#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<stdlib.h>

#include<time.h>

int level=0,ec\_iwd,rep=0,pr=0,ec\_iht,lc\_x,rc\_x,rc\_y,frc\_x,frc\_y,difx,dify,large,carwidth,posn,carheight,left\_x,midx,right\_x,maxx,i,maxy,horizon\_y,dx,dy,l;

double xfact,yfact,incx,incy,dxx,treeheight,treewidth,ftreeheight,ftreewidth,dyy,rt\_y,rt\_x,incr\_rx,incr\_ry,incr\_lx,incr\_ly,r\_ht,l\_ht,r\_wd,l\_wd,lt\_x,lt\_y;

int set()

{

maxy=getmaxy();

maxx=getmaxx();

horizon\_y=maxy/2;

midx=maxx/2;

left\_x=midx/1.5;

right\_x=midx\*1.3;

treeheight=horizon\_y/4;

treewidth=maxx/64;

//setting the dda values for right tree

rt\_x=right\_x;

rt\_y=horizon\_y;

dx=maxx-rt\_x;

dy=maxy-rt\_y;

l=(dx>dy)?dx:dy;

dxx=(double)dx/l;

dyy=(double)dy/l;

//setting up tree magnification variables

ftreeheight=treeheight\*3;

ftreewidth=treewidth\*1;

incr\_rx=(ftreewidth-treewidth)/(maxx-right\_x);

incr\_ry=(ftreeheight-treeheight)/(maxy-horizon\_y);

incr\_lx=(ftreewidth-treewidth)/(maxx-right\_x);

incr\_ly=(ftreeheight-treeheight)/(maxy-horizon\_y);

r\_ht=treeheight;

r\_wd=treewidth;

l\_ht=(ftreeheight-treeheight)/2;

l\_wd=(ftreewidth-treewidth)/2;

//setting initial co-ordinates of left tree

lt\_x=left\_x/2;

lt\_y=horizon\_y+((maxy-horizon\_y)/2);

//Setting players car width and height

carwidth=maxx/8;

carheight=horizon\_y/3;

posn=maxx/2;

//setting up right side enemys car

rc\_x=(midx+right\_x)/2;

lc\_x=(left\_x+midx)/2;

rc\_y=horizon\_y;

frc\_x=midx+(midx/2);

frc\_y=maxy;

difx=frc\_x-rc\_x;

dify=frc\_y-rc\_y;

large=(difx>dify)?difx:dify;

incx=(float)difx/large;

incy=(float)dify/large;

//setting up enemy cars height and width

ec\_iwd=carwidth/3;

ec\_iht=carheight/10;

//setting up magnification variables of enemy car

xfact=(double)((carwidth/1.5)-ec\_iwd)/(frc\_x-rc\_x);

yfact=(double)((carheight\*1)-ec\_iht)/(maxy-horizon\_y);

//printf("%d / %d, %2f",caheight-ec\_iht,maxy-horizon\_y,yfact);

return 0;

}

int drawroad()

{

line(0,horizon\_y,maxx,horizon\_y);

line(midx,horizon\_y,midx,maxy);

line(left\_x,horizon\_y,0,maxy);

line(right\_x,horizon\_y,maxx,maxy);

return 0;

}

int drawtree(int x,int y,int clr,double height,double width)

{

float floor,one\_inner,baseheight,bspace;

int h=height,w=width;

if(clr==1)

{

setcolor(15);

}

else

{

setcolor(0);

}

//rectangle(x-width,y-height,x+width,y);

bspace=(w\*2)/3;

baseheight=h/4;

one\_inner=w\*2/3;

floor=h/8;

line(x-w+bspace,y,x+w-bspace,y);

line(x-w+bspace,y,x-w+bspace,y-baseheight);

line(x-w+bspace,y-baseheight,x-w,y-baseheight);

line(x-w,y-baseheight,x-w+one\_inner,y-baseheight-2\*floor);

line(x-w+one\_inner,y-baseheight-2\*floor,x-w+(w/10),y-baseheight-floor);

line(x-w+(w/10),y-baseheight-floor,x,y-h);

line(x,y-h,x+w-(w/10),y-baseheight-floor);

line(x+w-(w/10),y-baseheight-floor,x+w-one\_inner,y-baseheight-2\*floor);

line(x+w-one\_inner,y-baseheight-2\*floor,x+w,y-baseheight);

line(x+w,y-baseheight,x+w-bspace,y-baseheight);

line(x+w-bspace,y,x+w-bspace,y-baseheight);

setcolor(15);

return 0;

}

int sidetrees()

{

rt\_x+=dxx;

rt\_y+=dyy;

lt\_x-=dxx;

lt\_y+=dyy;

r\_ht+=incr\_ry;

r\_wd+=incr\_rx;

l\_ht+=incr\_ly;

l\_wd+=incr\_lx;

incr\_rx+=0.001;

incr\_ry+=0.001;

incr\_lx+=0.001;

incr\_ly+=0.001;

if(rt\_x>=(maxx-r\_wd))

