|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**BÁO CÁO**

**MÔN HỌC ĐỒ HỌA MÁY TÍNH**

**THIẾT KẾ GAME RẮN SĂN MỒI VỚI MÃ NGUỒN MỞ OPEN GL**

Ngành : **CÔNG NGHỆ THÔNG TIN**

Môn Học : **ĐỒ HỌA MÁY TÍNH**

Sinh viên thực hiện : Tên : Lê Văn Tâm

MSSV : 1711060286

Tên : Phan Tấn Tài

MSSV : 1711060286

Tên : Lê Thành Thuận

MSSV : 1711060286

TP. Hồ Chí Minh, 2020

|  |  |
| --- | --- |
| logomobile.png | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**BÁO CÁO**

**MÔN HỌC ĐỒ HỌA MÁY TÍNH**

**THIẾT KẾ GAME RẮN SĂN MỒI VỚI MÃ NGUỒN MỞ OPEN GL**

Ngành : **CÔNG NGHỆ THÔNG TIN**

Môn Học : **ĐỒ HỌA MÁY TÍNH**

Sinh viên thực hiện : Tên : Lê Văn Tâm

MSSV : 1711060286

Tên : Phan Tấn Tài

MSSV : 1711060286

Tên : Lê Thành Thuận

MSSV : 1711060286

Giảng viên hướng dẫn :

TP. Hồ Chí Minh, 2020

# LỜI MỞ ĐẦU

Trân trọng cảm ơn giảng viên – thầy Nguyễn Đông Phương là giảng viên khoa Công Nghệ Thông Tin – Trường đại học HUTECH, đã tạo điều kiện học tập tốt nhất để chúng em có thể hoàn thành báo cáo lần này.

Ngày nay, nhắc tới “đồ họa máy tính” hay “xử lý ảnh” thì nhiều người nghĩ chắc chắn nó có liên quan về Photoshop. “Thật vậy Computer Vision” thường khiến sinh viên nói chung, những người chưa tham gia, tìm hiểu sẽ khiến nó hơi dễ gây nhầm lẫn với việc chụp và xử lý ảnh chụp “Photoshop”. Vậy khái niệm “Computer vision” sau khi tham gia quá trình học tập, chúng em đã hiểu nó đơn giản là “Thị giác máy tính”, nó rất có ích trong việc tạo ra các chương trình máy tính có khả năng diễn dịch ( interpret ) hình ảnh. Có thể hiểu nó giúp máy tính có thể hiểu được nội dung, ý nghĩa đằng sau một bức ảnh.

Ngoài ra, chúng em còn có thể hiểu được tầm quan trong, các thức vận hành của mã nguồn mở “OPENGL”. OPEN GL là giao diện lập trình ứng dụng đồ họa (API), hỗ trợ rất tốt chúng em trong quá trình thiết kế - xây dựng ứng dụng.

Chính vì thế, chúng em quyết định thiết kế lại ứng dụng (game) “Rắn săn mồi”, xử dụng mã nguồn mở Open GL, với các kiến thức trên lớp lý thuyết và thực hành và kết hợp với tài liệu tham khảo bên ngoài.

Do chưa có nhiều kinh nghiệm nghiên cứu, thực hành nên ứng dụng thực hiện sẽ còn nhiều thiếu sót, chúng em mong nhận được đóng góp ý kiến của thầy để ứng dụng có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn !

Sinh viên thực hiện

**Lê Văn Tâm**

# CHƯƠNG 1. TỔNG QUAN

## 1.1. Tổng quan về vấn đề được nghiên cứu

### 1.1.1. Khảo sát thực trạng :

Hiện nay, trò chơi “Rắn săn mồi” chắc chắn rất quen thuộc với tất cả mọi người, kể cả từ người già đến trẻ nhỏ.

Sơ lược về ý tưởng, nguồn gốc : Rắn săn mồi là tên gọi chung của một thể loại game trong đó người chơi sẽ điều khiển một đoạn thẳng có chiều dài bằng cách lựa chọn đầu vào ( input ) là hướng đi của đầu con rắn làm sao cho đối tượng có thể ăn thức ăn có trên map ( ngẫu nhiên ) một cách nhiều nhất có thể và tránh đâm đầu vào tường ( rào cản ), và việc dễ dàng chơi, nên game đã có hàng tram phiên bản cho nhiều nền tảng, mà nổi bật nhất chính là trên thiết bị di động của Nokia – đã nổi nên như một hiện tượng lớn.

### 1.1.2. Đánh giá

#### 1.1.2.1. Ưu điểm

Không cần phải thiết kế một thuật toán mới.

Ứng dụng nhỏ, gọn, không chiếm nhiều dung lượng.

Thể hiện được hết yêu cầu sử dụng thư viện Open GL.

#### 1.1.2.2. Nhược điểm

Chính vì không cần thiết kế thuật toán mới, nên ứng dụng còn đơn giản.

Giao diện thiết kế chưa thực sự bắt mắt.

### 1.2.1. Mô tả ứng dụng

#### 1.2.1.1. Phạm vi

##### Thông tin tổng quan :

* Thư viện : Sử dụng thư viện “glew” và “freeglut” để dùng các tiện ích của OPEN GL. Xâu dựng thêm thư viện, chứa các tiện ích muốn thiết kế.
* Tên ứng dụng : Snake Game ( Rắn săn mồi ).

##### Mục tiêu

* Người dùng ( game thủ ) có thể mở ứng dụng, và chạy trên nền windows.
* Điều khiển hướng đi của đối tượng ( con rắn ) lên, xuống, sang trái và sang phải.
* Tính điểm và hiển thị kết thúc trò chơi.

1. Lợi ích mang lại

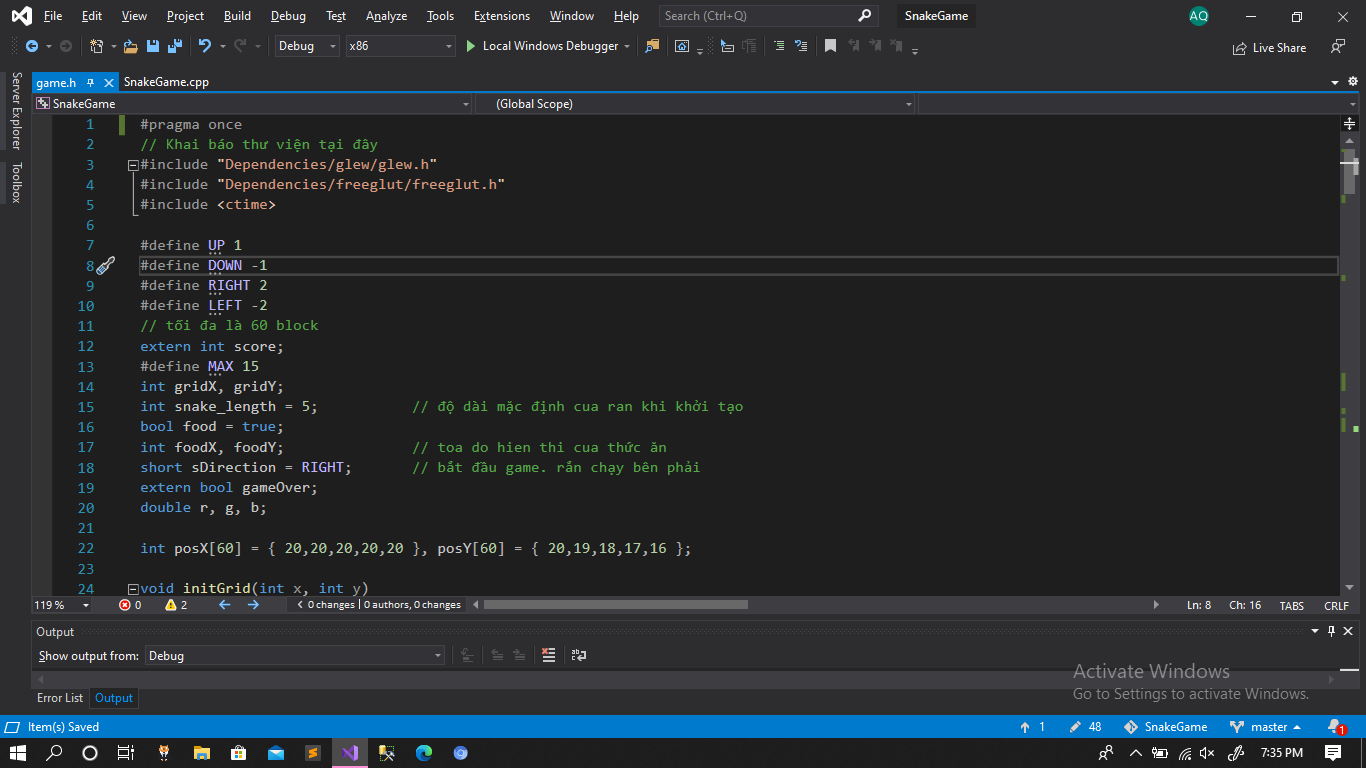
* Giúp chúng em hiểu sâu hơn về môn học.
* Cung cấp kiến thức về đồ họa.
* Khả năng tự tìm hiểu, tự học tốt hơn.
* Phát triển khả năng tư duy, giải quyết vấn đề.

