

### Source Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <graphics.h>

#include <conio.h>


//global variables

int x,y,xline1,xline2,trackLength,speed,readhighscore=0;

long int score;

char msg[100];

struct enemycarPosition{

int x1,x2,y1,y2;

} e;

struct mycarPosition{

int x1,x2,y1,y2;

} m;

struct enemycarPosition1 {

int x1,x2,y1,y2;

} e1;


struct highsc{

char name[20];
```

```

long int hscore;

}h[5];


struct colors{

int bgcolor;

int textcolor;

int ingame_bgcolor1;

int ingame_bgcolor2;

int ingame_textcolor;


}c;

int carShape;

struct previous_player{

char name[20];

int len;

}pp;


//user-defined functions

void setgraphics();//set the graphics

void startgame();//start game

void loadingscreen();//show loading screen

void mainmenu();//show main menu

void draw(long int,int);//draw path, display score and speed

void mycar(int);//spawn player car accord to car position

void enemycar(int *);//spawn and move enemy car 1

void enemycar1(int *);//spawn and move enemy car 2

```

```
int collisionDetection(struct mycarPosition mc,struct enemycarPosition ec);//check collision detection
of enemy car 1 with player car

int collisionDetection1(struct mycarPosition mc,struct enemycarPosition1 ec);//check collision
detection of enemy car 2 with player car

int scoreincrease(long int,int);//increase player car speed accord to score

void pause();//show paused screen

void gameover(long int);//show game over screen

void displayControls();//display player controls

void showHighscores();//display highscores

void readHighscore();//read highscores from file

void showCredits();//show credits screen

void takeuserdetails(long int);//take user details

int updateHighscore();//update the highscores in file

int checkifscoredHigh(long int);//check if player scored high

int readgamedata();//read current game data

void defaultColor();//set default color

void colorSelect(int ,int);//set color

int updategamedata();//update current game data

void customize();//show customize screen

void customizeCar();//show customize car screen

void customizeTheme();//show customize theme screen

void resetsettings();//reset all settings to default

void resetgamedata();//reset game data to default

void resetHighscore();//reset highscore to 0
```

```

int main(){

setgraphics();

x=getmaxx();
y=getmaxy();

if (readgamedata()==0){//if game is launched first time
resetgamedata();

} else

defaultColor();

loadingscreen();

mainmenu();

closegraph();

return 0;

}

```

```

void defaultColor(){

setbkcolor(c.bgcolor);

setcolor(c.textcolor);

}

void colorSelect(int a,int b){

setbkcolor(a);

setcolor(b);

}

```

```

void setgraphics() {

```

```

int gd=DETECT,gm;

initgraph(&gd,&gm,"C://turboc3//bgi");

}


void loadingscreen(){

int i=0;

cleardevice();

settextstyle(4,0,5);

outtextxy(x/2-200,y/2,"A 2D Car Racing Game");

delay(1000);


settextstyle(3,0,1);

outtextxy(x/2-100,y-120,"LOADING....");


while (i!=200){

rectangle(x/2-100,y-85,x/2-100+i,y-75);

rectangle(x/2-100,y-85,x/2+100,y-75);

i++;

delay(10);

}

}


void mainmenu(){

int selection=0;