{

rt\_x=right\_x;

rt\_y=horizon\_y;

r\_ht=treeheight;

r\_wd=treewidth;

incr\_rx=(ftreewidth-treewidth)/(maxx-right\_x);

incr\_ry=(ftreeheight-treeheight)/(maxy-horizon\_y);

}

if(lt\_x<=l\_wd)

{

lt\_x=left\_x;

lt\_y=horizon\_y;

l\_ht=treeheight;

l\_wd=treewidth;

incr\_lx=(ftreewidth-treewidth)/(maxx-right\_x);

incr\_ly=(ftreeheight-treeheight)/(maxy-horizon\_y);

}

return 0;

}

//Drawing our cAAAR

int mycar(int x,int clr)

{

int tirewidth,rx,lx,y,h,w;

float dy,aarsa\_y,l\_aarsa\_x,r\_aarsa\_x;

if(clr==1)

{setcolor(15);}

else

{setcolor(0);}

//outtextxy(20,2,"Your points:%d %d %d",level,rep,pr);

//rectangle(x-carwidth,maxy-carheight-5,x+carwidth,maxy-5);

y=maxy-5;

h=carheight;

w=carwidth;

//rectangle(x-w,y-h,x+w,y);

rectangle(0,0,getmaxx(),getmaxy());

dy=y-(h\*2/3);

rx=x-(w\*3/4);

lx=x+(w\*3/4);

tirewidth=w/5;

r\_aarsa\_x=x+w;

aarsa\_y=y-(h/3);

l\_aarsa\_x=x-w;

line(r\_aarsa\_x,aarsa\_y,r\_aarsa\_x+tirewidth,aarsa\_y-tirewidth);

line(r\_aarsa\_x+tirewidth,aarsa\_y-tirewidth,r\_aarsa\_x,aarsa\_y-tirewidth);

line(l\_aarsa\_x,aarsa\_y,l\_aarsa\_x-tirewidth,aarsa\_y-tirewidth);

line(l\_aarsa\_x-tirewidth,aarsa\_y-tirewidth,l\_aarsa\_x,aarsa\_y-tirewidth);

line(rx,dy,lx,dy);

line(x-w,y-(h/3),rx,dy);

line(x+w,y-(h/3),lx,dy);

line(x-w,y-(h/3),rx,y-h);

line(x+w,y-(h/3),lx,y-h);

line(rx,dy,rx,y-h);

line(lx,dy,lx,y-h);

line(rx,y-h,lx,y-h);

//transformers

line(x-w,y-(h/10),x-w,y-(h/3));

line(x-w,y-(h/10),x-w+(w/10),y-(h/10));

line(x-w+(w/10),y-(h/10),x-w+(w/10),y);

line(x-w+(w/10),y,x-w+(w/10)+tirewidth,y);

line(x-w+(w/10)+tirewidth,y,x-w+(w/10)+tirewidth,y-(h/10));

line(x-w+(w/10)+tirewidth,y-(h/10),x+w-(w/10)-tirewidth,y-(h/10));

// transformers 2

line(x+w,y-(h/10),x+w,y-(h/3));

line(x+w,y-(h/10),x+w-(w/10),y-(h/10));

line(x+w-(w/10),y-(h/10),x+w-(w/10),y);

line(x+w-(w/10),y,x+w-(w/10)-tirewidth,y);

line(x+w-(w/10)-tirewidth,y,x+w-(w/10)-tirewidth,y-(h/10));

line(x+w,y-(h/10),x-w,y-(h/10));

line(x+w,y-(h/3),x-w,y-(h/3));

setcolor(15);

return 0;

}

int draw\_enemycar(int x, int y,int height,int width,int clr)

{

int tireheight,tirewidth,dy,h,w;

float vargap,rgap,topgap;

if(clr==1)

{setcolor(15);}

else

{setcolor(0);}

//rectangle(x-width,y-height,x+width,y);

h=height;

w=width;

//rectangle(x-w,y-h,x+w,y);

rectangle(0,0,getmaxx(),getmaxy());

dy=h/2;

topgap=(w\*40/100);

rgap=(w\*25/100);

vargap=(w\*45/100);

tirewidth=w/5;

tireheight=h/7;

// windshield

line(x-w+rgap,y-((h\*2)/3),x+w-rgap,y-((h\*2)/3));

line(x-w+rgap,y-((h\*2)/3),x-w+vargap,y-h+(h/10));

line(x-w+vargap,y-h+(h/10),x+w-vargap,y-h+(h/10));

line(x+w-vargap,y-h+(h/10),x+w-rgap,y-((h\*2)/3));

