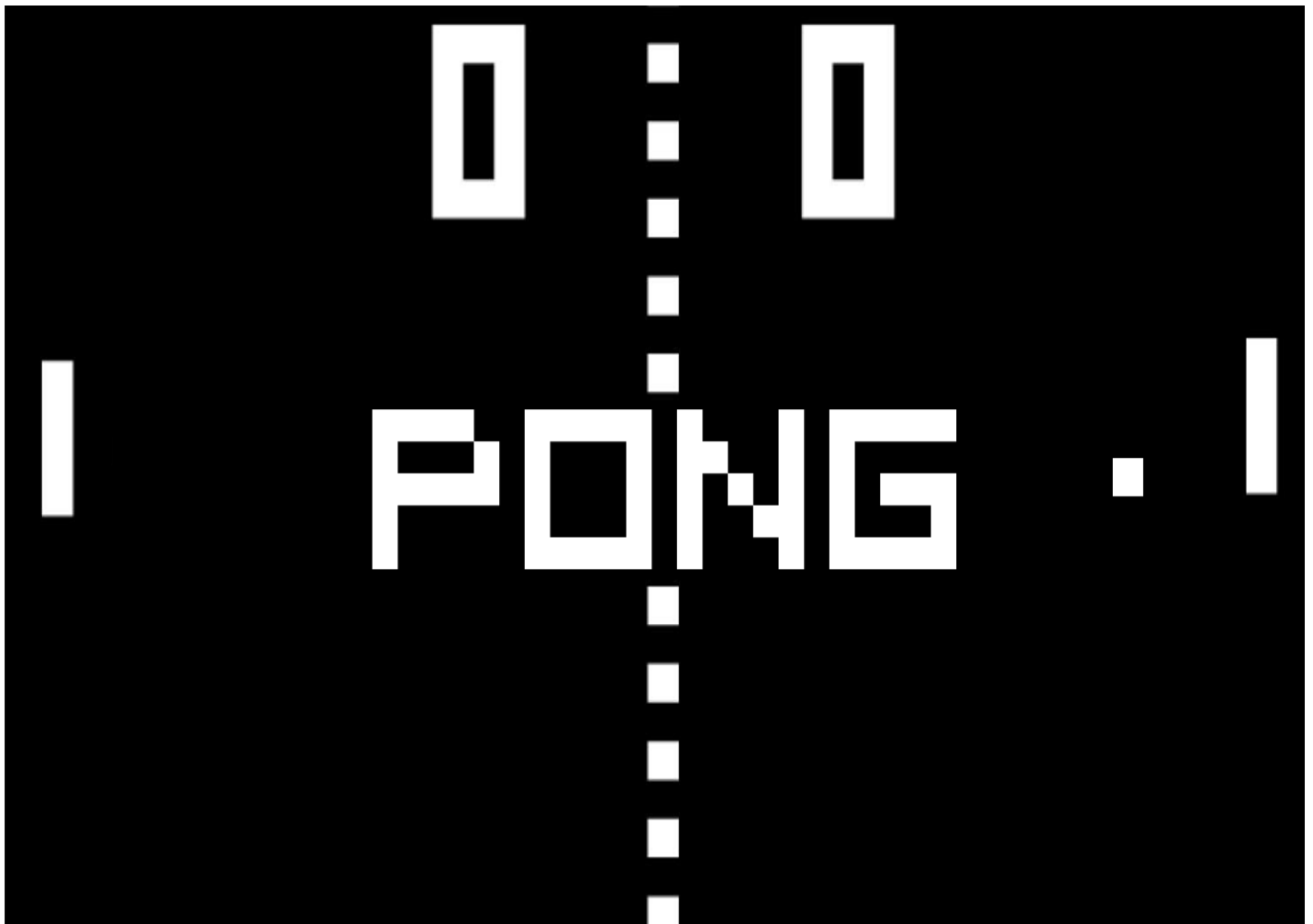


Báo cáo đồ án 1

Pong



Họ và tên: Phan Minh Hưng
MSSV: 1512225

I. Sơ nét về game Pong

Pong là một trong những trò chơi điện tử đầu tiên. Đây là một trò chơi thể thao bóng bàn có đồ họa hai chiều đơn giản. Trò chơi ban đầu được sản xuất bởi Atari, phát hành nó vào năm 1972. Allan Alcorn đã tạo ra Pong như một bài tập huấn luyện được giao cho ông bởi người đồng sáng lập Atari Nolan Bushnell. Bushnell dựa trên ý tưởng về một trò chơi bóng bàn điện tử có trong Magnavox Odyssey; Magnavox sau đó đã kiện Atari vì vi phạm bằng sáng chế. Người đồng sáng lập của Bushnell và Atari Ted Dabney đã rất ngạc nhiên bởi chất lượng công việc của Alcorn và quyết định sản xuất trò chơi.