cleardevice();

```

```
settextstyle(3,0,3);
```

```
defaultColor();
```

```
while(1){
```

```
cleardevice();
```

```
score=0;//initial score
```

```
speed=1;//initial speed
```

```
outtextxy(x/2-100,100,"Play");
```

```
outtextxy(x/2-100,150,"Controls");
```

```
outtextxy(x/2-100,200,"HighScores");
```

```
outtextxy(x/2-100,250,"Customize");
```

```
outtextxy(x/2-100,300,"Credits");
```

```
outtextxy(x/2-100,350,"Exit");
```

```
circle(x/2-110,125,5);
```

```
circle(x/2-110,175,5);
```

```
circle(x/2-110,225,5);
```

```
circle(x/2-110,275,5);
```

```
circle(x/2-110,325,5);
```

```
circle(x/2-110,375,5);
```

```
setfillstyle(SOLID_FILL,c.textcolor);
```

```
floodfill(x/2-110,125+50*selection,c.textcolor);
```

```
switch(getch()){
```

```
case 'W':
```

```
case 'w':
```

```
if (selection==0)
```

```
selection=5;
```

```
else
```

```
selection--;
```

```
break;
```

```
case 's':
```

```
case 'S':
```

```
if (selection==5)
```

```
selection=0;
```

```
else
```

```
selection++;
```

```
break;
```

```
case 13://enterkey - CR
```

```
if (selection==0){
```

```
startgame();
```

```
}
```

```
else if(selection==1){
```

```
displayControls();
```

```
}
```

```
else if(selection==2){
```

```
showHighscores();
```

```

    }

    else if(selection==3){

        customize();

    }

    else if (selection==4){

        showCredits();

    }

    else {

        exit(0);

    }


    break;

    default:

    break;

    }


}

}

}

void startgame(){

    int i,mycarLocation,collision=0,activateEnemy1=0;

    int *enemycarPos,*enemycarPos1;


    cleardevice();

    randomize();

```



```

defaultColor();

for(i=3;i!=0;i--){

cleardevice();

sprintf(msg,"Starting in %d...",i);

outtextxy(x/2-50,100,msg);

delay(500);

}

colorSelect(c.ingame_bgcolor1,c.ingame_textcolor);

settextstyle(0,0,1);

xline1=x/2-75;

xline2=x/2+75;

trackLength=xline2-xline1;

mycarLocation=random(100)%3;

*enemycarPos=0;

*enemycarPos1=0;

while(1){

cleardevice();

draw(score,speed);

enemycar(enemycarPos);

if (activateEnemy1==1){

enemycar1(enemycarPos1);

*enemycarPos1=*enemycarPos1+speed+1;

collision=collisionDetection(m,e)+collisionDetection1(m,e1);

```

```
}else  
  
collision=collisionDetection(m,e);  
  
mycar(mycarLocation);  
  
  
  
if (collision==1){  
gameover(score);  
}  
  
  
  
if(kbhit()){  
switch(getch()){  
case 'A':  
case 'a':  
if(mycarLocation!=0)  
mycarLocation--;  
break;  
case 'd':  
case 'D':  
if(mycarLocation!=2)  
mycarLocation++;  
break;  
case 'p':  
case 'P':  
pause();
```

```

break;

case 'X':

case 'x':

exit(0);

break;

default:

break;


}

}

*enemycarPos=*enemycarPos+speed+1;

score=score+speed*1;

if (*enemycarPos>y/2)

activateEnemy1=1;

speed=speedincrease(score,speed);//increase speed

delay(10);


}

}


void mycar(int carLocation){

m.x1=xline1+(trackLength/3*carLocation)+3;

m.x2=xline1+(trackLength/3*(carLocation+1))-3;

m.y1=y-3;

m.y2=y-50;

```

```

if (carShape==0){//default shape

rectangle(m.x1,m.y1-35,m.x2,m.y1);//big block

rectangle(m.x1+10,m.y2+13,m.x2-10,m.y1);//middle block

rectangle(m.x1+5,m.y2,m.x2-5,m.y2+13);//front block

setfillstyle(SOLID_FILL,c.ingame_textcolor);

bar(m.x1+3,m.y2+3,m.x1+5,m.y2+13);//left tyre

bar(m.x2-3,m.y2+3,m.x2-5,m.y2+13);//right tyre

}

else if (carShape==1){//rectangle shape

rectangle(m.x1,m.y1,m.x2,m.y2);

} else if(carShape==2){//bar

setfillstyle(SOLID_FILL,c.ingame_textcolor);

bar(m.x1,m.y1,m.x2,m.y2);

} else if (carShape==3){//ellipse

ellipse((m.x1+m.x2)/2,(m.y1+m.y2)/2,0,360,(m.x2-m.x1)/2,(m.y1-m.y2)/2);