1. Các bước để xây dựng hoàn thành ứng dụng

* Lập kế hoạch tìm hiểu thuật toán
* Phân tích các func cần sử dụng trong Open GL với tài liệu trên Microsoft.
* Bắt đầu thiết kế.
* Kiểm tra lại ứng dụng.
* Soạn tài liệu và báo cáo.

# CHƯƠNG 2. KẾT QUẢ THỰC NGHIỆM

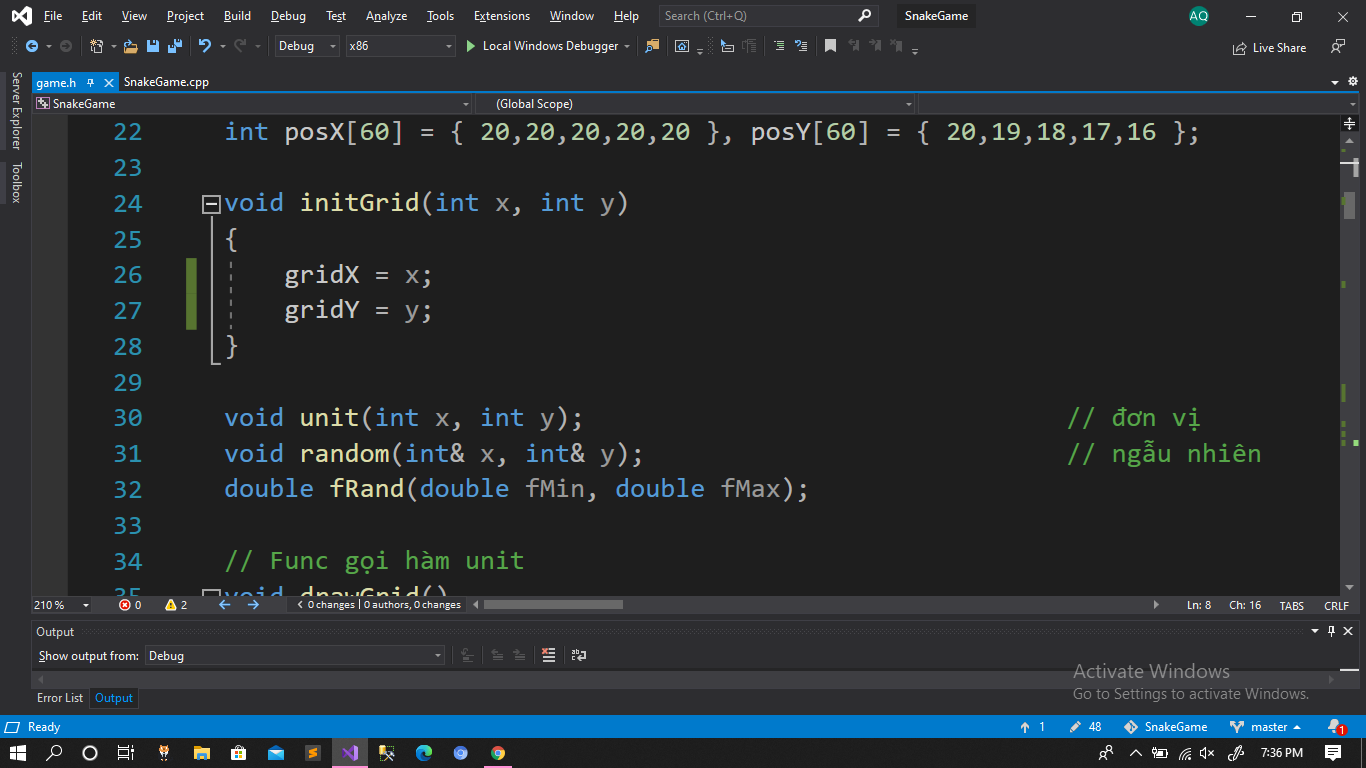
1. Giới thiệu & giải thích



( Hình 1 : Khai báo thư viện của OpenGL )

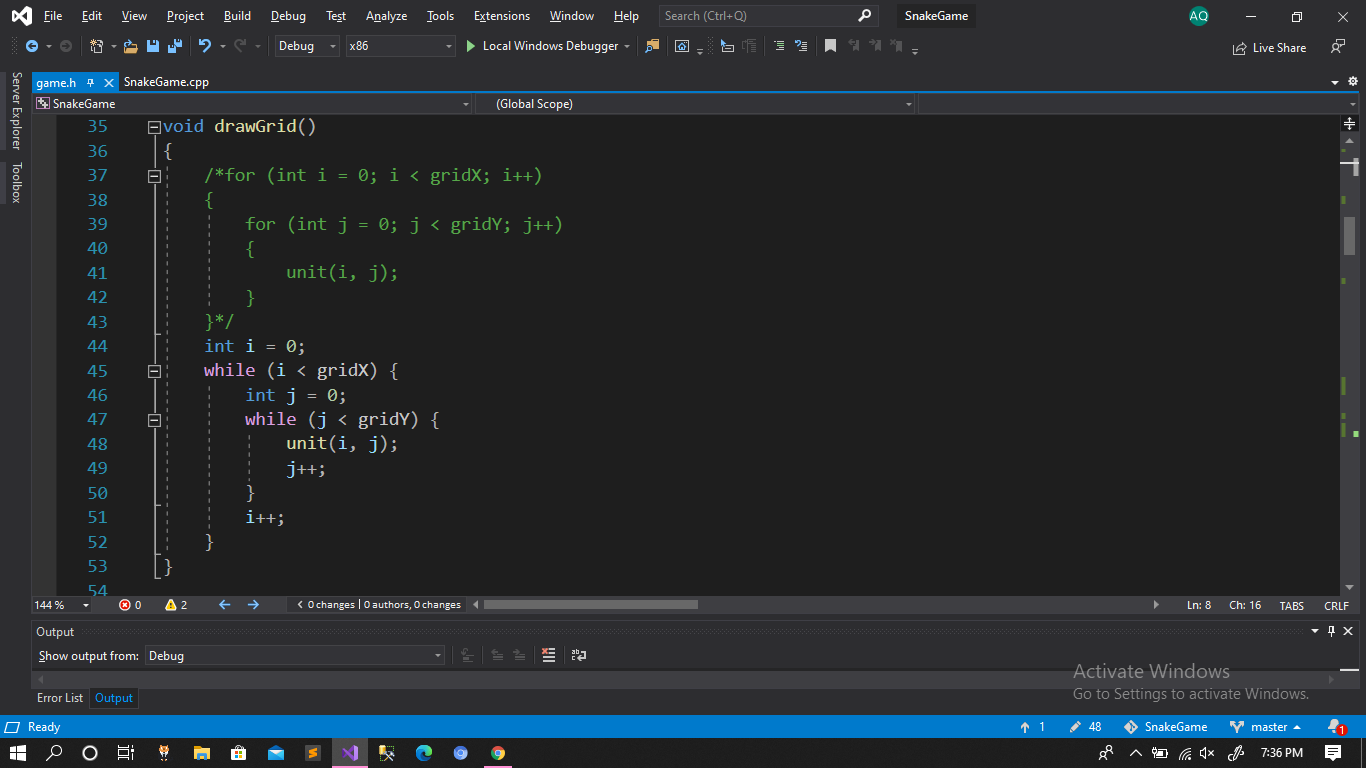
Giải thích :

* Khai báo hằng số : UP, DOWN, RIGHT, LEFT, MAX ( độ dài tối đa của rắn )
* Khai báo biến toàn cục : gridX,gridY ( tọa độ ), snake\_length ( độ dài mặc định của rắn khi khởi tạo ), foodX,foodY ( tọa độ thức ăn ), sDirection = RIGHT ( hướng đi mới khởi tạo sẽ là hướng về bên phải ), food = true, gameOver, r-g-b ( red – green – blue ), mảng



Hình 2 : Khởi tạo lưới

* Khai báo hàm khởi tạo Grid ( lưới )
* Hàm không trả về giá trị, chỉ gán gridX = x, grid Y = y;
* Được sử dụng để gọi lại trong snakeGame.cpp.