Pong là một trò chơi thể thao hai chiều mô phỏng bóng bàn. Người chơi điều khiển một mái chèo trong trò chơi bằng cách di chuyển nó theo chiều dọc bên trái hoặc bên phải màn hình. Họ có thể cạnh tranh với một người chơi khác đang điều khiển một mái chèo thứ hai ở phía đối thủ. Người chơi sử dụng mái chèo để đánh bóng qua lại. Mục tiêu là để mỗi người chơi đạt được mười một điểm trước đối thủ; điểm kiếm được khi một người không trả lại bóng cho người khác.



II. Nội dung đồ án và báo cáo tiến trình

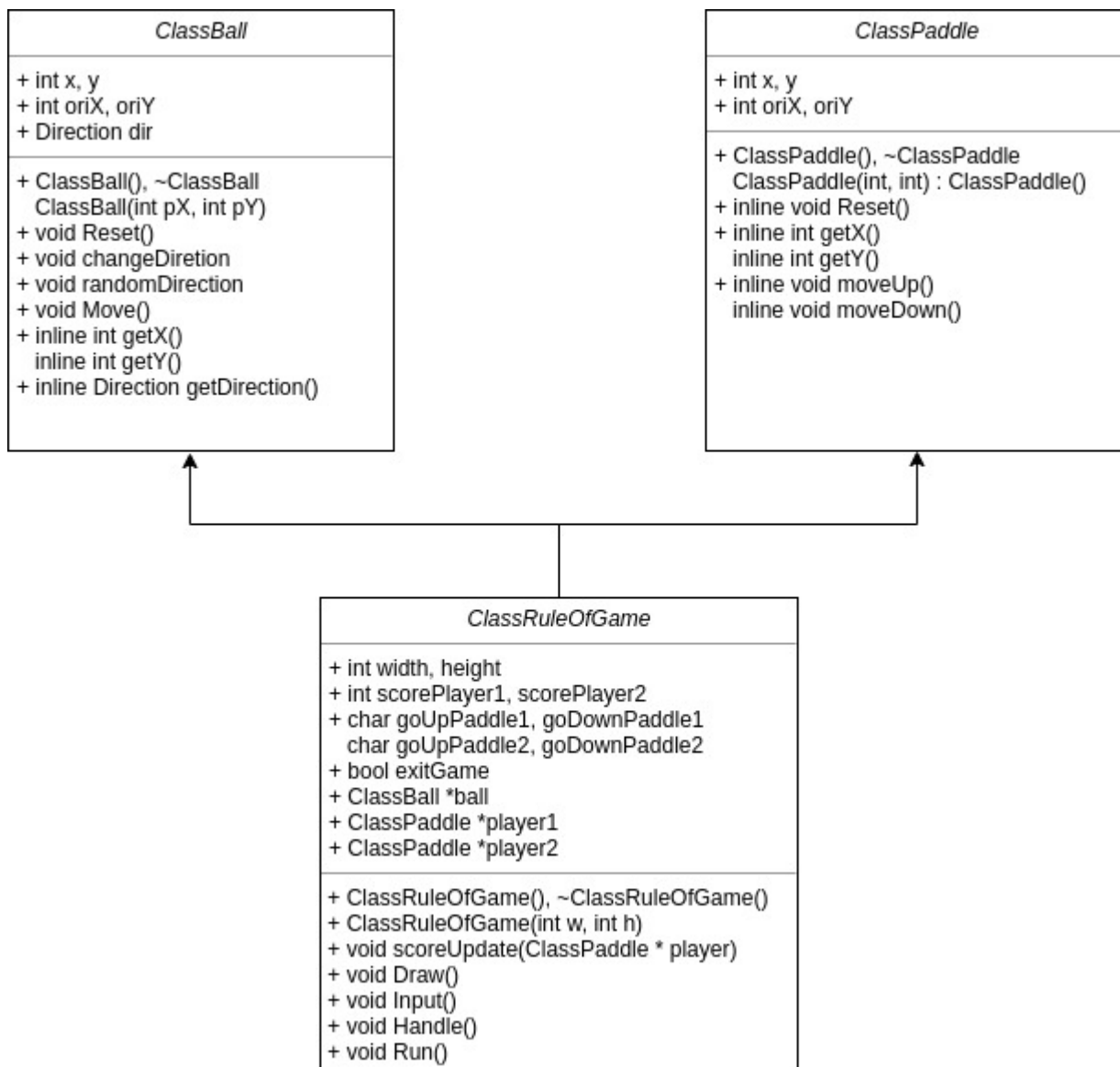
- Xây dựng và phát triển game Pong gồm 2 người chơi đối kháng với nhau, với mỗi người chơi sẽ điều khiển thanh trượt để đỡ lấy quả bóng để quả bóng văng về phía người chơi còn lại. Mỗi lần người chơi đỡ được quả bóng sẽ làm cho quả bóng tăng tốc lên (10%) và quả bóng sẽ di chuyển trong sân chơi, Game 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.

