BÁO CÁO ĐỒ ÁN 1

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Giảng viên hướng dẫn: Nguyễn Thành An

Sinh viên thực hiện:

1. 18120529 - Phan Văn Võ Quyền
2. 18120530 – Lê Thị Như Quỳnh

Khoa Công nghệ Thông tin
Đại học Khoa học Tự nhiên TP HCM
Tháng 11/2019

I. MIÊU TẢ ĐỒ ÁN VÀ MỨC ĐỘ HOÀN THÀNH ĐỒ ÁN:

1. Miêu tả đồ án:

Bạn sẽ phát triển một game cho 2 người chơi đối kháng với nhau, với mỗi người chơi sẽ điều khiển thanh trượt để đỡ lấy quả bóng để quả bóng văng về phía người chơi còn lại. Mỗi lần người chơi đỡ được quả bóng sẽ làm cho quả bóng tăng tốc lên (10%) và quả bóng sẽ di chuyển trong sân chơi, Game kết thúc khi một người chơi không hứng được bóng.Trong quá trình chơi quả bóng có thể đổi hướng được nhờ vào va chạm vào thành sân đấu,sự đổi hướng là tuân thủ theo nguyên tắc phản xạ trong vật lý. Các chức năng bạn làm như:

➢ Chức năng cơ bản:

- Hiện màn hình chơi (2d)
- Điều khiển thanh trượt qua lại để hứng bong (2d)
- Xử lý tình huống khi bóng va chạm váo biên sân hay thanh trượt của người chơi(2d)
- Xử lý thắng thua và kết thúc game (2d)
- Tăng tốc độ khi va chạm vào thanh trượt người chơi (2d)

➢ Các trường nâng cao cộng điểm:

- Thiết kế giao diện với màu sắc đẹp (1d)
- Cho thanh trượt chạy tự động để hứng quả bóng (2d)

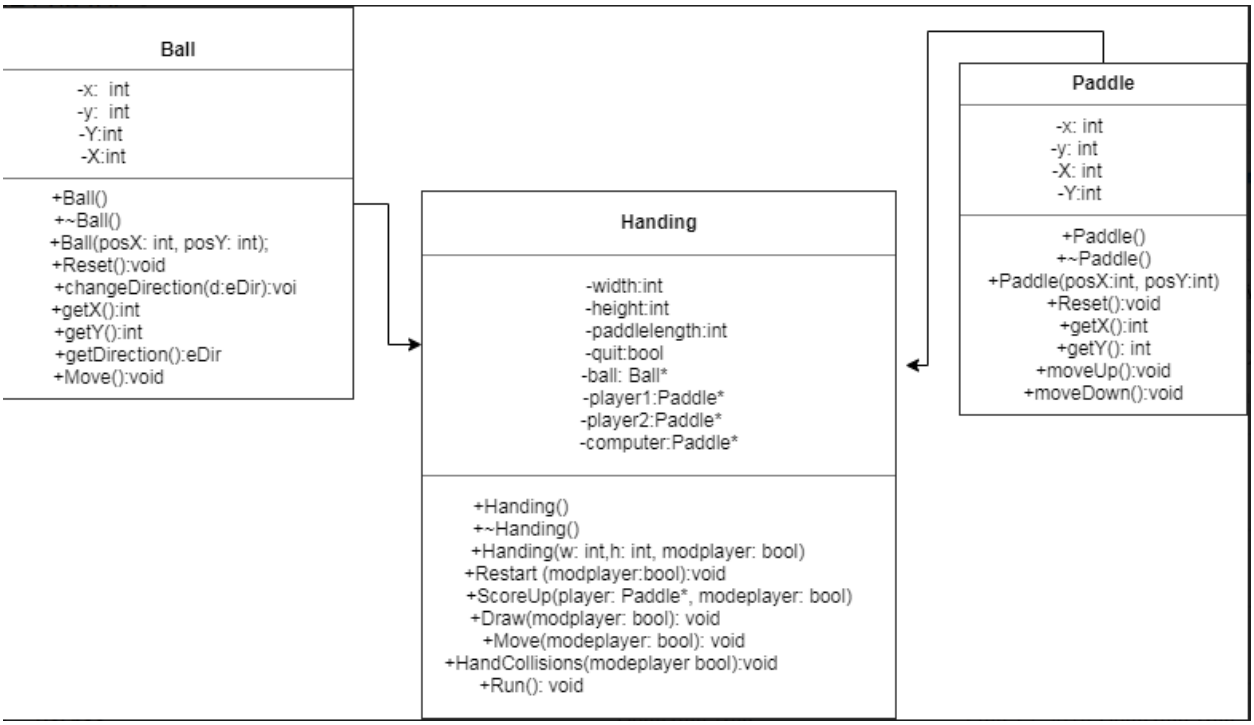
2. Mức độ hoàn thành:

Yêu cầu	Mức độ hoàn thành đồ án
Hiện màn hình chơi	100%
Điều khiển thanh trượt qua lại để hứng bong	100%
Xử lý tình huống khi bóng va chạm váo biên sân hay thanh trượt của người chơi	100%
Xử lý thắng thua và kết thúc game	100%
Tăng tốc độ khi va chạm vào thanh trượt người chơi	100%
Thiết kế giao diện với màu sắc đẹp	70%
Cho thanh trượt chạy tự động để hứng quả bóng	100%

II. PHÂN CHIA CÔNG VIỆC:

1. Phan Văn Võ Quyền – 18120529
   - Viết class Ball
   - Viết hàm Draw()
   - Viết hàm Move()
   - Viết hàm ScoreUp()
2. Lê Thị Như Quỳnh – 18120530
   - Viết class Paddle
   - Viết hàm HandConllusions()
   - Viết hàm Reset()
   - Viết hàm Run()