Hình 3 : Vẽ lưới

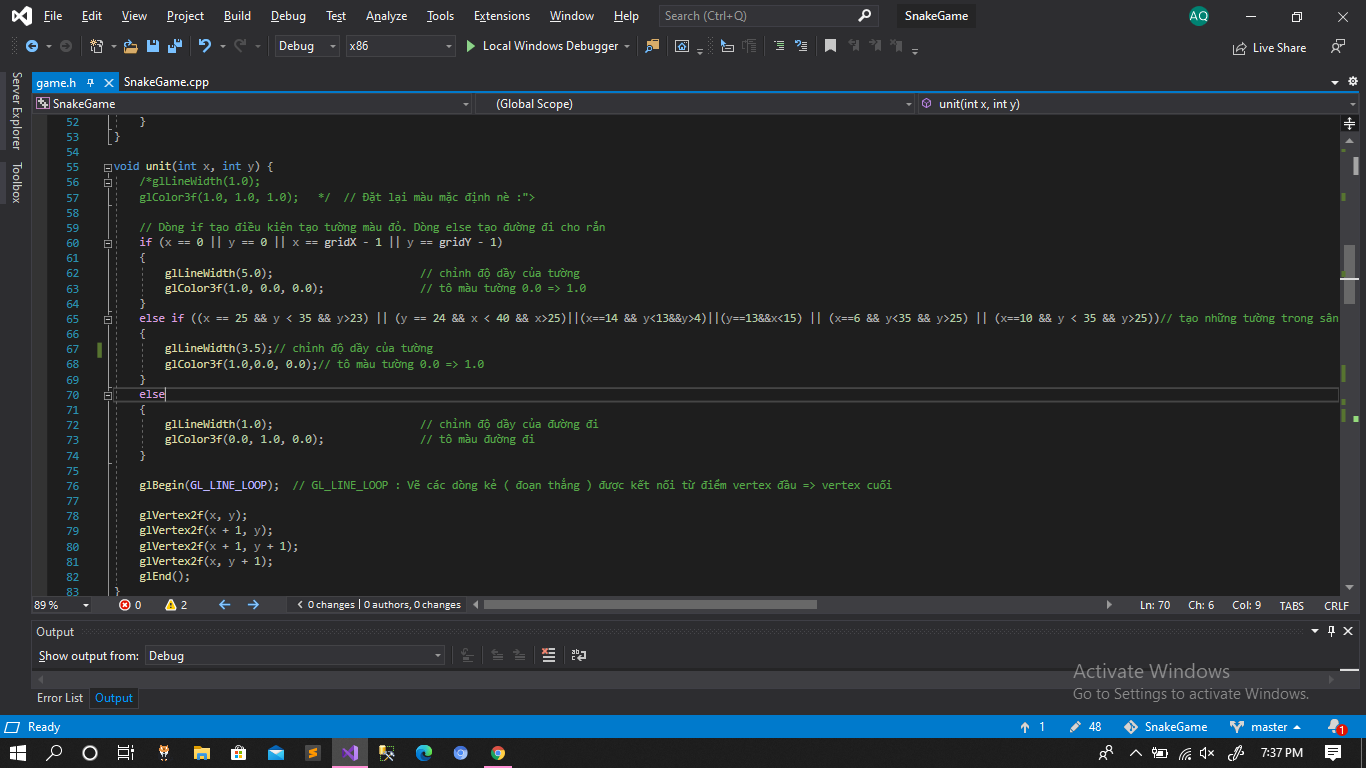
- Vòng while:

+i tọa độ tại vị trí 0, sau đó gắn điều kiện cho tọa độ đó nhỏ hơn tọa độ x.

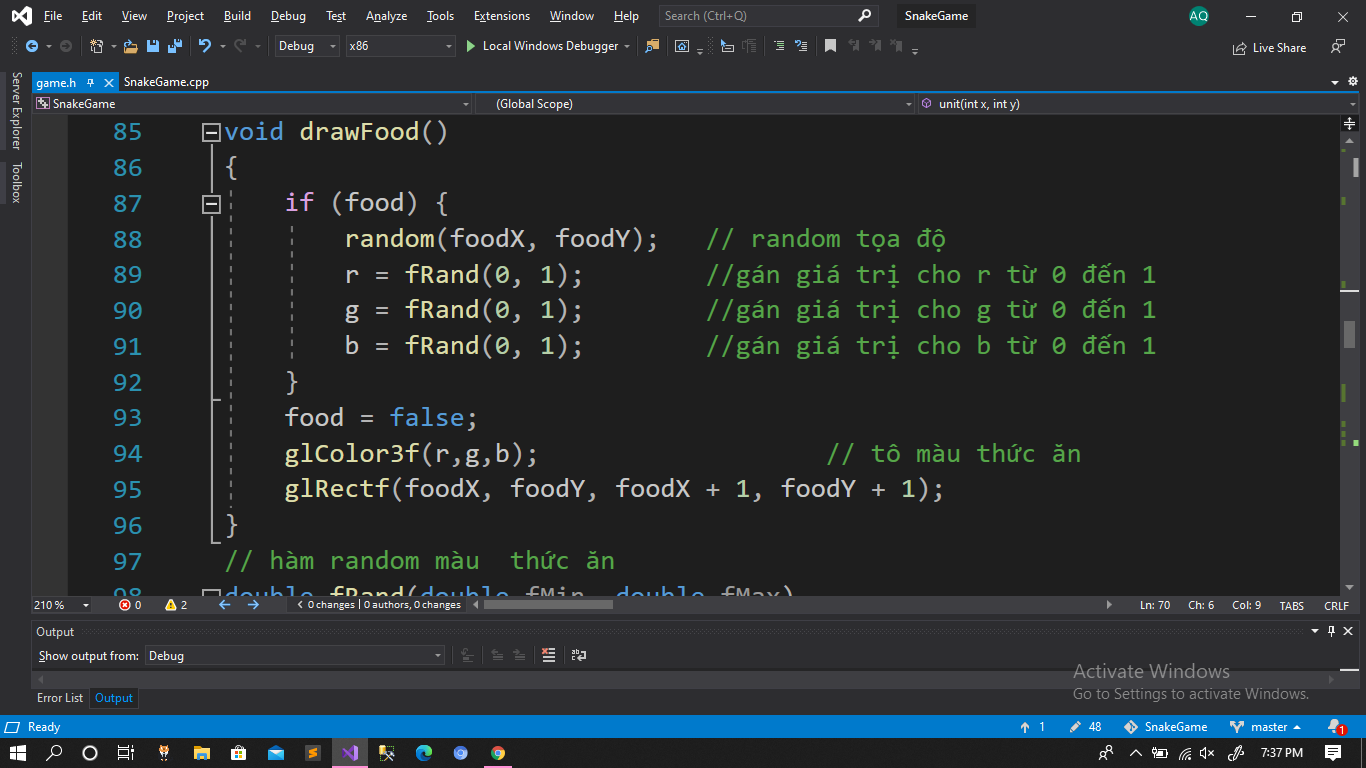
+Thỏa điều kiện sẽ thực hiện tương tự đối với biến được gọi là j so với tọa độ y. Thỏa điều kiện sẽ gọi hame tạo lưới unit.

+Và biến I và j sẽ tăng dần cho đến khi không thỏa điều kiện nữa.

