



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

NHÓM TT

ĐỒ ÁN BÁO CÁO GIỮA KÌ

|Đề tài|

PONG GAME

ĐOÀN MINH TÂN - 18120234

HUYỀN HỒ THANH TRÀ - 18120245

Thành phố Hồ Chí Minh – 11/2019



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

NHÓM TT

ĐỒ ÁN BÁO CÁO GIỮA KÌ

|Đề tài|

PONG GAME

|Giảng viên hướng dẫn|

Ths.Nguyễn Thành An

Thành phố Hồ Chí Minh – 11/2019

LỜI CẢM ƠN

Trong thời gian làm đồ án cuối kì, nhóm em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè. Nhóm em xin gửi lời cảm ơn chân thành đến Th.s Nguyễn Thành An, giảng viên môn Lập trình hướng đối tượng Trường Đại học Khoa Học Tự Nhiên TP HCM, người đã tận tình hướng dẫn, chỉ bảo nhóm em trong suốt quá trình làm đồ án. Em cũng xin chân thành cảm ơn các thầy cô giáo trong trường Đại học Khoa Học Tự Nhiên TP HCM nói chung đã dạy dỗ cho nhóm em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ nhóm em trong suốt quá trình học tập. Cuối cùng, nhóm em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên nhóm em trong suốt quá trình học tập và hoàn thành đồ án giữa kì.

MỤC LỤC

I. CÁCH TỔ CHỨC:	6
1. Menu:	6
2. Paddle:	8
3. Ball:	9
4. Border:	10
5. Score:.....	10
6. Header Manager:.....	11
II. CÁCH CHƠI:	12
1. Chế độ 1 người chơi:	13
2. Chế độ 2 người chơi:	13

DANH MỤC HÌNH

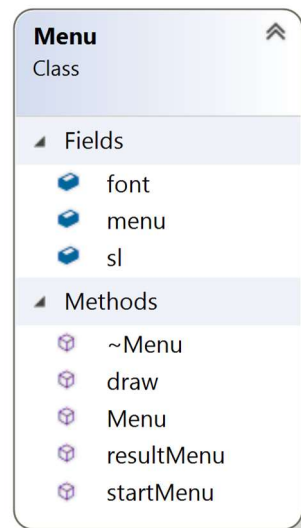
Hình 1:UML của class Menu	6
Hình 2: Constructor Menu.....	6
Hình 3: Cài đặt hàm startMenu	7
Hình 4: UML của class Paddle	8
Hình 5: UML của class Ball	9
Hình 6: UML của class Border	10
Hình 7: UML của class Score.....	11
Hình 8: Menu chính	12
Hình 9: Giao diện khi đang chơi.....	13

I. CÁCH TỔ CHỨC:

Đồ án được chia thành nhiều đối tượng để dễ quản lý: Border, Paddle, Ball, Manager, Score, Menu.

1. Menu:

Được tổ chức và cài đặt như sau:



Hình 1: UML của class Menu

Gồm có:

- Thuộc tính: *con trỏ menu* (để giữ cái text có trong menu), *font* (font chữ cho từng text trong menu), *sl* (số lượng text có trong menu).
- Phương thức:
 - + *constructor Menu()*: dùng để thiết lập các thuộc tính ban đầu cũng như lấy font từ file đưa vào trong chương trình.

```

Menu::Menu()
{
    if (!font.loadFromFile("pong.ttf"))
    {
        cout << "Font file not found!" << endl;
        exit(-1);
    }
    menu = NULL;
    sl = 0;
}
  
```

Hình 2: Constructor Menu

+ `void startMenu()`: để cài đặt cho menu ban đầu (menu đầu tiên khi vào game):

```
void Menu::startMenu()
{
    sl = 3; //số lượng Text có trong start menu
    menu = new Text[sl];

    //thiết lập các thuộc tính cho text
    //thiết lập font
    menu[0].setFont(font);
    //thiết lập nội dung của text
    menu[0].setString("PONG");
    //thiết lập size chữ
    menu[0].setCharacterSize(200);
    //thiết lập màu chữ
    menu[0].setFillColor(Color::White);
    //thiết lập vị trí tâm của text.
    menu[0].setOrigin(Vector2f(menu[0].getGlobalBounds().width / 2, menu[0].getGlobalBounds().height / 2));
    //thiết lập vị trí của text trên cửa sổ.
    menu[0].setPosition(sf::Vector2f(screenX / 2, screenY / (sl + 1) * 1));

    menu[1].setFont(font);
    menu[1].setFillColor(sf::Color::White);
    menu[1].setString("PLAY");
    menu[1].setOrigin(Vector2f(menu[1].getGlobalBounds().width / 2, menu[1].getGlobalBounds().height / 2));
    menu[1].setPosition(sf::Vector2f(screenX / 2, screenY / (sl + 1) * 2.5));

    menu[2].setFont(font);
    menu[2].setFillColor(sf::Color::White);
    menu[2].setString("QUIT");
    menu[2].setOrigin(Vector2f(menu[2].getGlobalBounds().width / 2, menu[2].getGlobalBounds().height / 2));
    menu[2].setPosition(sf::Vector2f(screenX / 2, screenY / (sl + 1) * 3));
}
```

