The Truck Rally

C++ Project (Game)

A.Krishnakanth XII-A 2016-17

ACKNOLEDGEMENT

I wish to express my sincere gratitude to the school management, for providing me an opportunity to do my project work on C++.

I sincerely thank Mrs. Gayathri, Computer Science teacher for her guidance and encouragement in carrying out this project work. I also wish to express my gratitude to my friends and family who rendered their help during the period of my project work.

INDEX

MINIMUM REQUIREMENTS

HARDWARE REQUIREMENTS

1. 80386-SX 16MHz processor
2. 4 MB of memory (RAM)
3. 10 MB of free space on a HDD
4. 512 KB of VGA compatible graphics card (VRAM).

SOFTWARE REQUIREMENTS

1. Borland Turbo C++ Compiler Installed
2. MS-DOS version 4.00/Windows (95/98)/Linux (x32)
3. Keyboard with Keyboard Driver Installed
4. VGA Graphic Driver Installed (BGI)

PROGRAM DEFINITION

1. This program is a game where user or player is a truck.
2. The main motive of the game is to tackle as much as trucks as possible to gain more score.
3. The difficulty would increase as each time cycle completes with suitable bonus.
4. The efficiency of the program is so remarkable that the fancy graphical and processing lags have been reduced to a great extent to make player more comfortable.
5. This program would fit in any screen as proper coding’s are given.

CONTROLS

1. Press ‘A’ to move left
2. Press ‘D’ to move right
3. Press ‘Esc’ to exit the game.

RULES AND REGULATIONS

1. Avoid hitting opponent trucks
2. Avoid going out of roads
3. After every time cycle, Speed would be increased and bonus would be given
4. After a certain point of time, speed would remain constant while still you would be awarded with bonus points.

DEFINITIONS (CLASS AND FUNCTIONS)

CLASS: SCO

1. To store score and name into an object
2. And then to store the object into an file
3. Helps in displaying Hi-Scores
4. Contains functions to enter, summarize and return information.

FUNCTION: RESET()

1. Initializes the default values to the respective variables.

FUNCTION: BORDERS()

1. Draws borders of red and blue.

FUNCTION: TITLE()

1. Displays the fancy title screen with hi-score of the game

FUNCTION: SCORESAVE()

1. Save the score and name into an object and then into a file

FUNCTION: HISCORE()

1. To return Hi-score

FUNCTIONS: ENE(),ENE1() AND ENE2()

1. Initializes random non-overlapping x-coordinate values for the opponent trucks

FUNCTION: ENE12()

1. To invoke ene(), ene1() and ene2()

FUNCTION: CONTROL()

1. Move the player by getting controls
2. Regulates automatic movement of opponent trucks
3. Exits the program when pressed ‘ese’ button

FUNCTION: MOVE()

1. Move opponent truck forward by 10 units

FUNCTION: BGROUND()

1. Draws Roads and green grasses on screen

FUNCTION: CAR()

1. Draws player truck with respect to movements

FUNCTIONS: CAR1(),CAR2() AND CAR3()

1. Draws opponent trucks with respect to movements
2. Calls respective ene() function when y-coordinate value goes beyond maximum y-coordinate value of the screen.

FUNCTION: DRAW()

1. Invokes and regulates BGround(),car(), car1(), car2() and car3() efficiently
2. Draws and regulates Time Bar and Time Value
3. After every time cycle speed is increased
4. After every time cycle bonus score is awarded.

FUNCTION: OUT()

1. Exits the game screen when truck crashes or truck going out of roads.
2. Invokes ScoreSave() function
3. Closes and clears graphical screen.
4. Ask user for retry

FUNCTION: MUSIC)

1. To Play the desired music when invoked.

FUNCTION: MAIN()

1. Initiates Graphics
2. Check for graphic initialization errors
3. Displays Controls and rules.
4. Regulates and navigates the game's functions.

