

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP HCM
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----

BẢN BÁO CÁO

ĐỒ ÁN GIỮA KÌ

Giảng viên: Nguyễn Thành An

Sinh viên 1: Nguyễn Ngọc Đức

MSSV 1: 18120165

Sinh viên 2: Nguyễn Viết Dũng

MSSV 2: 18120167

Hồ Chí Minh, 2019

BÁO CÁO ĐỒ ÁN GAME PING PONG

I. Danh sách thành viên

Họ và tên	MSSV	Chữ ký
Nguyễn Ngọc Đức	18120165	
Nguyễn Viết Dũng	18120167	

II. Danh sách mục tiêu:

- **Đã hoàn thành:**
 - Hiện màn hình chơi.
 - Save game, bảng thành tích.
 - Xử lý thắng thua và kết thúc game.
 - Tăng tốc độ khi va chạm vào thanh trượt người chơi và đổi hướng bóng khi va chạm vật.
 - Ý tưởng thêm :
 - Vật chắn cố định trong game.
 - Đa dạng về cách chơi trong 5 màn.
 - Thanh trượt chạy tự động để hứng bóng.
 - Thiết kế giao diện với màu sắc đẹp.
 - Nâng cao độ khó của game khi tạo ra màn chơi có đường đến đích cho quả bóng nếu quả bóng chạm đúng điểm cần chạm.

III. Công việc của mỗi thành viên, các phương thức và thuộc tính:

1. Công việc của mỗi thành viên:

Đức:

- Xây dựng bóng: xây dựng khối vật, màu sắc, khởi tạo vị trí, reset vị trí mỗi màn, tốc độ bóng.
- Xây dựng vợt: khởi tạo vị trí, độ lớn, màu sắc, di chuyển.
- Xây dựng tiện ích cho game, các phím nhập vào và kiểm tra phím nhập.
- lưu màn.
- Thanh trượt chạy tự động để hứng bóng

Dũng:

- Tạo điểm số và cách thức kết thúc trò chơi.
- Viết báo cáo.
- Làm video.
- Tạo và lưu trữ bảng điểm.

Chung:

- Trình bày lưu đồ thuật toán.
- Chú thích.
- Khởi tạo gạch, tạo vị trí gạch, phương hướng di chuyển của bóng khi va chạm.
- Phát triển màn chơi và phần thưởng.

2. Các phương thức và thuộc tính:

+ Thuộc tính:

- Class CRacket:

- . float x, y // tọa độ góc dưới bên trái của vợt dạng khối chữ nhật
- . int rwidth; // độ lớn của vợt
- . int red, green, blue // chỉ số 3 màu chủ đạo để tổ hợp màu cho vợt

- Class CBall:

- . float pos_x, pos_y // tọa độ tâm của bóng dạng khối vuông
- . float go_x, go_y // vector (go_x, go_y) là vector hướng di chuyển bóng (độ lớn bằng 1)
- . int size // độ lớn của bóng
- . float speed // tốc độ bóng

- Class CScore:

- . int score // điểm người chơi

- Class CBonus:

- float x,y : vị trí của đồ thưởng.
- int type: loại đồ

```
// 1:L: racket_width x2
// 2:H: ball_speed /2
// 3:X2: score x2
// 4:PD: destroy a random column
// 5:PN: destroy a random row
// 6:S: racket_width /2
// 7:K: ball_speed x2
// 8:X0.5: score /2
```

- **CBXH:**

- **String** name: tên của người chơi
- **Int** score : điểm của người chơi

- **Function_Library:**

- **Const int** width : tạo độ rộng cho màn hình trò chơi : 1000 px;
- **Const int** height : tạo độ cao cho màn hình trò chơi : 600 px
- **Const** interval : độ trễ : 1000/1000
- **Define** 5 phím : S,P,O,H,C
- **Khởi tạo cơ bản và giới hạn cho vợt :**
 - Min width: 25
 - Max width : 400
 - Width :100
 - Height : 30
 - Speed : 15
- **Khởi tạo cơ bản ban đầu cho bóng**
 - Vị trí x,y của bóng (giữa màn hình; độ cao của vợt)
 - Hướng đi vector của bóng(0,1)(bóng đi thẳng lên trên)
 - Độ lớn của bóng: 10
 - Min speed :3
 - Max speed :15
 - Speed : 7

- **CBall(2):**

- **Int** x,y: vị trí mới của bóng theo tọa độ x,y
- **Int** OriginalX, originallyY: vị trí cũ của bóng theo tọa độ x,y
- **eDir**: chỉnh hướng bóng
 - 0: dừng bóng
 - 1: di chuyển trái
 - 2 : di chuyển trái trên
 - 3 : di chuyển trái dưới
 - 4 : di chuyển phải
 - 5 : di chuyển phải trên
 - 6 : di chuyển phải dưới

- **cPaddle:**

- **int** x,y : vị trí mới của vợt
- **int** OriginalX, originally: vị trí cũ của vợt

- **cGameManager:**

- **int** width,height: độ rộng, chiều cao của màn hình trò chơi

- `int` score1, score2: điểm cũ, điểm của quá trình 2 người chơi
- `char` up1, down1, up2, down2: điều chỉnh hướng di chuyển cho 2 người chơi
- `bool` quit: kiểm tra điều kiện kết thúc màn chơi
- `cBall`: tạo bóng:
- `cPaddle` player1, player2: tạo vợt cho 2 người chơi

- **Source:**

- `int` choose: lựa chọn giữa máy và người chơi
- `const int` hang, `const int` cot: số lượng brick tồn tại trong hang và cột
- `int` countvacham : số lần va chạm với gạch(sử dụng cho màn 4)
- `int` counvacham2 : số lần va chạm với vợt (sử dụng cho phần tính điểm)
- `vector<CBonus>` ListBonus: tạo mảng lưu trữ những bonus có trên màn hình

+ **Phương thức:**

- **Class CRacket:**

- . `float` GetPos_x(); // trả về x
- . `float` GetPos_y(); // trả về y
- . `int` getwidth(); //trả về width
- . `int` setwidth(int); //đặt lại giá trị cho độ lớn của vợt
- . `void` InitBottom(); // khởi tạo các giá trị thuộc tính ban đầu cho vợt dưới
- . `void` RandomColor(); // tạo tổ hợp màu ngẫu nhiên cho vợt (trừ màu đen)
- . `void` Show(); // hiển thị vợt ra cửa sổ game
- . `void` Move(int); // di chuyển vợt trái phải với giá trị vào là 0/1

- **Class CBall:**

- . `float` GetPos_x(); // trả về pos_x
- . `float` GetPos_y(); // trả về pos_y
- . `float` GetDir_x(); // trả về go_x
- . `float` GetDir_y(); // trả về go_y
- . `float` GetSpeed(); // trả về speed
- . `void` SetPosition(float, float); // thay đổi vị trí bóng
- . `void` SetDirection(float, float); // thay đổi hướng bay của bóng
- . `void` SetSpeed(float); // thay đổi tốc độ bóng
- . `void` Init(); // Khởi tạo giá trị thuộc tính ban đầu cho bóng
- . `void` Move(); // thực hiện di chuyển bóng
- . `void` Show(); // hiển thị bóng trên cửa sổ game

- **Class CScore:**

- . `int` GetScore(); //trả về score_right
- . `void` Init(); // khởi tạo tỉ số ban đầu (0)
- . `void` Goal(int); // tăng điểm cho người chơi khi phá gạch
- . `void` Show(); // hiển thị tỉ số trên cửa sổ game

- **Class CBonus:**

```
. float getpos_x();           // trả về tọa độ x
. float getpos_y();           // trả về tọa độ y
. float setpos_y(float);       // trả về tọa độ y
. int getType();               // trả về loại bonus
. void Init(float x, float y); // tạo vị trí random cho vật phẩm
. void move();                 // di chuyển vật phẩm
. void show();                 // vẽ lên màn hình
```

- **Class CBXH:**

```
. void Input();                // nhập vào bảng xếp hạng từ file
. void Output();               // viết lại file bảng xếp hạng mới
. string getname();            // lấy tên để đưa vào bảng xếp hạng
. int getScore();              // lấy điểm
. void setname(string);        // đặt tên mới
```

- **Class cBall(2):**

- cBall(int, int) // khởi tạo vị trí cho bóng
- void reset() // trả về vị trí cho bóng
- void changeDirection(eDir) // thay đổi hướng đi của bóng
- void randomDirection() // tạo hướng đi ngẫu nhiên cho bóng
- int getX(); // lấy tọa độ x của bóng
- int getY() // lấy tọa độ y của bóng
- eDir getDirection() // lấy hướng đi của bóng
- void Move() // di chuyển bóng
- cBall(); // khởi tạo bóng
- ~cBall(); // phá hủy bóng
- friend ostream& operator<<(ostream&, cBall);

- **Class CGameManager:**

- cGameManager(int, int) // khởi tạo với độ rộng màn hình console
- ~cGameManager() // xóa 2 vệt và bóng
- Void Draw() // xóa màn hình và vẽ màn hình mới
- Void Logic() // xử lý va chạm
- Void Input() // khởi tạo đưa vào vị trí bóng và vệt
- Void Run() // thực hiện chương trình
- Int Flip() // khởi tạo random cho hướng di chuyển bắt đầu

- **cPaddle:**

- cPaddle() // khởi tạo vệt
- cPaddle(int, int) // tạo vệt với vị trí chính xác tọa độ x y
- void reset() // khởi tạo lại khi ghi điểm

- int getX() // lấy tọa độ x của vợt
- int getY() // lấy tọa độ y của vợt
- void moveUp() // di chuyển lên
- void MoveDown() // di chuyển xuống
- friend ostream& operator<<(ostream&, cPaddle)

- **Chung:**

```
. void vector2d_norm(float& x, float& y); // chuẩn hóa vector (x,y) về độ lớn bằng 1
. std::string int2str(int x); // chuyển số nguyên x thành kí tự
. void drawText(float x, float y, std::string text); //hiển thị chuỗi kí tự text trên cửa sổ game tại tọa độ (x,y)
. void Init(); // khởi tạo mặc định các biến vợt, bóng, điểm số
. void draw(); // hiển thị các đối tượng trên cửa sổ game
. void keyboard(int k); // điều khiển vợt trong game
. void updateBall(); // xử lý các tình huống bóng như di chuyển, chạm vợt, chạm tường,...
. void update(int value); // sự thay đổi các biến, khung hình theo thời gian
. void enable2D(int width, int height); // cài đặt chế độ 2D của OpenGL
. void delay (int)
. void randomLevel4() //tạo vị trí random cho 5 viên gạch ở màn 4
. void InitBrick() //khởi tạo gạch cho các màn
. int checkEndGame() // kiểm tra kết thúc game
. int checkVacham(Cball,int,int) //kiểm tra va chạm với viên được đưa vào hàm
. void Init() //khởi tạo game
. void updateBXH() //cập nhật lại bảng xếp hạng sau khi người chơi điểm cao
. void draw() //vẽ màn hình game và chạy qua mỗi bước
. void keyboard() //kiểm tra phím được nhập để thực hiện phương thức
. void updateBonus() //cập nhật màn hình game với bonus được đưa vào
. void updateBall() //cập nhật vị trí bóng
. void update(int) //thực hiện các hàm qua các bước nhập, vẽ hình.
. void enable2D(int width,int height) //mở màn hình game và khởi tạo các phương thức của thư viện openGL
```

III. Sơ đồ lớp :

cBall(2)
<pre>int x, y; int originalX, originalY; eDir direction;</pre>
<pre>cBall(int, int); void Reset(); void changeDirection(eDir); void randomDirection(); int getX(); int getY(); eDir getDirection(); void Move(); cBall(); ~cBall(); friend ostream& operator <<(ostream&, cBall);</pre>



cBall
<pre>float pos_x; float pos_y; float go_x; float go_y; int size; float speed;</pre>
<pre>float GetPos_x(); float GetPos_y(); float GetDir_x(); float GetDir_y(); float GetSpeed(); void SetPosition(float, float); void SetDirection(float, float); void SetSpeed(float); void Init(); void Move(); void Show();</pre>

cPaddle
<pre>int x, y; int originalX, originalY;</pre>
<pre>cPaddle(); cPaddle(int , int); void Reset(); int getX(); int getY(); void moveUp(); void moveDown(); friend ostream & operator << (ostream&, cPaddle);</pre>



CRacket
<pre>float x; float y; int red; int green; int blue;</pre>
<pre>float GetPos_x(); float GetPos_y(); int GetWidth(); void SetWidth(int); void InitBottom(); void RandomColor(); void Show(); void Move(int);</pre>

cGameManager
<pre> int width, height; int score1, score2; char up1, down1, up2, down2; bool quit; cBall* ball; cPaddle* player1; cPaddle* player2; </pre>
<pre> cGameManager(int, int); ~cGameManager(); void Draw(); void Logic(); void Input(); void Run(); int flip(); </pre>

CBonus
<pre> float x; float y; int type; </pre>
<pre> float GetPos_x(); float GetPos_y(); void SetPos_y(float); int GetType(); void Init(float x, float y); void move(); void show(); </pre>

CBXH
<pre> string name; int score; </pre>
<pre> void input(); void output(); string getName(); int getScore(); void setName(string); void setScore(int); </pre>

CScore
<pre> int score; </pre>
<pre> int getScore(); void Init(long long); void Goal(float); void Show(); </pre>