* Được gọi trong snakeGame.cpp để khởi tạo Grid ( lưới )

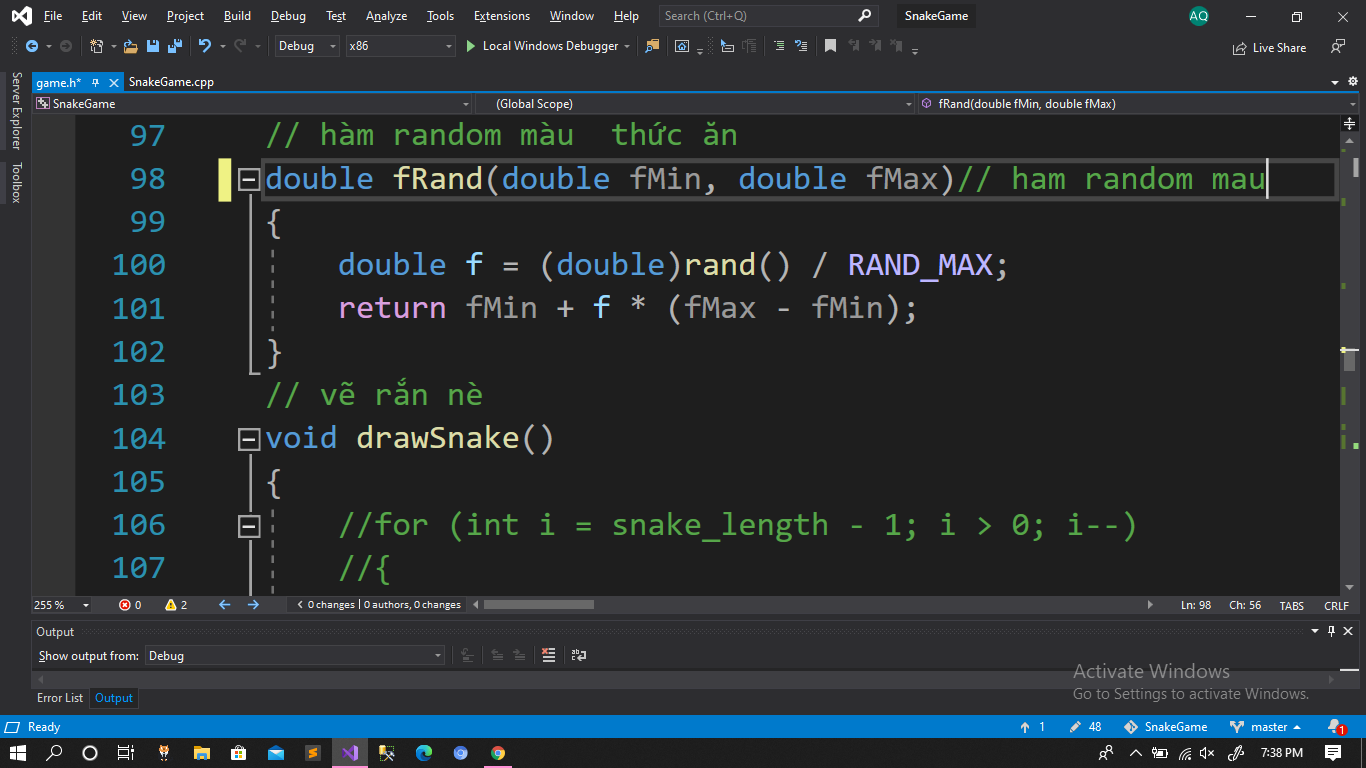
Hình 4 : Xử lý đơn vị trong bản đồ

* Dòng if thứ nhất xử lý điều kiện nếu tọa độ của x,y có giá trị là 0 hoặc có giá trị gridX-1, gridY-1 ( giá trị cuối cùng trong mảng ) thì đường kẻ có độ dày là 5.0, có màu đỏ
* Dòng if thứ hai xử lý điều kiện vẽ độ dày đường kẻ và tô màu tường vật cản
* Dòng else sẽ chỉnh độ dày của đường đi và tô màu đường đi
* glBegin(GL\_LINE\_LOOP) : là một method của OpenGL để vẽ các đường kẻ
* glVertex2f dùng để liên kết các đỉnh tọa độ x và y với nhau.