SOURCE CODE

```
#include <conio.h>

#include <fstream.h>

#include <ctype.h>

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <dos.h>

#include <string.h>

#include <stdlib.h>


int x1,x2,x3,x4,y2,y3,y4,i=0,speed=30;

unsigned long int score=0;

float t=150;

char value[20],ch,ch1;


class sco

{

        int s1;

        char name[20];

        public:

                sco()

                {       s1=0;
```

```
            strcpy(name,NULL);
        }
    void summary()
    {
        clrscr();
        cout<<"\n\t\t\t  Summary\n\t\t\t-----------";
        cout<<"\n\nName: "<<name<<"\nScore: "<<value;
        cout<<"\n\n\nPress any key to exit.........";
        getch();
    }
    void save()
    {
        cout<<"Enter Name (Max. 20 characters): ";
        gets(name);
        s1=score;
    }
    int ret()
    {
        return s1;
    }
}sco1,sco2;
```

```
void music()

{

        sound (330);delay(100);

        sound (330);delay(300);

        sound (330);delay(300);

        sound (262);delay(100);

        sound (330);delay(300);

        sound (392);delay(700);

        sound (196);delay(700);

        sound (196);delay(125);

        sound (262);delay(125);

        sound (330);delay(125);

        sound (392);delay(125);

        sound (523);delay(125);

        sound (660);delay(125);

        sound (784);delay(575);

        sound (660);delay(575);

        sound (207);delay(125);

        sound (262);delay(125);

        sound (311);delay(125);

        sound (415);delay(125);

        sound (523);delay(125);

        sound (622);delay(125);
        sound (830);delay(575);
        sound (622);delay(575);
        sound (233);delay(125);
        sound (294);delay(125);
        sound (349);delay(125);
        sound (466);delay(125);
        sound (587);delay(125);
        sound (698);delay(125);
        sound (932);delay(575);
        sound (932);delay(125);
        sound (932);delay(125);
        sound (932);delay(125);
        sound (1046);
        nosound();
}

void Reset()
{
        score=0;
        speed=30;
        t=150;
        y2=0;
        y3=-70;
        y4=-200;
}

void Borders()          //To draw borders
{
        cleardevice();
        setfillstyle(SOLID_FILL,RED);
        bar(0,0,getmaxx(),10);
        setfillstyle(SOLID_FILL,BLUE);
        bar(0,getmaxy()-10,getmaxx(),getmaxy());
}

void title()
{
        int gd=DETECT,gm;
        clrscr();
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        for(int i=0;i<10;i++)
        {
        cleardevice();
        setfillstyle(SOLID_FILL,RED);
        bar(50,150,600,230);
        rectangle(50,150,600,230);
```

```
settextstyle(GOTHIC_FONT,HORIZ_DIR,5);
outtextxy(80,160,"        Truck Rally");
delay(100);
cleardevice();
setfillstyle(SOLID_FILL,BLUE);
bar(50,150,600,230);
rectangle(50,150,600,230);
settextstyle(GOTHIC_FONT,HORIZ_DIR,5);
outtextxy(80,160,"        Truck Rally");
delay(100);
cleardevice();
setfillstyle(SOLID_FILL,CYAN);
bar(50,150,600,230);
rectangle(50,150,600,230);
settextstyle(GOTHIC_FONT,HORIZ_DIR,5);
outtextxy(80,160,"        Truck Rally");
delay(100);
cleardevice();
setfillstyle(SOLID_FILL,MAGENTA);
bar(50,150,600,230);
rectangle(50,150,600,230);
settextstyle(GOTHIC_FONT,HORIZ_DIR,5);
outtextxy(80,160,"        Truck Rally");
```

```
        delay(100);
        }
}

void ScoreSave()
{
        cout<<"\n===============================================================================\n\n";
        cout<<"\t\t\tScoreboard\n\t\t\t-------------\n\n";
        sco1.save();
        ofstream fout("Score_Truck.dat",ios::binary|ios::app);
        fout.write((char*)&sco1,sizeof(sco1));
        sco1.summary();
        fout.close();
        getch();

}

int hiscore()
{
        ifstream fin("Score_Truck.dat",ios::binary|ios::beg);
        int max;
        for(int i=0;i<100;i++)
        {
                fin.read((char*)&sco2,sizeof(sco2));
                if(sco2.ret()>max)
                        max=sco2.ret();
        }
        fin.close();
        return max;
}

void ene()
{
        X:randomize();
        int a=210+(rand()%190);
        x2=a;
        y2=0;
}

void ene1()
{
        randomize();
        int a=210+random(190);
        x3=a;
        y3=-70;
        randomize();
}

void ene2()
{
        randomize();
        int a;
        X:a=230+(random(100)+(rand()%100));
        if(a>400)
             goto X;
        x4=a;
        if((x1-20)<=x4&&x4<=(x1+20))
                goto X;
        if((x2-20)<=x4&&x4<=(x2+20))
                goto X;
        y4=-200;
        randomize();
}
void ene12()
{
        ene2();
        ene1();
        ene();
```

```
}


void move()
{
        y4+=10;
        y3+=10;
        y2+=10;
}
int Control()
{
        if(!kbhit())
        {       delay(speed);
                return 0;
        }
        else
        {
                switch(getch())
                {
                        case'd':case'D':x1+=10;break;
                        case'a':case'A':x1-=10;break;
                        case 27:getch();closegraph();exit(0);break;
                }
        }
```

```
                return 0;
}

void BGround()
{
        /*              GRASS               */
        setfillstyle(9,GREEN);
        bar3d(0,0,getmaxx()/3,getmaxy(),0,0);
        setfillstyle(9,GREEN);
        bar3d(2*getmaxx()/3,0,getmaxx(),getmaxy(),0,0);

        /*              ROAD                        */
        setfillstyle(SOLID_FILL,8);
        bar3d(getmaxx()/3,0,2*getmaxx()/3,getmaxy(),0,0);


}
void car()
{
        /*              Tyres               */
        setfillstyle(1,BLACK);
        bar3d(x1+1,getmaxy()-38,x1+5,getmaxy()-32,0,0);
        setfillstyle(1,BLACK);
        bar3d(x1+25,getmaxy()-38,x1+29,getmaxy()-32,0,0);
        /*          Body      */
        setfillstyle(1,YELLOW);
        bar3d(x1+5,getmaxy()-10,x1+25,getmaxy()-40,0,0);

        /*          Container */
        setfillstyle(4,BLUE);
        bar3d(x1,getmaxy()-30,x1+30,getmaxy(),0,0);
}
void car1()
{
        /*          Tyres             */
        setfillstyle(1,BLACK);
        bar3d(x2+1,y2+2,x2+5,y2+7,0,0);
        setfillstyle(1,BLACK);
        bar3d(x2+25,y2+2,x2+29,y2+7,0,0);

        /*          Body      */
        setfillstyle(1,RED);
        bar3d(x2+5,y2-10,x2+25,y2+10,0,0);

        /*          Container */
        int a=random(13);
```

```
	setfillstyle(11,a);
	bar3d(x2,y2-40,x2+30,y2,0,0);

	if(y2>getmaxy())
		ene();
}

void car2()
{
	/*		Tyres			*/
	setfillstyle(1,BLACK);
	bar3d(x3+1,y3+2,x3+5,y3+7,0,0);
	setfillstyle(1,BLACK);
	bar3d(x3+25,y3+2,x3+29,y3+7,0,0);

	/*		Body		*/
	setfillstyle(1,GREEN);
	bar3d(x3+5,y3-10,x3+25,y3+10,0,0);

	/*		Container */
	randomize();
	int a=random(13)+1;
	setfillstyle(11,a);
```