} else{//defaultShape

rectangle(m.x1,m.y1-35,m.x2,m.y1);//big block

```

```

rectangle(m.x1+10,m.y2+13,m.x2-10,m.y1);//middle block

rectangle(m.x1+5,m.y2,m.x2-5,m.y2+13);//front block

setfillstyle(SOLID_FILL,c.ingame_textcolor);

bar(m.x1+3,m.y2+3,m.x1+5,m.y2+13);//left tyre

bar(m.x2-3,m.y2+3,m.x2-5,m.y2+13);//right tyre

}

}

void enemycar(int *i){

static int enemycarLoc;

if (*i==0){

enemycarLoc=random(100)%3;

e.x1=xline1+3+(trackLength/3)*enemycarLoc;

e.x2=xline1+(trackLength/3*(enemycarLoc+1))-3;

}

e.y1=*i+3;

e.y2=*i+48;

if (carShape==0){//default shape

rectangle(e.x1,e.y1,e.x2,e.y1+33);//big block

rectangle(e.x1+5,e.y1+33,e.x2-5,e.y2);//front block

setfillstyle(SOLID_FILL,c.ingame_textcolor);

bar(e.x1+3,e.y1+33,e.x1+5,e.y2-3);//left tyre

bar(e.x2-5,e.y1+33,e.x2-3,e.y2-3);//right tyre

}

else if (carShape==1){//rectangle shape

```

```

rectangle(e.x1,e.y1,e.x2,e.y2);

} else if(carShape==2){//bar

setfillstyle(SOLID_FILL,c.ingame_textcolor);
bar(e.x1,e.y1,e.x2,e.y2);

} else if (carShape==3){//ellipse

ellipse((e.x1+e.x2)/2,(e.y1+e.y2)/2,0,360,(e.x2-e.x1)/2,(e.y2-e.y1)/2);

} else{//defaultShape

rectangle(e.x1,e.y1,e.x2,e.y1+33);//big block
rectangle(e.x1+5,e.y1+33,e.x2-5,e.y2);//front block
setfillstyle(SOLID_FILL,c.ingame_textcolor);
bar(e.x1+3,e.y1+33,e.x1+5,e.y2-3);//left tyre
bar(e.x2-5,e.y1+33,e.x2-3,e.y2-3);//right tyre

}

if(e.y1>y)
*i=-speed-1;
}

```

```

void enemycar1(int *j){
static int enemycarLoc1;

if (*j==0){
enemycarLoc1=random(100)%3;

e1.x1=xline1+3+(trackLength/3)*enemycarLoc1;
e1.x2=xline1+(trackLength/3*(enemycarLoc1+1))-3;
}

e1.y1=*j+3;
e1.y2=*j+48;

if (carShape==0){//default shape

rectangle(e1.x1,e1.y1,e1.x2,e1.y1+33);//big block
rectangle(e1.x1+5,e1.y1+33,e1.x2-5,e1.y2);//front block
setfillstyle(SOLID_FILL,c.ingame_textcolor);
bar(e1.x1+3,e1.y1+33,e1.x1+5,e1.y2-3);//left tyre
bar(e1.x2-5,e1.y1+33,e1.x2-3,e1.y2-3);//right tyre
}

else if (carShape==1){//rectangle shape

rectangle(e1.x1,e1.y1,e1.x2,e1.y2);

} else if(carShape==2){//bar

setfillstyle(SOLID_FILL,c.ingame_textcolor);
bar(e1.x1,e1.y1,e1.x2,e1.y2);

```

```

} else if (carShape==3){//ellipse

ellipse((e1.x1+e1.x2)/2,(e1.y1+e1.y2)/2,0,360,(e1.x2-e1.x1)/2,(e1.y2-e1.y1)/2);

} else{//defaultShape

rectangle(e1.x1,e1.y1,e1.x2,e1.y1+33);//big block
rectangle(e1.x1+5,e1.y1+33,e1.x2-5,e1.y2);//front block
setfillstyle(SOLID_FILL,c.ingame_textcolor);
bar(e1.x1+3,e1.y1+33,e1.x1+5,e1.y2-3);//left tyre
bar(e1.x2-5,e1.y1+33,e1.x2-3,e1.y2-3);//right tyre

