#include <stdlib.h> //for rand()

#include "glut.h"

#include "GL.H"

#include <stdio.h>

#include <conio.h>

#include <math.h> //for drawing circle

#include <Windows.h>//For Sound

#include <mmsystem.h>//For Sound

#define M\_PI 3.14159265358979323846f

int Scores = 0;

int g\_Width = 600;

int g\_Height = 600;

int X = 290;

int Y = 1;

int WD1 = 40;

int WD = 30;

int HT = 50;

int C = 280;

int D = 275;

int car2\_x = 360;

int car2\_y = -200;

int car5\_x = 360;

int car5\_y = -500;

int car3\_x = 215;

int car3\_y = -200;

int car6\_x = 215;

int car6\_y = -10;

int car4\_x = 290;

int car4\_y = -50;

//global mouse position

int mouse\_X = 0; //xcod

int mouse\_Y = 0; //ycodz

int bar1\_x[4] = { 0, 0, 20, 20 };

int bar1\_y[4] = { 250, 350, 350, 250 };

int bar2\_x[4] = { 580, 580, 600, 600 };

int bar2\_y[4] = { 250, 350, 350, 250 };

int c\_x = 300, c\_y = 300, c\_speed = 5, c\_direction = 0;

//Scores

int player1 = 0, player2 = 0;

int chances = 3;

int chance\_counter = 0;

GLuint image1;

void init();

void myMouseFunction(int button, int state, int mouseX, int mouseY);

void myKeyboardFunction(unsigned char key, int mouseX, int mouseY);

void mySpecialKeysFunction(int key, int x, int y);

void myReshapeFunction(int width, int height);

void myTimerFunction(int val);

void myDisplayFunction();

GLuint loadTexture(char \*name);

void draw(GLuint id, int x, int y, int wd, int ht);

void LoadResources()

{

image1 = loadTexture("saleem.tga");

}

//Drawing a Circle

void draw\_circle(int x, int y, int radius) {

float angle;

glColor3f(0.0, 0.6, 0.0);

glLineWidth(1.0f);

}

//Printing in the GLUT Window

void renderBitmapString(

float x,

float y,

char \*string) {

char \*c;

glRasterPos2f(x, y);

for (c = string; \*c != '\0'; c++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18, \*c);

}

}

//Reset all variables to restart game

void reset() {

//Bar1

bar1\_x[0] = 0;

bar1\_x[1] = 0;

bar1\_x[2] = 20;

bar1\_x[3] = 20;

bar1\_y[0] = 250;

bar1\_y[1] = 350;

bar1\_y[2] = 350;

bar1\_y[3] = 250;

//Bar2

bar2\_x[0] = 580;

bar2\_x[1] = 580;

bar2\_x[2] = 600;

bar2\_x[3] = 600;

bar2\_y[0] = 250;

bar2\_y[1] = 350;

bar2\_y[2] = 350;

bar2\_y[3] = 250;

//Circle Position

c\_x = 300, c\_y = 300, c\_speed = 0, c\_direction = 0;

//Scores

player1 = 0, player2 = 0;

chances = 3;

}

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

{

glutInit(&argc, argv); //initilize the GLUT libraray

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA); //has four colour components red, green,blue, and alpha

glutInitWindowSize(600, 600); //size of window

glutInitWindowPosition(0, 0); //poistion of window

glutCreateWindow("My First OpenGL Appliction, Hooo Yeah");

LoadResources();

init();

glutMouseFunc(myMouseFunction);

glutKeyboardFunc(myKeyboardFunction);

glutReshapeFunc(myReshapeFunction);

glutDisplayFunc(myDisplayFunction);

glutTimerFunc(33, myTimerFunction, 0);

// glutFullScreen(); // make the screen fullscreen

glutSetCursor(GLUT\_CURSOR\_INFO); //Displays the type of cursor u want

glutMainLoop();

return 0;

}

void init(void)

{

glClearColor(5.0, 0.1, 0.0, 0.0); //background color of openGl window

// ^red, green, blue, alpha(opaquenss)

glMatrixMode(GL\_PROJECTION); //glMatrixMode ? specify which matrix is the current matrix ????

glLoadIdentity(); //glLoadIdentity ? replace the current matrix with the identity matrix ????

glOrtho(0.0, g\_Width, 0.0, g\_Height, -1.0, 1.0); //set (0,0) on left top

}

void myMouseFunction(int button, int state, int mouseX, int mouseY)