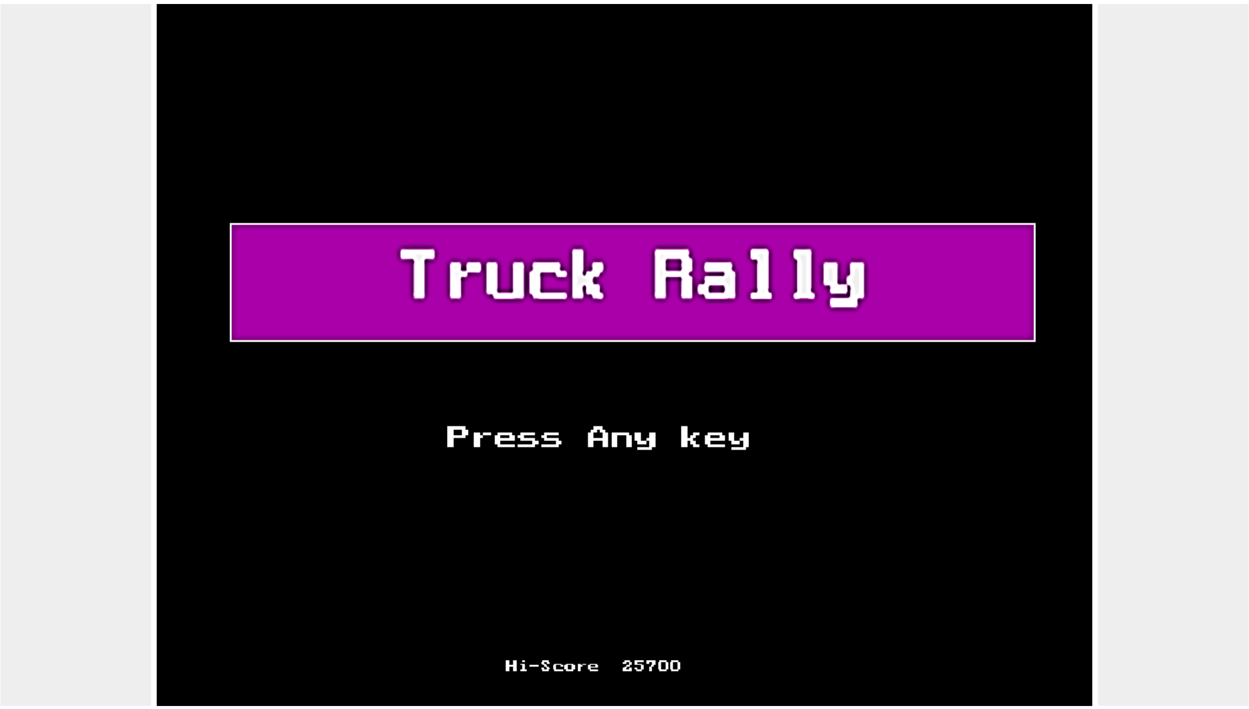
```
        bar3d(x3,y3-40,x3+30,y3,0,0);

        if(y3>getmaxy())
                ene1();
}
void car3()
{
        /*              Tyres                   */
        setfillstyle(1,BLACK);
        bar3d(x4+1,y4+2,x4+5,y4+7,0,0);
        setfillstyle(1,BLACK);
        bar3d(x4+25,y4+2,x4+29,y4+7,0,0);

        /*              Body            */
        setfillstyle(1,MAGENTA);
        bar3d(x4+5,y4-10,x4+25,y4+10,0,0);

        /*              Container */
        randomize();
        int a=random(13)+2;
        setfillstyle(11,a);
        bar3d(x4,y4-40,x4+30,y4,0,0);
        if(y4>getmaxy())
                ene2();
}

void draw()
{
        BGround();
        outtextxy(465,100,"Press Esc to exit...");
        outtextxy(440,10,"Score: ");
        sprintf(value,"%d",score);
        outtextxy(500,10,value);
        car();
        car1();
        car2();
        car3();
        setfillstyle(1,RED);
        bar3d(480,150,500,400,0,0);
        setfillstyle(1,WHITE);
        if(t>400)
        {
                score+=50;
                t=150;
                speed-=10;
        if(speed<0)
            speed=0;
        sound (932);delay(20);
        sound (1046);nosound();

    }
    bar3d(480,t,500,400,0,0);
}

int out()
{
    int a1=x1-30,a2=x1+30;
    if((x1+7)<=(getmaxx()/3)||(x1)>=(2*(getmaxx()/3))-20)
    {
            getch();
            cleardevice();
            closegraph();
            cout<<"\nOh ! Ride only on roads !\nGame
Over ! ;(\n\n\nPress any key to proceed....";
            getch();
            ScoreSave();
            cout<<"\n\nDo you wish to restart (Y/N)";
            cin>>ch1;
            tolower(ch1);
            if(ch1=='y'||ch1=='Y')
                return 1;
            return 0;
        }
    if(y2>=(getmaxy()-35))
    {
        if(x2>a1 && x2<a2)
        {
            getch();
            closegraph();
            cout<<"\nOh ! You crashed on Truck 1!\nGame Over ! ;(\n\n\nPress any key to proceed....";
            getch();
            ScoreSave();
            cout<<"\n\nDo you wish to restart (Y/N)";
            cin>>ch1;
            tolower(ch1);
            if(ch1=='y'||ch1=='Y')
                return 1;
            return 0;
        }
    }
    if(y3>=(getmaxy()-35))
    {
        if(x3>a1 && x3<a2)
        {
            getch();
            closegraph();
            cout<<"\nOh ! You crashed on Truck 2!\nGame Over ! ;(\n\n\nPress any key to proceed....";
            getch();
            ScoreSave();
            cout<<"\n\nDo you wish to restart (Y/N)";
            cin>>ch1;
            tolower(ch1);
            if(ch1=='y'||ch1=='Y')
                return 1;
            return 0;
        }
    }
    if(y4>=(getmaxy()-35))
    {
        if(x4>a1 && x4<a2)
        {
            getch();
```

```
                cleardevice();
                closegraph();
                cout<<"\nOh ! You crashed on Truck 3 !\nGame Over ! ;(\n\n\nPress any key to proceed....";
                getch();
                ScoreSave();
                cout<<"\n\nDo you wish to restart (Y/N)";
                cin>>ch1;
                tolower(ch1);
                if(ch1=='y'||ch1=='Y')
                    return 1;
                return 0;
            }
    }
    return 0;
}

void main()
{
    int gd= DETECT, gm, err,i;
    clrscr();
    cout<<"\n\n Initiating Graphics....";
    delay(500);
```