III. SƠ ĐỒ LỚP:



IV. HƯỚNG DẪN SỬ DỤNG VÀ CHẠY CHƯƠNG TRÌNH:

1. Giải thích hàm:

➢ Thư viện hàm Window và khai báo biến toàn cục:

```
extern int speed;
extern bool modeplayer;
enum eDir { STOP = 0, LEFT = 1, UPLEFT = 2, DOWNLEFT = 3, RIGHT = 4, UPRIGHT = 5,
DOWNRIGHT = 6 };

using namespace std;
void SetConsoleSize(int width, int height);
void removeCursor();
void gotoxy(int x, int y);
void TextColor(int x);
void WindowInit();
void ModeMenu(bool &modeplayer);
```

Giải thích: speed biến tốc độ, mỗi khi bóng va chạm vào thanh trượt thì biến tốc độ sẽ giảm xuống. SetConsoleSize hàm thay đổi kích thước màn hình console. removeCursor hàm nhằm mục đích di chuyển con trỏ màn hình.gotoxy hàm đưa con trỏ đến vị trí x, y trong màn hình console. TextColor hàm vẽ màu. WindowInit gọi hàm thay đổi màn hình console. ModeMenu hàm hiển thị danh sách lựa chon khi bắt đầu game.

➢ Quả Bóng:

```
class Ball
{
private:
	// tọa độ
	int x; int y;
	// tọa độ gốc
	int X; int Y;
	eDir direction;
public:
	Ball();
	~Ball();
	Ball(int posX, int posY);
	void Reset();
	void changeDirection(eDir d);
	//void randomDirection();
	int getX();
	int getY();
	eDir getDirection();
	void Move();
};
```

Thuộc tính: tọa độ hoành x, tọa độ tung y, biến X, Y nhằm mục đích giữ tọa độ cho bóng ở vị trí bắt đầu game. Direction biến thể hướng bay của bóng.

Phương Thức: Ball() hàm khởi tạo mặc định. ~Ball() hàm xóa bóng.

Ball(int posX, int posY); hàm khởi tạo có tham số, gán giá trị tọa độ ban đầu cho bóng. biến originalX,originalY lần lượt giữ giá trị cuả tọa đậu ban đâu, khởi tạo giá trị cho tạo động bóng x = posX; y = posY; khởi tạo hướng bóng ban đầu: đứng yên.

```
X = posX;
Y = posY;
x = posX;
y = posY;
direction = STOP;
```

void Reset(); nhằm mục đích lấy lại tạo độ ban đầu của bóng, hướng bóng đứng yên.

```
void Ball::Reset()
{
	x = X;
	y = Y;
	direction = STOP;
}
```

void changeDirection(eDir d); hàm thay đổi hương di chuyển của bóng.

```
void Ball::changeDirection(eDir d)
{
	direction = d;
}
```

int getX(); int getY(); lần lượt hai hàm lấy giá trị tọa độ của bóng.
void Move();: hàm di chuyển của bóng dựa trên hướng di chuyển của bóng.

```
void Ball::Move()
{
	switch (direction)
	{
	case STOP:
			break;
	case LEFT:
// vòng for tạo độ trễ cho việc cập nhật tọa độ x, coi như đang gián tiếp làm thay đổi tốc độ di chuyển của bóng
			for (int i = 0; i < speed; i++)
			{
			}
			x--;
			break;

	case RIGHT:
			 for (int i = 0; i < speed; i++)
			 {
			 }
			 x++;
			 break;
```

```
        case UPLEFT:
                    for (int i = 0; i < speed; i++)
                    {
                    }
                    x--;
                    y--;
                    break;

        case DOWNLEFT:
                    for (int i = 0; i < speed; i++)
                    {
                    }
                    x--;
                    y++;
                    break;

        case UPRIGHT:
                    for (int i = 0; i < speed; i++)
                    {
                    }
                    x++;
                    y--;
                    break;

        case DOWNRIGHT:
                    for (int i = 0; i < speed; i++)
                    {
                    }
                    x++;
                    y++;
                     break;

        default:
                    break;
                    }
}
```

- Thanh Trượt:

```
  class Paddle
{
private:
        int x; int y;
        int X; int Y;
public:
        Paddle();
        ~Paddle();
        Paddle(int posX, int posY);
        void Reset();
        int getX();
        int getY();
        void moveUp();
        void moveDown();
};
```

Thuộc tính: Thuộc tính: tọa độ hoành x, tọa độ tung y, biến X, Y nhằm mục đích giữ tọa độ ban đầu cho thanh ở vị trí bắt đầu game.
Phương Thức: Paddle(); hàm khởi tạo mặc định
~Paddle():hàm xóa thanh.
Paddle(int posX, int posY):hàm khởi tạo có hai tham số cho thanh.
void Reset(): hàm lấy lại giá trị tọa độ ban đầu cho thanh.

```
void Paddle::Reset()
{
	x = X;
	y = Y;
}
```

int getX(); int getY(); Hai hàm lấy giá trị tọa độ của thanh
void moveUp(); void moveDown();: hàm di chuyển lê xuống thanh

- Các hàm xử lý:

```
class Handing
{
private:
	int width, height;
	int score1, score2;
	int paddleLength;
	bool quit;

	Ball *ball;
	Paddle *player1;
	Paddle *player2;
	Paddle *computer;
public:
	Handing();
	~Handing();
	Handing(int w, int h, bool modepalyer);
	// Hàm khởi tạo game ban đầu
	void Restart(bool modeplayer);
	// Hàm tính điểm
	void ScoreUp(Paddle *player, bool modeplayer);
	// Hàm vẽ khung chơi
	void Draw(bool modeplayer);
	// hàm kiểm tra đầu vào
	void Move(bool modeplayer);
	// hàm xử lí va chạm
	void HandCollisions(bool modeplayer);
	// hàm chạy game
	void Run();
};
```

Thuộc Tính: width, height: chiều rộng và chiều cao của màn hình chơi. score1, score2: điểm số của hai người chơi. paddleLength chiều dài của thanh trượt. quit: biến kiểm tra có kết thúc game hay không. Ball: bóng trong game. player1: người

chơi thứ nhất. player2: người chơi thứ hai nếu chọn chế độ hai người chơi. Computer: người chơi là máy, nếu chọn chế độ chơi với máy.