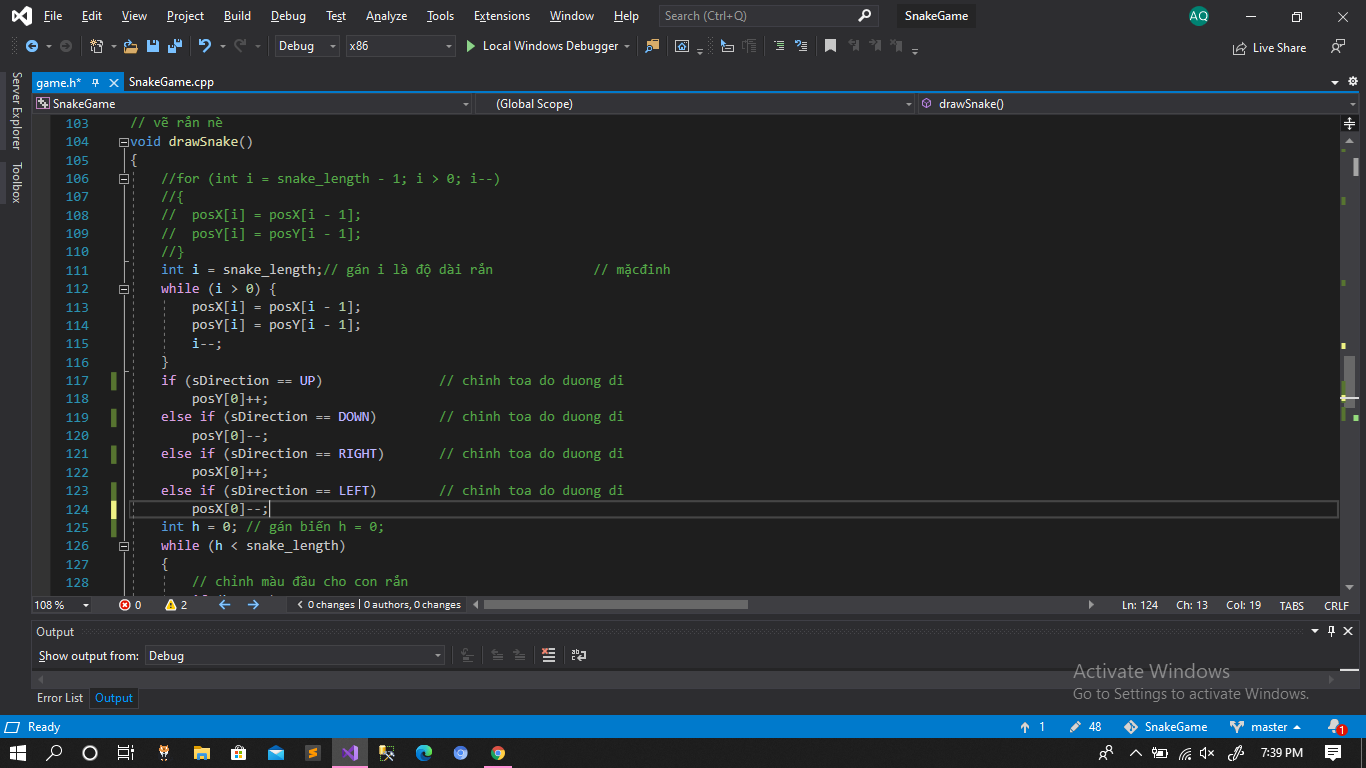
Hình 5 : Vẽ thức ăn cho rắn

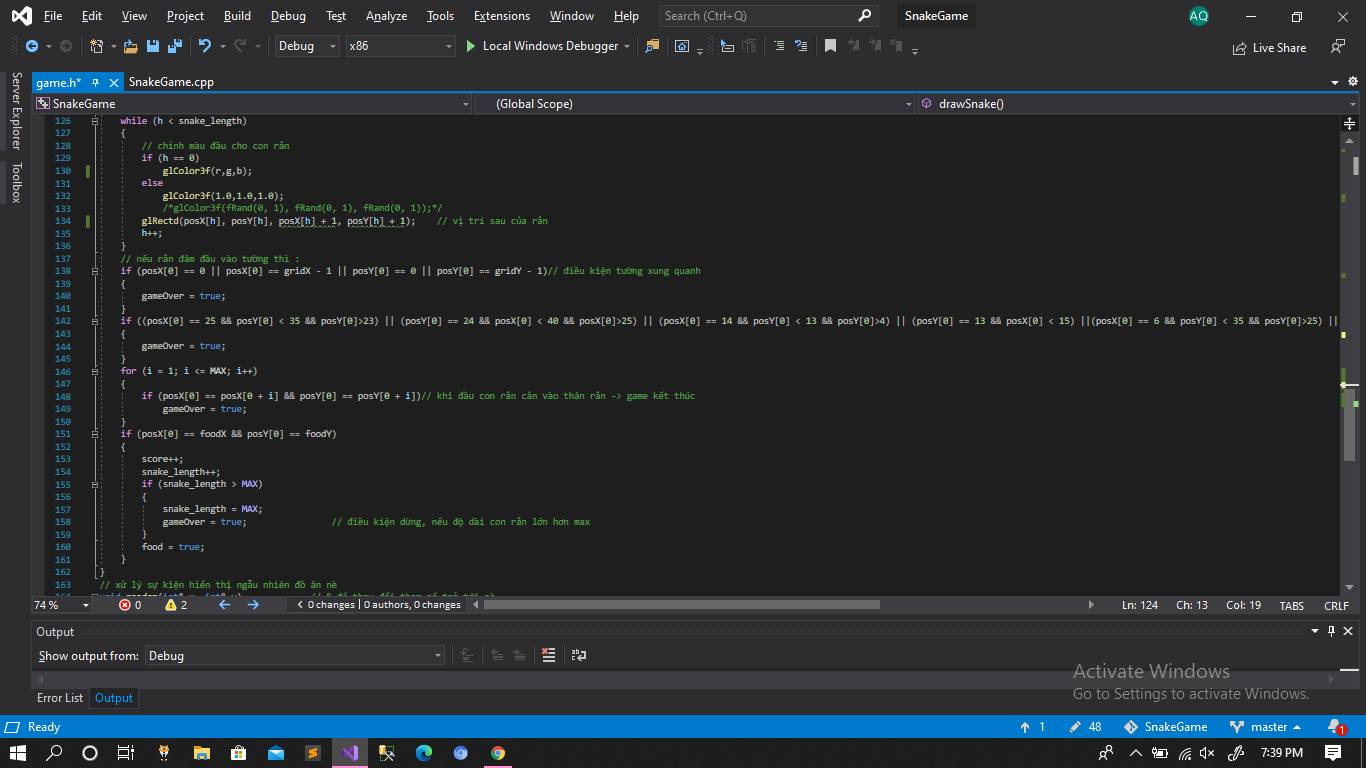
* Thức ăn sẽ được hiển thị tự do trong bản đồ.
* Trường hợp thức ăn có giá trị true thì xử lý gán ngẫu nhiên giá trị của foodX, foodY.
* R,g,b sẽ sinh ngẫu nhiên giá trị từ 0 => 1 để mỗi lần rắn ăn xong, thức ăn sẽ có màu mới.
* Food = false : sau khi vẽ xong sẽ không vẽ tiếp cho tới khi rắn ăn
* glColor3f : dùng để tô màu thức ăn.
* glRectf : thức ăn sẽ là một hình chữ nhật



Hình 6 : Xử lý sinh màu ngẫu nhiên

* Hàm gồm hai tham số truyền vào, dữ liệu trả về kiểu double.
* F có giá trị ngẫu nhiên và được chia lấy phân số.
* Trả về kết quả là tham số thứ 1 + với biến f nhân với (tham số 2 – tham số 1).





Hình 7 : Vẽ rắn

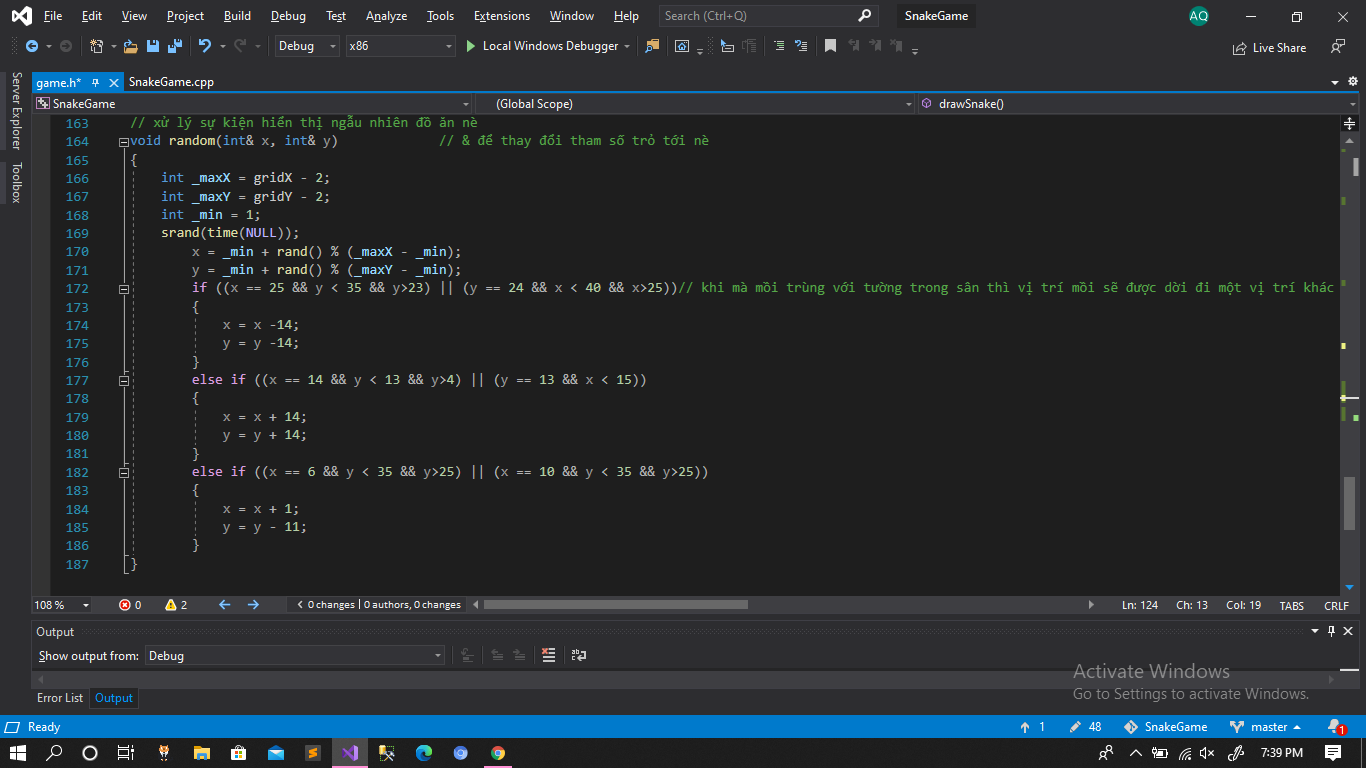
-Vòng while:

+ I là biến độ dài của rắn.

+ Vẽ phần thân của rắn với từ tọa độ posX và posY.