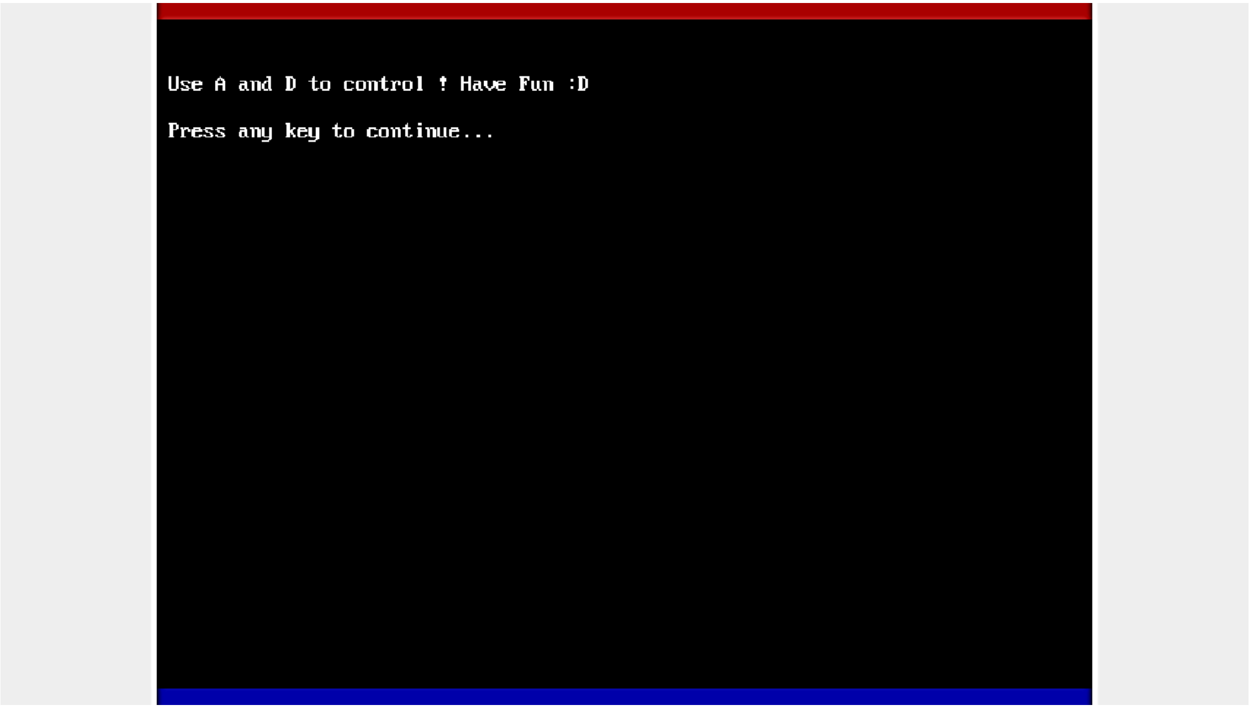
```
START:initgraph(&gd, &gm, "C:\\TC\\BGI");
err = graphresult();
if (err!= grOk)
{
        cout<<"\n\t\t\t  Error\n\t\t\t==========";
        cout<<"\n Graphics error: "<<grapherrormsg(err);
        cout<<"\n\n Press any key to halt......";
        getch();
        exit(1);
}
cleardevice();
cout<<"\n Graphics Initiated...\n Graphic Result is "<<err;
delay(1000);
cleardevice();
Reset();
title();
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
outtextxy((getmaxx()/2)-120,(getmaxy()/2)+50,"Press Any key");
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
outtextxy((getmaxx()/2)-80,getmaxy()-30,"Hi-Score");
char HiScore[10];
sprintf(HiScore,"%d",hiscore());
outtextxy((getmaxx()/2),getmaxy()-30,HiScore);
```

```
	getch();

	settextstyle(DEFAULT_FONT,HORIZ_DIR,1);

	music();

	cleardevice();

	Borders();

	cout<<"\n\n\n Use A and D to control !\n Do not Hit other Trucks !\n Never go out of bounds ! Have Fun :D\n\n Press any key to continue...";

	getch();

	cleardevice();

	x1=(getmaxx()/2)-10;

	ene12();

	for(i=0;ch!='x';i++)

	{

		draw();

		Control();

		move();

		if(out()==1)

			goto START;

		score+=10;

		t+=(0.5);

	}

	closegraph();

}
```