Phương Thức:

Handing(): phương thức tạo lập mặc định.

~Handing(): giải phóng cho ô nhớ đã xin cấp phát cho bóng, thanh trượt:

```
Handing::~Handing()
{
	delete ball, player1, computer;
}
```

Handing(int w, int h, bool modepalyer): hàm khởi tạo game, khởi tạo giá trị cho thuộc tính. Điểm số bằng 0. chiều dài của thanh trượt bằng 6. Xin cấp phát và Khởi tạo biến bong, player1, tùy vào chế độ chởi thì xin cấp phát và khởi tạo cho biến player2 hay computer.

```
Handing::Handing(int w, int h, bool modepalyer)
{
	srand(time(NULL));
	quit = false;
	score1 = score2 = 0;
	paddleLength = 6;
	width = w; height = h;
	ball = new Ball(w / 2, h / 2);
	player1 = new Paddle(1, h / 2 - 3);
	if (modepalyer)
		player2 = new Paddle(w - 2, h / 2 - 3);
	else
		computer = new Paddle(w - 2, h / 2 - 3);
}
```

Restart(bool modeplayer): hàm nhằm lấy lại giá trị ban đầu cho bóng, thanh trượt.

```
void Handing::Restart(bool modeplayer)
{
	system("cls");
	ball->Reset();
	player1->Reset();
	if (modeplayer)
	{
		player2->Reset();
	}
	else
		computer->Reset();
	TextColor(0);
	Draw(modeplayer);
}
```

ScoreUp: Hàm tăng điểm cho người chơi, In ra màn hình ai là người thắng, cập nhật số điêrm của người thắng. Nếu người chơi nhập Q để thoát game, không chơi nữa, còn nhấn các nút còn lại, thì game sẽ sang màn mới.

```
void Handing::ScoreUp(Paddle *player, bool modeplayer)
{
	if (player == player1)
	{
		TextColor(10);
		gotoxy(width / 2 - 6, height / 2); cout << "PLAYER1 WON";
		score1++;
		// cập nhật điểm ngay lập tức
		gotoxy(width - 60, height + 3);
		cout << "Player1's score: " << score1;
		TextColor(14);
		gotoxy(width / 2 - 13, height / 2 + 1);
		int temp = toupper(_getch());
		if (temp == 'Q')
			quit = true;
		if (quit)
			return;
	}
	// modeplayer == true thì thực hiện thao tác liên quan đến 2 người chơi
	if (modeplayer)
	{
		if (player == player2)
		{
			TextColor(11);
			gotoxy(width / 2 - 6, height / 2); cout << "PLAYER2 WON";
			score2++;
			// cập nhật điểm ngay lập tức
			gotoxy(width - 17, height + 3);
			cout << "Player2's score: " << score2;
			TextColor(14);
			gotoxy(width / 2 - 13, height / 2 + 1);
			int temp = toupper(_getch());
			if (temp == 'Q')
				quit = true;
			if (quit)
				return;
		}
	}
	else
	{
		if (player == computer)
		{
			TextColor(11);
			gotoxy(width / 2 - 6, height / 2); cout << "COMPUTER WON";
			score2++;
			// cập nhật điểm ngay lập tức
			gotoxy(width - 20, height + 3);
			cout << "Computer's score: " << score2;
```

```
                    TextColor(14);
                    gotoxy(width / 2 - 13, height / 2 + 1);
                    int temp = toupper(_getch());
                    if (temp == 'Q')
                          quit = true;
                    if (quit)
                          return;
             }
      }

      Restart(modeplayer);
}
```

Draw(bool modeplayer): hàm vẽ khung chơi, modeplayer là chế độ chơi, nhằm mục đích vẽ khung chơi thích hợp cho chế độ đó, vẽ hướng dân chơi, điểm của các người chơi

void Move(bool modeplayer): Hàm xử lí di chuyển, cho bóng chạy.

Nếu chế độ chơi là chơi với máy thì người chơi sử dụng phím:

W: để di chuyển lên.

S:để di chuyển xuống

Nếu chế đổ chơi là chơi hai người, thì người chơi sử dụng phím:

Người chơi 1:

W: để di chuyển lên.

S: để di chuyển xuống.

Người chơi 2:

O để di chuyển lên.

L đê di chuyên xuống.

Ngoài ra, ở cả hai chế độ chơi còn có các phím.

Q: đẻ thoát game

L: chơi lại màn đang chơi.

