#include<Windows.h>

#include<GL/gl.h>

#include<GL/freeglut.h>

#include<iostream>

using namespace std;

float Mx, Bx = -1, By = 0 ;

int Dx = 1, Dy = 0, W, H ;

struct block

{

int Damaged = 0; float X = 0.0; float Y = 0.0;

}Brick;

block B‬‬[28]; int Toob = 0;

int win;

void Mouse(int o, int p)

{

Mx = (float)o / W \* 2 - 1;

}

void ResetGame();

void DispTimer(int v)

{

glutTimerFunc(10, DispTimer, 0);

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_QUADS);

glColor3f(1, 0, 0);

glVertex2d(Mx - 0.1, -0.87);

glColor3f(1, 0, 0);

glVertex2f(Mx - 0.1, -0.93);

glColor3f(1, 0, 0);

glVertex2f(Mx + 0.1, -0.93);

glColor3f(1, 0, 0);

glVertex2f(Mx + 0.1, -0.87);

if (Dx == 1)

Bx += 0.01;

else

Bx -= 0.01;

if (Dy == 1)

By += 0.01;

else

Bx -= 0.01;

if (Bx > 1)

Dx = 0;

else if (Bx < -1)

Dx = 1;

if (By > 1)

Dy = 0;

else if (By < -1)

Dy = 1;

if (Bx < Mx + 0.1 && Bx > Mx - 0.1 && By < -0.87 && By > -0.39)

Dy = 1;

glEnd();

glBegin(GL\_QUADS);

glColor3f(0, 1, 0);

glVertex2f(Bx - 0.02, By + 0.02);

glVertex2f(Bx - 0.02, By - 0.02);

glVertex2f(Bx + 0.02, By - 0.02);

glVertex2f(Bx + 0.02, By + 0.02);

glEnd();

glBegin(GL\_QUADS);

for (int r = 0; r < 28; r++)

{

if (B‬‬[r].Damaged == 0)

{

glColor3f(1, 0, 0);

glVertex2f(B‬‬[r].X, B‬‬[r].Y);

glColor3f(0, 1, 0);

glVertex2f(B‬‬[r].X, B‬‬[r].Y - 0.1);

glColor3f(1, 0, 0);

glVertex2f(B‬‬[r].X + 0.2, B‬‬[r].Y - 0.1);

glColor3f(0, 1, 0);

glVertex2f(B‬‬[r].X + 0.2, B‬‬[r].Y);

}

}

glEnd();

glBegin(GL\_QUADS);

for (int r = 0; r < 28; r++)

{

if (B‬‬[r].Damaged == 0)

{

if (B‬‬[r].Y > By - 0.02 && B‬‬[r].Y - 0.1 < By + 0.02 && B‬‬[r].X < Bx + 0.02 && B‬‬[r].X + 0.2 > Bx - 0.02)

{

if (Dx == 1)

Dx = 0;

else

Dx = 1;

if (Dy == 1)

Dy = 0;

else

Dy = 1;

B‬‬[r].Damaged = 1;

Toob++;

}

glColor3f(1, 0, 0);

glVertex2f(B‬‬[r].X, B‬‬[r].Y);

glColor3f(0, 0, 1);

glVertex2f(B‬‬[r].X, B‬‬[r].Y - 0.1);

glColor3f(1, 0, 1);

glVertex2f(B‬‬[r].X + 0.2, B‬‬[r].Y - 0.1);

glColor3f(0, 1, 1);

glVertex2f(B‬‬[r].X + 0.2, B‬‬[r].Y);

}

}

glEnd();

if (By < -1 || Toob >= 28)

ResetGame();

glFlush();

glutSwapBuffers();

}

void ResetGame()

{

for (int r = 0; r < 28; r++)

B‬‬[r].Damaged = 0;

Bx = -1;

By = 0;

Dx = 1;

Dy = 0;

Mx = 0;

glutWarpPointer(W = 1, H = 1);

}

void size(int Width, int Height)

{

W = Width;

H = Height;

glViewport(0, 0, Width, Height);

}

void keys(unsigned char K, int X, int Y)

{

if (K == 27)

glutDestroyWindow(win);

}

void Dummy() {}

int main(int argc, char\*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);

win = glutCreateWindow("nagar");

glutFullScreen();

glutSetCursor(GLUT\_CURSOR\_NONE);

glClearColor(0, 0, 0, 1);

glutReshapeFunc(size);

glutPassiveMotionFunc(Mouse);

glutMotionFunc(Mouse);

glutDisplayFunc(Dummy);

glutKeyboardFunc(keys);

glutTimerFunc(10, DispTimer, 0);

for (int r = 0; r < 28; r++)

B‬‬[r].Damaged = 0;

float tmpX = -1;

for (int r = 0; r < 10; r++)

{

B‬‬[r].X = tmpX;

B‬‬[r].Y = 1;

tmpX += 0.2;

}

tmpX = -0.8;

for (int r = 10; r < 18; r++)

{

B‬‬[r].X = tmpX;

B‬‬[r].Y = 0.9;

tmpX += 0.2;

}

tmpX = -0.6;

for (int r = 18; r < 24; r++)

{

B‬‬[r].X = tmpX;

B‬‬[r].Y = 0.8;

tmpX += 0.2;

}

tmpX = -0.4;

for (int r = 24; r < 28; r++)

{

B‬‬[r].X = tmpX;

B‬‬[r].Y = 0.7;

tmpX += 0.2;

}

typedef struct

{

float z, n;

int Damaged;

};

glutMainLoop();

return 0;

}