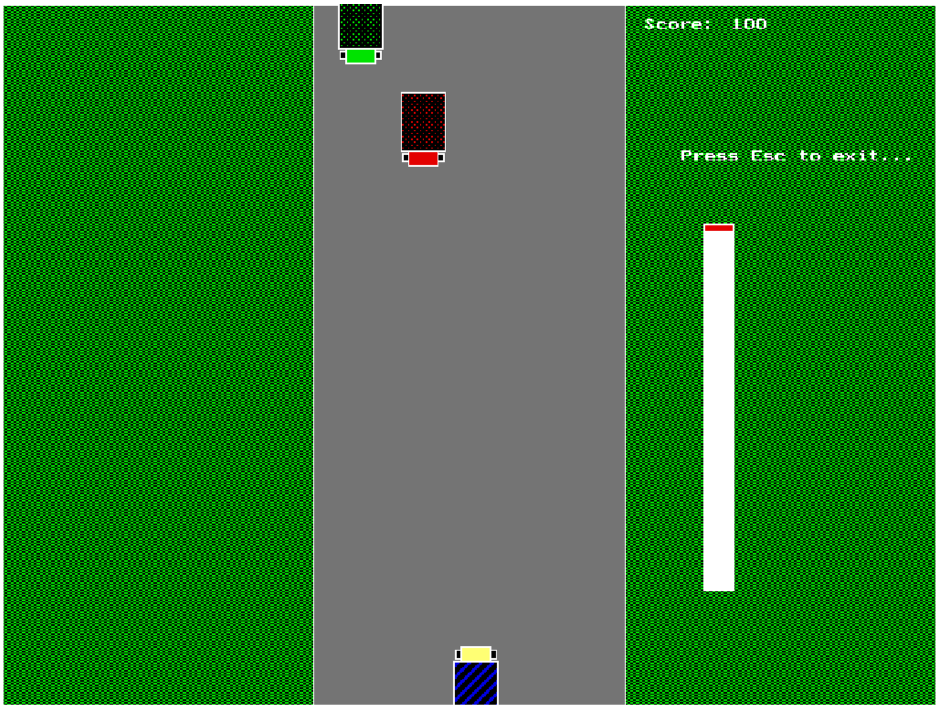
OUTPUT SCREENS



```
Truck Rally

Press Any key

Hi-Score  25700
```

