Group Members:

Name : Sakshi Patil	Sap ID :60004180091
Name : Samiksha Patil	Sap ID :60004180092

Batch: B2

COMPUTER GRAPHICS MINI PROJECT

Topic:

CAR RACING GAME

Abstract:

The project is created to demonstrate concepts of Computer Graphics. It is car racing Game made using C programming language.

The player can control the car through arrow keys. User has to avoid car to hit any obstacles. Here, we have various graphics.h functions such as setfillstyle, setcolor, outtextxy, etc.

To erase the previous content on screen cleardevice function is used. After running the code one can see two options on the screen ie. Play and Exit. One can start playing the game by clicking the play button. If the user wants to exit the the user has to click exit. If the user's game gets over (as the car hits the obstacle) then he/she is redirected to the same view where he/she can play again or exit. While the user is playing the game, the user can see their score at the top right part of the screen. User's score is displayed at the center when the game is over. User can move the car to the left using left arrow key and to the right using right arrow key. The user has to try his best to avoid car to hit the obstacles.

Explanation of Functions in the code :

Play():

If the play option is selected by user then this function is called. The car is initialized at 300 and life is initialized to 1.
1st obstacle starts from 0. 2nd obstacle starts 20 behind the 1st obstacle. 3rd one starts 40 behind the 1st one. For positioning of x axis a random integer through randax function is called and assigned under a certain range so that obstacles are randomly placed .This loop continues till life is 0 ie. game is over. the car can be moved to left and right using left arrow key and right arrow key through switch case of ASCII values.

Menu():

This function gives you the option to play the game or exit. Here we use switch case to select the option the user wants to enter. If the user selects play then the play function is called and if the user selects exit then the program is closed.

randx() :

It uses random function to choose a random value for random obstacle. If the x axis distance is less than 25 the obstacle lies outside the lane. So we add 25 to keep the values in the lane. Also this function allows the car to move to the limits of the lane.

right() :

This function is called to perform right direction movement

left() :

This function is called to perform left direction movement

bgmove():

Handles graphics of the car and it's movements as well as adds graphics and colour to ground and lane .

op1(),op2(),op3() :

Adds graphics and colour to obstacles

check():

To check the collision of the car along with the obstacles . If car collided then over() is called.

over():

After collision this function is called to show the game over and scores are displayed.

