học: Thực hành Phương pháp Lập trình Hướng Đối Tượng**

Thành phố Hồ Chí Minh - 2019

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Nguyễn Huy Tú**

**Nguyễn Duy Vũ**

**18120254**

**18120264**

**ĐỒ ÁN GIỮA KỲ**

**| Đề tài |**

**TRÒ CHƠI BÓNG BÀN**

**| Giáo viên hướng dẫn |**

**ThS. Nguyễn Thành An**

**Môn học: Thực hành Phương pháp Lập trình Hướng Đối Tượng**

Thành phố Hồ Chí Minh – 2019

|  |
| --- |
| **LỜI CẢM ƠN** |

Qua thời gian học tập và rèn luyện tại trường đại học Khoa Học Tự Nhiên, Thành phố Hồ Chí Minh, được sự chỉ bảo và giảng dạy nhiệt tình của quý thầy cô khoa Công nghệ Thông tin, vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình nghiên cứu bài báo cáo mà còn là hành trang để nhóm em bước vào đời một cách vững chắc và tự tin.

Với lòng biết ơn sâu sắc nhất, nhóm xin được gửi đến quý thầy cô ở khoa Công nghệ Thông tin trường đại học Khoa Học Tự Nhiên, đặc biệt là ThS. Nguyễn Thành An, người đã tận tình hướng dẫn, giúp đỡ nhóm hoàn thành chuyên đề báo cáo này, lời cảm ơn chân thành nhất. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì bài báo cáo này khó lòng mà hoàn thiện được.

Nhóm cũng muốn gửi lời cảm ơn đến gia đình, bạn bè, và người thân đã luôn ủng hộ và tạo điều kiện tốt nhất để nhóm có thể tập trung nghiên cứu đề tài này.

Trong quá trình thực tập cũng như làm báo cáo, do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên nhóm khó tránh khỏi sai sót. Nhóm rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn để nhóm học hỏi được nhiều kinh nghiệm và hoàn thành tốt hơn trong bài báo cáo sau này.

Trân trọng.

TP.HCM, ngày 01 tháng 11 năm 2019

**Sinh viên thực hiện**

|  |
| --- |
| **MỤC LỤC** |

[**LỜI CẢM ƠN** 3](#_Toc25674439)

[**MỤC LỤC** 4](#_Toc25674440)

[**BÁO CÁO ĐỒ ÁN** 5](#_Toc25674441)

[**Thông tin nhóm** 5](#_Toc25674442)

[**Thông tin đồ án** 5](#_Toc25674443)

[**Sản phẩm** 5](#_Toc25674444)

[**PHÂN CÔNG THÀNH VIÊN** 6](#_Toc25674445)

[**TIẾN TRÌNH ĐỒ ÁN** 7](#_Toc25674446)

[**PHƯƠNG THỨC & THUỘC TÍNH** 8](#_Toc25674447)

[**MÃ NGUỒN** 11](#_Toc25674448)

[**SƠ ĐỒ LỚP** 21](#_Toc25674449)

[**LƯU ĐỒ THUẬT TOÁN** 22](#_Toc25674450)

[**HƯỚNG DẪN SỬ DỤNG** 23](#_Toc25674451)

[**TÀI LIỆU THAM KHẢO** 26](#_Toc25674452)

# **BÁO CÁO ĐỒ ÁN**

### **Thông tin nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Email | Vai trò |
| 18120264 | Nguyễn Duy Vũ | Vu38988@gmail.com | Developer |
| 18120254 | Nguyễn Huy Tú | 18120254@student.hcmus.edu.vn | Developer |

### **Thông tin đồ án**

|  |  |
| --- | --- |
| Tên quy trình | Tên |
| IDE | Visual Studio 2017 |
| Báo cáo | Microsoft Word |
| Giao diện | Console Prompt |
| Tên | Trò chơi bóng bàn |
| Product Owner | Thầy Nguyễn Thành An |

### **Sản phẩm**

Đồ án này viết một chương trình trò chơi bóng bàn (ping pong) trên nền console với những chức năng được yêu cầu bởi thầy Nguyễn Thành An.