}

if(e1.y1>y)
*j=-speed-1;
}

int collisionDetection(struct mycarPosition mc,struct enemycarPosition ec ){
if (ec.x1>mc.x2){
return 0;
}
else{
if(ec.x2<mc.x1) {
return 0;
} else{

```



```

if(ec.y1>mc.y1){
return 0;

}else{

if(ec.y2<mc.y2){
return 0;

}else{
return 1;

}

}

}

}

}

}

```

```

int collisionDetection1(struct mycarPosition mc,struct enemycarPosition1 ec ){

if (ec.x1>mc.x2){

return 0;

}

else{

if(ec.x2<mc.x1) {

return 0;

}else{

if(ec.y1>mc.y1){

return 0;

}else{

if(ec.y2<mc.y2){

```

```
return 0;
```

```
}else{
```

```
return 1;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void draw(long int sc,int sp){
```

```
line(xline1,0,xline1,y);
```

```
line(xline2,0,xline2,y);
```

```
sprintf(msg, "Score : %ld ",sc);
```

```
outtextxy(xline2+20,y/2,msg);
```

```
sprintf(msg,"Speed: %d",sp);
```

```
outtextxy(xline1-100,y/2,msg);
```

```
}
```

```
int speedincrease(long int sc ,int sp){
```

```
switch(sp){
```

```
case 1:
```

```
if (sc>1000)
```

```
sp++;
```

```
break;

case 2:

if (sc>2000)

sp++;

break;

case 3:

if (sc>4000)

sp++;

break;

case 4:

if (sc>8000)

sp++;

break;

case 5:

if (sc>16000)

sp++;

break;

case 6:

if (sc>32000)

sp++;

break;

case 7:

if (sc>64000)

sp++;

break;

case 8:
```

```

if (sc>120000)

sp++;

break;

case 9:

if (sc>240000)

sp++;

break;

default:

break;

}

return sp;


}


void pause(){

int the_end=0;


setfillstyle(SOLID_FILL,c.ingame_bgcolor2);//darkgray


bar(xline1,y/2-210,xline2,y/2-180);


//pause symbol

line(xline1+50,y/2-200,xline1+50,y/2-190);//straight line

line(xline1+50,y/2-200,xline1+60,y/2-195);//back slash line

```

```
line(xline1+50,y/2-190,xline1+60,y/2-195);//front slash line
```

```
outtextxy(xline1+70,y/2-200,"Paused");
```

```
while(!the_end){
```

```
switch(getch()){
```

```
case 'p':
```

```
case 'P':
```

```
the_end=1;
```

```
break;
```

```
case 'm':
```

```
case 'M':
```

```
mainmenu();
```

```
break;
```

```
case 'x':
```

```
case 'X':
```

```
exit(0);
```

```
break;
```

```
default:
```

```
break;
```

```
}
```

```
}
```

```
}
```

```
void gameover(long int sc){
```

```
setfillstyle(SOLID_FILL,c.ingame_bgcolor2);
```

```

bar(xline1,y/2-210,xline2,y/2-180);

outtextxy(xline1+40,y/2-200,"Game Over");

delay(10);

getch();

cleardevice();

sprintf(msg,"Score: %ld",score);

outtextxy(10,100,msg);

if (checkifscoredHigh(sc)==1) {

outtextxy(10,130,"Congrats! You made your way to the Highscore Table.");

outtextxy(10,150,"Press any key to continue.");

getch();

takeuserdetails(score);

}

else{

outtextxy(10,130,"Sorry! You couldn't made your way to the Highscore Table.");

outtextxy(10,150,"Press any key to return to mainmenu.");

getch();

}

mainmenu();