_kbhit(): nhằm nhận biết có phím nhập vào. Temp mang giá trị của kí tự vừa nhập, rồi so sanh để di chuyển thanh trượt. Nếu bóng đang trọng trang thái đứng yên thì random bóng để di chuyển. Xử lí thanh computer chạy tự động, nếu bóng chạy về hướng computer, và tọa độ giảm thì computer sẽ chạy lên, tọa độ tăng thì computer sẽ chạy xuống.

```
void Handing::Move(bool modeplayer)
{
      ball->Move();



      // kiểm tra bắt phím, phục vụ di chuyển paddle
```

```
	if (_kbhit())
	{
		int temp = toupper(_getch());

		if (temp == 'W')
			if (player1->getY() > 0)
				player1->moveUp();
		if (temp=='S')
			if (player1->getY() + paddleLength - 1 < height - 1)
				player1->moveDown();
		if (modeplayer)
		{

			if (temp=='O')
				if (player2->getY() > 0)
					player2->moveUp();
			if (temp=='L')
				if (player2->getY() + paddleLength - 1 < height - 1)
					player2->moveDown();
		}
		// nếu là q thì thoát game
		if (temp=='Q')
		{
			quit = true;
		}
		// nếu là r thì chơi ván mới, ghi nhận thành tích ván trước
		if (temp=='R')
		{
			Restart(modeplayer);
		}
	}

	if (ball->getDirection() == STOP) {
		eDir a= (eDir)((rand() % 6) + 1);
		ball->changeDirection(a);
}

	// Hàm xử lý thanh tự động
	if (!modeplayer)
	{
			if (ball->getDirection() == RIGHT)
			{
				if (ball->getY() > computer->getY() + paddleLength - 1)
				{
					if (computer->getY() + paddleLength - 1 < height -
1)
						computer->moveDown();
				}
				else if (ball->getY() < computer->getY())
				{
					if (computer->getY() > 0)
						computer->moveUp();
				}
			}
```

```
                    if (ball->getDirection() == DOWNRIGHT && ball->getY() > computer-
>getY() + paddleLength - 1)
                    {
                          computer->moveDown();
                    }
                    if (ball->getDirection() == UPRIGHT && ball->getY() < computer-
>getY())
                    {
                          if (computer->getY() > 0)
                                computer->moveUp();
                    }

      }
}
```

HandCollisions: hàm xử lí va chạm.
+Đầu tiên xử lí: khi bóng va chạm với người chơi 1, nếu tọa độ của bóng ứng bằng với tọa đọ của một điểm trên thanh trượt, thì bóng sẽ đổi hướng bay theo cơ chế vật lý, riêng với hướng bay thẳng thì sẽ random lại hướng bay khác. nếu thanh trượt hứng được bóng thì speed giảm xuống tương đương tốc độ tăng lên.
+Bước hai: Xem chế độ chơi với máy hay là hai người chơi: còn việc xử lí thì giống với xử lí với ngưởi chơi một
+Bước ba: xử lí chạm biên. Thay đổi hướng bongs theo cơ chết vật lý cho bóng mỗi khi bóng chạm biên.
+Bước 4: xử lí thăng thua. Nếu tọa độ hoành của bòng tọa độ của biên phải thì ngưởi chơi 1 thằng. Nếu tọa đọo của hoành của bòng với biên bên trái thì người chơi 2 thắng.

```
void Handing::HandCollisions(bool modeplayer)
{
      // Kiểm tra va chạm với người chơi 1, xử lí chuyển hướng, người chơi đánh
trúng mới tăng speed lên
      for (int i = 0; i < 6; i++)
      {
            if (ball->getX() == player1->getX() + 1)
                  if (ball->getY() == player1->getY() + i)
                  {

                        // xử lí theo nguyên tắc phản xạ vật lí
                        if (ball->getDirection() == LEFT) {
                              eDir a = eDir(rand() % 3 + 4);
                              ball->changeDirection(a);
                        }
                        if (ball->getDirection() == UPLEFT)
                              ball->changeDirection(UPRIGHT);
                        if (ball->getDirection() == DOWNLEFT)
```

```
                                ball->changeDirection(DOWNRIGHT);
                        // khi đánh trúng bóng thì tăng speed
                        speed = speed * 0.7;
                }
        }

        // Kiểm tra va chạm giữa người chơi 2 và bóng, xử lí chuyển hướng, người chơi đánh trúng mới tăng speed lên
        if (modeplayer)
        {

                for (int i = 0; i < 6; i++)
                {
                        if (ball->getX() == player2->getX() - 1)
                                if (ball->getY() == player2->getY() + i)
                                {

                                        if (ball->getDirection() == RIGHT)
                                        {
                                                eDir a = eDir(rand() % 3 + 1);
                                                ball->changeDirection(a);
                                        }
                                        if (ball->getDirection() == UPRIGHT)
                                                ball->changeDirection(UPLEFT);
                                        if (ball->getDirection() == DOWNRIGHT)
                                                ball->changeDirection(DOWNLEFT);
                                        speed = speed * 0.7;
                                }
                }

        }
        // kiểm tra va chạm giữ computer và bóng
        else
        {

                for (int i = 0; i < 6; i++)
                {
                        if (ball->getX() == computer->getX() - 1)
                                if (ball->getY() == computer->getY() + i)
                                {

                                        if (ball->getDirection() == RIGHT)
                                        {
                                                eDir a = eDir(rand() % 3 + 1);
                                                ball->changeDirection(a);
                                        }
                                        if (ball->getDirection() == UPRIGHT)
                                                ball->changeDirection(UPLEFT);
                                        if (ball->getDirection() == DOWNRIGHT)
                                                ball->changeDirection(DOWNLEFT);
                                        speed = speed * 0.7;
                                }
                }
        }
```

```
	// xử lí va chạm biên
	if (ball->getY() == height - 1)
		ball->changeDirection(ball->getDirection() == DOWNRIGHT ? UPRIGHT :
UPLEFT);
	else if (ball->getY() == 0)
		ball->changeDirection(ball->getDirection() == UPRIGHT ? DOWNRIGHT :
DOWNLEFT);

	// chạm biên phải thì player1 thắng, tăng điểm
	if (ball->getX() == width - 1)
		ScoreUp(player1, modeplayer);
	// chạm biên trái thì palyer2 hoặc computer thắng, tùy vào mode chọn
	else if (ball->getX() == 0)
	{
		if (modeplayer)
		{
			ScoreUp(player2, modeplayer);
		}
		else
		{
			ScoreUp(computer, modeplayer);
		}
	}
}
```

void Handing::Run(): Hàm xử lí game.Bao gồm in màn hình game đầu tiên lên. Quit kết thúc game, khi người chơi muốn kết thúc game. Tạo một vòng lặp, bên trong sẽ sử lí di chuyển của bóng và thanh, xử lí va chạm, vẽ lại màn hình xử lí.