Code:

```
 #include<stdio.h>
 #include<conio.h>
 #include<dos.h>
 #include<stdlib.h>
 #include<graphics.h>
 #define LEFT 75
 #define RIGHT 77
 #define UP 72
 #define DOWN 80
 #define ENTER 13
 int score=0;
 struct race
 {
int cp, init, op1, op2, op3;
 };
 struct race car;
  int randx( void )
 {
int x = random( 55*8 );
if ( x < 25*8 )
{
x = x + ( 25 *8 );
}
return ( x );
 }
 void right()
 {
if ( car.cp < 420 )
{
car.cp += 10;
}
return ;
 }
 void left()
 {
if ( car.cp > 180 )
```

```
{
car.cp -= 10;
}
return ;
}
 void bgmove()
 {      setcolor(RED);
outtextxy(520,5,"SCORE=");
gotoxy(75,1);
printf("%d",score);
setfillstyle(1,GREEN);
bar3d(0,0,160,getmaxy(),0,0);
setfillstyle(1,GREEN);
bar3d(480,0,getmaxx(),getmaxy(),0,0);
if ( car.init > 5 )
{
car.init = 1;
}
setcolor( RED );
setfillstyle( SOLID_FILL,WHITE );
sector( car.cp + 22, 330, 0, 180, 19, 25 );
setfillstyle( SOLID_FILL,RED );
bar3d( car.cp+2, 333, car.cp + 37, 360, 4, 4 );
setfillstyle( SOLID_FILL,7);
bar(car.cp,336,car.cp+1,341);
bar(car.cp,350,car.cp+1,355);
bar(car.cp+42,336,car.cp+44,341);
bar(car.cp+42,350,car.cp+44,355);

for ( int i = car.init;i < 70;i += 5 )
{
setcolor(BLUE );
setfillstyle( 1, 8);
bar( 16 * 8, i * 8, 20 * 8, ( i + 2 ) * 8 );
bar( 60 * 8, i * 8, 64 * 8, ( i + 2 ) * 8 );
}
return ;
 }
 void opp1( int x1 )
 {
if ( car.op1 < 600 )
{
car.op1 += 10;
```

```
setfillstyle( 1, RED );
bar3d( x1, car.op1, x1 + 30, car.op1 + 30, 2, 3 );
}
else
{
car.op1 = 0;
}
return ;
 }
 void opp2( int x2 )
 {
if ( car.op2 < 600 )
{
car.op2 += 10;
setfillstyle( 1, YELLOW );
bar3d( x2, car.op2, x2 + 30, car.op2 + 30, 2, 3 );
}
else
{
car.op2 = 0;
}
return ;
 }
 void opp3( int x3 )
 {
if ( car.op3 < 600 )
{
car.op3 += 10;
setfillstyle( 1, BLUE );
bar3d( x3, car.op3, x3 + 30, car.op3 + 30, 2, 3 );
}
else
{
car.op3 = 0;
}
return ;
 }
 int check( int x, int op, int cp )
 {
int a[ 31 ], b[ 31 ], c[ 40 ], d[ 50 ];
for ( int i = 0;i < 30;i++ )
{
a[ i ] = x + i;
```

```
}
for ( i = 0;i < 30;i++ )
{
b[ i ] = op + i;
}
for ( i = 0;i < 40;i++ )
{
c[ i ] = cp + i;
}
for ( i = 0;i < 50;i++ )
{
d[ i ] = 300 + i;
}
for ( i = 0;i < 40;i++ )
{
for ( int j = 0;j < 50;j++ )
{
for ( int k = 0;k < 30;k++ )
{
if ( a[ k ] == c[ i ] && b[ k ] == d[ j ] )
return 1;
}
}
}
return 0;
 }
void over( void )
 {
for ( int i = 1;i < 35;i++ )
{
delay( 20 );
settextstyle( 7, 0, 5 );
setcolor( YELLOW );
outtextxy( 150, 100, "!!!GAME OVER!!!" );
settextstyle(3,0,3);
setcolor(RED);
outtextxy(200,200,"SCORE=");
gotoxy(40,14);
printf("%d",score);
}
delay( 2000 );
score=0;
getch();
```