}

void displayControls(){

cleardevice();

settextstyle(3,0,4);

```

```

outtextxy(x/2-100,100,"CONTROLS");

settextstyle(3,0,2);

outtextxy(100,150,"Upward");

outtextxy(400,150,"W");

outtextxy(100,175,"Downward");

outtextxy(400,175,"S");

outtextxy(100,200,"Left");

outtextxy(400,200,"A");

outtextxy(100,225,"Right");

outtextxy(400,225,"D");

outtextxy(100,250,"Pause/Resume");

outtextxy(400,250,"P");

outtextxy(100,275,"Direct Exit (while playing)");

outtextxy(400,275,"X");

outtextxy(100,300,"Mainmenu (when paused)" );

outtextxy(400,300,"M");

outtextxy(100,400,"Press any key to return to mainmenu.");


getch();

mainmenu();


}


void showCredits(){

cleardevice();

settextstyle(3,0,4);

```

```

outtextxy(x/2-100,100,"CREDITS");

settextstyle(3,0,2);

outtextxy(100,150,"This game is developed by:");

outtextxy(125,175,"-Rohan Dhimal");

outtextxy(125,200,"-Prahlad Neupane");

outtextxy(125,225,"-Puskar Humagain");

outtextxy(125,250,"-Bibek Dhital");

outtextxy(100,275,"Special thanks to: Kamal Rana Sir");


outtextxy(100,350,"Press any key to return to mainmenu.");


getch();

mainmenu();

}


void showHighscores(){

int i;

cleardevice();

if (readhighscore==0)

readHighscore();

settextstyle(3,0,4);

outtextxy(x/2-100,100,"HighScore");

```



```

settextstyle(3,0,3);

for (i=0;i<5;i++){
    sprintf(msg,"%d. %s",i+1,h[i].name);
    outtextxy(100,150+i*25,msg);
    sprintf(msg,"%ld",h[i].hscore);
    outtextxy(400,150+i*25,msg);
}

outtextxy(100,150+i*25+100,"Press any key to return to mainmenu.");
getch();
}

void readHighscore(){

size_t elements_read;

FILE *fp;

fp=fopen("highscore.bin","rb");

if (fp==NULL){
    resetHighscore();
} else{

elements_read=fread(&h,sizeof(struct highsc),5,fp);

fclose(fp);

if (elements_read==0){
    outtextxy(x/2-50,y/2,"Error");
}
}
}

```

```

    getch();

    mainmenu();

}

readhighscore=1;

}

}

void takeuserdetails(long int sc){

char inputbuf[20];

int input_pos = pp.len;

char c;

int the_end = 0;

strcpy(inputbuf,pp.name);

do {

cleardevice();

outtextxy(10,140,"Enter your name:");

bar(10,160,300,200);

outtextxy (15,175, inputbuf);

c = getch();

switch (c)

{

```

```

case 8: /* backspace */

if (input_pos)

{

input_pos--;

inputbuf[input_pos] = 0;

}

break;

case 13: /* return */

the_end = 1;

break;

case 27: /* Escape = Abort */

inputbuf[0] = 0;

the_end = 1;

break;

default:

if (input_pos < 20-1 && c >= ' ' && c <= '~')

{

inputbuf[input_pos] = c;

input_pos++;

inputbuf[input_pos] = 0;

}

}

} while (!the_end);

strcpy(pp.name,inputbuf);

updategamedata();

```

```

h[4].hscore=sc;

strcpy(h[4].name,inputbuf);

if (updateHighscore()==0){
    outtextxy(10,220,"Unable to update Highscore.");
    outtextxy(10,240,"Press any key to return to mainmenu.");
} else {
    outtextxy(10,220,"Highscore successfully updated.");
    outtextxy(10,240,"Press any key to return to mainmenu.");

}

getch();

}

int updateHighscore(){
    char ch[20];
    int i,j=4;
    long int k;
    size_t elements_written;
    FILE *fp;

    for(i=3;i>0,j>0;i--){
        if (h[j].hscore>h[i].hscore){

```

```

k=h[j].hscore;

h[j].hscore=h[i].hscore;

h[i].hscore=k;


strcpy(ch,h[j].name);

strcpy(h[j].name,h[i].name);

strcpy(h[i].name,ch);


j=i;