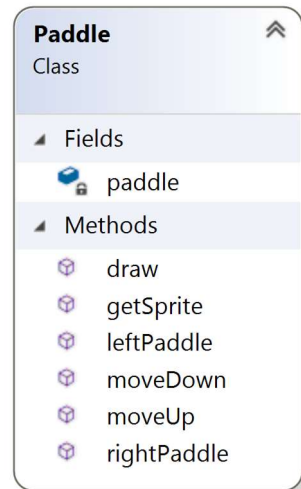
Hình 3: Cài đặt hàm `startMenu`

```
+ void Menu::draw(RenderWindow& window)
{
    //in từng phân tử (text) của menu ra cửa sổ
    for (int i = 0; i < sl; i++){
        window.draw(menu[i]);
    }
}
➔ Dùng để in menu (các text có trong menu ra cửa sổ).

+ void Menu::resultMenu(int& winner)
//in kết quả thắng thua ra màn hình
{
    //Thiết lập các thuộc tính cho các text trong Menu kết
    quả , trong đó có thông báo thắng thua, Thoát (Quit) hay chơi
    lại (Replay)
    // ..... Chi tiết nằm bên trong file source code.....
}
```

2. Paddle:

Paddle : Thanh trượt để hứng banh (tượng trưng cho người chơi). Được tổ chức và cài đặt như sau:



Hình 4: UML của class Paddle

Gồm có:

- Các thuộc tính: `RectangleShape` paddle (để chỉ thanh cuộn có thuộc tính là một hình chữ nhật).
- Các phương thức:

```
+ void leftPaddle()
```

```
{
```

```
// thiết lập kích thước, màu sắc, vị trí cho thanh trượt bên trái.
```

```
}
```

```
+ Tương tự như thế ta có void Paddle::rightPaddle() cho thanh trượt bên phải.
```

```
+ void Paddle::draw(RenderWindow& window)
```

```
//vẽ thanh trượt
```

```
+ void Paddle::moveUp(Border& topBorder)
```

```
//di chuyển thanh trượt lên// di chuyển với khoảng cách bằng 1/2 chiều dài của thanh trượt
```

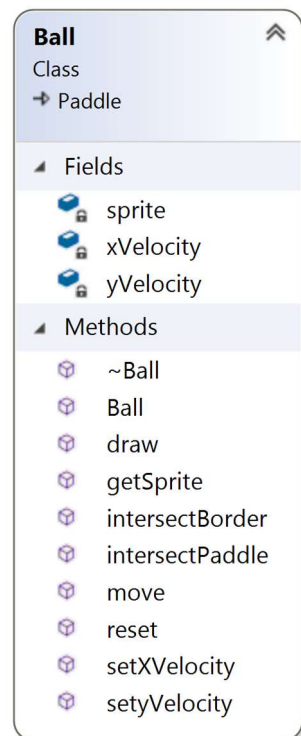
```
+ void Paddle::moveDown(Border& botBorder)
```

```
//di chuyển thanh trượt xuống. di chuyển với khoảng cách bằng 1/2 chiều dài của thanh trượt.
```

```
+ RectangleShape getSprite() { return paddle; };// lấy kiểu đối tượng.
```


3. Ball:

Đối tượng Ball được tổ chức và cài đặt như sau:



Hình 5: UML của class Ball

Gồm có:

- Các thuộc tính;
 - + prite : với kiểu `CircleShape` (hình tròn)
 - + float `xVelocity` và `yVelocity` (vận tốc theo x và y của quả banh)
- Các phương thức sau:


```

+ void setXVelocity(float x):
//thiết lập giá trị vận tốc theo X

+ void setyVelocity(float y);
//thiết lập giá trị vận tốc theo Y

+ CircleShape getSprite() { return sprite; };
//lấy kiểu đối tượng của Ball

+ void intersectBorder(){
    // xử lý va chạm của quả bóng với các bức tường.
}
+ void intersectPaddle(){

```

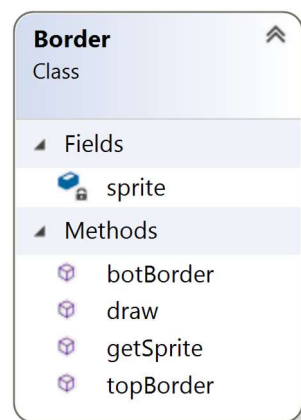
```

        // xử lý va chạm của quả bóng với hai thanh cuộn.
    }
+ void draw(RenderWindow& window); // vẽ quả bóng ra cửa sổ game
+ void move(); // thay đổi tọa độ của Ball → di chuyển Ball
+ void reset(); // Đưa quả bóng về trạng thái ban đầu.