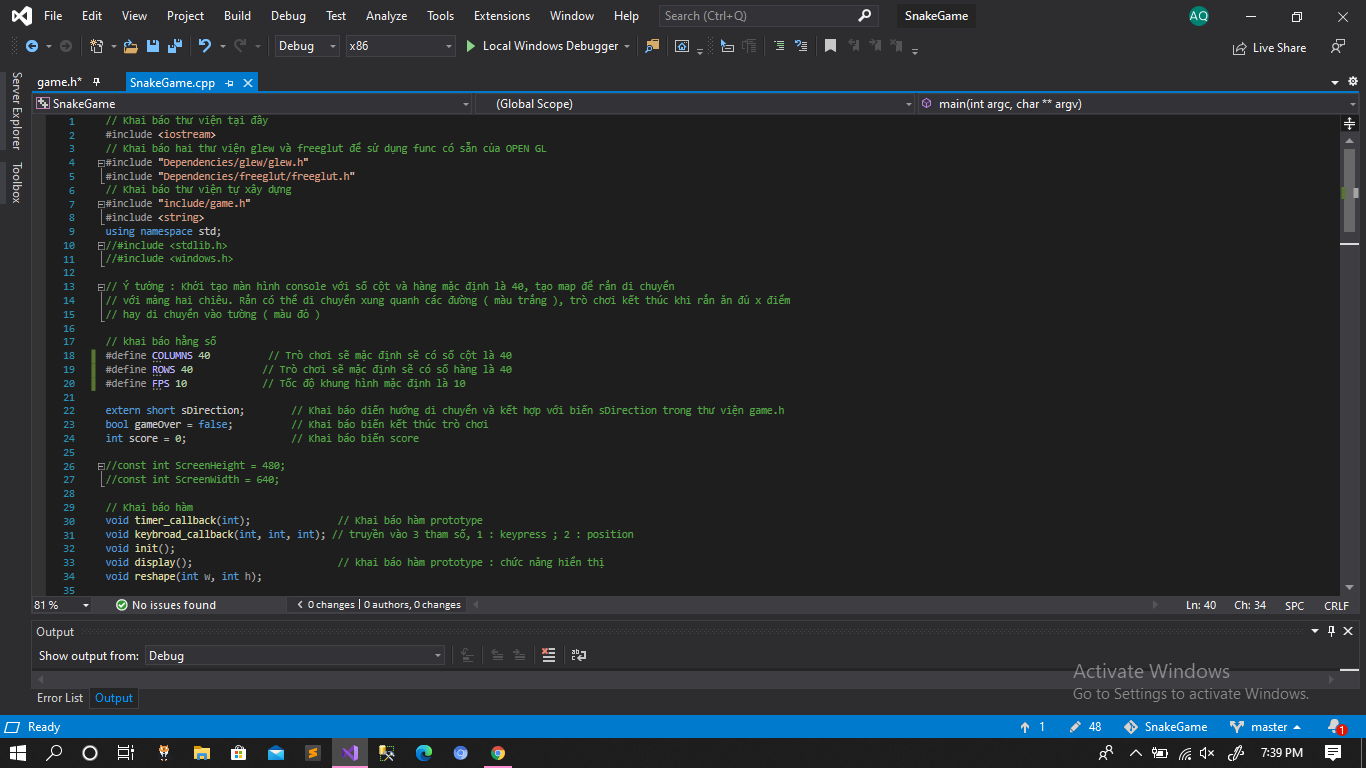
+Sẽ dừng vẽ khi biến I =0.

* Câu lệnh rẽ nhánh xử lý thay đổi tọa độ của rắn.
  + Nếu rắn đi lên thì tương đương y sẽ tăng ngược lại đi xuống y giảm
  + Nếu rắn sang phải thì tương đương x sẽ tăng và ngược lại sang trái x sẽ giảm
* Vòng while :
  + h là biến đầu của rắn, với vị trí đang trỏ tới là 0.
  + Vị trí đầu của rắn sẽ có màu ngẫu nhiên với thức ăn.
  + Phần thân của rắn sẽ trỏ từ vị trí 1 đến vị trí cuối cùng sẽ to màu trắng.
* glRectd : Dùng để vẽ con rắn hình chữ nhật
* h++ : tăng dần tới khi nhỏ hơn độ dài của con rắn.
* Điều kiện dừng
  + Nếu đầu rắn ( posX,posY ) mà có tọa độ là 0 hoặc tọa độ là điểm cuối cùng của lưới thì game sẽ kết thúc ( gameOver = true ).
  + Nếu đầu rắn ( posX,posY ) mà có tọa độ bằng tọa độ của chướng ngại vật thì game sẽ kết thúc.
  + Nếu game rắn vượt quá độ dài tối đa ( MAX ) thì game sẽ kết thúc, game thủ thắng trò chơi.
* Xử lý sự kiện rắn ăn mồi
  + Score : Mỗi lần rắn ăn trúng thức ăn thì điểm số sẽ tăng lên một đơn vị
  + Snake\_length : Mỗi lần ăn trúng thức ăn thì mảng rắn sẽ tăng dần.
  + Nếu rắn đạt độ dài tối đa thì game kết thúc.
  + Đặt lại thức ăn là true để vẽ thức ăn



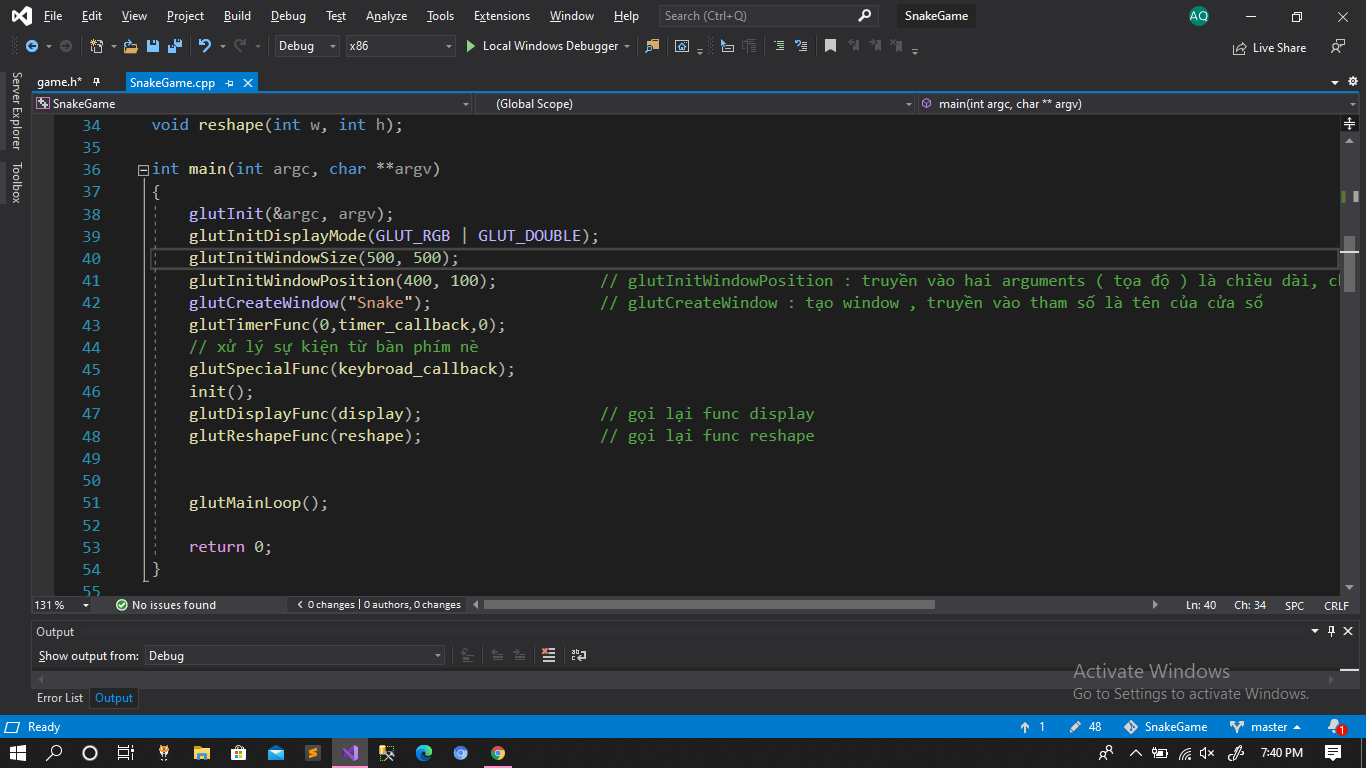
Hình 8 : Xử lý dữ liệu sinh tọa độ ngẫu nhiên

* Khai báo biến -maxX : tọa độ lớn nhất X sẽ nhỏ hơn tọa độ của lưới
* Khai báo biến -maxY : tọa độ lớn nhất Y sẽ nhỏ hơn tọa độ của lưới
* Srand(time(NULL)) : xử lý ngẫu nhiên không lặp lại vị trí trước đó.
* X, Y sẽ thay đổi ngẫu nhiên trong khoảng min tới \_maxX, \_maxY.
* Câu lệnh điều kiện : Khi mồi sinh ngẫu nhiên trùng với tường thì sẽ dời đi vị trí khác.