} else {

break;

}

}


fp=fopen("highscore.bin","wb");

if (fp==NULL){

return 0;

} else {

elements_written=fwrite(&h,sizeof(struct highsc),5,fp);

fclose(fp);

if (elements_written==0){

return 0;

} else {

return 1;

```

```
}
```

```
}
```

```
}
```

```
int checkifscoredHigh(long int sc){
```

```
if (readhighscore==0)
```

```
readHighscore();
```

```
if (sc>h[4].hscore)
```

```
return 1;
```

```
else
```

```
return 0;
```

```
}
```

```
int readgamedata(){
```

```
size_t elements_written;
```

```
int n=0;
```

```
FILE *fp;
```

```
fp=fopen("gamedata.bin","rb");
```

```
if (fp==NULL){
```

```
return 0;
```

```
} else{
```

```
elements_written=fread(&c,sizeof(struct colors),1,fp);
```

```
if (elements_written==0){
```

```

return 0;

} else {

n++;

}

elements_written=fread(&carShape,sizeof(int),1,fp);

if (elements_written==0){

return 0;

} else {

n++;

}

elements_written=fread(&pp,sizeof(struct previous_player),1,fp);

if (elements_written==0){

return 0;

} else {

n++;

}

if (n==3){

return 1;

} else {

return 0;

}

}

```

```
}
```

```
int updategamedata(){
```

```
FILE *fp;
```

```
fp=fopen("gamedata.bin","wb");
```

```
if (fp==NULL){
```

```
return 0;
```

```
}else{
```

```
fwrite(&c,sizeof(struct colors),1,fp);
```

```
fwrite(&carShape,sizeof(int),1,fp);
```

```
fwrite(&pp,sizeof(struct previous_player),1,fp);
```

```
return 1;
```

```
}
```

```
}
```

```
void customize(){
```

```
int i=0;
```

```
while(1){
```

```
defaultColor();
```

```
cleardevice();
```

```
settextstyle(3,0,4);
```

```
outtextxy(x/2-50,100,"Customize");
```



```
settextstyle(3,0,3);  
outtextxy(125,150,"Theme");  
outtextxy(125,200,"Car Shape");  
outtextxy(125,250,"Reset all Settings");  
outtextxy(125,300,"Return");  
circle(100,165,3);  
circle(100,215,3);  
circle(100,265,3);  
circle(100,315,3);  
setfillstyle(SOLID_FILL,c.textcolor);  
floodfill(100,165+i*50,c.textcolor);
```

```
switch(getch()){  
case 'W':  
case 'w':  
if(i!=0)  
i--;  
break;  
case 's':  
case 'S':  
if(i!=3)  
i++;  
break;  
case 'm':  
case 'M':
```

```
mainmenu();

break;

case 13:

if (i==0)

customizeTheme();

else if(i==1)

customizeCar();

else if (i==2){

resetsettings();

}


else

mainmenu();

break;

default:

break;

}

}

}

void customizeTheme(){

struct colors c1;

int i=0,j;
```

```

c1=c;

while(1){
cleardevice();
defaultColor();
settextstyle(3,0,4);
outtextxy(x/2-50,100,"Theme");

settextstyle(3,0,3);
outtextxy(125,150,"Dark Ocean");
outtextxy(125,200,"Gray Mountain");
outtextxy(125,250,"Return");
circle(100,165,3);
circle(100,215,3);
circle(100,265,3);
setfillstyle(SOLID_FILL,c.textcolor);
floodfill(100,165+i*50,c.textcolor);

if (c.bgcolor==1){
outtextxy(400,150,"(Current Theme)");
}else{
outtextxy(400,200,"(Current Theme)");
}

switch(getch()){

```

```
case 'W':  
  
case 'w':  
  
if(i!=0)  
  
i--;  
  
break;  
  
case 's':  
  
case 'S':  
  
if(i!=2)  
  
i++;  
  
break;  
  
case 'm':  
  
case 'M':  
  
mainmenu();  
  
break;  
  
case 13:  
  
if (i==2){  
  
customize();  
  