```

4. Border:

Đối tượng Border được tổ chức và cài đặt như sau:



Hình 6: UML của class Border

Gồm có:

- Các thuộc tính: 1 biến sprite mang kiểu `RectangleShape`.
- Các phương thức:

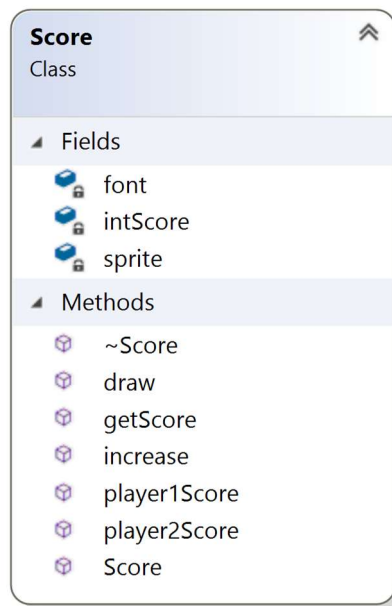
```

+ void topBorder(); // thiết lập tường trên
+ void botBorder(); // thiết lập tường dưới
+ RectangleShape getSprite() { return sprite; }; // hàm lấy kiểu mà đối tượng đang giữ
+ void draw(RenderWindow& window); // vẽ border ra cửa sổ game

```

5. Score:

Đối tượng score được tổ chức và quản lý như sau:



Hình 7: UML của class Score

Gồm:

- Các thuộc tính: font (lưu font chữ cho điểm hiển thị trên màn hình của sổ, được đọc từ file), intScore (lưu điểm người chơi), sprite (kiểu text để xuất ra màn hình của sổ game)
- Các phương thức:
 - + Constructor `Score::Score()`: để thiết lập ban đầu cho các thuộc tính, cũng như là load font từ file vào chương trình.
 - + `void player1Score();` //thiết lập vị trí xuất hiện cho điểm người chơi 1 (người chơi bên trái).
 - + `void player2Score();` //thiết lập vị trí xuất hiện cho điểm người chơi 2 (người chơi bên phải).
 - + `void increase();` // tăng điểm của người chơi lên 1 điểm
 - + `void draw(RenderWindow& window);` //in điểm của người chơi ra màn hình
 - + `int getScore() { return intScore; };` //hàm lấy điểm của người chơi