- Yêu cầu và kết quả đạt được:

- + Hiện màn hình chơi : Hoàn thành
- + Điều khiển thanh trượt: Hoàn thành
- + Xử lý va chạm: Hoàn thành
- + Xử thắng thua và kết thúc game: Hoàn thành
- + Tăng tốc độ khi va chạm thanh trượt: Chưa hoàn thành.

III. Cài đặt chương trình

1. Sơ đồ UML



2. Cài đặt chương trình:

a. Xây dựng ClassBall:

```
using namespace std;
enum Direction { STOP = 0, LEFT = 1, UPLEFT = 2, DOWNLEFT = 3, RIGHT = 4, UPRIGHT = 5,
DOWNRIGHT = 6 };

class ClassBall
{
private:
    int x, y;
    int oriX, oriY;
    Direction dir;

public:
    //Khởi tạo mặc định
    ClassBall::ClassBall()
    {}

    //Khởi tạo với tọa độ của Ball
    ClassBall::ClassBall(int pX, int pY)
    {
        oriX = pX;
        oriY = pY;
        x = pX;
        y = pY;
        dir = STOP;
    }

    //Reset vị trí của Ball
    void ClassBall::Reset()
    {
        x = oriX; y = oriY;
        dir = STOP;
    }

    //Thay đổi hướng di chuyển của Ball
    void ClassBall::changeDirection(Direction d)
    {
        dir = d;
    }

    //Ngẫu nhiên hướng di chuyển của Ball
    void ClassBall::randomDirection()
    {
        dir = (Direction)((rand() % 6) + 1);
    }

    //Hàm xác định hướng di chuyển của Ball
    void ClassBall::Move()
    {
        switch (dir)
        {
            case STOP:
                break;
            case LEFT:
                x--;
                break;
            case RIGHT:
```

```

        x++;
        break;
    case UPLEFT:
        x--; y--;
        break;
    case DOWNLEFT:
        x--; y++;
        break;
    case UPRIGHT:
        x++; y--;
        break;
    case DOWNRIGHT:
        x++; y++;
        break;
    default:
        break;
    }
}

//Lấy tọa độ của Ball
inline int ClassBall::getX()
{
    return x;
}

inline int ClassBall::getY()
{
    return y;
}

//Lấy hướng của Ball
inline Direction ClassBall::getDirection()
{
    return dir;
}

//Hàm hủy
ClassBall::~ClassBall()
{
}

};