{

mouse\_X = mouseX;

mouse\_Y = mouseY;

if (button == 1 && state == 0) //Click

{

printf("mouse clicked \n");

printf("\nmouseX: %d mouseY: %d, State: %d", mouseX, mouseY, state);

}

}

void myKeyboardFunction(unsigned char key, int mouseX, int mouseY)

{

if (D > 15)

{

if (key == '2') {

D = D - 20;

}

}

if (D<500)

{

if (key == '8') {

D = D + 20;

}

}

if (D > 2 && D<550)

{

if (key == '4' && C>190) {

C = C - 5;

}

if (key == '6' && C<370) {

C = C + 5;

}

}

if (key == 's') {

//Start game

c\_speed = 5;

}

//Restart game

if (key == 'p') {

reset();

}

}

void myReshapeFunction(int width, int height)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

g\_Width = width;

g\_Height = height;

glViewport(0, 0, g\_Width, g\_Height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, g\_Width, g\_Height, 0.0, -1.0, 1.0); //change the (0,0) to top left

}

void myTimerFunction(int val)

{

glutTimerFunc(33, myTimerFunction, 0);

myDisplayFunction();

// car move up

{

if (c\_speed == 5)

{

//row 2 cars

if (Y > -200 && Y <= 620) {

Y += 3;

if (Y >= 615)

Y = -50;

}

//row 3 cars

if (car2\_y > -201 && car2\_y <= 620) {

car2\_y += 2;

if (car2\_y >= 615)

car2\_y = -200;

}

//row 1 cars

if (car3\_y >= -201 && car3\_y <= 600) {

car3\_y += 1.4;

if (car3\_y >= 615)

car3\_y = -150;

}

//row 2 cars

if (car4\_y > -51 && car4\_y <= 620) {

car4\_y += 2.3;

if (car4\_y >= 615)

car4\_y = -450;

} // row 3 cars

if (car5\_y > -501 && car5\_y <= 620) {

car5\_y += 1.7;

if (car5\_y >= 610)

car5\_y = -100;

}

// row 1 cars

if (car6\_y > -11 && car6\_y <= 630) {

car6\_y += 1.3;

if (car6\_y >= 622)

car6\_y = -8;

}

}

}

}

int i = 0;

void myDisplayFunction()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.2, 1.0);

glVertex2f(150, 0);

glVertex2f(150, 600);

glVertex2f(450, 600);

glVertex2f(450, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 3.2, 0.0);

glVertex2f(170, 0);

glVertex2f(170, 600);

glVertex2f(430, 600);

glVertex2f(430, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.0, 0.0);

glVertex2f(190, 0);

glVertex2f(190, 600);

glVertex2f(410, 600);

glVertex2f(410, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 6.2, 1.0);

glVertex2f(265, 0);

glVertex2f(265, 600);

glVertex2f(270, 600);

glVertex2f(270, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.0, 0.0);

glVertex2f(335, 0);

glVertex2f(335, 600);

glVertex2f(340, 600);

glVertex2f(340, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 6.2, 1.0);

glVertex2f(335, 0);

glVertex2f(335, 600);

glVertex2f(340, 600);

glVertex2f(340, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 6.2, 0.0);

glVertex2f(X, Y);

glVertex2f(X, Y + HT);

glVertex2f(X + (WD / 2), Y + (1.33\*HT));

glVertex2f(X + WD, Y + HT);

glVertex2f(X + WD, Y);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 3.2, 0.0);

glVertex2f(car2\_x, car2\_y);

glVertex2f(car2\_x, car2\_y + HT);

glVertex2f(car2\_x + (WD / 2), car2\_y + (1.33\*HT));

glVertex2f(car2\_x + WD, car2\_y + HT);

glVertex2f(car2\_x + WD, car2\_y);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.0, 3.0);

glVertex2f(car3\_x, car3\_y);

glVertex2f(car3\_x, car3\_y + HT);

glVertex2f(car3\_x + (WD / 2), car3\_y + (1.33\*HT));

glVertex2f(car3\_x + WD, car3\_y + HT);

glVertex2f(car3\_x + WD, car3\_y);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex2f(car4\_x, car4\_y);

glVertex2f(car4\_x, car4\_y + HT);

glVertex2f(car4\_x + (WD / 2), car4\_y + (1.33\*HT));