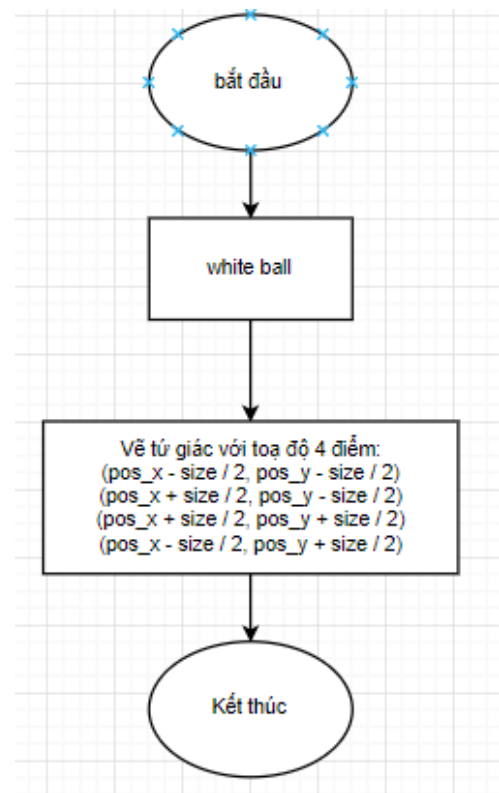
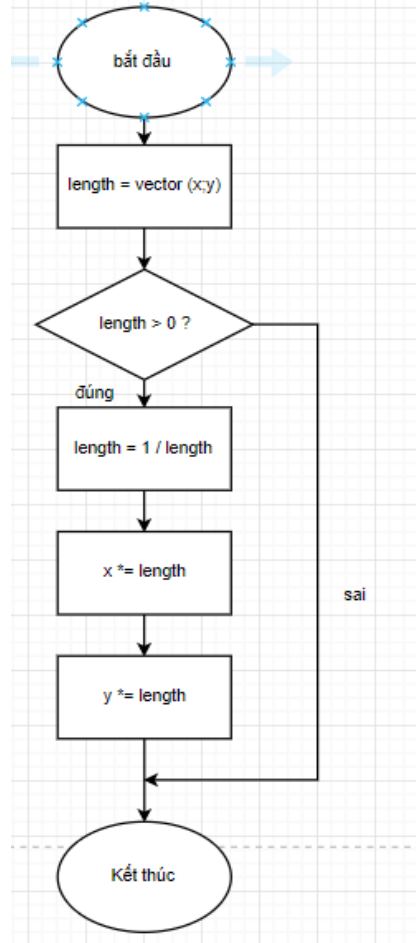
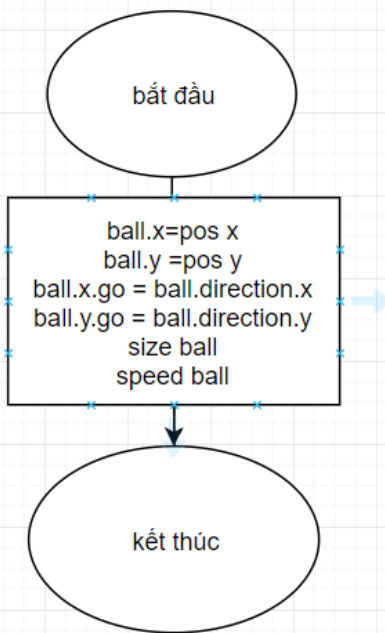
IV. LƯU ĐỒ THUẬT TOÁN

○ CBall

init : khởi tạo

vector2d_norm

show

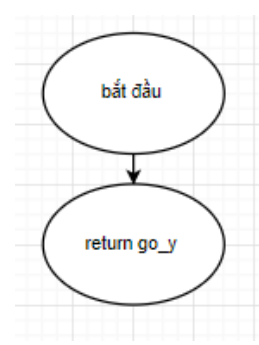
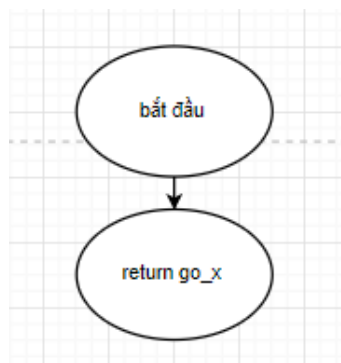
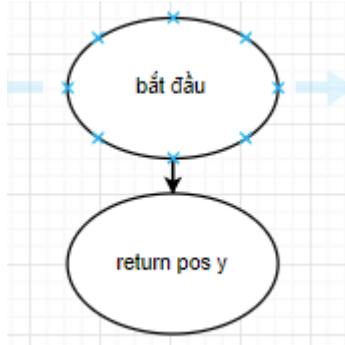
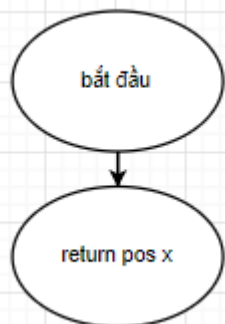