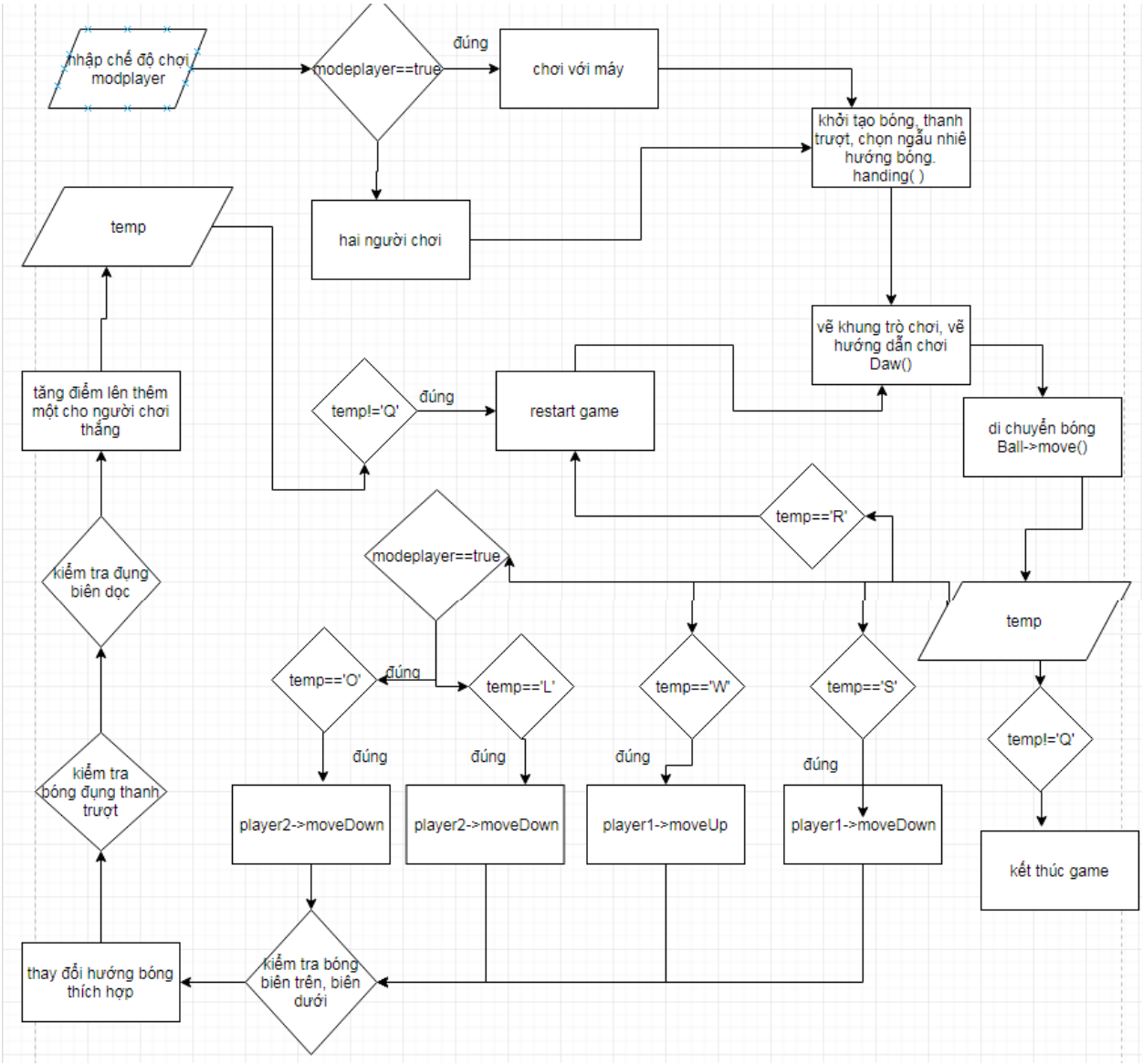
```
// vẽ khung chơi ban đầu, cùng in thông tin
	Draw(modeplayer);

	// thực hiện while để chạy game, chạy đến khi qiut == true thì dừng
	while (!quit)
	{
		srand(time(NULL));
		// kiểm tra đầu vào
		Move(modeplayer);

		// xử lí các va chạm
		HandCollisions(modeplayer);
		// vẽ lại màn hình sau một xử lí
		Draw(modeplayer);
			}
	// in chiến thắng chung cuộc
	if (quit)
	{
		system("cls");
		gotoxy(30, 15);
		if (modeplayer)
		{
```

```
                        if (score1 > score2)
                                cout << "END GAME - WINNER: PLAYER1 - CONGRATULATION" <<
endl;
                        else if (score1 < score2)
                                cout << "END GAME - WINNER: PLAYER1 - CONGRATULATION" <<
endl;
                        else
                                cout << "END GAME - TIE" << endl;
                        gotoxy(30, 16);
                }
                else
                {
                        if (score1 > score2)
                                cout << "END GAME - WINNER: PLAYER1 - CONGRATULATION" <<
endl;
                        else if (score1 < score2)
                cout << "END GAME - WINNER: COMPUTER - CONGRATULATION" << endl;
                        else
                                cout << "END GAME - TIE" << endl;
                        gotoxy(30, 16);
                }
        }
}
```