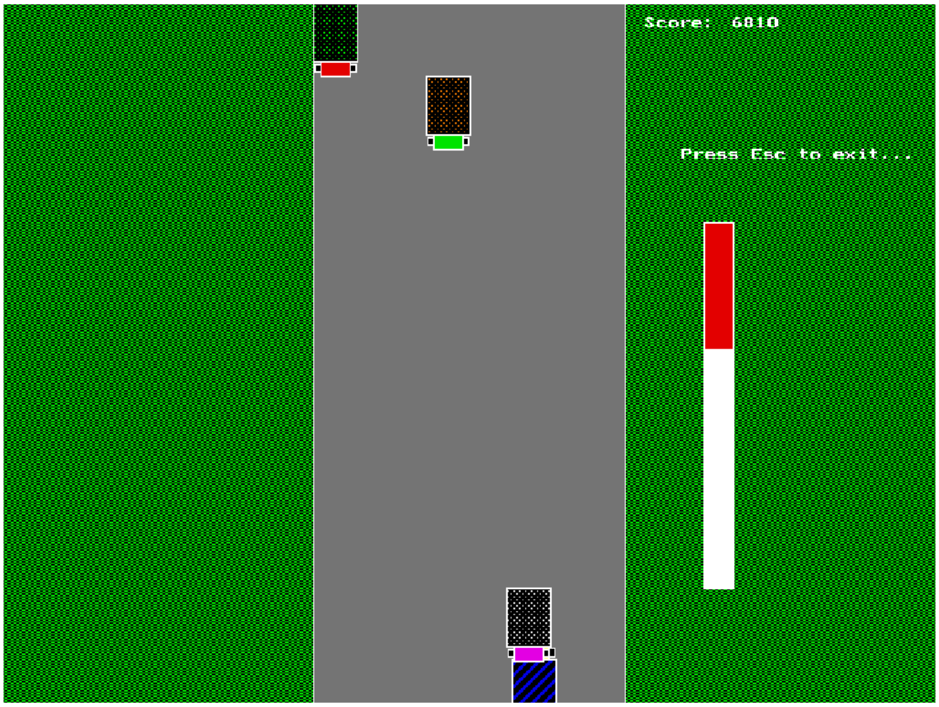


```
Use A and D to control ! Have Fun :D

Press any key to continue...
```