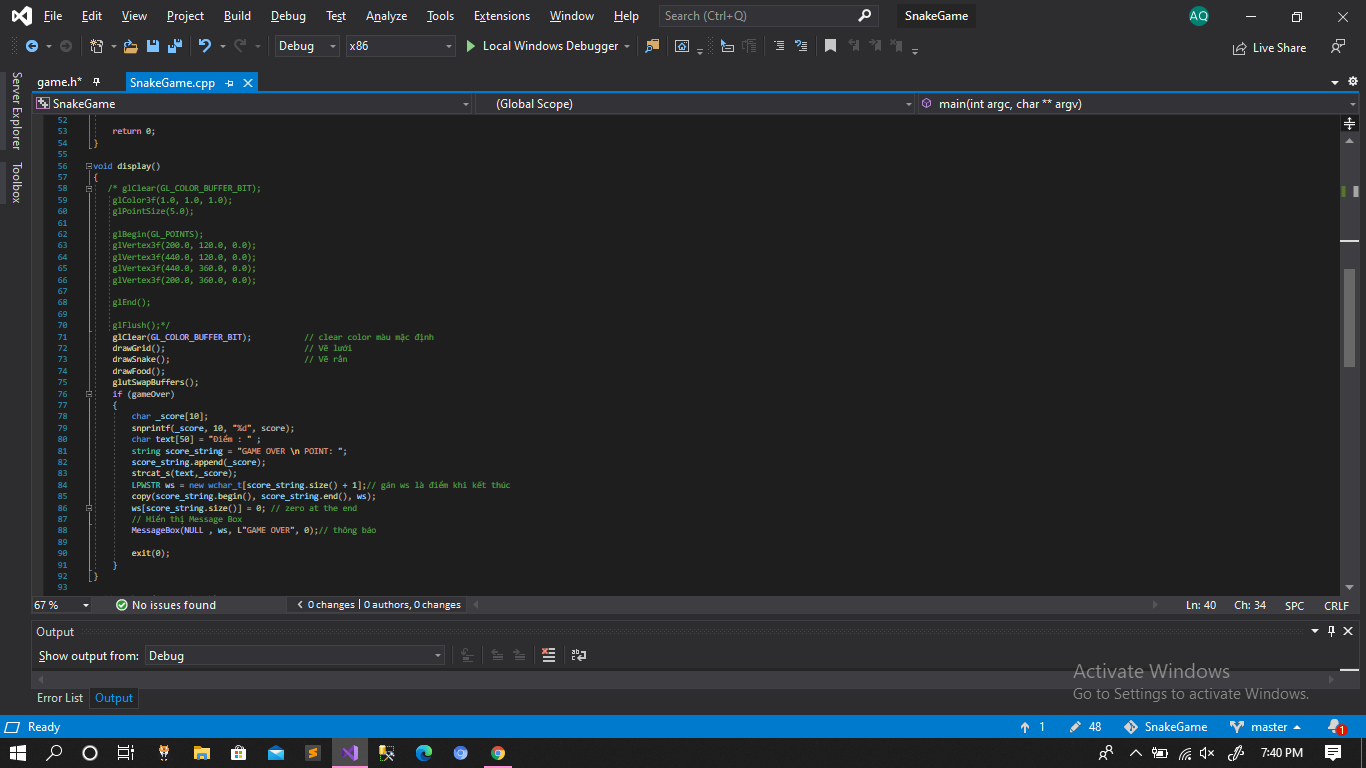
Hình 9 : Khai báo thư viện, gọi lại hàm, khai báo hằng số, biến toàn cục

* Khai báo thư viện có sẵn trorng C/C++, khai báo đường dẫn thư viện glew, freeglut để sử dụng phương thức của OPENGL.
* Khai báo COLUMS, ROWS có giá trị mặc định là 40,40.
* Khai báo FPS là tốc độ khung hình mỗi giây ( Frame per second ).
* Khai báo hướng di chuyển của rắn.
* Khai báo biến trò chơi kết thúc.
* Khai báo điểm
* Khai báo các hàm gọi lại, hàm khởi tạo.



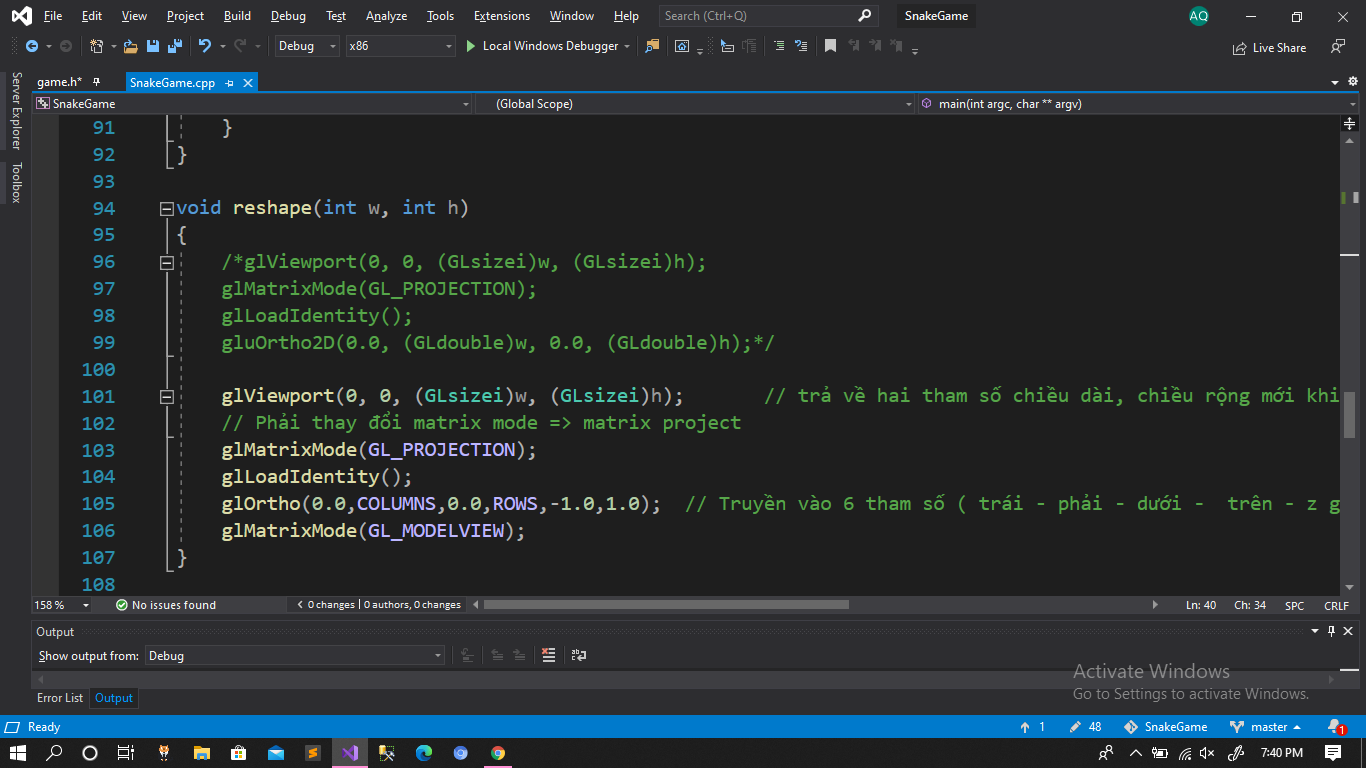
Hình 10 : Khai báo hàm main

* glutInit : Khởi tạo thư viện GLUT.
* glutInitDisplayMode : Tạo cửa sổ
* glutInitWindowSize : Tạo kích cỡ cửa sổ
* glutInitWindowPosition : Tạo vị trí hiển thị
* glutCreateWindow : Tạo cửa số có tên là Snake
* glutTimeFunc : gọi lại hàm timer\_callback mỗi 0 milisecond.
* gltSpecialFunc() : Đặt lệnh gọi lại bàn phím cho cửa sổ.
* Init() : Hàm khởi tạo
* glutDisplayFunc() : gọi lại hàm display
* glutReshapeFunc() : gọi lại hàm reshape
* glutMainLoop() : Lặp lại các vòng xử lý của GLUT.