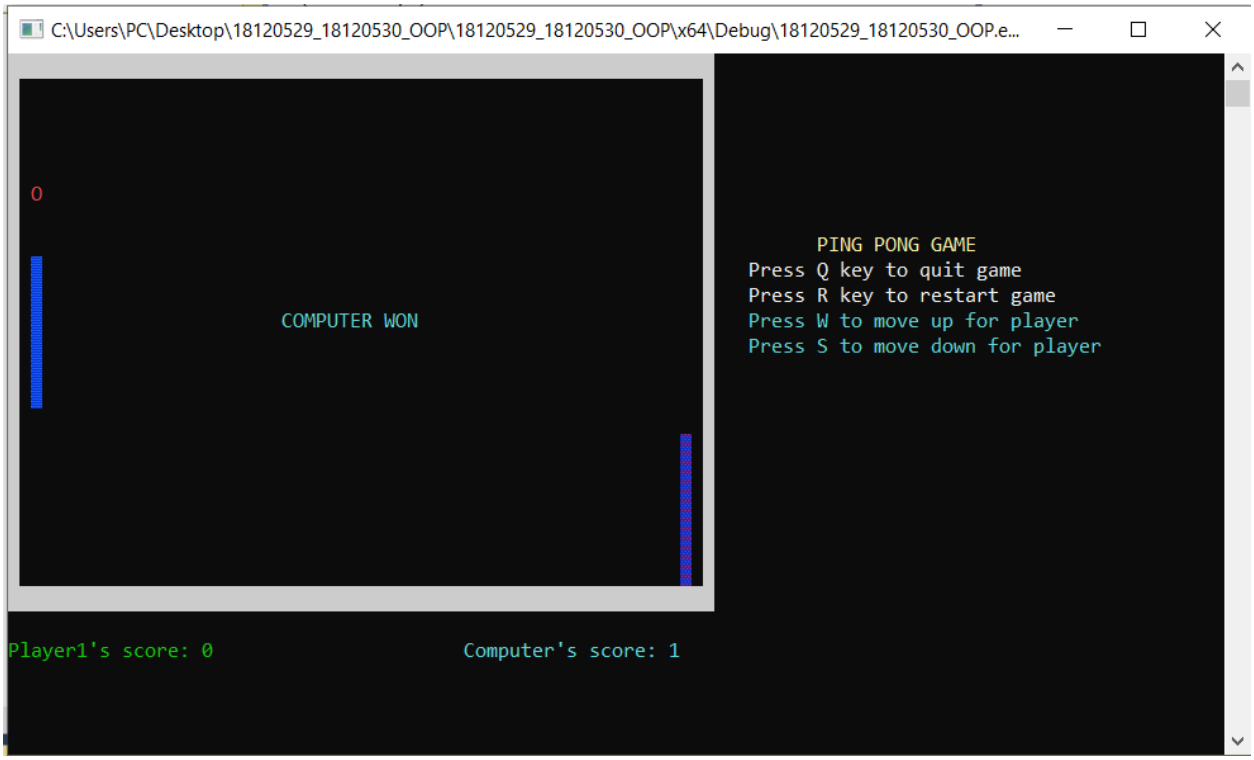
V. LƯU ĐỒ THUẬT TOÁN:



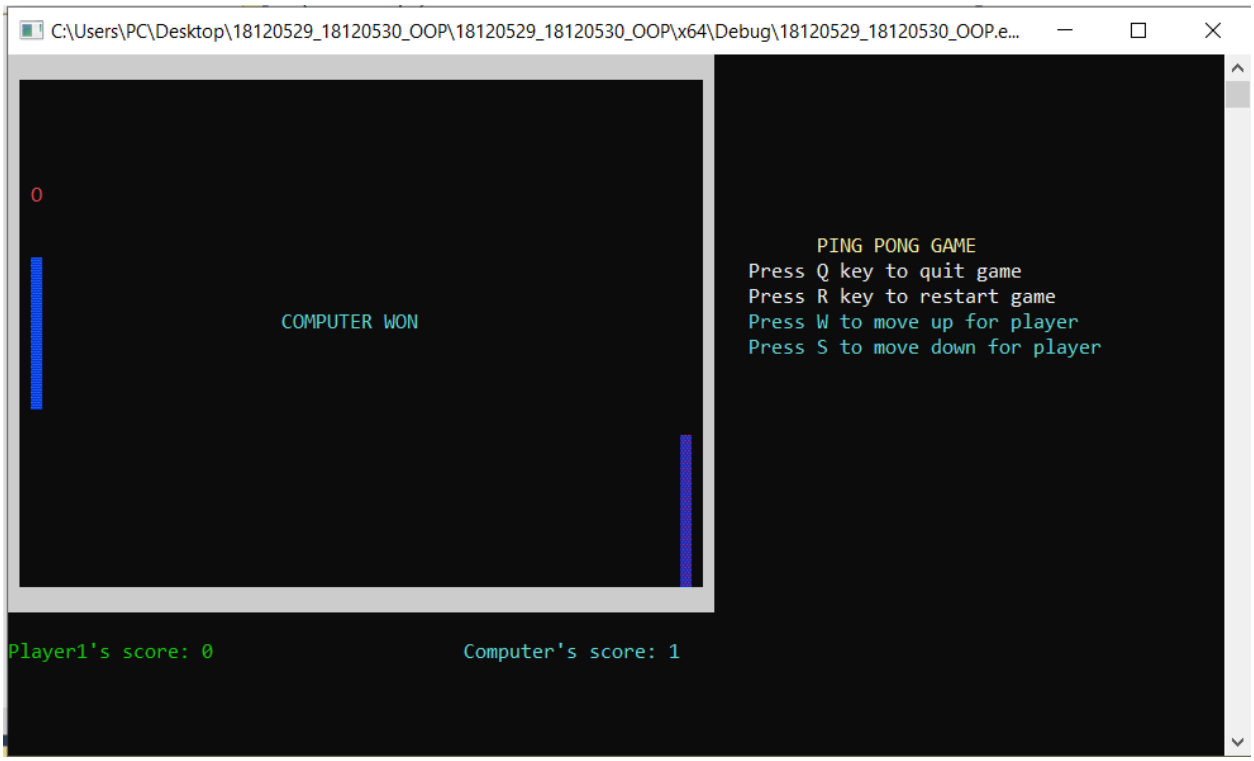
VI. MỘT SỐ HÌNH ẢNH TRONG GAME VÀ VIDEO HƯỚNG DẪN:

- Video hướng dẫn chơi: https://drive.google.com/file/d/1lnM7M8ofFCgmcRtfAveXQ9dA6baWG0Iy/view?usp=sharing
- Chế độ người chơi với máy:

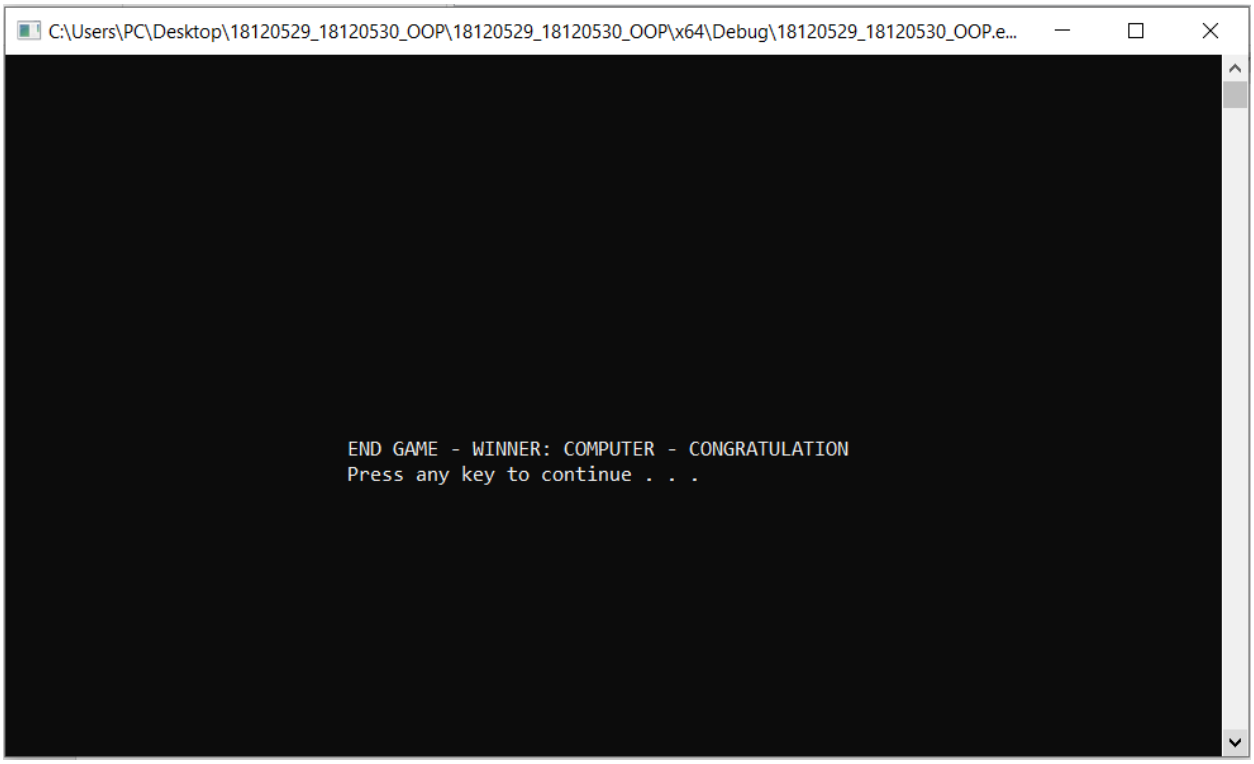
Bắt đầu game



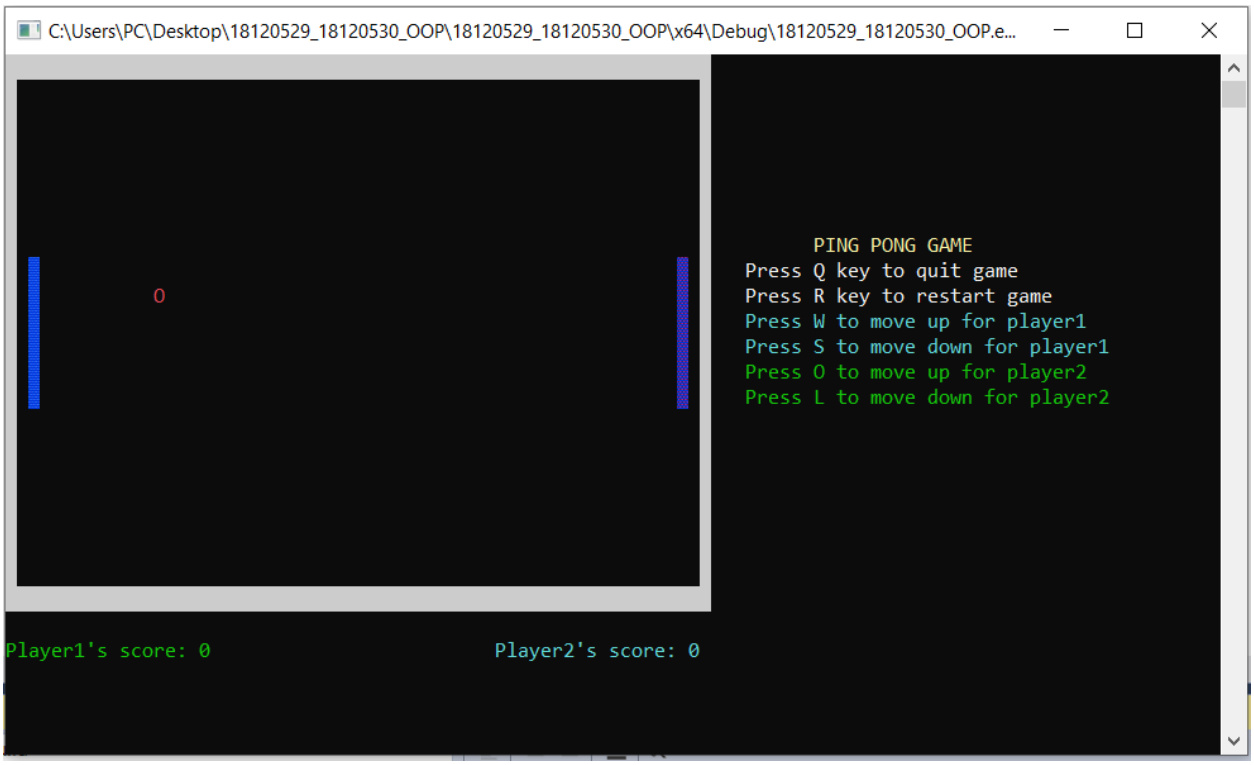