```

b. Xây dựng ClassPaddle:

```
class ClassPaddle
{
private:
    int x, y;
    int oriX, oriY;
public:
    //Khởi tạo mặc định
    ClassPaddle::ClassPaddle()
    {
        x = y = 0;
    }

    //Khởi tạo mặc định với 2 tham số
    ClassPaddle::ClassPaddle(int pX, int pY) : ClassPaddle()
    {
        oriX = pX;
        oriY = pY;
        x = pX;
        y = pY;
    }

    //Reset vị trí của Paddle
    inline void ClassPaddle::Reset()
    {
        x = oriX;
        y = oriY;
    }

    //Lấy vị trí của Paddle
    inline int ClassPaddle::getX()
    {
        return x;
    }
    inline int ClassPaddle::getY()
    {
        return y;
    }

    //Di chuyển Paddle
    inline void ClassPaddle::moveUp()
    {
        y -= 2;
    }
    inline void ClassPaddle::moveDown()
    {
        y += 2;
    }
};
```

c. Xây dựng ClassRuleOfGame:

```
#include "ClassPaddle.h"

class ClassRuleOfGame
{
private:
    int width, height;
    int scorePlayer1, scorePlayer2;
    char goUpPaddle1, goDownPaddle1, goUpPaddle2, goDownPaddle2;
    bool exitGame;
    ClassBall *ball;
    ClassPaddle *player1;
    ClassPaddle *player2;
public:
    //Khởi tạo hàm quản lý luật chơi
    ClassRuleOfGame(int w, int h)
    {
        srand(time(NULL));
        exitGame = false;
        goUpPaddle1 = 'w'; goUpPaddle2 = 72;
        goDownPaddle1 = 's'; goDownPaddle2 = 80;
        scorePlayer1 = scorePlayer2 = 0;
        width = w; height = h;
        ball = new ClassBall(w / 2, h / 2);
        player1 = new ClassPaddle(1, h / 2 - 3);
        player2 = new ClassPaddle(w - 2, h / 2 - 3);
    }

    //Hàm hủy
    ~ClassRuleOfGame()
    {
        delete ball, player1, player2;
    }

    //Cập nhật điểm cho người chơi
    void scoreUpdate(ClassPaddle * player)
    {
        if (player == player1)
        {
            scorePlayer1++;
            if (scorePlayer1 == 10)
            {
                scorePlayer1 = 0;
                cout << "Player 1 Win!" << endl;
                system("pause");
            }
        }
        else if (player == player2)
        {
            scorePlayer2++;
            if (scorePlayer2 == 10)
            {
                scorePlayer2 = 0;
                cout << "Player 2 Win!" << endl;
                system("pause");
            }
        }

        ball->Reset();
    }
}
```

```

        player1->Reset();
        player2->Reset();
    }

    //Hàm vẽ giao diện trò chơi
    void Draw()
    {
        //Xuất điểm cho người chơi
        cout << "Score Player 1: " << scorePlayer1 << endl << "Score Player 2: " <<
scorePlayer2 << endl;

        system("cls");
        for (int i = 0; i < width + 2; i++)
            cout << "\xB2";
        cout << endl;

        for (int i = 0; i < height; i++)
        {
            for (int j = 0; j < width; j++)
            {
                int ball_x = ball->getX();
                int ball_y = ball->getY();
                int player1_x = player1->getX();
                int player2_x = player2->getX();
                int player1_y = player1->getY();
                int player2_y = player2->getY();

                if (j == 0)
                    cout << "\xB2";

                //Vẽ ball
                if (ball_x == j && ball_y == i)
                    cout << "0";

                //Vẽ paddle
                else if (player1_x == j && player1_y == i)
                    cout << "\xDB"; //player1
                else if (player1_x == j && player1_y + 1 == i)
                    cout << "\xDB"; //player1
                else if (player1_x == j && player1_y + 2 == i)
                    cout << "\xDB"; //player1
                else if (player1_x == j && player1_y + 3 == i)
                    cout << "\xDB"; //player1

                else if (player2_x == j && player2_y == i)
                    cout << "\xDB"; //player2
                else if (player2_x == j && player2_y + 1 == i)
                    cout << "\xDB"; //player1
                else if (player2_x == j && player2_y + 2 == i)
                    cout << "\xDB"; //player1
                else if (player2_x == j && player2_y + 3 == i)
                    cout << "\xDB"; //player1
                else
                    cout << " ";

                if (j == width - 1)
                    cout << "\xB2";
            }
            cout << endl;
        }
    }