# **PHÂN CÔNG THÀNH VIÊN**

Sinh viên thực hiện đồ án:

* Nguyễn Huy Tú – 18120254
* Nguyễn Duy Vũ -18120264

Phân công:

|  |  |
| --- | --- |
| **Tên chức năng** | **Thành viên thực hiện** |
| Thiết kế class diagram | Nguyễn Duy Vũ |
| Điều khiển thanh điều khiển | Nguyễn Duy Vũ |
| Xử lý hướng đi của bóng | Nguyễn Duy Vũ |
| Nhận phím sự kiện từ bàn phím | Nguyễn Duy Vũ |
| Hiển thị menu, giao diện | Nguyễn Huy Tú |
| Hiển thị di chuyển của bóng và thanh | Nguyễn Huy Tú |
| Tính điểm người chơi | Nguyễn Huy Tú |
| Xử lý va chạm, thắng thua | Nguyễn Huy Tú |

# **TIẾN TRÌNH ĐỒ ÁN**

Những câu đã làm được:

* Hiện màn hình chơi.
* Điều khiển thanh trượt qua lại để hứng bóng.
* Xử lý tình huống khi bóng va chạm vào biên sân hay thanh trượt của người chơi.
* Xử lý thắng thua và kết thúc game.
* Thiết kế giao diện với màu sắc đẹp.

Những câu chưa làm được:

* Tăng tốc độ khi va chạm vào thanh trượt người chơi.
* Cho thanh trượt chạy tự động dể hứng quả bóng.

# **PHƯƠNG THỨC & THUỘC TÍNH**

1. **Lớp cGameManger:**

*Các thuộc tính:*

int width, height;//Chiều rộng, Chiều cao của game

char up1, down1, up2, down2;//Các nút điều khiển

bool quit;//Điều kiện để thoát game

int score1, score2; // Điểm số

cBall \*ball; // Bóng

cPaddle \*player1; // Thanh điều khiển

cPaddle \*player2;

Lớp điều khiển Game, lớp này sẽ bao gồm lớp **cBall** và lớp **cPaddle**. width và height là 2 thuộc tính thể hiện chiều dài và chiều rộng của khung chơi. Bóng sẽ phản xạ hoặc kết thúc game khi va đập khung chơi. up1, down1, up2, down2 là nút điều khiển thanh lên, xuống của người chơi 1 và 2 (mặc định người chơi 1 là W, S, người chơi 2 là phím mũi tên lên, xuống). quit là biến kiểm tra xem người chơi có muốn thoát không. \*ball, \*player1, \*player2 lần lượt là bóng, thanh điều khiển của người chơi 1, thanh điều khiển của người chơi 2 mà game sẽ quản lý.

*Các phương thức:*

cGameManger(int, int); // Hàm tạo

~cGameManger(); // Hàm hủy

void ScoreUp(cPaddle\*); // Cộng điểm

void DrawBorder(); // Vẽ khung xung quanh

void Draw(); // Vẽ bóng và thanh điều khiển

void Input(); // Nhận sự kiện từ bàn phím

void Logic(); // Xử lý va chạm

void Run(); // Chạy game

void CleanPaddle(); // Xóa thanh điều khiển sau khi kết thúc game

Phương thức khởi tạo cGameManger(int, int) với 2 tham số int truyền vào là chiều dài và chiều rộng của khung chơi.

Phương thức void ScoreUp(cPaddle\*) tăng score của người chơi 1 hoặc 2 (tùy vào con trỏ được truyền vào).

Phương thức void DrawBorder() vẽ khung chơi (thành phần tĩnh).

Phương thức Draw() vẽ bóng và thanh điều khiển (thành phần động).

Phương thức void Input() nhận nút ấn của người chơi thành lệnh.

Phương thức void Logic() xử lý các va chạm, thoát game, tính điểm.

Phương thức void Run() tạo vòng lặp sử dụng các phương thức ở trên để chơi game. Phương thức void CleanPaddle() làm sạch màn hình khi kết thúc game.