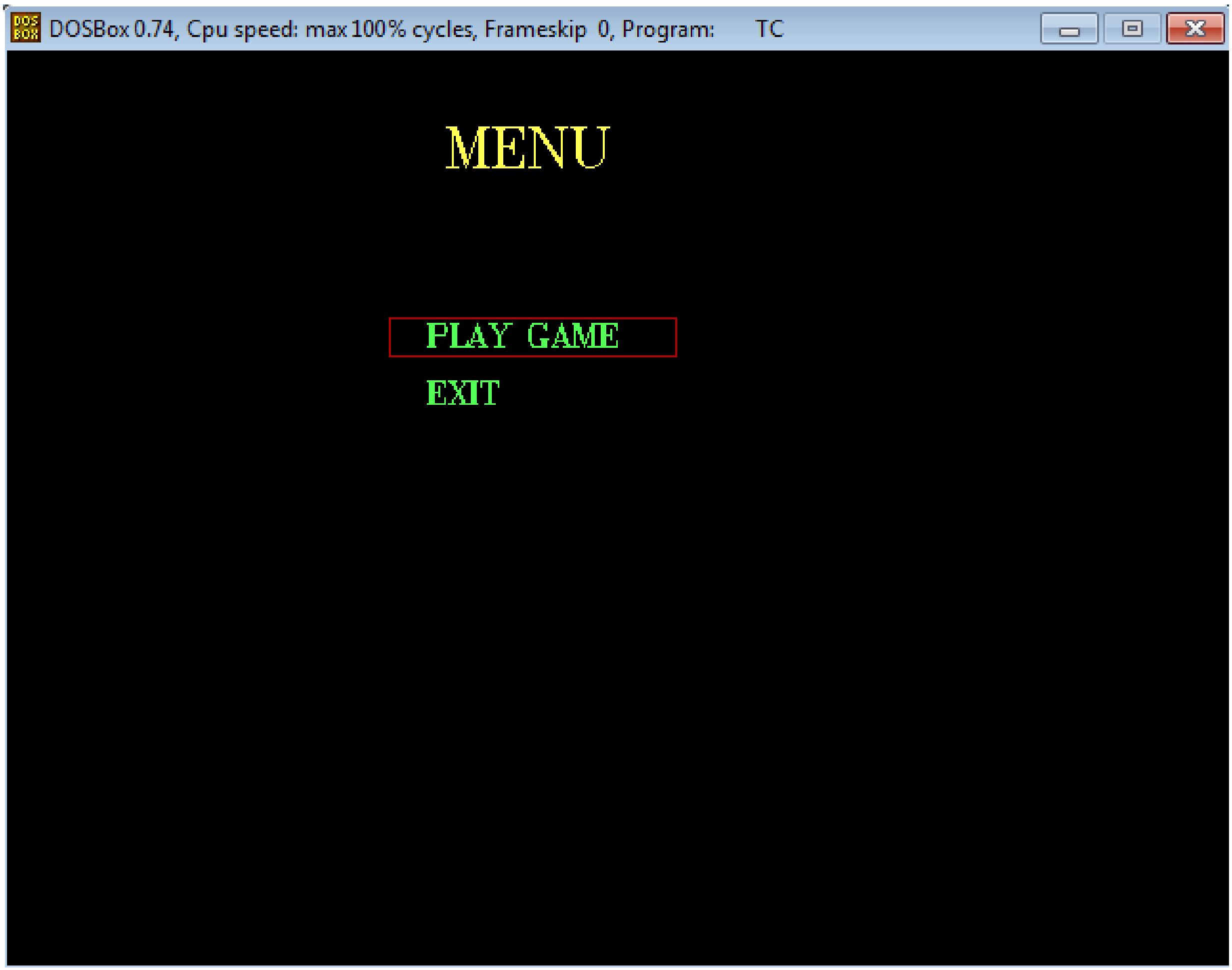
```
return ;
 }
  int play( void )
 {
car.cp = 300;
car.init = 0;
car.op1 = 0;
car.op2 = -20;
car.op3 = -40;
int life = 1, m, x1, x2, x3, s;
 re:
while ( life )
{
cleardevice();
if ( car.op1 == 0 ) //
{
x1 = randx();
car.op1++;
}
else
{
score++;
opp1( x1 );
}
if ( car.op2 == 0 )
{
x2 = randx();
car.op2++;
}
else if ( car.op2 < 1 )
{
car.op2++;
}
else
{
opp2( x2 );
}
if ( car.op3 == 0 )
{
x3 = randx();
car.op3++;
}
else if ( car.op3 < 1 )
```

```
{car.op3++;}
else
{
opp3( x3 );
}
setcolor( 10 );
bgmove();
if ( check( x2, car.op2, car.cp ) || check( x1, car.op1, car.cp ) || check( x3,
car.op3, car.cp ) )
{
life--;
if ( life == 0 )
{
over();
}
car.op1 = 1;
car.op2 = -16;
car.op3 = -32;
goto re;
}
if ( kbhit() )
{
m = getch();
switch ( m )
{
case LEFT: left();
break;
case RIGHT: right();
break;
case 'q': exit( 0 );
} }
else
{
delay( 100 );
} }
getch();
return ( 0 );
 }
 void menu()
 {
int s = 1, t = 0, a, p = 140;
 do
{
```

```
do
{
cleardevice();
setcolor( YELLOW );
settextstyle( 1, 0, 4 );
outtextxy( 230, 30, "MENU" );
settextstyle( 1, 0, 1 );
setcolor( 10 );
outtextxy( 220, 137, "PLAY GAME" );
outtextxy( 220,167, "EXIT" );
setcolor( RED );
rectangle( 200, p, 350, p + 20 );
a = getch();
switch ( a )
{
case UP: if ( p > 140 )
{
p -= 30;
s--;
}
break;
case DOWN: if ( p < 180 )
{
p += 30;
s++;
};
break;
case ENTER: a = 1;
break;
default: break;
}
}
while ( a != 1 );
switch ( s )
{
case 1: play();
break;
case 2: exit( 0 );
break;
}
}
while ( t == 0 );
 }
 void main()
 {
int gdriver = DETECT, gmode;
initgraph( &gdriver, &gmode, "C:\\TC\\bgi" );
cleardevice();

menu();
getch();
 }
```