glVertex2f(car4\_x + WD, car4\_y + HT);

glVertex2f(car4\_x + WD, car4\_y);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(2.0, 0.0, 0.0);

glVertex2f(car5\_x, car5\_y);

glVertex2f(car5\_x, car5\_y + HT);

glVertex2f(car5\_x + (WD / 2), car5\_y + (1.33\*HT));

glVertex2f(car5\_x + WD, car5\_y + HT);

glVertex2f(car5\_x + WD, car5\_y);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(5.0, 1.3, 1.0);

glVertex2f(car6\_x, car6\_y);

glVertex2f(car6\_x, car6\_y + HT);

glVertex2f(car6\_x + (WD / 2), car6\_y + (1.33\*HT));

glVertex2f(car6\_x + WD, car6\_y + HT);

glVertex2f(car6\_x + WD, car6\_y);

glEnd();

// our van

glBegin(GL\_POLYGON);

glColor3f(0.0, 5.2, 1.0);

glVertex2f(C, D);

glVertex2f(C, D + HT);

glVertex2f(C + WD1, D + HT);

glVertex2f(C + WD1, D);

glVertex2f(C + (WD1 / 2), D - (HT / 3));

glEnd();

int r = 0;

// ||

if (D - (HT / 3) >= Y && D - (HT / 3) <= Y + (1.33\*HT)

&& C + (WD1 / 2) <= X + WD && C + (WD1 / 2) >= X) {

r = 1;

glColor3f(0.0, 6.2, 0.0);

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

if (Y <= 300 && Y >= 297) {

Scores += 1;

}

if (car2\_y <= 300 && car2\_y >= 297) {

Scores += 1;

}

if (car3\_y <= 300 && car3\_y >= 297) {

Scores += 1;

}

if (car4\_y <= 300 && car4\_y >= 297) {

Scores += 1;

}

if (car5\_y <= 300 && car5\_y >= 297) {

Scores += 1;

}

if (car6\_y <= 300 && car6\_y >= 297) {

Scores += 1;

}

if (D - (HT / 3) >= car2\_y && D - (HT / 3) <= car2\_y + (1.33\*HT) && C + (WD1 / 2) <= car2\_x + WD && C + (WD1 / 2) >= car2\_x) {

r = 1;

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

if (D - (HT / 3) >= car3\_y && D - (HT / 3) <= car3\_y + (1.33\*HT) && C + (WD1 / 2) <= car3\_x + WD && C + (WD1 / 2) >= car3\_x) {

r = 1;

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

if (D - (HT / 3) >= car4\_y && D - (HT / 3) <= car4\_y + (1.33\*HT) && C + (WD1 / 2) <= car4\_x + WD && C + (WD1 / 2) >= car4\_x) {

r = 1;

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

if (D - (HT / 3) >= car5\_y && D - (HT / 3) <= car5\_y + (1.33\*HT) && C + (WD1 / 2) <= car5\_x + WD && C + (WD1 / 2) >= car5\_x) {

r = 1;

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

if (D - (HT / 3) >= car6\_y && D - (HT / 3) <= car6\_y + (1.33\*HT) && C + (WD1 / 2) <= car6\_x + WD && C + (WD1 / 2) >= car6\_x) {

r = 1;

reset();

//PlaySound("starwars.wav", NULL, SND\_ASYNC | SND\_FILENAME | SND\_LOOP);

PlaySound(LPCWSTR("F:\\swish.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

}

draw(image1, 0, 100, 100, 90);

char score[1000];

glColor3f(1.3, 2.2, 0.9);

renderBitmapString(5, 130, "Scores:");

//renderBitmapString(80, 130, itoa(Scores, score, 10));

//renderBitmapString(200, 20, itoa(player2, score, 10));

renderBitmapString(2, 70, "Try to Avoid .");

renderBitmapString(2, 100, "Incoming Cars.");

renderBitmapString(2, 40, "Use keys 2,4,6,8.");

//renderBitmapString(10, 40, "Player 1 uses arrow keys, player 2 uses q & z");

glColor3f(0.0, 1.2, 2.0);

if (r == 1) {

renderBitmapString(100, 200, "GAME OVER, Welldone! You Deserve a SAMOSA");

glColor3f(1.0, 0.0, 0.0);

renderBitmapString(230, 230, "From SIR SHAMAYL");

}

glutSwapBuffers(); //brings the openGl window on the front

//printf("\n.");

}