1. **Lớp cBall:**

*Các thuộc tính:*

int x, y; // Tọa độ bóng

int prex, prey; // Tọa độ trước (dùng cho hiển thị)

int originalX, originalY; //Vị trí gốc của bóng

eDir direction; //Hướng di chuyển của bóng

int velocity; //Vận tốc của bóng

Thuộc tính x, y là tọa độ của bóng. prex, prey là tọa độ trước đó của bóng, dùng để xóa tọa độ trước khi bóng di chuyển (có tọa độ mới), khiến bóng trông như di chuyển. originalX, originalY lưu giữ tọa độ ban đầu của bóng, giúp bóng Reset đúng nơi. direction là hướng di chuyển của bóng, là kiểu enum, velocity là vận tốc bóng.

*Các phương thức:*

cBall(int, int);//Constructor

int getX(); // Lấy thuộc tính protected

int getY();

int getPreX();

int getPreY();

eDir getDirection();

void Reset();//Đặt lại bóng

void changeDirection(eDir);//Đổi hướng di chuyển của bóng

void randomDirection();//Ngâu nhiên hướng di chuyển của bóng

void Move();//Di chuyển bóng (thay đổi tọa độ)

Phương thức khởi tạo cBall(int, int); với 2 tham số truyền vô là tọa độ x, y của bóng. Các phương thức get dùng để lấy thuộc tính protected của lớp.

Phương thức Reset() đặt lại tọa độ x, y theo tọa độ origi sau khi xử lý thắng thua.

Phương thức changeDirection(eDir) đổi hướng chuyển động của bóng khi gặp vật cản. Phương thức randomDirection() tạo ngẫu nhiên hướng của bóng khi bắt đầu trò chơi.

Phương thức Move() di chuyển bóng tùy theo hướng của nó.

1. **Lớp cPaddle:**

*Các thuộc tính:*

int x, y; // Tọa độ thanh

int prex, prey;// Tọa độ trước (dùng cho hiển thị)

int originalX, originalY;//Vị trí gốc của thanh

x, y là tọa độ của thanh (điểm chính giữa của thanh). prex, prey giống như thuộc tính của lớp **cBall**, dùng cho hiển thị. originalX, originalY giữ vị trí ban đầu của thanh.

*Các phương thức:*

cPaddle(int posX, int posY);// Hàm tạo

int getX(); // Lấy các thuộc tính protected

int getY();

int getPreX();

int getPreY();

void Reset(); // Đặt lại thanh

void moveUp();//Di chuyển thanh lên (thay đổi tọa độ)

void moveDown();//Di chuyển thanh xuống (thay đổi tọa độ)

Phương thức khởi tạo cPaddle(int posX, int posY), với tham số truyền vô là tạo độ x, y của thanh.

Phương thức get lấy cái thuộc tính protected.

Phương thức Reset() đưa thanh về vị trí ban đầu khi bắt đầu 1 game mới.

Phương thức moveUp(), moveDown() tăng, giảm tọa độ thanh, giúp thanh di chuyển lên, xuống.

# **MÃ NGUỒN**

File cBall.h

#ifndef \_BALL\_H\_

#define \_BALL\_H\_

#include <iostream>

#include <time.h>

#include <conio.h>

using namespace std;

enum eDir { STOP = 0, LEFT = 1, UPLEFT = 2, DOWNLEFT = 3, RIGHT = 4, UPRIGHT = 5, DOWNRIGHT = 6 };//Các di chuyển của bóng

class cBall

{

protected:

int x, y; // Tọa độ bóng

int prex, prey; // Tọa độ trước (dùng cho hiển thị)

int originalX, originalY; //Vị trí gốc của bóng

eDir direction; //Hướng di chuyển của bóng

int velocity; //Vận tốc của bóng

public:

cBall(int, int);//Constructor

int getX(); // Lấy thuộc tính protected

int getY();

int getPreX();

int getPreY();

eDir getDirection();

void Reset();//Đặt lại bóng

void changeDirection(eDir);//Đổi hướng di chuyển của bóng

void randomDirection();//Ngâu nhiên hướng di chuyển của bóng

void Move();//Di chuyển bóng (thay đổi tọa độ)

};

#endif

File cBall.cpp

#include "cBall.h"

cBall::cBall(int posX, int posY)//Constructor{

originalX = posX;

originalY = posY;

x = posX; y = posY;

direction = STOP;

velocity = 1;