Output:

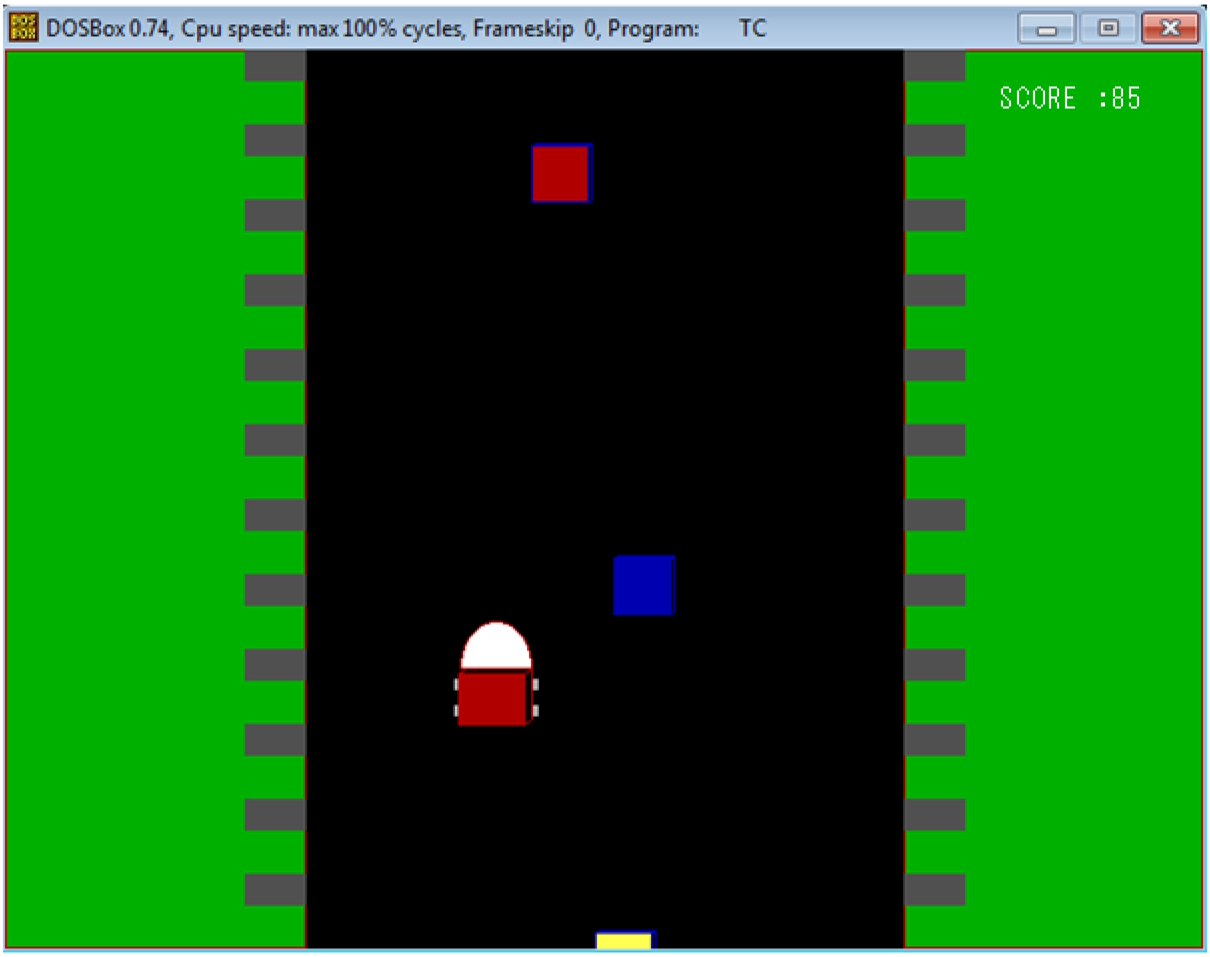


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

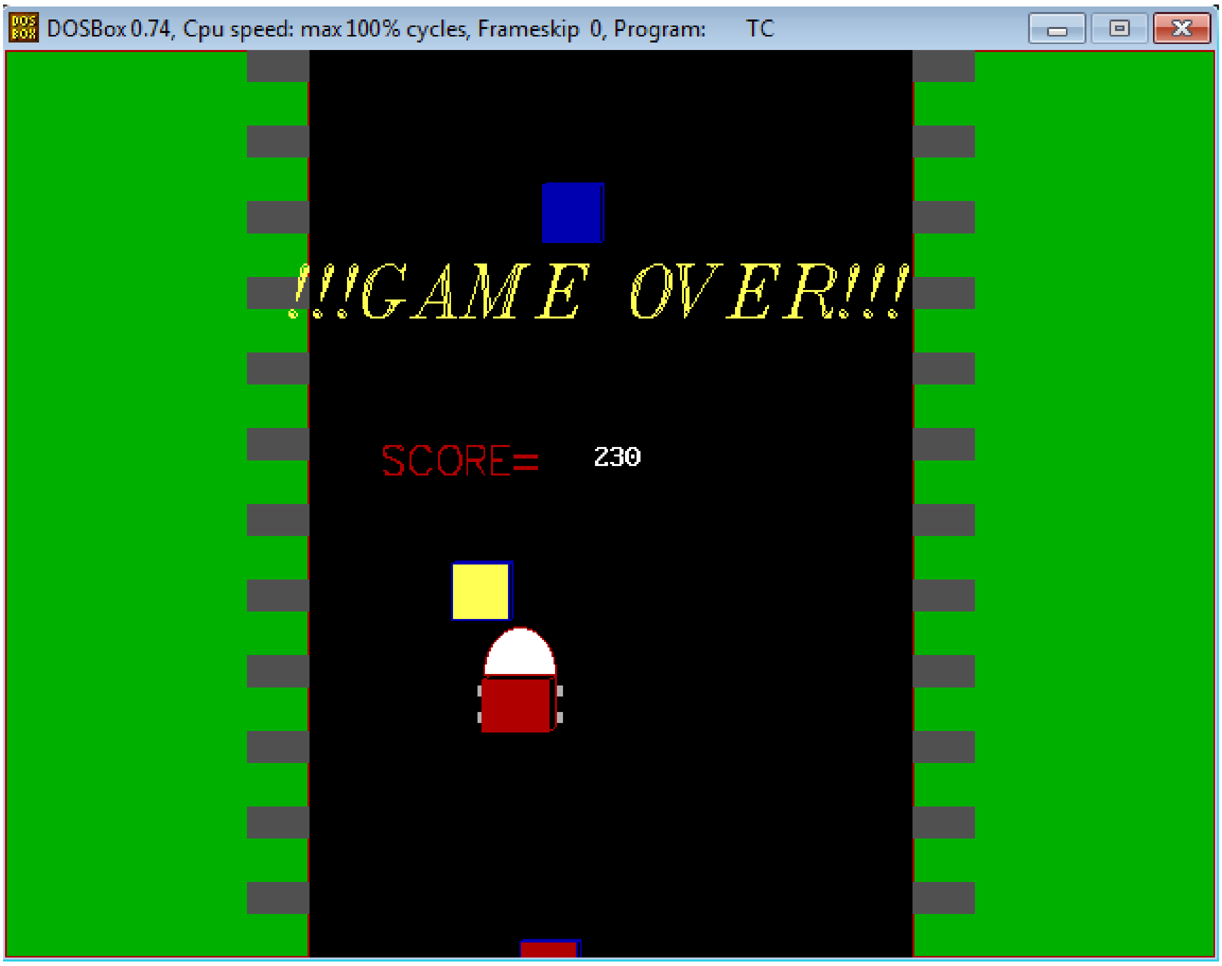
MENU

PLAY GAME

EXIT



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
SCORE :85



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
!!!GAME OVER!!!
SCORE= 230

Conclusion :

Thus we have implemented car racing game using computer graphics in C.