Khi máy thắng



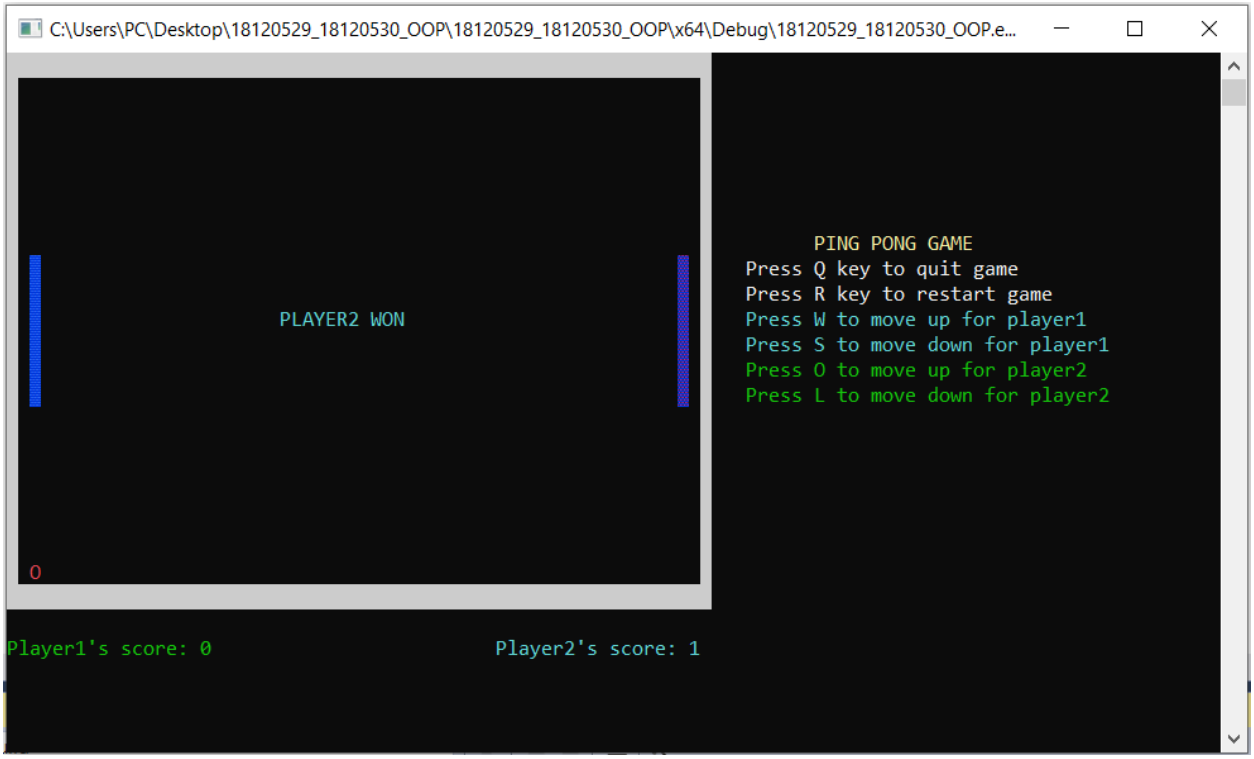
Kết thúc khi ấn nút Q, in ra người thắng chung cuộc



- Chế độ hai người chơi với nhau:
  Bắt đầu game:



Khi người chơi 2 thắng:



Ấn nút Q và kết thúc game, in ra người chiến thắng