}

void cBall::Reset(){

x = originalX; y = originalY;//x, y được gán bằng giá trị ban đầu

direction = STOP;//Ball không di chuyển

}

void cBall::changeDirection(eDir d){

direction = d;

}

void cBall::randomDirection(){

direction = (eDir)((rand() % 6) + 1);

}

int cBall::getX(){

return x;

}

int cBall::getY(){

return y;

}

int cBall::getPreX(){

return prex;

}

int cBall::getPreY(){

return prey;

}

eDir cBall::getDirection(){

return direction;

}

void cBall::Move(){

prex = x;

prey = y;

switch (direction){

case STOP:

break;

case LEFT:

x-=velocity;

break;

case RIGHT:

x+=velocity;

break;

case UPLEFT:

x-=velocity; y-=velocity;

break;

case DOWNLEFT:

x-=velocity; y+=velocity;

break;

case UPRIGHT:

x+=velocity; y-=velocity;

break;

case DOWNRIGHT:

x+=velocity; y+=velocity;

break;

default:

break;

}

}

File cPaddle.h

#ifndef \_PADDLE\_H\_

#define \_PADDLE\_H\_

class cPaddle{

protected:

int x, y; // Tọa độ bóng

int prex, prey;// Tọa độ trước (dùng cho hiển thị)

int originalX, originalY;//Vị trí gốc của bóng

public:

cPaddle(int posX, int posY);// Hàm tạo

int getX(); // Lấy các thuộc tính protected

int getY();

int getPreX();

int getPreY();

void Reset(); // Đặt lại thanh

void moveUp();//Di chuyển thanh lên (thay đổi tọa độ)

void moveDown();//Di chuyển thanh xuống (thay đổi tọa độ)

};

#endif

File cPaddle.cpp

#include "cPaddle.h"

cPaddle::cPaddle(int posX, int posY) {

originalX = posX;

originalY = posY;

x = posX;

y = posY;

prex = posX;

prey = posY;

}

int cPaddle::getX() {

return x;

}

int cPaddle::getY() {

return y;

}

int cPaddle::getPreX() {

return prex;

}

int cPaddle::getPreY() {

return prey;

}

void cPaddle::Reset() {

prex = x;

prey = y;

x = originalX;

y = originalY;

}

void cPaddle::moveUp() {

prey = y;

y -= 1;

}

void cPaddle::moveDown() {

prey = y;

y += 1;

}

File cGameManger.h

#ifndef \_GAME\_H\_

#define \_GAME\_H\_

#include "cBall.h"

#include "cPaddle.h"

#include "DoHoa.h"

class cGameManger{

protected:

int width, height;//Chiều rộng, Chiều cao của game

char up1, down1, up2, down2;//Các nút điều khiển

bool quit;//Điều kiện để thoát game

int score1, score2; // Điểm số

cBall \*ball; // Bóng

cPaddle \*player1; // Thanh điều khiển

cPaddle \*player2;

public:

cGameManger(int, int); // Hàm tạo

~cGameManger(); // Hàm hủy

void ScoreUp(cPaddle\*); // Cộng điểm

void DrawBorder(); // Vẽ khung xung quanh

void Draw(); // Vẽ bóng và thanh điều khiển

void Input(); // Nhận sự kiện từ bàn phím

void Logic(); // Xử lý va chạm

void Run(); // Chạy game

void CleanPaddle(); // Xóa thanh điều khiển sau khi kết thúc game

};

#endif

File cGameManger.cpp

#include "cGameManger.h"

cGameManger::cGameManger(int w, int h) {

srand(time(NULL));//Khởi tạo cho các giá trị random

quit = false;

up1 = 'w'; up2 = 72;//72 là nút lên

down1 = 's'; down2 = 80;//80 là nút xuống trong bảng mã ASCII

width = w; height = h;

score1 = score2 = 0;

ball = new cBall(w / 2, h / 2);

player1 = new cPaddle(2, h / 2);

player2 = new cPaddle(w - 2, h / 2);

}

cGameManger::~cGameManger() {

delete ball, player1, player2;

}

void cGameManger::ScoreUp(cPaddle\* player) {

if (player == player1)

score1++;

else if (player == player2)

score2++;

ball->Reset();

player1->Reset();

player2->Reset();