```



```

        for (int i = 0; i < width + 2; i++)
            cout << "\xB2";
        cout << endl;
    }

    void Input()
    {
        ball->Move();

        int ball_x = ball->getX();
        int ball_y = ball->getY();
        int player1_x = player1->getX();
        int player2_x = player2->getX();
        int player1_y = player1->getY();
        int player2_y = player2->getY();

        if (_kbhit())
        {
            char current = _getch();
            if (current == goUpPaddle1)
                if (player1_y > 0)
                    player1->moveUp();
            if (current == goUpPaddle2)
                if (player2_y > 0)
                    player2->moveUp();
            if (current == goDownPaddle1)
                if (player1_y + 4 < height)
                    player1->moveDown();
            if (current == goDownPaddle2)
                if (player2_y + 4 < height)
                    player2->moveDown();

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

            //Thoát game bằng nút z
            if (current == 'z' || current == 'Z')
                exitGame = true;
        }
    }

    //Hàm xử lý va chạm
    void Handle()
    {
        int ball_x = ball->getX();
        int ball_y = ball->getY();
        int player1_x = player1->getX();
        int player2_x = player2->getX();
        int player1_y = player1->getY();
        int player2_y = player2->getY();

        //Xử lý ball va chạm paddle trái
        for (int i = 0; i < 4; i++)
            if (ball_x == player1_x + 1)
                if (ball_y == player1_y + i)
                    ball->changeDirection((Direction)((rand() % 3) + 4));
    }

```

```

        //Xử lý ball va chạm paddle phải
        for (int i = 0; i < 4; i++)
            if (ball_x == player2_x - 1)
                if (ball_y == player2_y + i)
                    ball->changeDirection((Direction)((rand() % 3) + 1));

        //Xử lý ball va chạm Wall dưới
        if (ball_y == height - 1)
            ball->changeDirection(ball->getDirection() == DOWNRIGHT ? UPRIGHT :
UPLEFT);

        //Xử lý ball va chạm Wall trên
        if (ball_y == 0)
            ball->changeDirection(ball->getDirection() == UPRIGHT ? DOWNRIGHT :
DOWNLEFT);

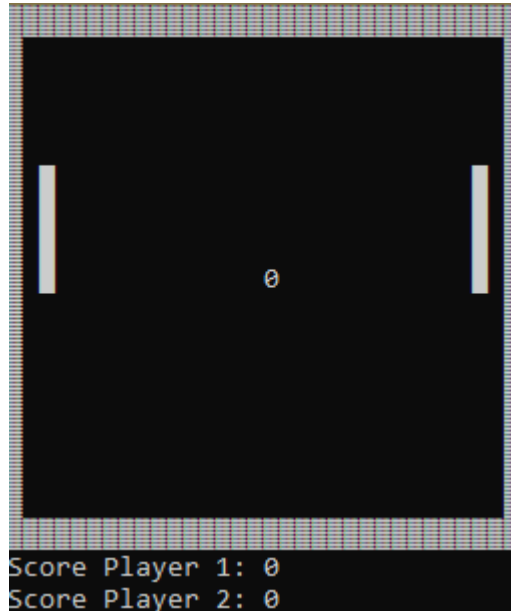
        //Xử lý ball va chạm Wall phải
        if (ball_x == width - 1)
            scoreUpdate(player1);
        //Xử lý ball va chạm Wall trái
        if (ball_x == 0)
            scoreUpdate(player2);
    }

    //Hàm run game
    void Run()
    {
        while (!exitGame)
        {
            Draw();
            Input();
            Handle();
        }
    }
};

```

3. Demo và hướng dẫn chơi game

a. Demo



b. Gameplay:

- Di chuyển: Di chuyển thanh điều hướng để đón lấy bóng
- + Player 1: w is up, s is down.
- + Player 2: ⬆ is up, ⬇ is down.
- Thoát game: phím z hoặc Z dùng để thoát trò chơi.

IV. Tự đánh giá và nhận xét:

- Đồ án hoàn thành 80% yêu cầu.
- Source đơn giản và dễ dàng cập nhật và nâng cấp chức năng về sau.
- Có tham khảo source code từ nhiều nguồn trên Internet.