6. Header Manager:

Được tổ chức như sau:

```

//tra ve 1 neu player1 thang, 2 neu player2 thang
int playWithPerson(RenderWindow& window);

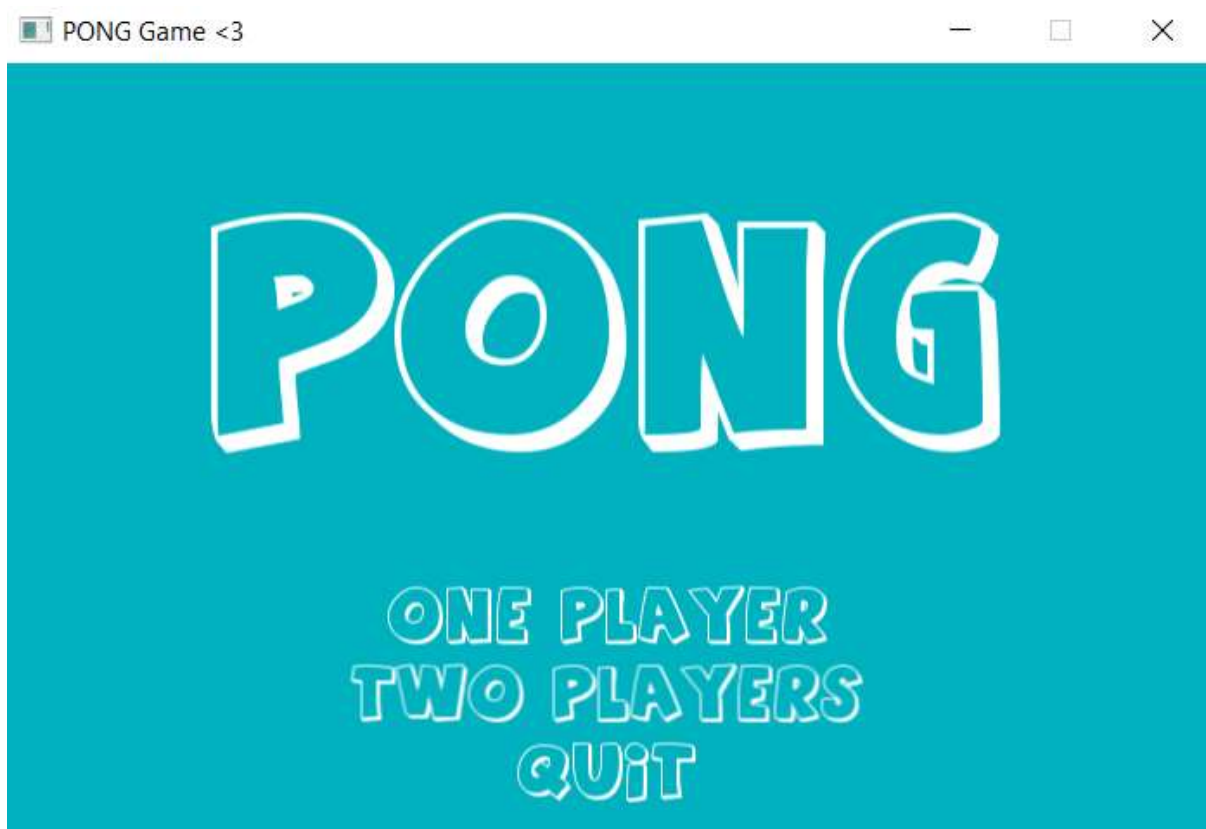
//tra ve 1 neu player1 thang, 2 neu player2 thang
//nguoi chơi là player2
  
```

```
int playWithComputer(RenderWindow& window);

void runStartMenu(RenderWindow& window, int& mode)
{
    //truyền vào biến kiểu nguyên mode
    //biến mode sẽ nhận giá trị là 1 nếu người chơi chọn chế
    độ ONEPLAYER
    //biến mode sẽ nhận giá trị là 2 nếu người chơi chọn chế
    độ TWOPLAYERS
}
void runResultMenu(RenderWindow& window, int& winner)
{
    //Nếu tham số winner truyền vào là 1 thì in "WIN" bên sân
    chơi của player1 ngược lại thì in bên sân chơi của player2,
    khi đó bên sân chơi của đối phương sẽ in chữ ""LOSE""
}
}
```

II. CÁCH CHƠI:

- Trong thư mục Release, chạy file Pong.exe.
- Mới vào, ta sẽ thấy một cửa sổ hiện ra như sau:



Hình 8: Menu chính

- Nếu click vào “ONE PLAYER” thì cửa sổ sẽ chuyển sang chế độ 1 người chơi và chơi tự động với “máy” – người chơi còn lại, được lập trình tự động.
- Nếu click vào “TWO PLAYERS” thì cửa sổ sẽ chuyển sang chế độ 2 người chơi:
 - + Với người chơi bên trái: điều khiển lên xuống bằng: nhấn “W” (lên), “S” (xuống).
 - + Với người chơi bên phải: nhấn điều hướng tên lên để đi lên và nhấn nút điều hướng xuống để đi xuống.
- Nếu click “QUIT” thì thoát game.

1. Chế độ 1 người chơi:

- Người chơi tự động sẽ được mặc định là người chơi bên trái, còn người chơi thật thì mặc định là người chơi bên phải.
- Cách tính điểm cực kì đơn giản: sau mỗi lần bắt trượt bóng thì điểm đối phương sẽ được tăng lên 1 điểm cứ tiếp tục như thế cho tới khi đạt được 5 điểm thì chiến thắng.

2. Chế độ 2 người chơi:

- Tương tự như chế độ một người chơi, nhưng đối thủ lại là một người chơi ảo được lập trình tự động, cách tính điểm và quyết định thắng thua cũng giống với chế độ 1 người chơi.
- Sau khi kết thúc game và kết quả hiển thị ra màn hình như sau:



Hình 9: Giao diện khi đang chơi

- Nếu muốn chơi lại thì click chọn “REPLAY”.
- Nếu muốn thoát game thì nhấn “QUIT” hoặc nhấn nút “X” góc phải bên trên cửa sổ.

PHỤ LỤC

Link video Demo:

https://drive.google.com/open?id=1Of4oyAqnCnFfQ2H7NtV_hlmxHLSi_c85