}

void cGameManger::DrawBorder() {

// ve thanh ngang ben tren

for (int i = 0; i <= width; i++) {

gotoxy(i, 0); textcolor(ColorCode\_Green); cout << "\xB2";

}

// ve thanh doc trai

for (int i = 1; i < height; i++) {

gotoxy(0, i); textcolor(ColorCode\_Green); cout << "\xB2";

}

// ve thanh doc phai

for (int i = 1; i < height; i++) {

gotoxy(width, i); textcolor(ColorCode\_Green); cout << "\xB2";

}

// ve thanh ngang duoi

for (int i = 0; i <= width; i++) {

gotoxy(i, height); textcolor(ColorCode\_Green); cout << "\xB2";

}

}

void cGameManger::Draw() {

int ballx = ball->getX();

int bally = ball->getY();

int prex = ball->getPreX();

int prey = ball->getPreY();

gotoxy(prex, prey); textcolor(ColorCode\_Yellow); cout << " ";

gotoxy(ballx, bally); textcolor(ColorCode\_Yellow); cout << "O";

int player1x = player1->getX();

int player1y = player1->getY();

int p1prex = player1->getPreX();

int p1prey = player1->getPreY();

int checkup1;

if (p1prey > player1y)

checkup1 = 1;

else if (p1prey < player1y)

checkup1 = 0;

else

checkup1 = 2;

for (int i = -4; i <=4; i++) {

gotoxy(player1x, player1y + i); textcolor(ColorCode\_Green); cout << "\xDB";

}

if (checkup1 == 1) { // di len

gotoxy(p1prex, p1prey + 4); textcolor(ColorCode\_Green); cout << " ";

}

else if (checkup1 == 0) { // di xuong

gotoxy(p1prex, p1prey - 4); textcolor(ColorCode\_Green); cout << " ";

}

int player2x = player2->getX();

int player2y = player2->getY();

int p2prex = player2->getPreX();

int p2prey = player2->getPreY();

int checkup2;

if (p2prey > player2y)

checkup2 = 1;

else if (p2prey < player2y)

checkup2 = 0;

else

checkup2 = 2;

for (int i = -4; i <= 4; i++) {

gotoxy(player2x, player2y + i); textcolor(ColorCode\_Green); cout << "\xDB";

}

if (checkup2 == 1) { // di len

gotoxy(p2prex, p2prey + 4); textcolor(ColorCode\_Green); cout << " ";

}

else if (checkup2 == 0) { // di xuong

gotoxy(p2prex, p2prey - 4); textcolor(ColorCode\_Green); cout << " ";

}

gotoxy(3, 43); textcolor(ColorCode\_Yellow); cout << "SCORE 1: " << score1 << endl;

gotoxy(67, 43); textcolor(ColorCode\_Yellow); cout << "SCORE 2: " << score2 << endl;

gotoxy(3, 45); textcolor(ColorCode\_DarkYellow); cout << " ";

gotoxy(3, 47); textcolor(ColorCode\_DarkGreen); cout << " ";

}

void cGameManger::Input() {

// goi den direction cua ball

ball->Move();

int player1x = player1->getX();

int player1y = player1->getY();

int player2x = player2->getX();

int player2y = player2->getY();

//Lấy các sự kiện điều khiển

if (\_kbhit()){

char current = \_getch(); // phim nhan cua nguoi choi

if (current == up1) {

if (player1y - 4 > 1)

player1->moveUp();

}

if (current == up2) {

if (player2y - 4 > 1)

player2->moveUp();

}

if (current == down1) {

if (player1y + 4 < height - 1)

player1->moveDown();

}

if (current == down2) {

if (player2y + 4 < height - 1)

player2->moveDown();

}

// luc bat dau thi cho random huong

if (ball->getDirection() == STOP)

ball->randomDirection();

// thoat game

if (current == 'q' || current == 'Q')

quit = true;

}

}

void cGameManger::CleanPaddle() {

for (int i=1;i<height;i++) {

gotoxy(2,i); cout << " ";

}

for (int i = 1; i < height; i++) {

gotoxy(width-2, i); cout << " ";

}

}

void cGameManger::Logic() {

int ballx = ball->getX();

int bally = ball->getY();

int player1x = player1->getX();

int player1y = player1->getY();

int player2x = player2->getX();

int player2y = player2->getY();