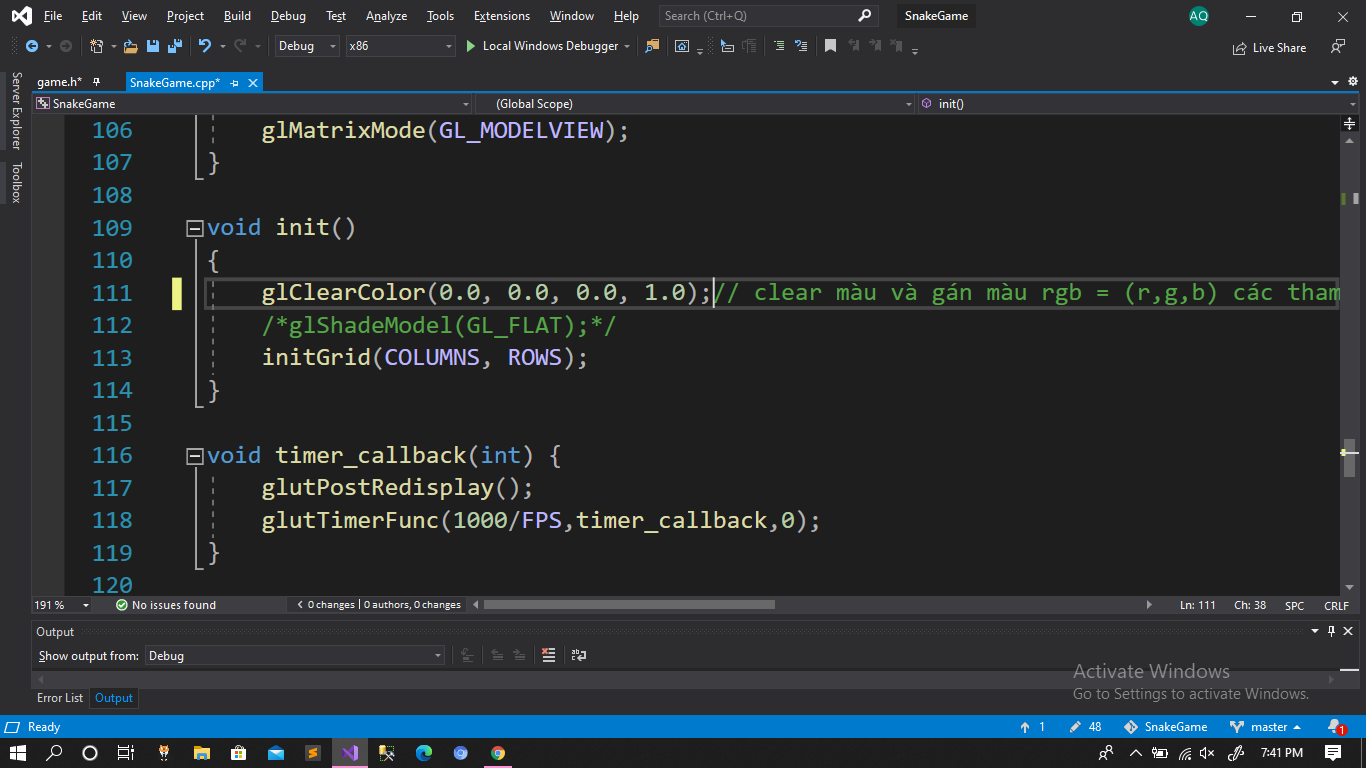
Hình 11 : Hàm hiển thị

* glClear : xóa màu mặc định.
* DrawGrid : Vẽ lưới .
* DrawSnake : Vẽ rắn.
* DrawFood : Vẽ thức ăn.
* IF GameOver:
  + Khai báo biến \_score.
  + Gán kiểu string cho biến score\_string.
  + Gán ws là biến điểm khi kết thúc game.
  + xuất ra thông báo GameOver và Point… biến Ponit được gán là ws.
  + Xuất ra thông báo bằng MessageBox.



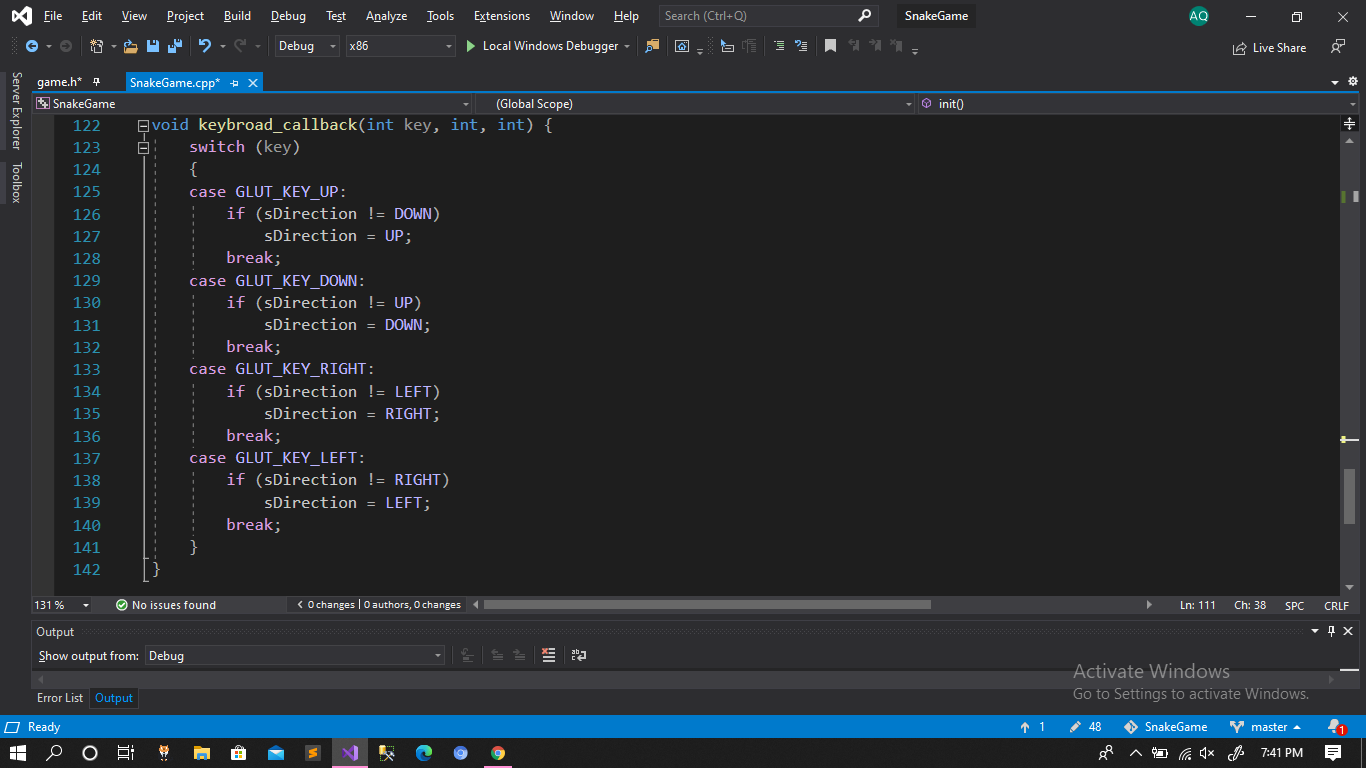
Hình 12 : Hàm Reshape

* glViewport : trả về tham số chiều dài, chiều rộng mới.
* glMatrixMode : Đặt lại ma trận
* glLoadIdentity : Thay thế ma trận hiện tại bằng ma trận nhận dạng
* glOrtho : Chuyền vào 6 tham số \* trái, phải,dưới, trên, khoảng các gần, khoảng cách xa.



Hình 12 : Hàm khởi tạo , hàm thời gian gọi lại

* glClearColor : đặt lại màu cho buffer.
* initGrid : truyền vào giá trị của hàng, cột để khởi tạo.
* glutPostRedisplay đánh dấu cửa sổ đang hiển thị là cửa sổ cần lặp lại.
* glutTimerFunc gọi lại hàm timer\_callback trong 1000 miliseconds / FPS ( mặc định ) với value là 0.



Hình 13 :Hàm gọi lại input từ bàn phím ( phím lên , xuống, qua trái, qua phải ).

* Biến key là tham số truyền vào.
* Xử lý rẽ nhánh :
  + TH Nếu nhấn phím mũi tên đi lên ( mà đầu của con rắn không phải là đang đi xuống thì chỉnh đường dẫn đi lên ) thoát.
  + TH Nếu nhấn phím mũi tên đi xuống ( mà đầu của con rắn không phải là đang đi lên thì chỉnh đường dẫn đi xuống) thoát.
  + TH Nếu nhấn phím mũi tên qua trái ( mà đầu của con rắn không phải là đang qua phải thì chỉnh đường dẫn qua trái thoát.
  + TH Nếu nhấn phím mũi tên qua phải ( mà đầu của con rắn không phải là đang qua trái thì chỉnh đường dẫn qua phải) thoát.