Set pos.x

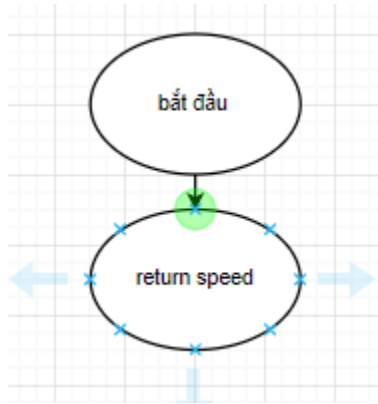
set pos.y

get pos.x

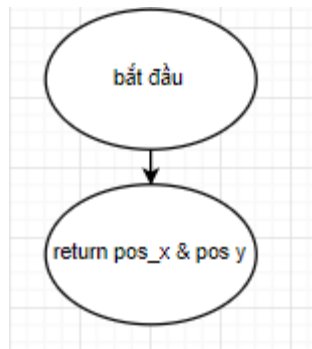
get pos.y



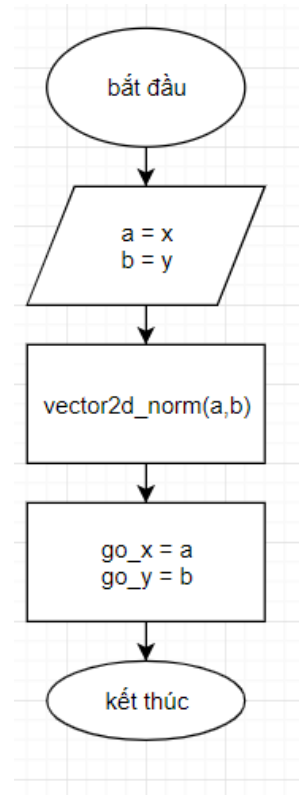
Get/set speed



set position

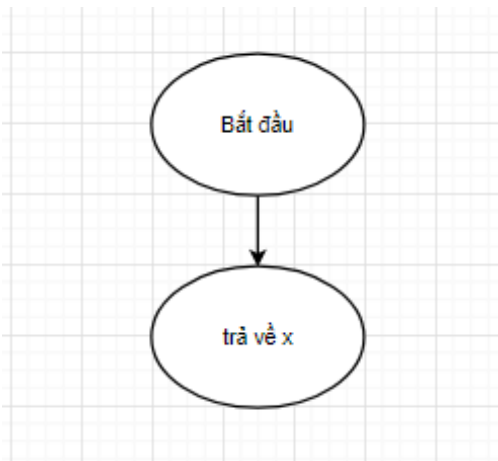


set direction



○ CRacket

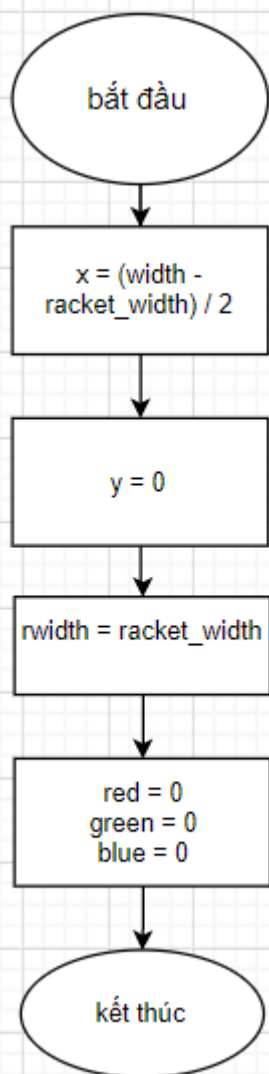
get pos.x



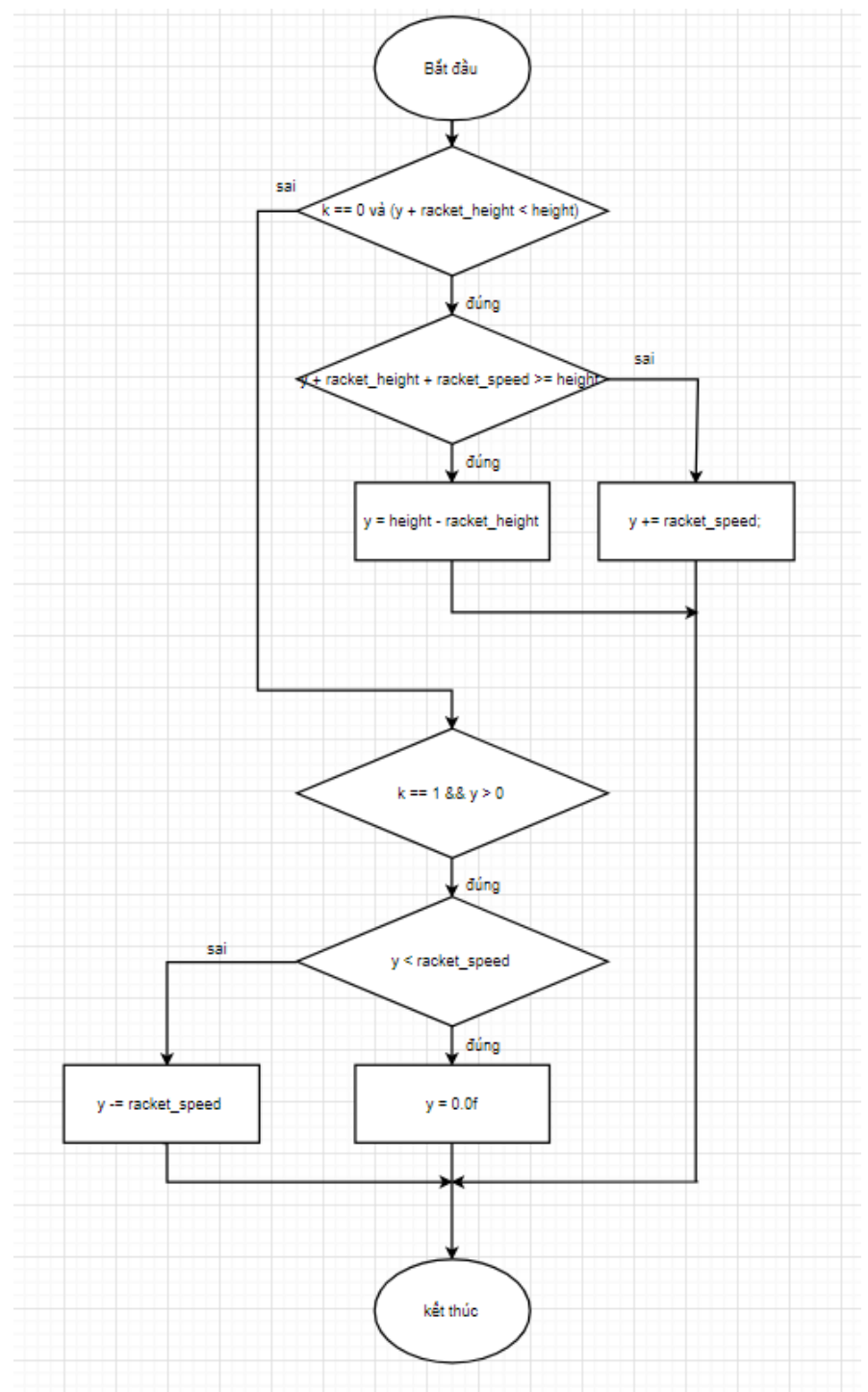
get pos.y



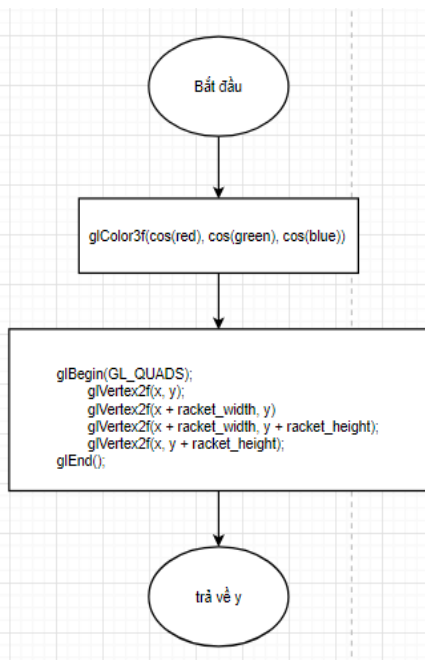
init_bottom



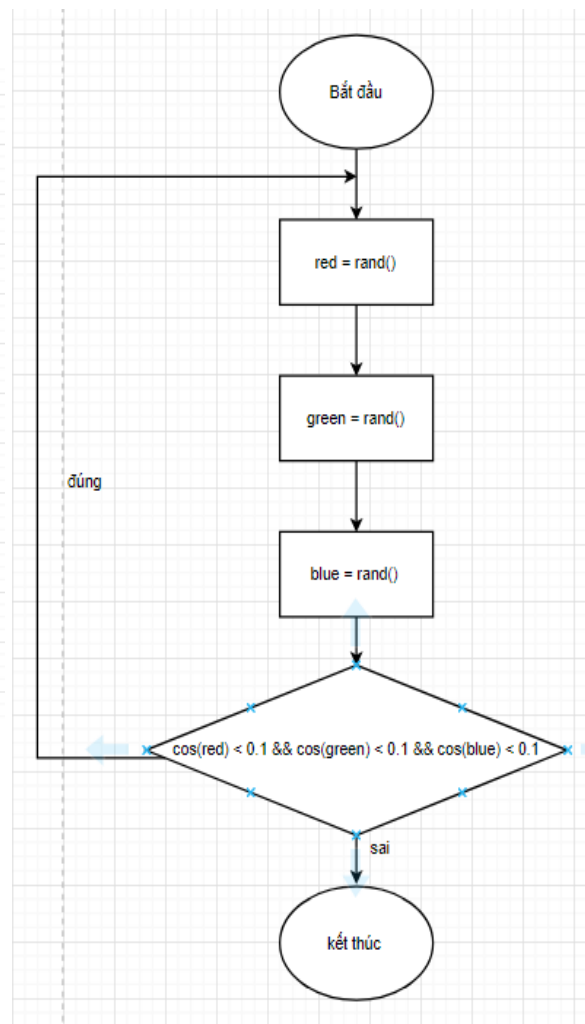
random color



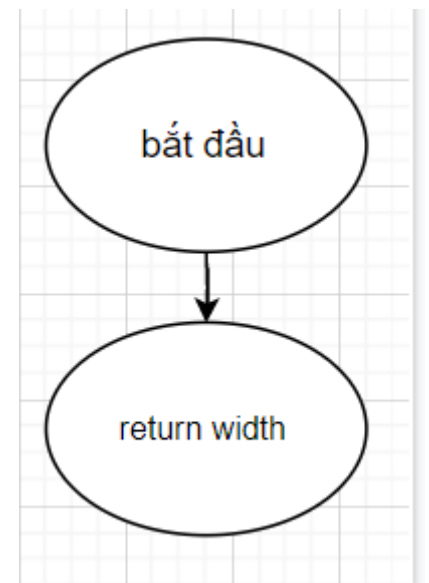
show



move

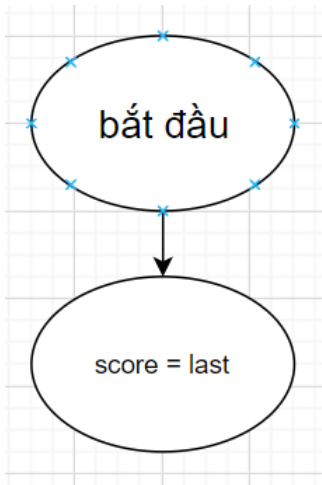


get/set width

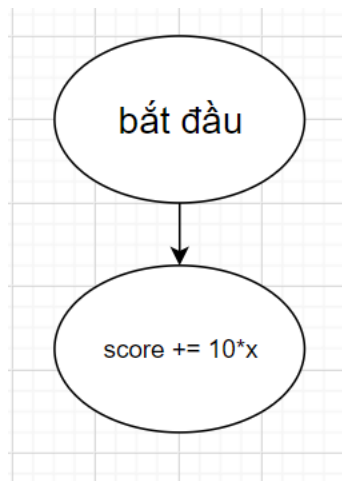


○ **CScore:**

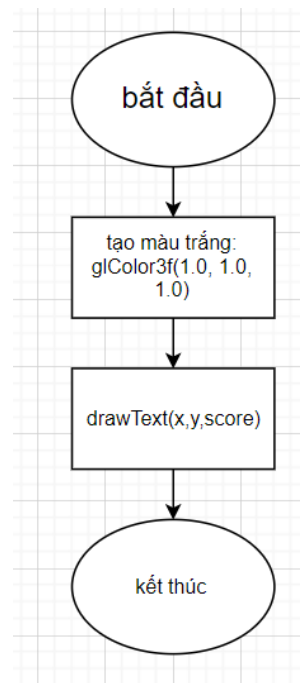
Init:



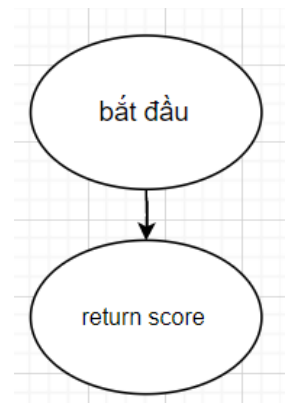
goal:



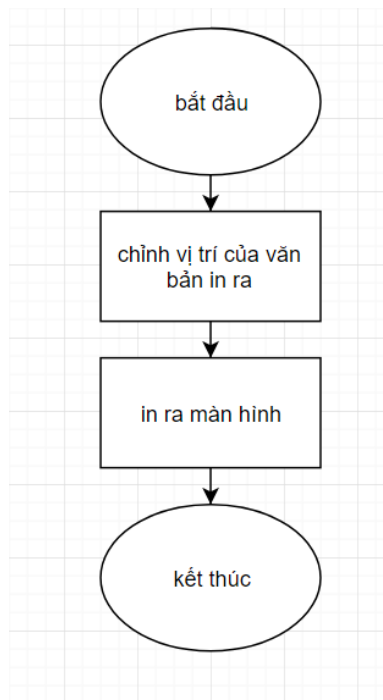
show:



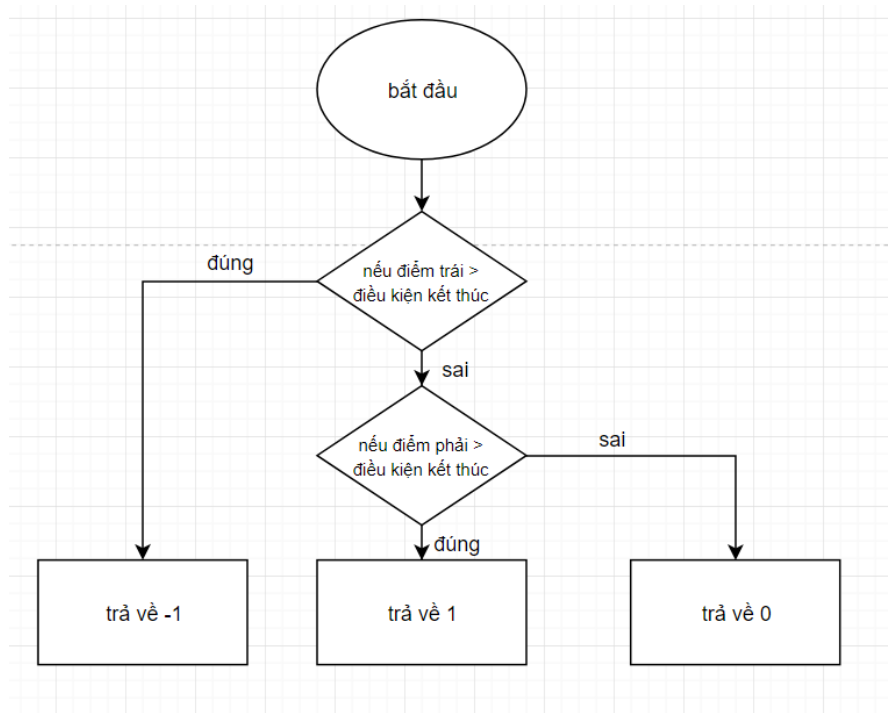
get score:



drawtext:

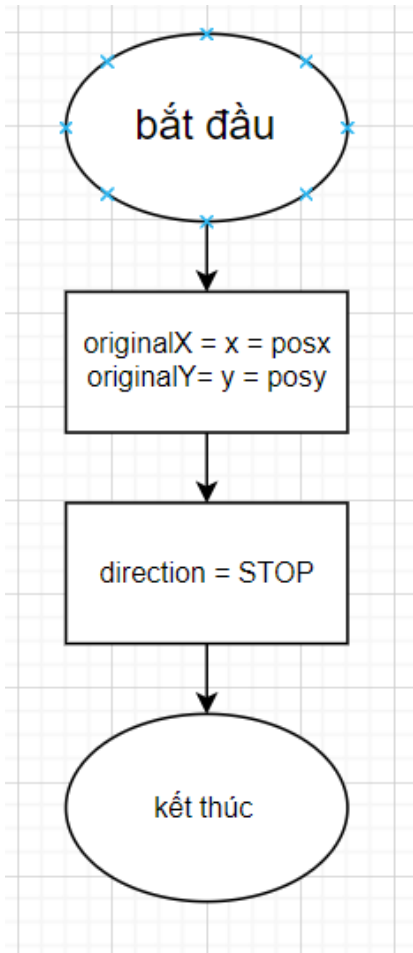


Isendgame:

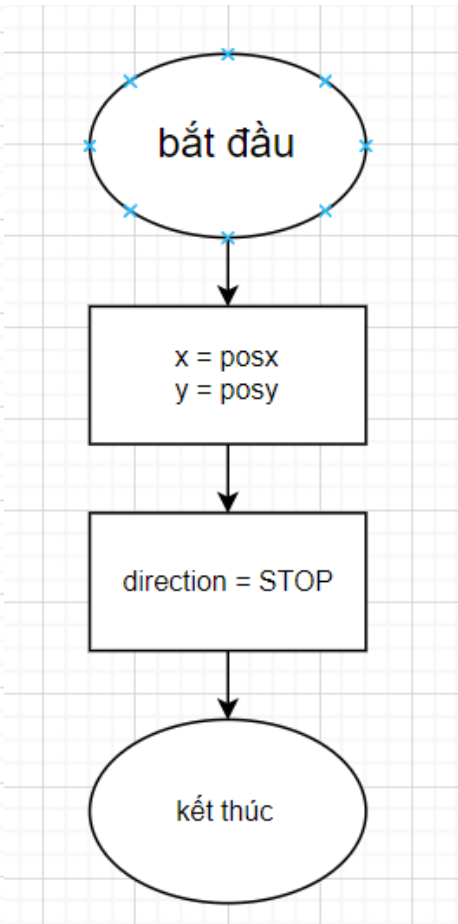


○ **cBall:**

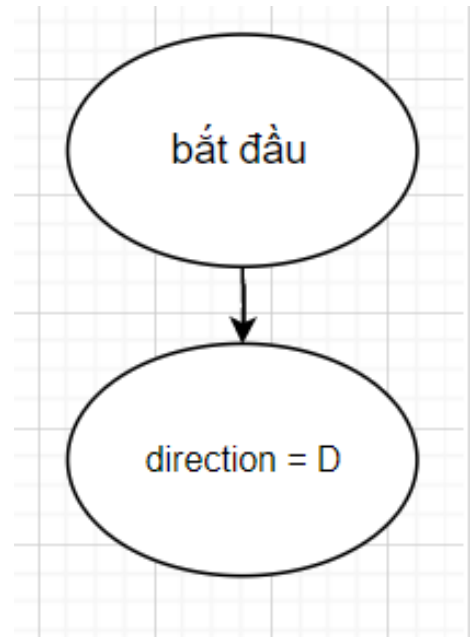
cBall



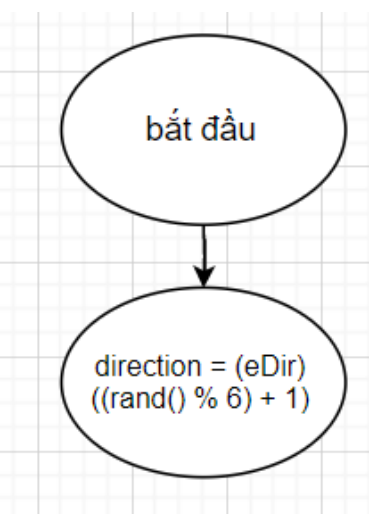
Reset:



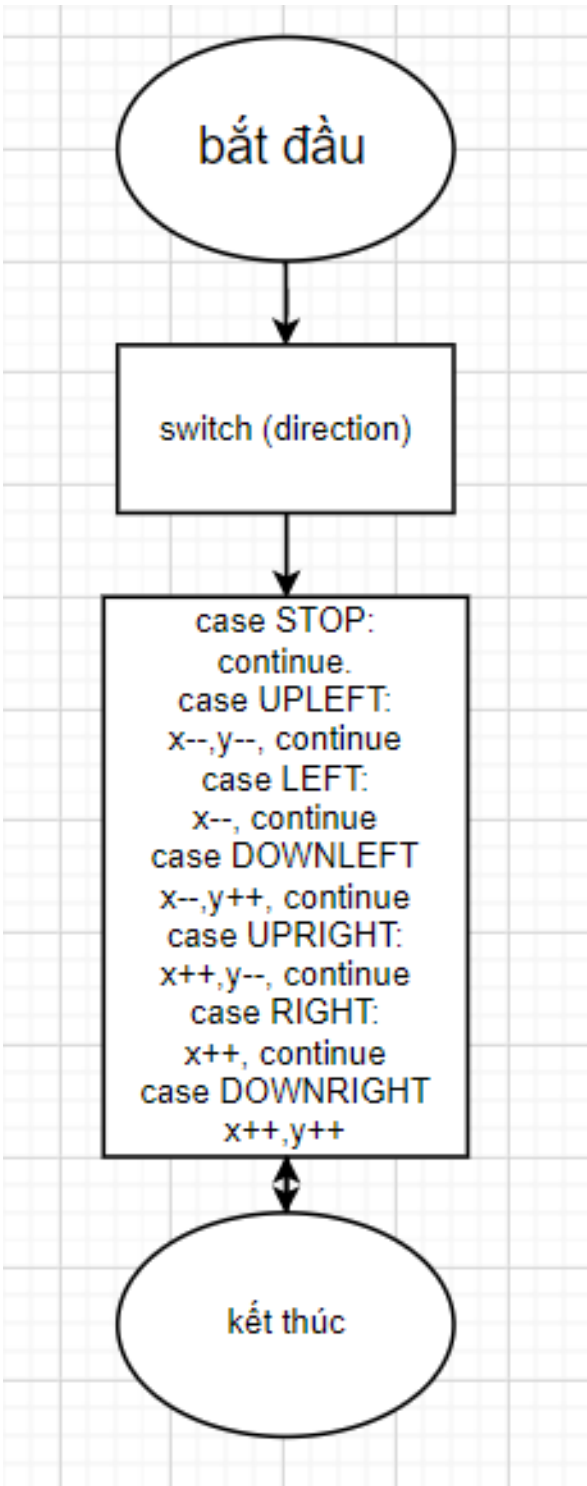
changeDirection:



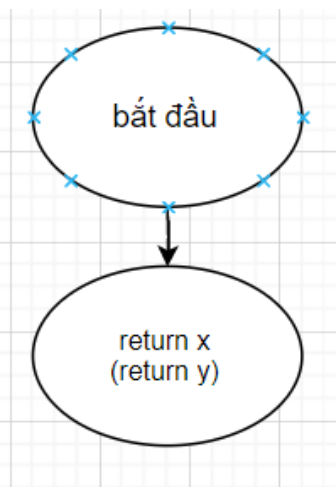
RandomDirection



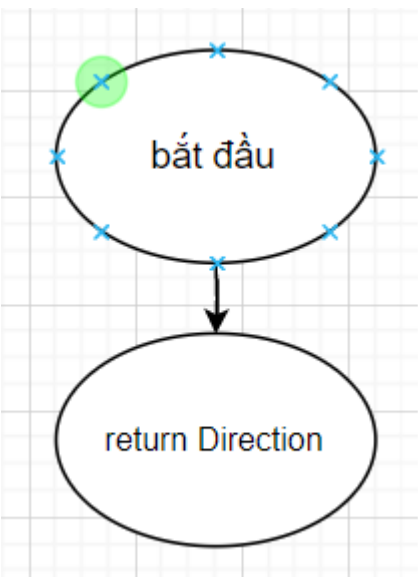
Move



Get x(get y)

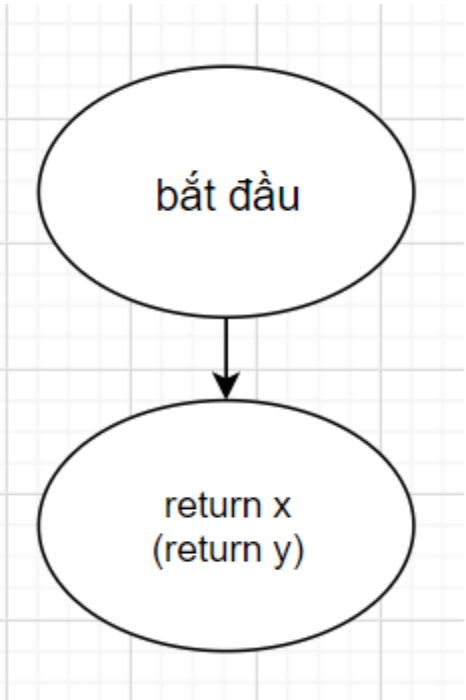


getDirection

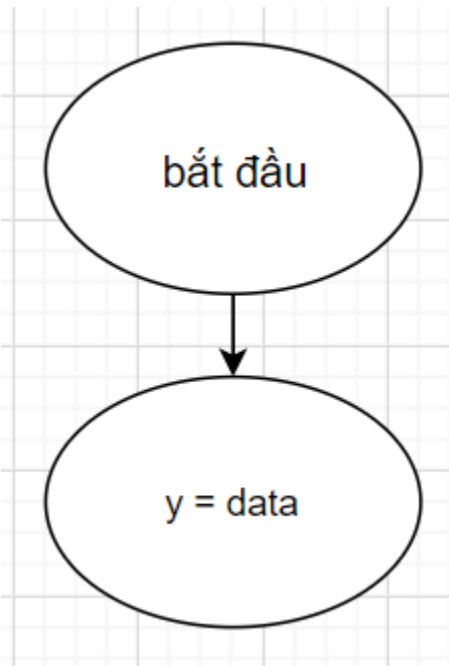


CBonus:

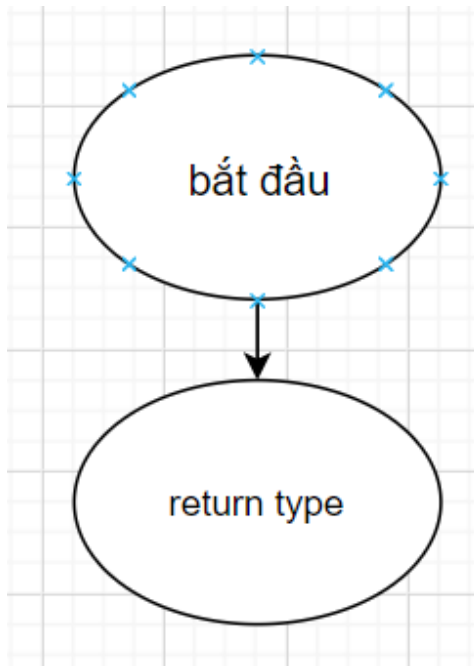
GetPos_x(GetPos_y)



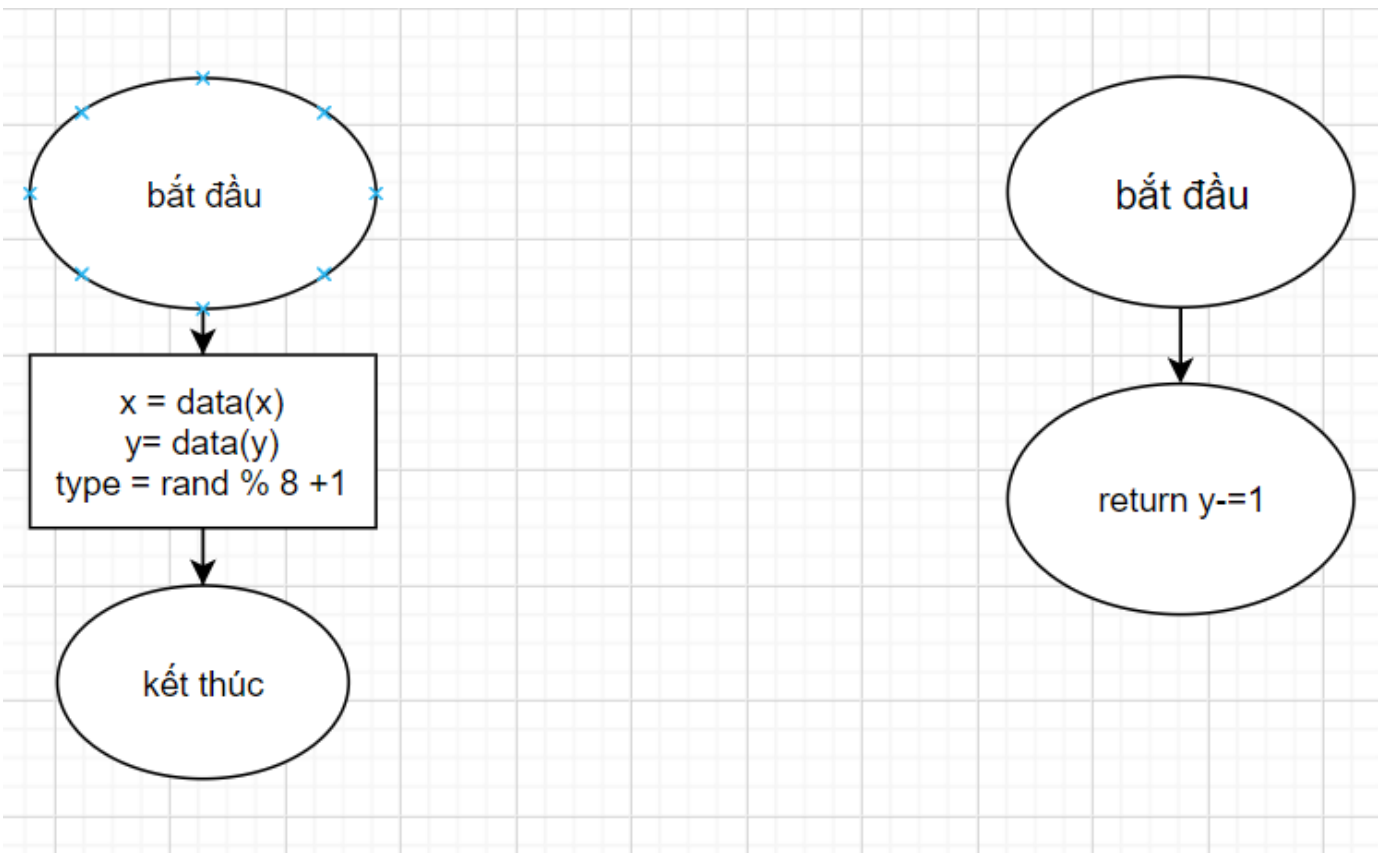
SetPos_y



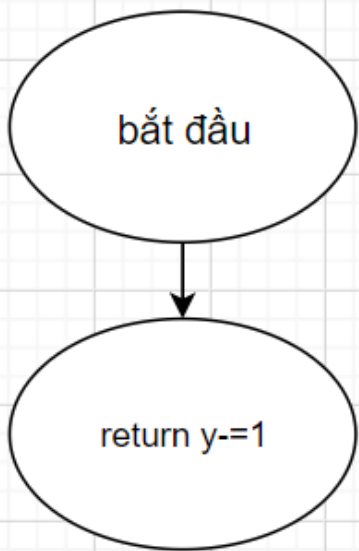
getType



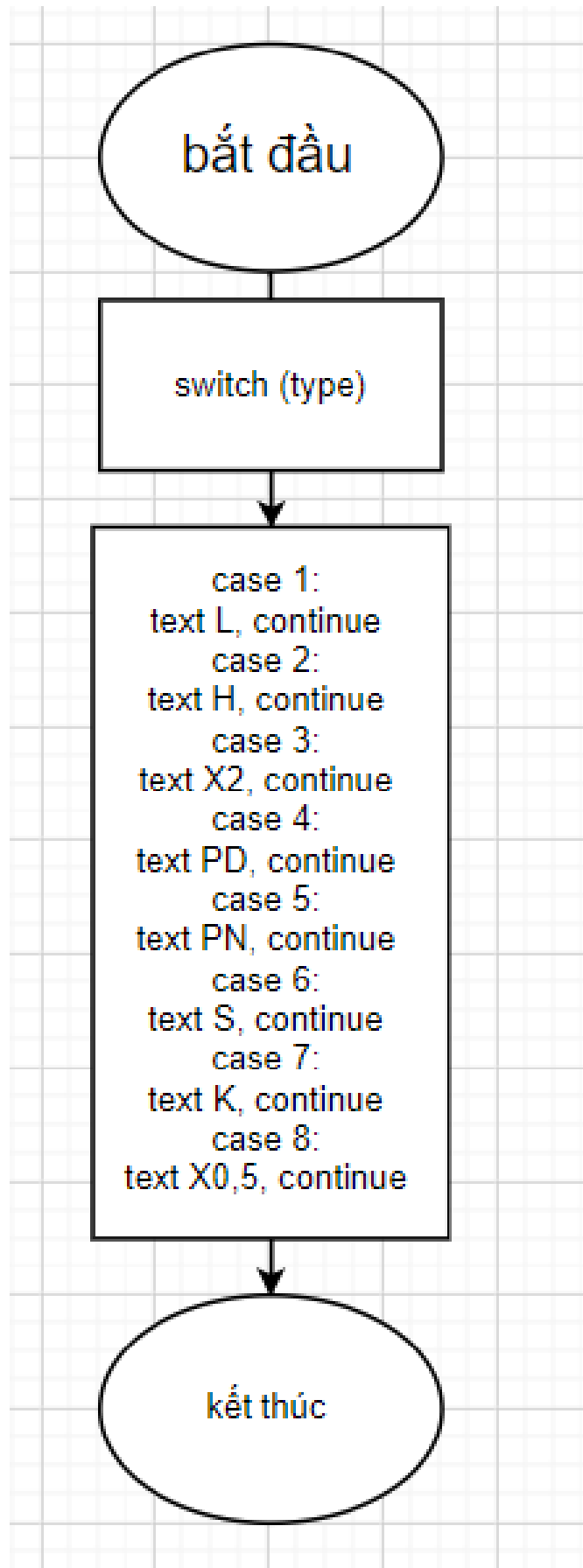
Init



Move

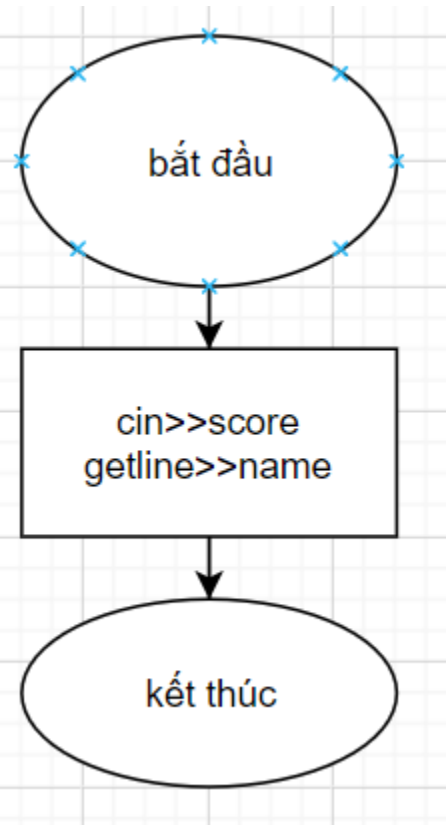


Show

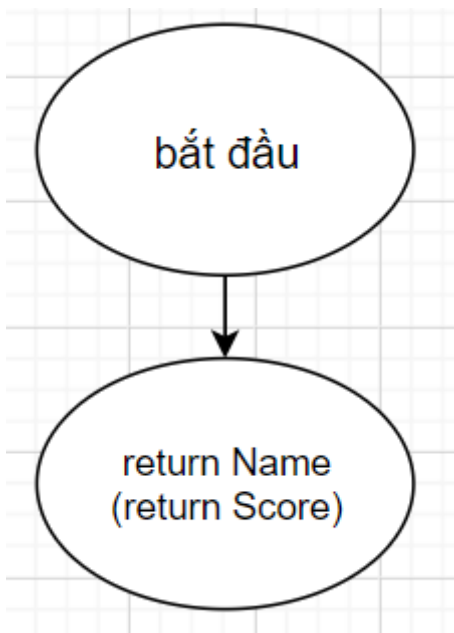


□ CBXH:

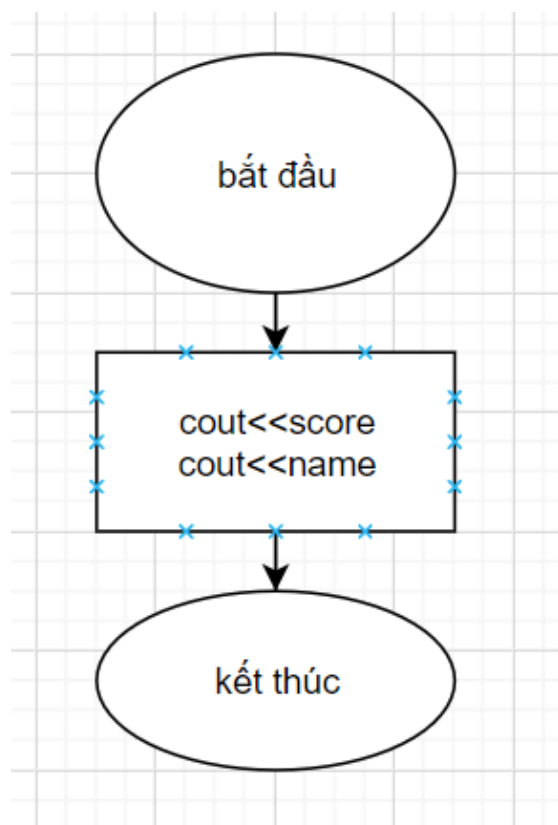
Input:



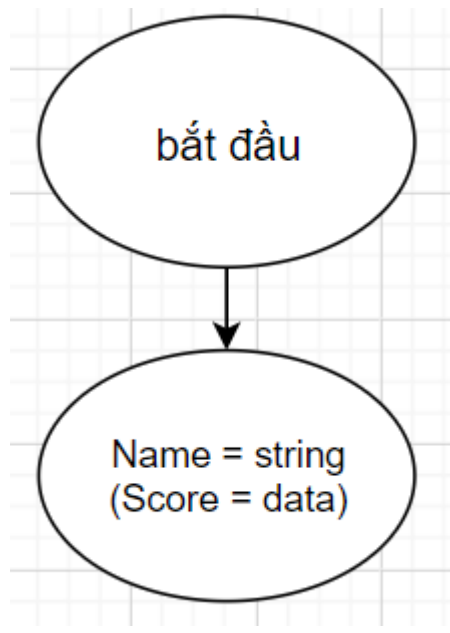
getName(getScore)



Output

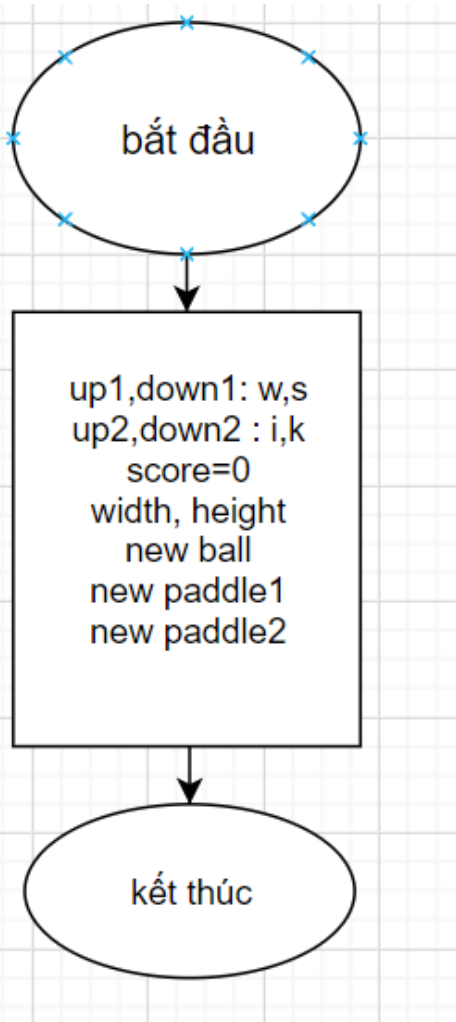


SetName(setScore)

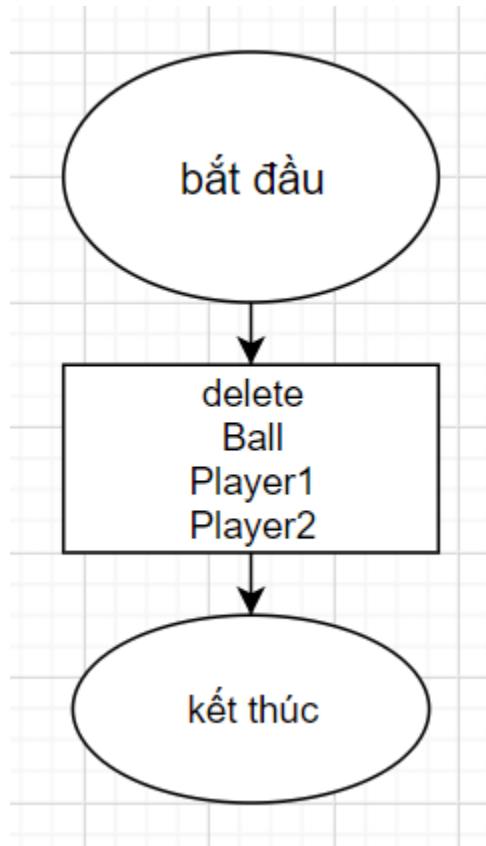


- **cGameManager:**

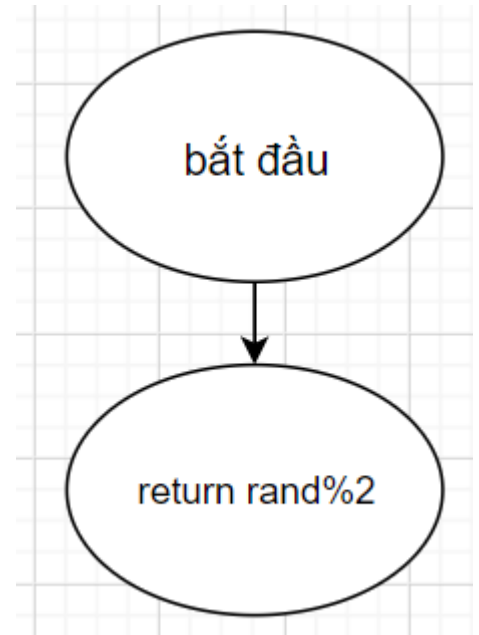
cGameManager:



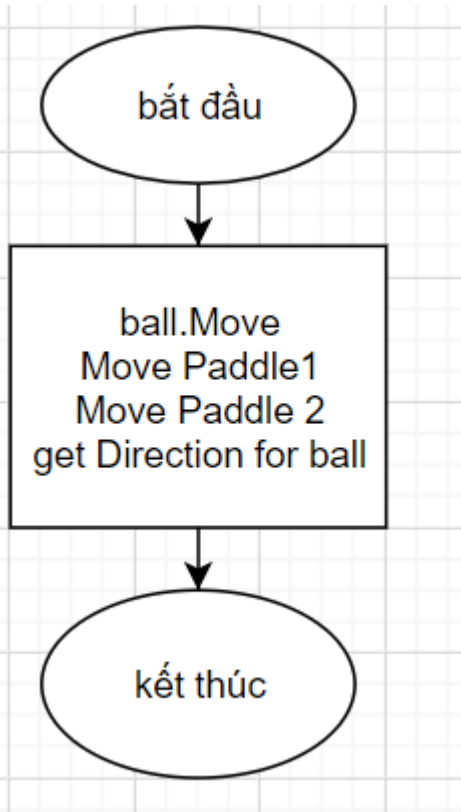
~cGameManager



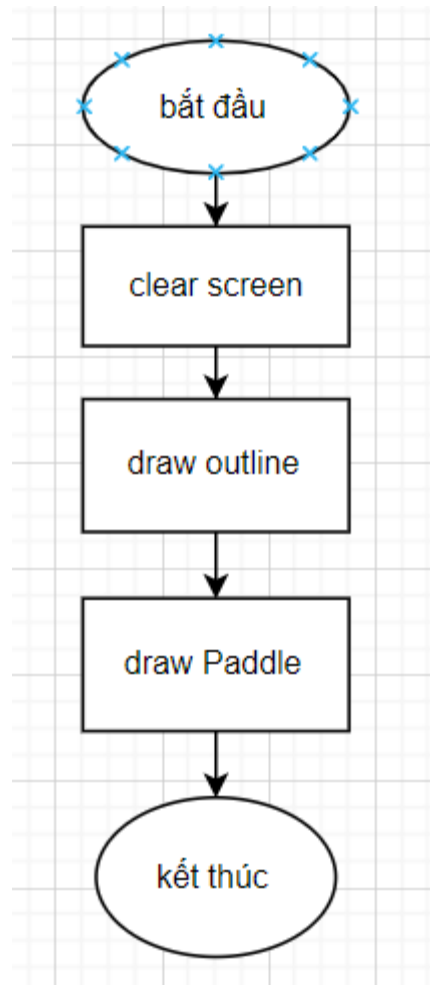
flip



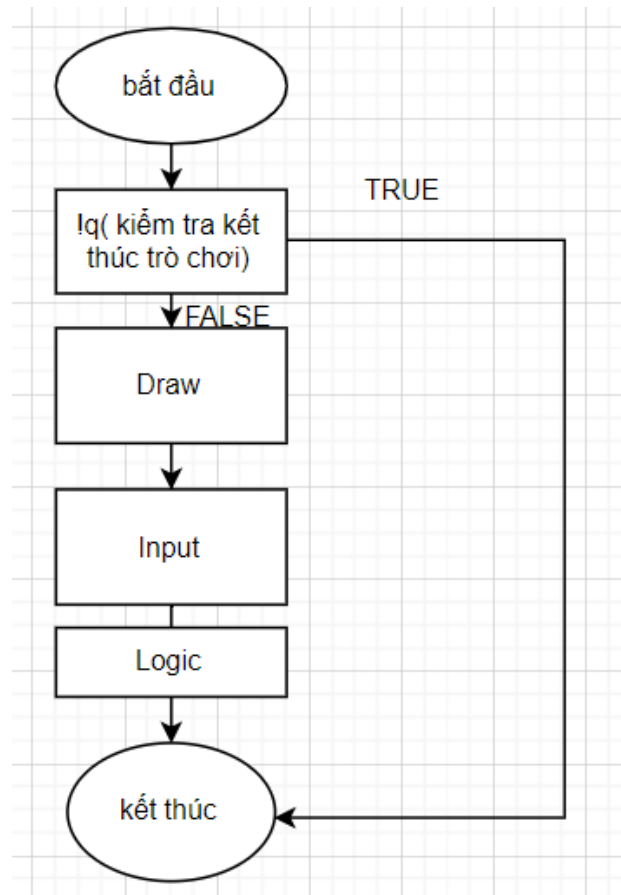
Input:



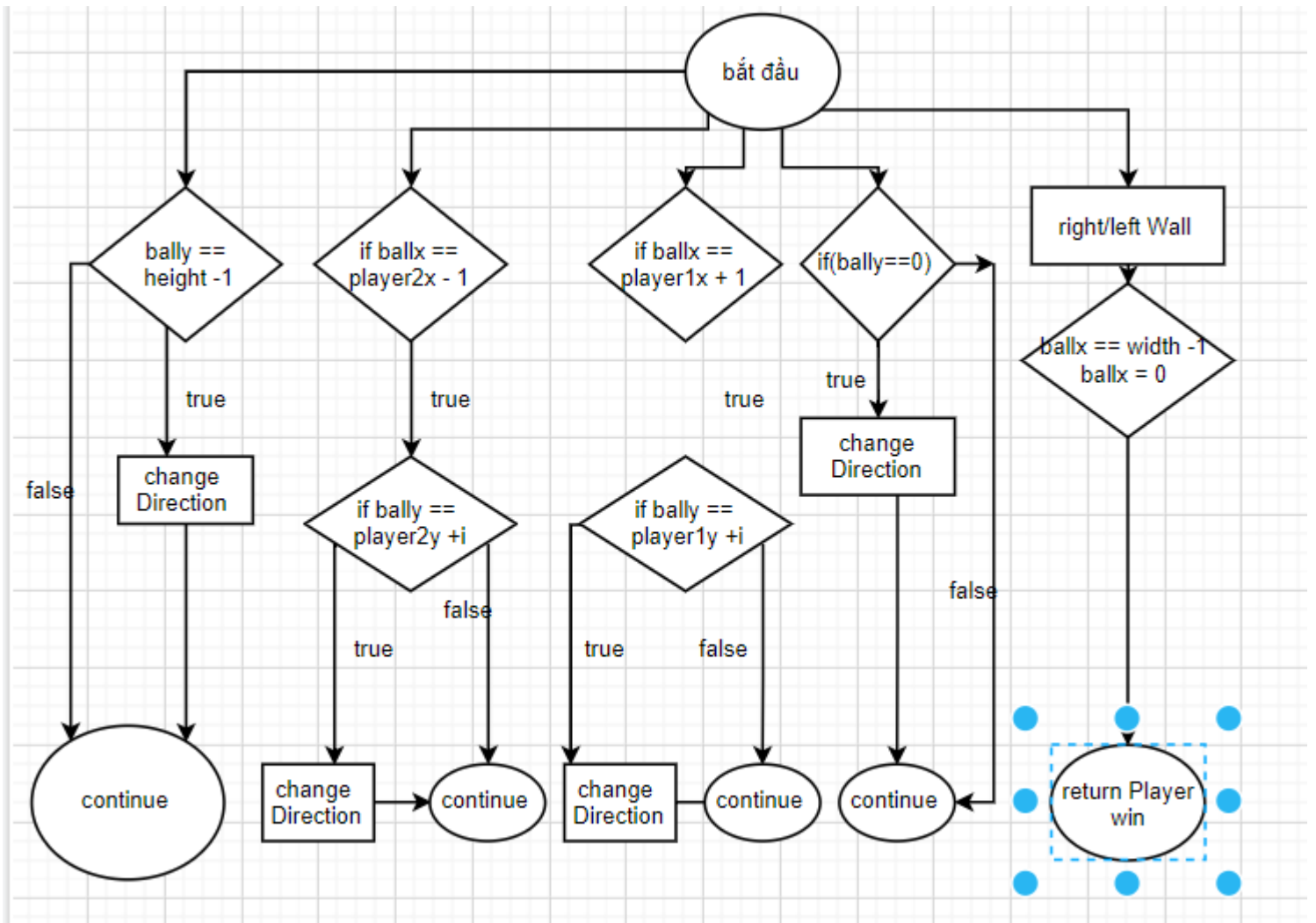
draw



Run

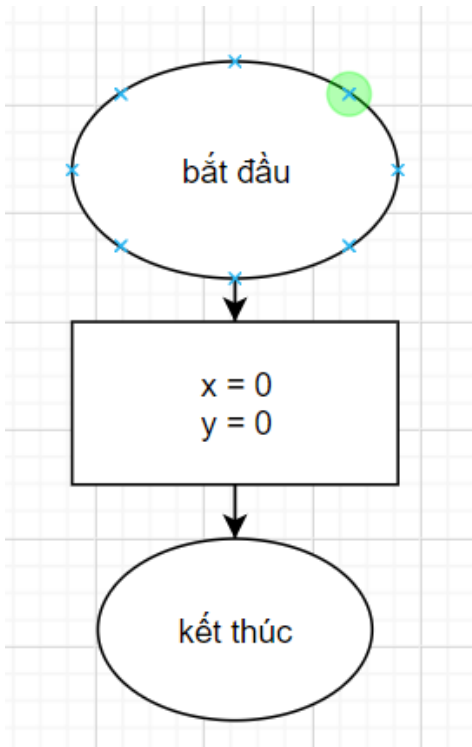


Logic

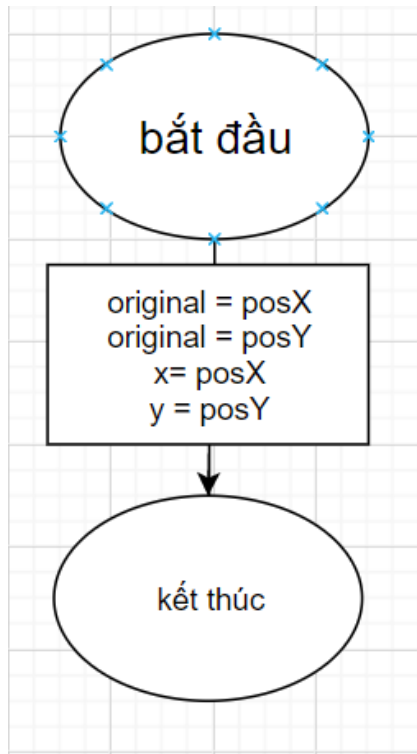


- **cPaddle:**

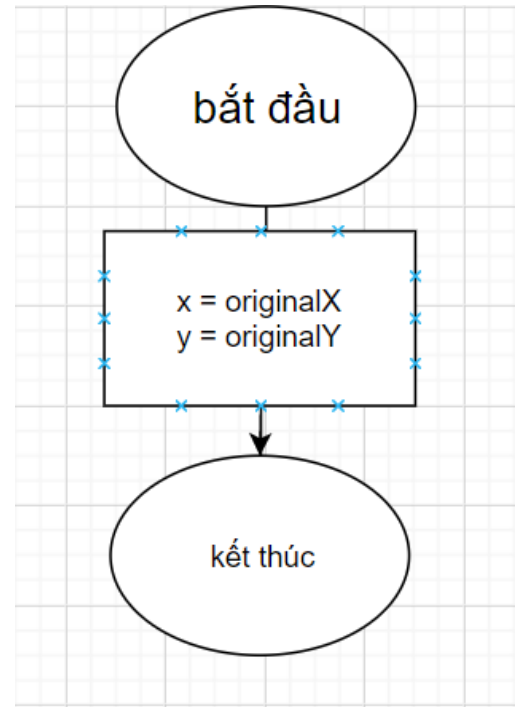
cPaddle:



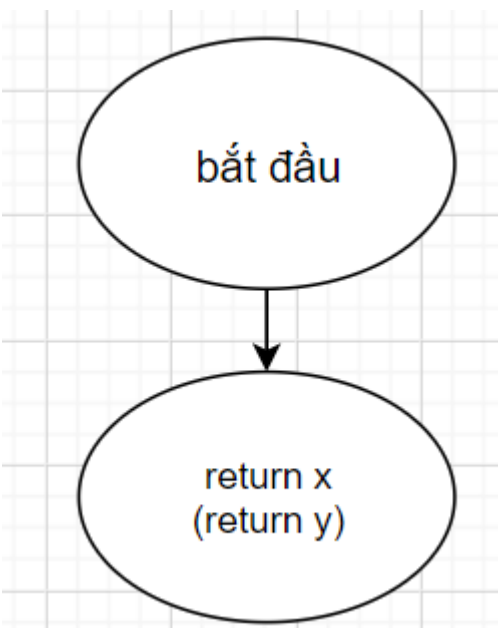
cPaddle:



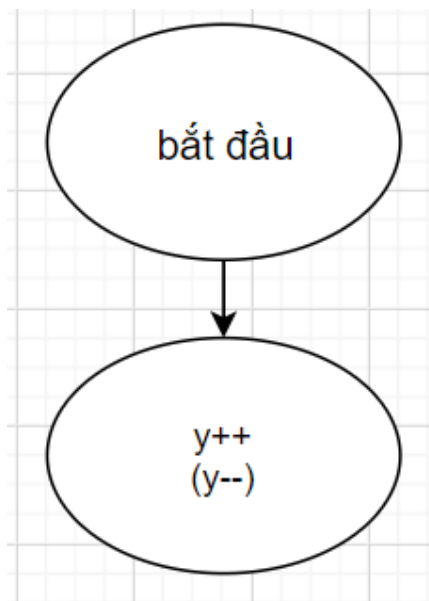
Reset:



Getx (Gety)

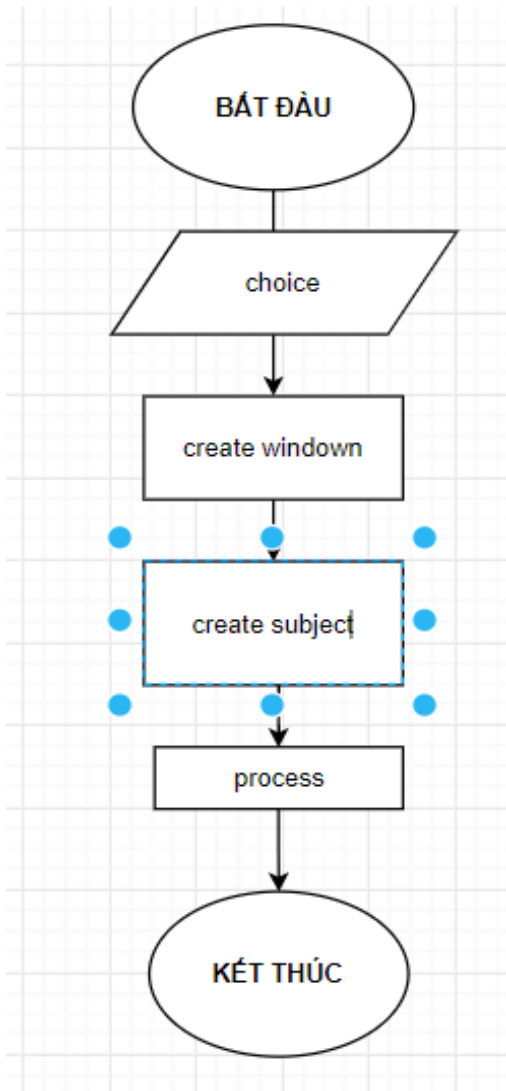


Moveup(MoveDown)

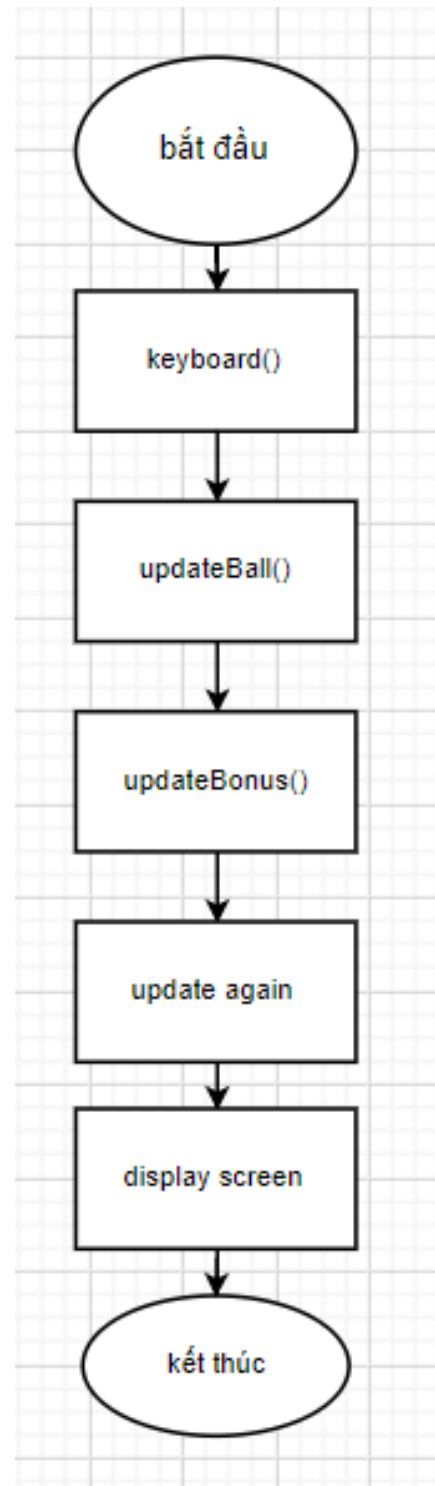


○ **source:**

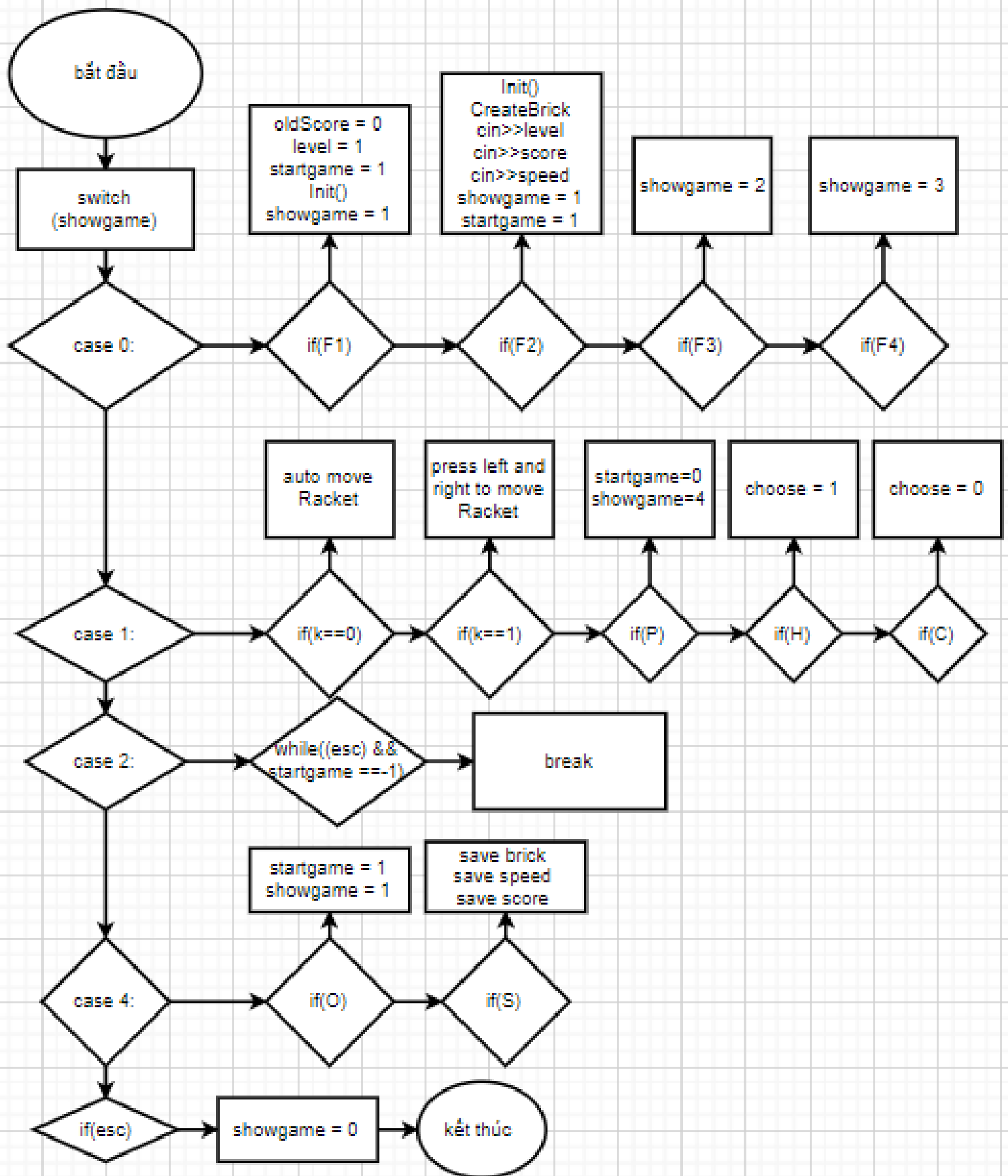
main:



update:



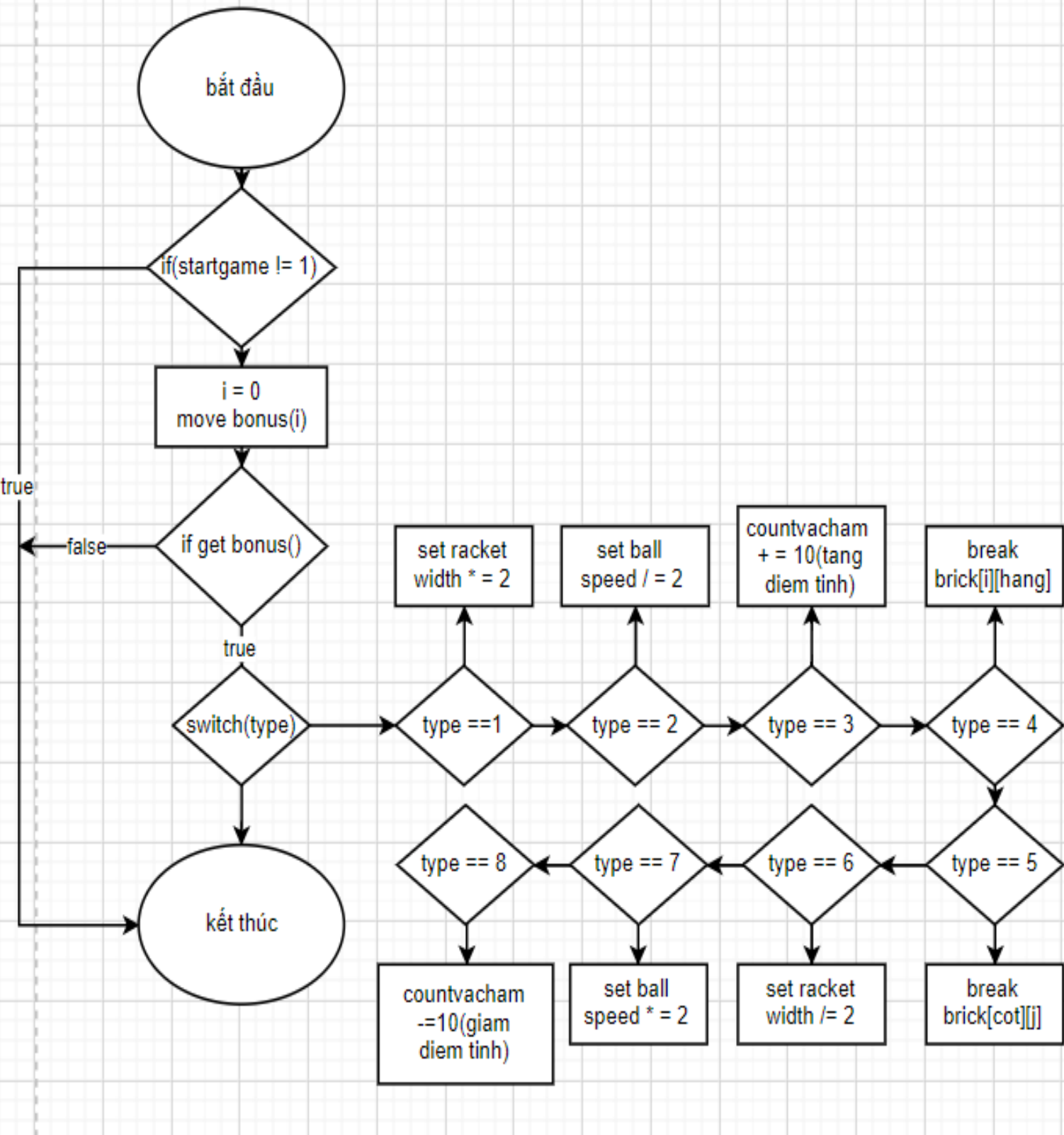
keyboard:



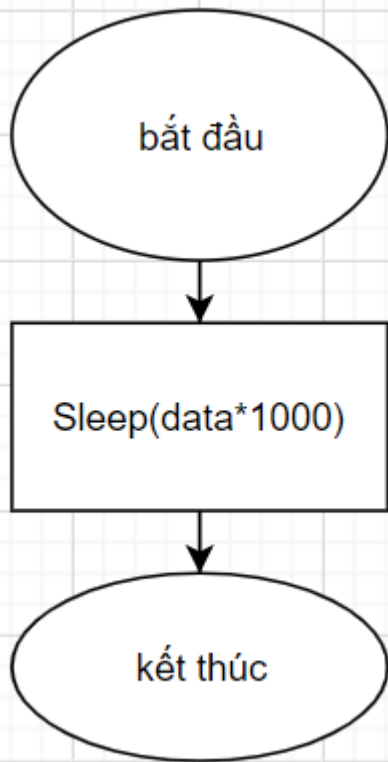
UpdateBall



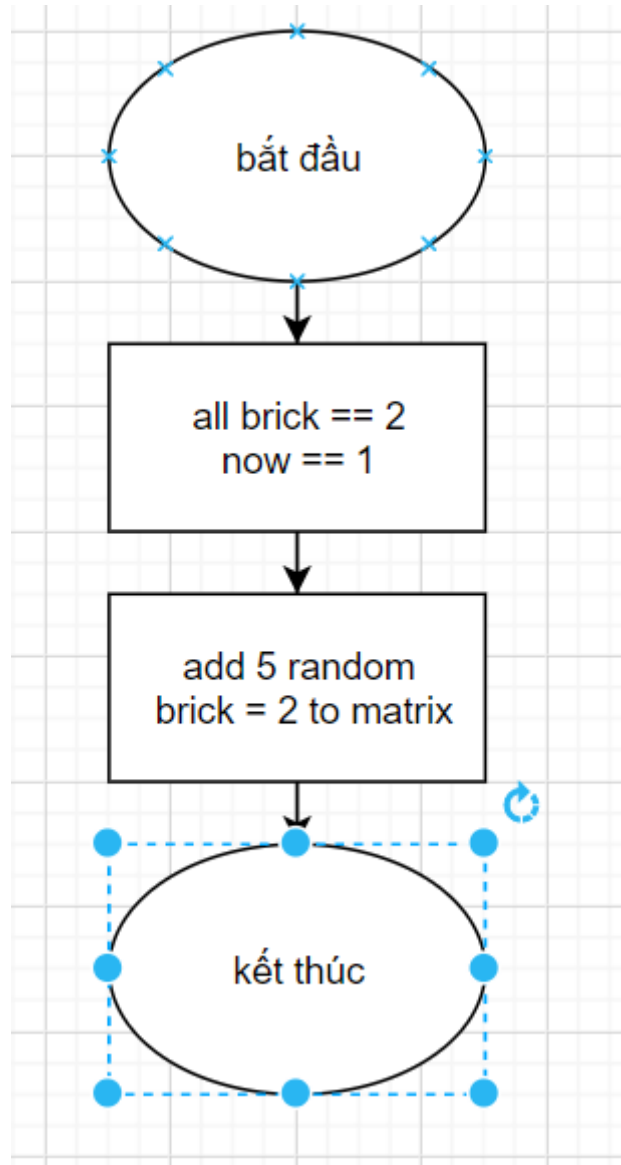
UpdateBonus:



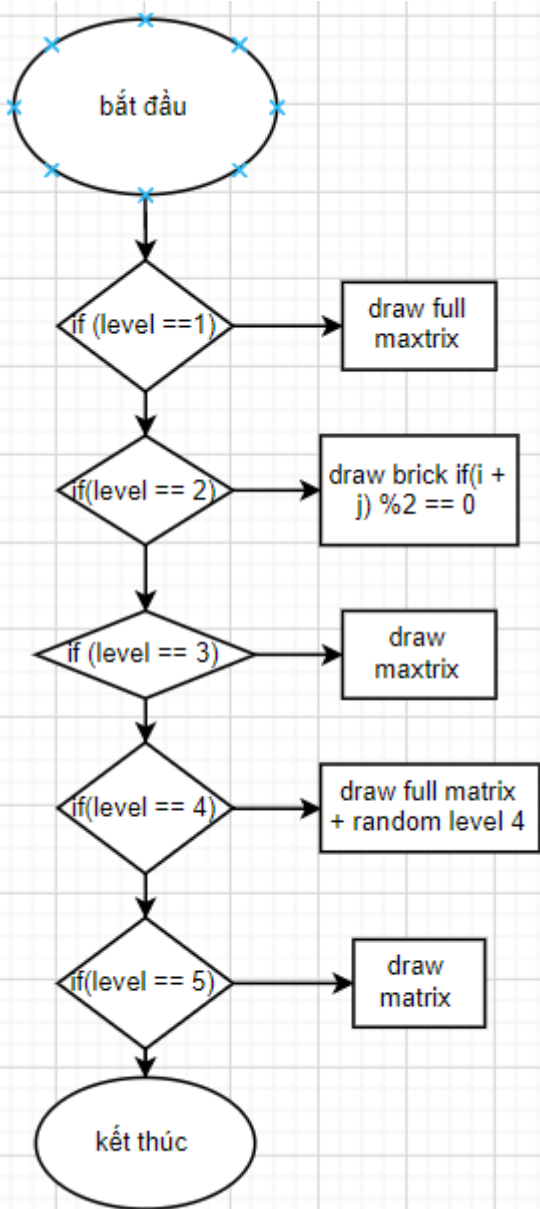
Delay:



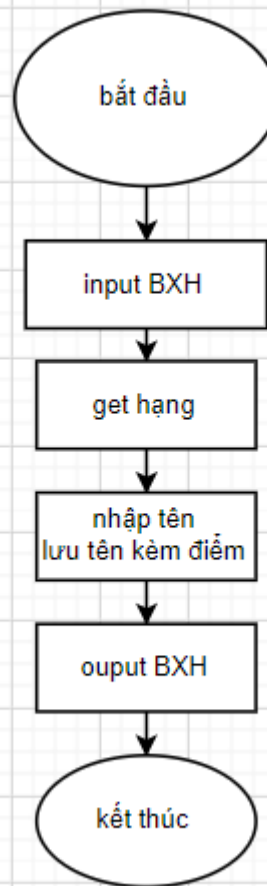
RandomLVL 4



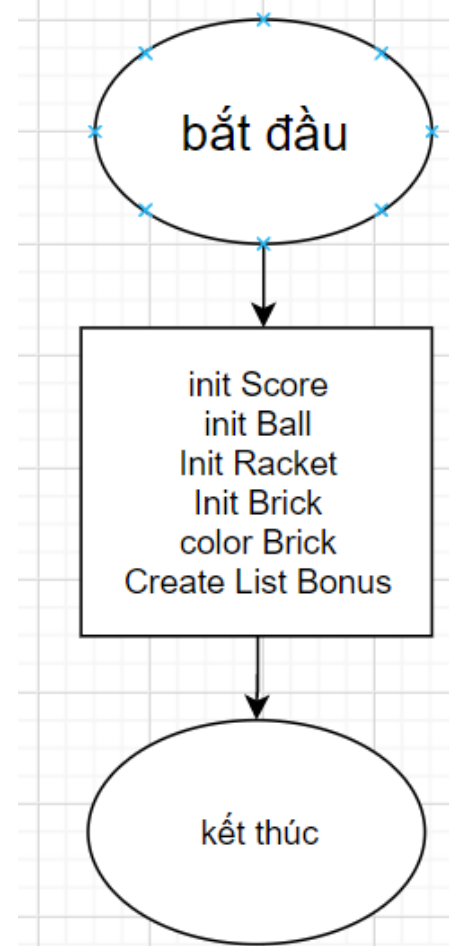
InitBrick



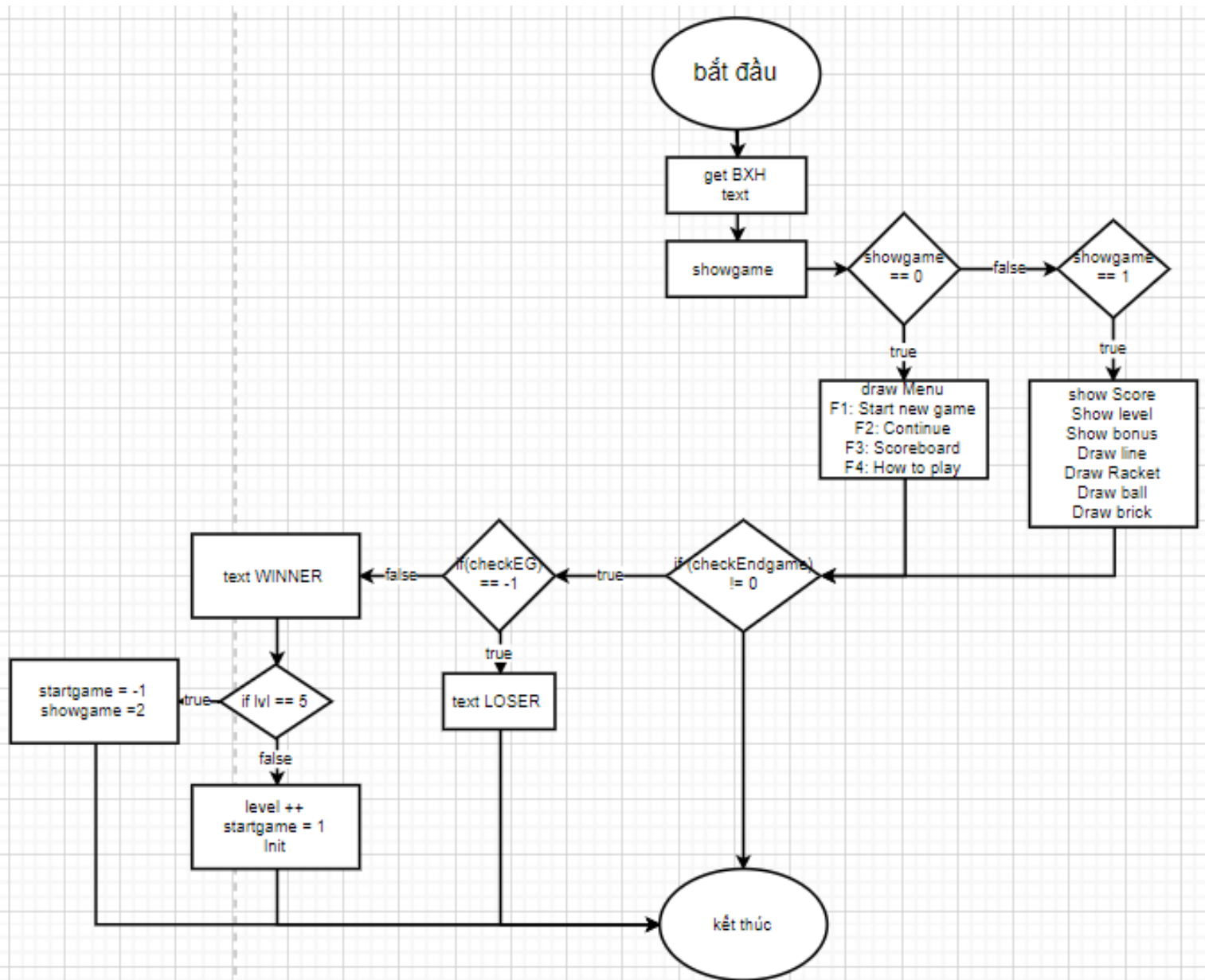
updateBXH



Init



draw



CheckVaCham

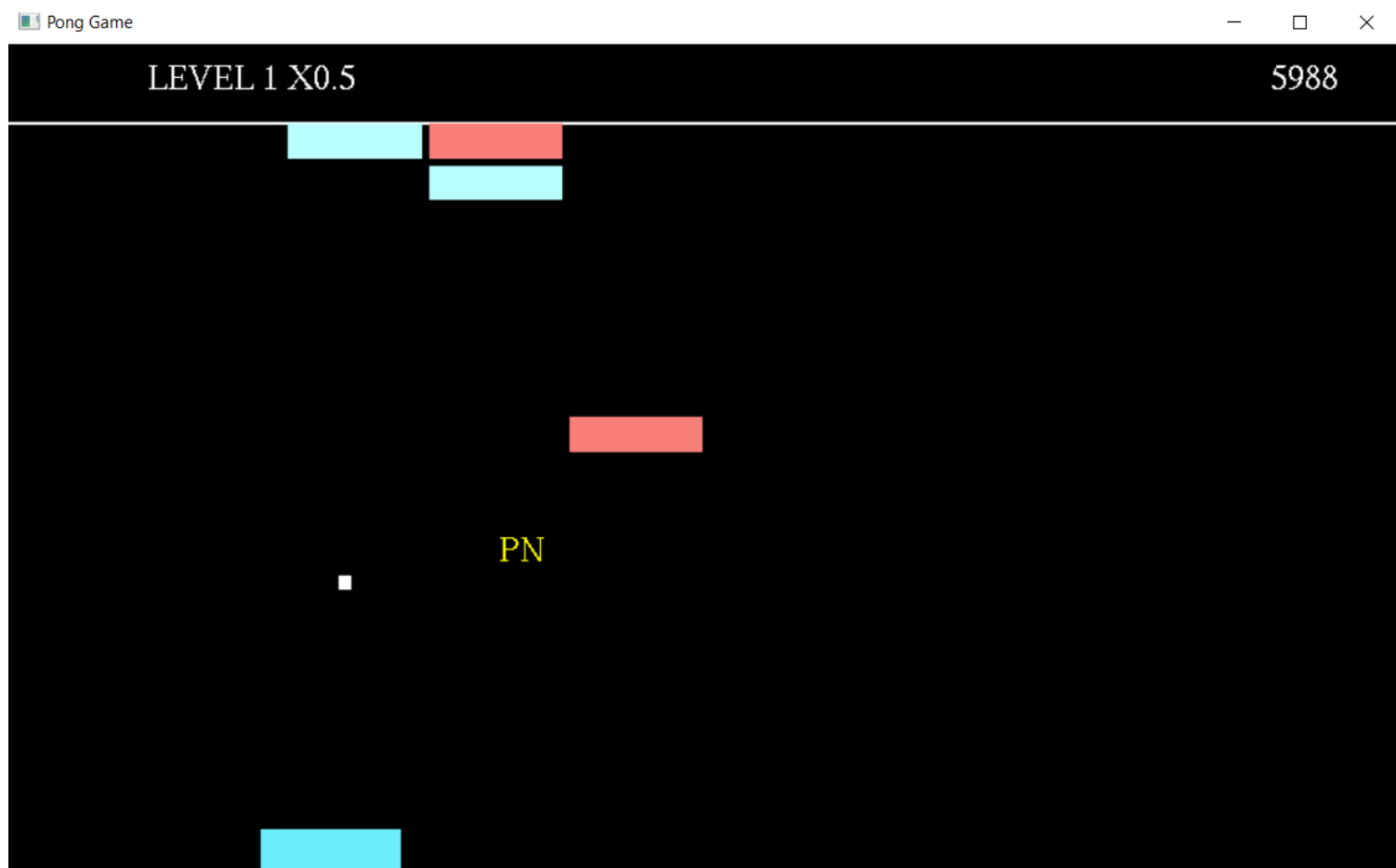


V. Hướng dẫn sử dụng:



Tại menu có 4 lựa chọn:

F1: chơi trò chơi mới

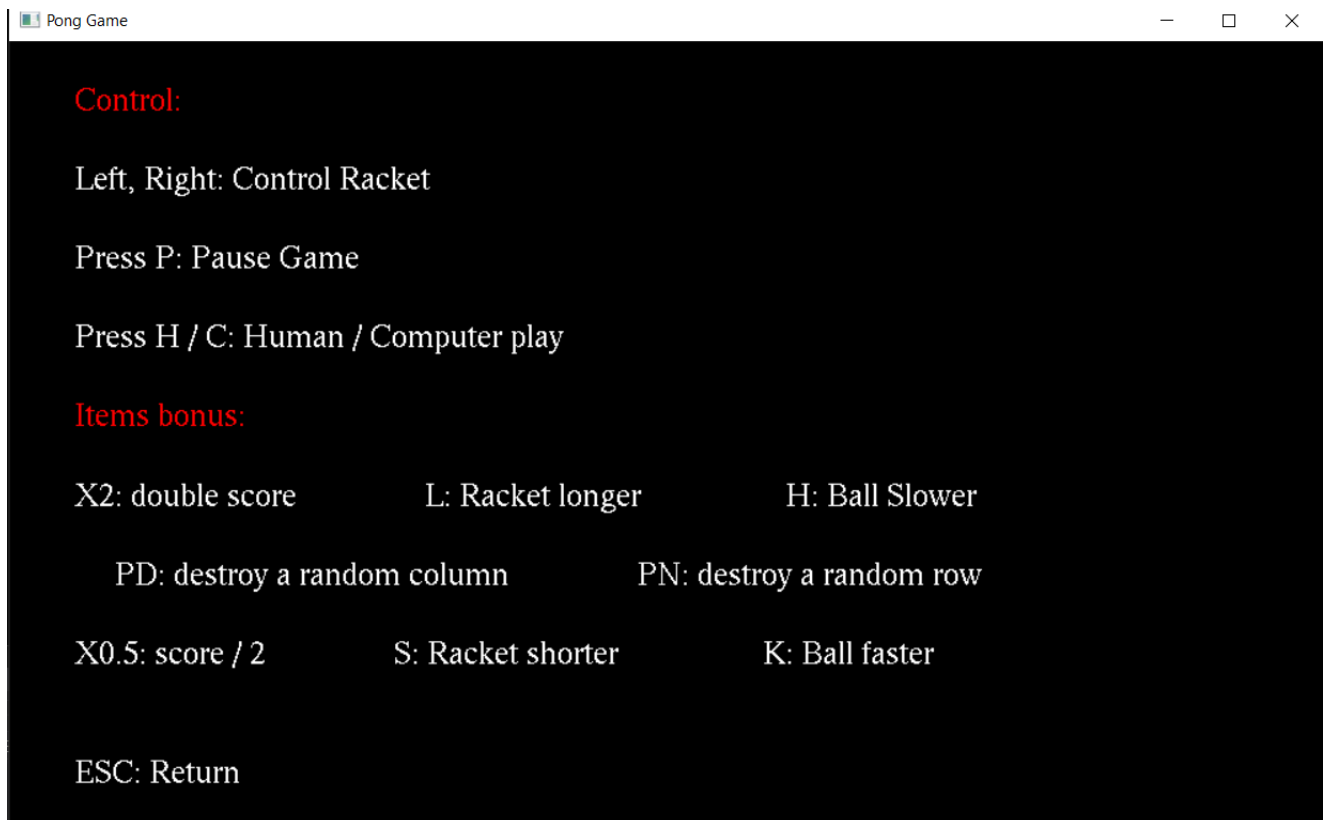


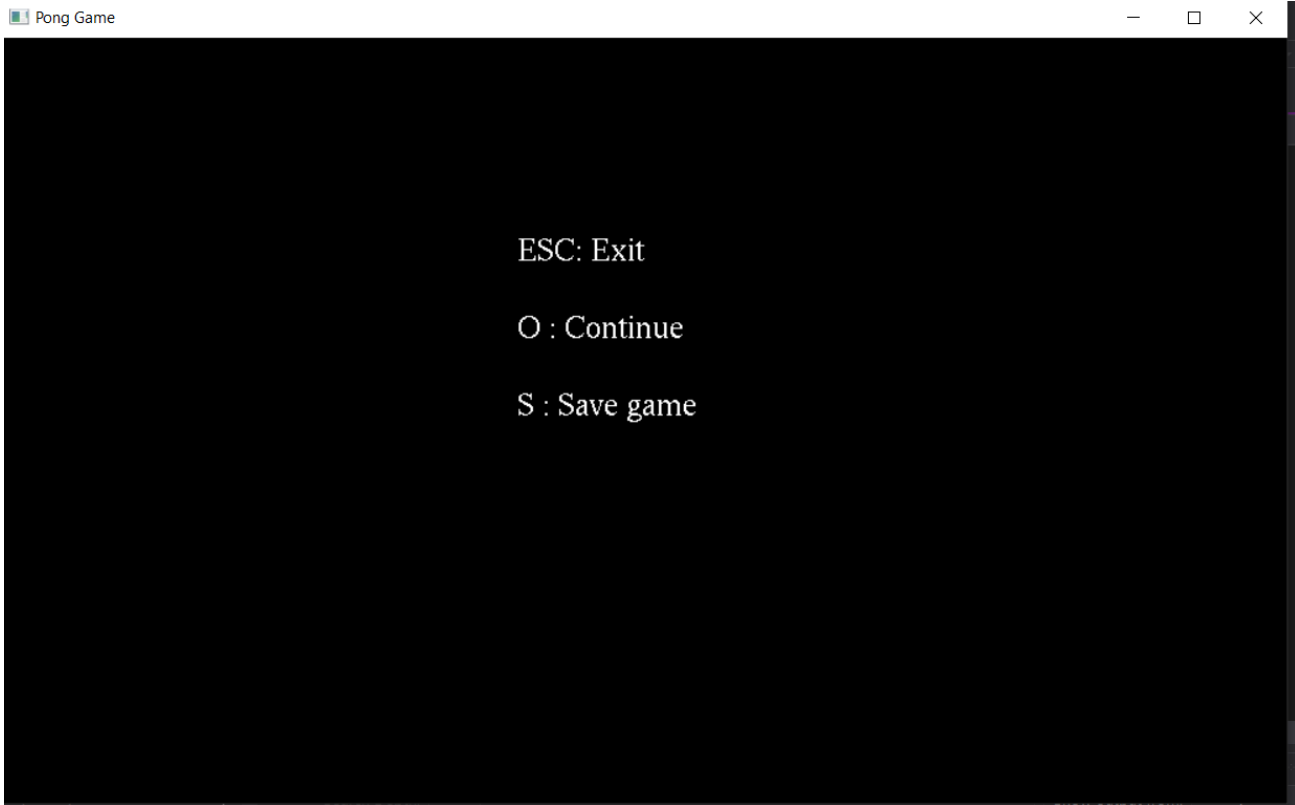
F2: tiếp tục trò chơi cũ
(nếu đã lưu)

F3: bảng xếp hạng

BANG XEP HANG		
AAA		174262
kai		142939
HEY		138976
Computer1		100000
TEST_2		97056
Computer2		90000
DUC		82670
Computer3		80000
BEST		71541
Computer4		70000
		Your score: 0
ESC: Return		

F4: hướng dẫn chơi





Trong quá trình chơi :

- ấn P để dừng game
 - ấn ESC để thoát
 - ấn O để tiếp tục trò chơi
 - ấn S để lưu trò chơi
- ấn H: người chơi
- ấn C: máy chơi
- phím trái phải để di chuyển trái phải
- Item:
 - X2: nhân đôi số điểm nhận được trong 10 lượt bóng chạm vợt
 - L: thanh vợt dài gấp đôi
 - H: bóng chậm đi
 - PD: phá hủy dòng bất kì
 - PN: phá hủy cột bất kì
 - X0.5: chia đôi số điểm nhận được trong 10 lượt bóng chạm vợt
 - S: thanh đỡ ngăn lại
 - K: bóng nhanh hơn

VI. Nhận xét đồ án

Nhóm 18120165-18120167

Nhận xét nhóm : 1512615

Class Ball:

- các hướng di chuyển chưa phù hợp, chỉ tồn tại 6 hướng
- hướng di chuyển chưa phù hợp với yêu cầu va chạm vật lí.
- Hàm di chuyển không rõ ràng bởi không ghi cách định hướng.
- Chưa có hàm tăng tốc cho bóng.

Class Paddle:

- Cách di chuyển và khởi tạo vị trí khó phát triển.

Class cGameManager:

- Liên tục vẽ và xóa làm cho game bị giật, hình ảnh không mượt mà.
- Sử dụng clear screen thay vì xóa hình ảnh của những vật thay đổi.
- Hàm logic là hàm va chạm tuy nhiên cách va chạm không phù hợp, không thể sử dụng để phát triển.

Nhận xét chung:

- Không hề chú thích cho code.
- Đồ án gọn gàng, đơn giản, dễ nắm bắt tuy nhiên thiếu sự đầu tư, trau chuốt. Hệ thống xây dựng game thiếu tính kế thừa, không có khả năng phát triển những chức năng cao cấp và phức tạp, không thể nâng cấp.