//left paddle

for (int i = -4; i <= 4; i++) {

if (ballx == player1x + 1) {

if (bally == player1y + i) {

if (bally < player1y) {

ball->changeDirection((eDir)(5)); // upright

}

else if (bally > player1y) {

ball->changeDirection((eDir)(6)); //downright

}

else if (bally == player1y) {

ball->changeDirection((eDir)(4)); //right

}

}

}

}

//right paddle

for (int i = -4; i <= 4; i++) {

if (ballx == player2x - 1) {

if (bally == player2y + i) {

if (bally > player2y) {

ball->changeDirection((eDir)(3)); //downleft

}

else if (bally < player2y) {

ball->changeDirection((eDir)(2)); //upleft

}

else if (bally == player2y) {

ball->changeDirection((eDir)(1)); //left

}

}

}

}

//bottom wall

if (bally == height - 1)

ball->changeDirection(ball->getDirection() == DOWNRIGHT ? UPRIGHT : UPLEFT);//Chạm tường trên thì đổi chiều

//top wall

if (bally == 1)

ball->changeDirection(ball->getDirection() == UPRIGHT ? DOWNRIGHT : DOWNLEFT);

//right wall

if (ballx == width - 1) {

ScoreUp(player1);

CleanPaddle();

gotoxy(3, 45); textcolor(ColorCode\_DarkYellow); cout << "Nguoi choi 1 chien thang";

gotoxy(3, 47); textcolor(ColorCode\_DarkGreen); cout << "Bam <Q> de thoat. Bam <ENTER> de choi tiep";

char c = \_getch();

if (c == 'Q' || c == 'q')

quit = true;

}

//left wall

if (ballx == 1) {

ScoreUp(player2);

CleanPaddle();

gotoxy(3, 45); textcolor(ColorCode\_DarkYellow); cout << "Nguoi choi 2 chien thang";

gotoxy(3, 47); textcolor(ColorCode\_DarkGreen); cout << "Bam <Q> de thoat. Bam <ENTER> de choi tiep";

char c = \_getch();

if (c == 'Q' || c == 'q')

quit = true;

}

}

void cGameManger::Run() {

DrawBorder(); // Vẽ khung

while (!quit){

Draw(); //Vẽ đồ họa

Input();//Các kí tự nhập khi chơi game

Logic();//Xử lý chuyển động của bóng

Sleep(50);

}

}

File DoHoa.h

#ifndef \_DOHOA\_H\_

#define \_DOHOA\_H\_

#define ColorCode\_Green 10

#define ColorCode\_Yellow 14

//--------------------------------------------------------

#include <windows.h>

#include <stdio.h>

#include <conio.h>

#include <iostream>

using namespace std;

void Menu(bool&);

void clrscr();

void resizeConsole(int width, int height);

void gotoxy(int column, int line);

void textcolor(int color);

#endif

File DoHoa.cpp

#include "DoHoa.h"

// Hàm thay đổi kích cỡ của khung cmd. Tham số truyền vào là chiều cao, chiều rộng.

void resizeConsole(int Width, int Height){

\_COORD coord;

coord.X = Width;

coord.Y = Height;

\_SMALL\_RECT Rect;

Rect.Top = 0;

Rect.Left = 0;

Rect.Bottom = Height - 1;

Rect.Right = Width - 1;

HANDLE Handle = GetStdHandle(STD\_OUTPUT\_HANDLE); // Get Handle

SetConsoleScreenBufferSize(Handle, coord); // Set Buffer Size

SetConsoleWindowInfo(Handle, TRUE, &Rect); // Set Window Size

}

// Hàm tô màu. Tham số truyền vào là mã màu

void textcolor(int color){

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), color);

}

// Hàm dịch chuyển con trỏ đến tọa độ x, y. Tham số truyền vào là tọa độ

void gotoxy(int column, int line){

COORD coord;

coord.X = column;

coord.Y = line;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coord);

}

// Hàm xóa màn hình.

void clrscr(){

CONSOLE\_SCREEN\_BUFFER\_INFO csbiInfo;

HANDLE hConsoleOut;

COORD Home = { 0,0 };