//headlights

circle(x-w+rgap,y-dy,(tireheight/2));

circle(x+w-rgap,y-dy,(tireheight/2));

//carborator stuff

rectangle(x-w+rgap+(tireheight/2),y-(tireheight\*3),x+w-rgap-(tireheight/2),y-tireheight-(h/10));

line(x-w+topgap,y-h,x-w,y-dy);

line(x-w+topgap,y-h,x+w-topgap,y-h);

line(x+w-topgap,y-h,x+w,y-dy);

line(x+w,y-dy,x+w,y-tireheight);

line(x+w,y-tireheight,x+w-(w/5),y-tireheight);

line(x+w-(w/5),y-tireheight,x+w-(w/5),y);

line(x+w-(w/5),y,x+w-(w/5)-tirewidth,y);

line(x+w-(w/5)-tirewidth,y,x+w-(w/5)-tirewidth,y-tireheight);

line(x+w-(w/5)-tirewidth,y-tireheight,x-w+(w/5)+tirewidth,y-tireheight);

line(x-w+(w/5)+tirewidth,y-tireheight,x-w+(w/5)+tirewidth,y);

line(x-w+(w/5)+tirewidth,y,x-w+(w/5),y);

line(x-w+(w/5),y,x-w+(w/5),y-tireheight);

line(x-w+(w/5),y-tireheight,x-w,y-tireheight);

line(x-w,y-tireheight,x-w,y-dy);

line(x-w,y-tireheight,x+w,y-tireheight);

setcolor(15);

return 0;

}

//welcome

int welcome(int color)

{

setcolor(color);

settextstyle(3,0,6);

outtextxy(100,2," Car Extreme Racing ");

settextstyle(6,0,2);

outtextxy(20,80," Use a and d Keys To go Left and Right");

outtextxy(20,140,"Press esc to Escape Any Time");

outtextxy(20,160,"Press Any Key To Continue ");

return 0;

}

int game\_over()

{

clrscr();

setcolor(15);

settextstyle(3,0,5);

outtextxy(50,2,"you just hit a car...");

outtextxy(50,75," Game Over...!!!!!! ");

/\*settextstyle(6,0,2);

outtextxy(20,80,"");

outtextxy(20,140,"Press esc to Escape Any Time");\*/

outtextxy(20,160,"Press Any Key To Continue ");

return 0;

}

int chk\_cont(int n)

{

if(n==pr)

{

rep++;

}

else

{

rep=0;

pr=n;

}

if(rep>rand()%3)

{

n=(n==0)?1:0;

rep=0;

}

return n;

}

int main(void)

{

int del=10,sensitivity=5,gm,gd=DETECT,car\_ok=1,side;

char ch,out;

double ecwd,echt,ecx,ecy;

srand(time(NULL));

side=rand()%2;

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

set();

ecx=(side==0)?lc\_x:rc\_x;

ecy=rc\_y;

ecwd=ec\_iwd;

echt=ec\_iht;

welcome(15);

ch=getch();

if(ch==27) exit(0);

welcome(0);

while(car\_ok)

{

rectangle(0,0,maxx,maxy);

drawroad();

drawtree(rt\_x,rt\_y,1,r\_ht,r\_wd);

drawtree(lt\_x,lt\_y,1,l\_ht,l\_wd);

mycar(posn,1);

draw\_enemycar(ecx,ecy,ecwd,echt,1);

delay(del);

draw\_enemycar(ecx,ecy,ecwd,echt,0);

mycar(posn,0);

drawtree(rt\_x,rt\_y,0,r\_ht,r\_wd);

drawtree(lt\_x,lt\_y,0,l\_ht,l\_wd);

sidetrees();

if(side==1)

{

ecx=ecx+incx;

}

else

{

ecx-=incx;

}

ecy+=incy;

ecwd+=xfact;

echt+=yfact;

if(ecy==maxy)

{

side=rand()%2;

side=chk\_cont(side);

ecx=(side==0)?lc\_x:rc\_x;

ecy=rc\_y;

ecwd=ec\_iwd;

echt=ec\_iht;

level++;

if(level>=10&&del>0)

{

del--;

sensitivity+=3;

level=0;

}

}

if(kbhit())

{

ch=getch();

if(ch==97)

{

posn-=sensitivity;

else if(ch==100)

{

posn+=sensitivity;

}

else if(ch==27)

{

car\_ok=0;

}

if((posn<carwidth)||(posn>maxx-carwidth)||(ecy>maxy-6-carheight&&(((ecx+ecwd>posn-carwidth)&&(ecx-ecwd<posn+carwidth))||((ecx-ecwd<posn+carwidth)&&(ecx+ecwd>posn-carwidth)))))

{

car\_ok=0;

game\_over();

}

}

loop:out=getch();

if(out==97||out==100)

{

goto loop;

}

closegraph();

return 0;

}