} else{  
  
outtextxy(100,400,"Updating");  
  
for (j=0;j<5;j++){  
  
outtextxy(300+j*10,400,".");  
  
delay(1000);  
  
}  
  
if (i==0){
```

```

c.bgcolor=1;//blue

c.textcolor=15;//white

c.ingame_bgcolor1=0;//black

c.ingame_bgcolor2=8;//darkgray

c.ingame_textcolor=15;//white


} else if (i==1){


c.bgcolor=15;//WHITE

c.textcolor=1;//BLUE

c.ingame_bgcolor1=7;//lightgray

c.ingame_bgcolor2=8;//darkgray

c.ingame_textcolor=15;//white


}


if (updategamedata()==0){

setcolor(RED);

outtextxy(100,425,"Error try again.");

c=c1;

} else{

outtextxy(100,425,"Successfully updated. Press any key.");

}

getch();

break;

default:

```

```
break;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void customizeCar() {
```

```
int carshape;
```

```
int i=0,j;
```

```
carshape=carShape;
```

```
while(1){
```

```
cleardevice();
```

```
defaultColor();
```

```
settextstyle(3,0,4);
```

```
outtextxy(x/2-50,100,"Car Shape");
```

```
settextstyle(3,0,3);
```

```
outtextxy(125,150,"Normal");
```

```
outtextxy(125,200,"Rectangle");
```

```
outtextxy(125,250,"Block");
```

```
outtextxy(125,300,"Circle");
```

```
outtextxy(125,350,"Return");
```

```

circle(100,165,3);

circle(100,215,3);

circle(100,265,3);

circle(100,315,3);

circle(100,365,3);


setfillstyle(SOLID_FILL,c.textcolor);

floodfill(100,165+i*50,c.textcolor);


outtextxy(400,150+carShape*50,"(Current Shape)");


switch(getch()){
case 'W':

case 'w':

if(i!=0)

i--;

break;

case 's':

case 'S':

if(i!=4)

i++;

break;

case 'm':

case 'M':

mainmenu();

```

```

break;

case 13:

if (i==4){

customize();

} else {

outtextxy(100,400,"Updating");

for (j=0;j<5;j++){

outtextxy(300+j*10,400,".");

delay(1000);

}

carShape=i;

if (updategamedata()==0){

setcolor(RED);

outtextxy(100,425,"Error try again.");

carShape=carshape;

} else {

outtextxy(100,425,"Successfully updated. Press any key.");

}

getch();

}

break;

default:

break;

}

}

```



```
}
```

```
void resetsettings(){
```

```
int i;
```

```
cleardevice();
```

```
defaultColor();
```

```
outtextxy(100,100,"Resetting all settings.");
```

```
for (i=0;i<5;i++){
```

```
outtextxy(350+i*25,100,".");
```

```
delay(1000);
```

```
}
```

```
resetHighscore();
```

```
resetgamedata();
```

```
outtextxy(100,125,"Successfully updated. Press any key.");
```

```
getch();
```

```
}
```

```
void resetgamedata()
```

```
{
```

```
c.bgcolor=1;//blue
```

```
c.textcolor=15;//white
```

```
c.ingame_bgcolor1=0;//black  
c.ingame_bgcolor2=8;//darkgray  
c.ingame_textcolor=15;//white
```

```
carShape=0;//regular shape
```

```
strcpy(pp.name,"Player");  
pp.len=6;
```

```
updategamedata();  
}
```

```
void resetHighscore(){  
    int i;  
    char name[20];  
    FILE *fp;
```

```
    for (i=0;i<5;i++)  
    {  
        strcpy(h[i].name,"Player");  
        h[i].hscore=0;  
    }
```

```
    fp=fopen("highscore.bin","wb");  
    if (fp!=NULL){
```

```
fwrite(&h,sizeof(struct highsc),5,fp);  
fclose(fp);  
  
}  
}
```