DWORD dummy;

hConsoleOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

GetConsoleScreenBufferInfo(hConsoleOut, &csbiInfo);

FillConsoleOutputCharacter(hConsoleOut, ' ', csbiInfo.dwSize.X \* csbiInfo.dwSize.Y, Home, &dummy);

csbiInfo.dwCursorPosition.X = 0;

csbiInfo.dwCursorPosition.Y = 0;

SetConsoleCursorPosition(hConsoleOut, csbiInfo.dwCursorPosition);

}

// Hiển thị Menu

void Menu(bool &play){

textcolor(ColorCode\_Yellow);

cout << R"(

\_\_\_\_\_\_

/ \_\_\_\_\_\_\_\_ \_\_\_\_\_ \_\_\_ \_\_\_

/ / \_\_/ \_\_ `/ \_\_ `\_\_ \/ \_ \

/ /\_/ / /\_/ / / / / / / \_\_/

\\_\_\_\_/\\_\_,\_/\_/ /\_/ /\_/\\_\_\_/

)";

textcolor(ColorCode\_Red);

cout << R"(

\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_ \_\_\_ \_\_\_

\_\_\_\_\_ /\_/ / /\ /\_\_/\ / /\ \_\_\_\_\_ \\_\ / /\ /\_\_/\

/ /::\ / /::\ \ \:\ / /:/\_ / /::\ / /::\ \ \:\

/ /:/\:\ / /:/\:\ \ \:\ / /:/ /\ / /:/\:\ / /:/\:\ \ \:\

/ /:/~/::\ / /:/ \:\ \_\_\_\_\_\\_\_\:\ / /:/\_/::\ / /:/~/::\ / /:/~/::\ \_\_\_\_\_\\_\_\:\

/\_\_/:/ /:/\:/\_\_/:/ \\_\_\:/\_\_/::::::::/\_\_/:/\_\_\/\:\ /\_\_/:/ /:/\:/\_\_/:/ /:/\:/\_\_/::::::::\

\ \:\/:/~/:\ \:\ / /:\ \:\~~\~~\\ \:\ /~~/:/ \ \:\/:/~/:\ \:\/:/\_\_\\ \:\~~\~~\/

\ \::/ /:/ \ \:\ /:/ \ \:\ ~~~ \ \:\ /:/ \ \::/ /:/ \ \::/ \ \:\ ~~~

\ \:\/:/ \ \:\/:/ \ \:\ \ \:\/:/ \ \:\/:/ \ \:\ \ \:\

\ \::/ \ \::/ \ \:\ \ \::/ \ \::/ \ \:\ \ \:\

\\_\_\/ \\_\_\/ \\_\_\/ \\_\_\/ \\_\_\/ \\_\_\/ \\_\_\/

)" << endl;

bool thoat = false;

char c = '0';

int luachon = 2;

while ((play == false) && (thoat == false)){

textcolor(ColorCode\_Green);

gotoxy(42, 35);

cout << "Bam <W> hoac <S> de di chuyen";

textcolor(ColorCode\_White);

gotoxy(52, 25); // 52 19

cout << "BAT DAU";

gotoxy(53, 27); // 52 21

cout << "THOAT";

while (c != char(13)){ // ENTER

textcolor(ColorCode\_Green);

gotoxy(49, 25 + luachon);

cout << "->"; // mui ten lua chon

gotoxy(60, 25 + luachon);

cout << "<-";

gotoxy(0, 0);

c = \_getch();

gotoxy(49, 25 + luachon); // xoa mui ten

cout << " ";

gotoxy(60, 25 + luachon); // xoa mui ten

cout << " ";

if (c == 's' || c == 'S'){

if (luachon < 2)

luachon += 2;

}

if (c == 'w' || c == 'W'){

if (luachon > 0)

luachon -= 2;

}

}

if (luachon == 0)

play = true;

if (luachon == 2){

play = false;

thoat = true;

}

luachon = 2;

c = ' ';

system("cls");

}

}

Hàm main.cpp

#include "cGameManger.h"

int main()

{

bool play = false;

resizeConsole(800, 600);//Vẽ màn hình trước khi vào game // windows width 105, height 50

cGameManger gameplay(80, 40);//Set dài rộng cho khung

Menu(play);

if (play == true)

gameplay.Run();

else

return 0;

}

# **SƠ ĐỒ LỚP**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **cGamManger** | | -width, height  -up1, up2, down1, down2  -quit  -cBall  -CPaddle | | -Draw()  -Input()  -Logic()  -Run() |  |  | | --- | | **cPaddle** | | -x,y  -prex,prey  -originalX, originallY | | -Reset()  -MoveUp()  -MoveDown() |  |  | | --- | | **cBall** | | -x,y  -prex,prey  -originalX, originallY  -direction  -velocity | | -Reset()  -ChangeDirection()  -RandomDirection()  -Move() | |

# **LƯU ĐỒ THUẬT TOÁN**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Bắt đầu |  |  | | --- | | ResizeConsole |  |  | | --- | | Menu |  |  | | --- | | Play |   true   |  | | --- | | Run |  |  | | --- | | Logic |  |  | | --- | | Draw |  |  | | --- | | Input |  |  | | --- | | Logic |   true   |  | | --- | | Quit | |

# **HƯỚNG DẪN SỬ DỤNG**

Chương trình được viết dưới dạng command line, đơn giản, trực quan và dễ dùng.

Người dùng bật file .exe lên để chơi. Sử dụng phím <W> hoặc <S> để lựa chọn thao tác, “THOAT” để thoát khỏi game, “BAT DAU” để chơi game.



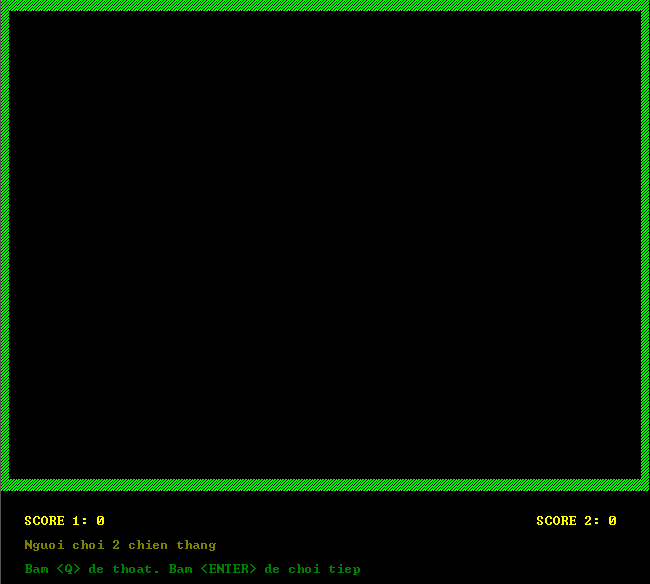
Bóng sẽ được cố định ở trung tâm sàn đấu. Ấn phím <W> hoặc <S> (người chơi 1) hay mũi tên lên hoặc xuống (người chơi 2) để bóng bắt đầu bật.



Khi một người chơi không đỡ được bóng, người còn lại sẽ thua, màn hình sẽ xuất ra thông báo người thắng cuộc, người thắng cuộc sẽ nhận thêm 1 score. Score 1 và Score 2 lần lượt thể hiện số lần thắng của người 1 và người 2.

Người chơi có thể chọn nhập phím <ENTER> để tiếp tục trò chơi hoặc ấn phím <Q> để thoát.

Trong quá trình chơi, nếu muốn thoát sớm hơn, có thể nhấn nút <Q> để thoát trước khi tìm ra người thắng cuộc.



Link video demo game: *https://drive.google.com/open?id=151Vz\_X06odW\_LmKzY8p-LNvtO82jDfhQ*

# **TÀI LIỆU THAM KHẢO**

[1] NVitanovic, C++ Tutorial 18 - Simple Snake Game (Part 1), <https://www.youtube.com/watch?v=E_-lMZDi7Uw&t=126s>

[2] NVitanovic, C++ Tutorial 18 - Simple Snake Game (Part 2), <https://www.youtube.com/watch?v=W1e5wO7XR2w&t=2s>

[3] NVitanovic, C++ Tutorial 18 - Simple Snake Game (Part 3), <https://www.youtube.com/watch?v=PSoLD9mVXTA>