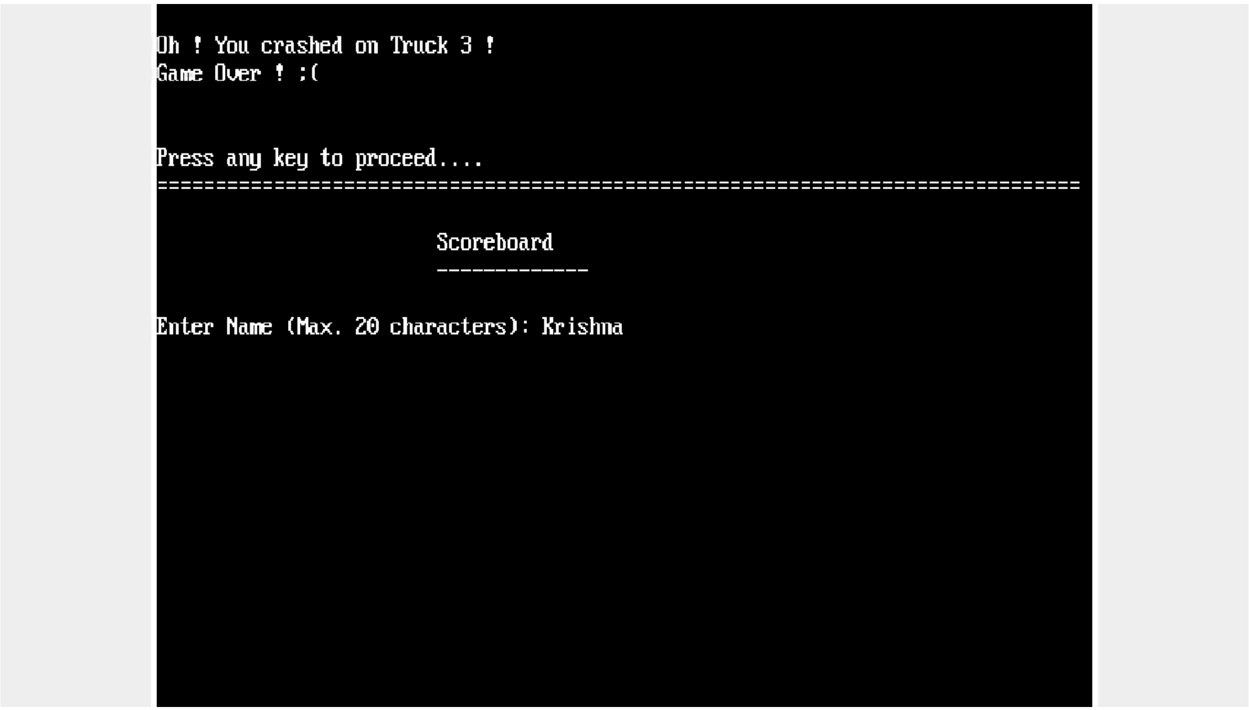
Score: 100

Press Esc to exit...

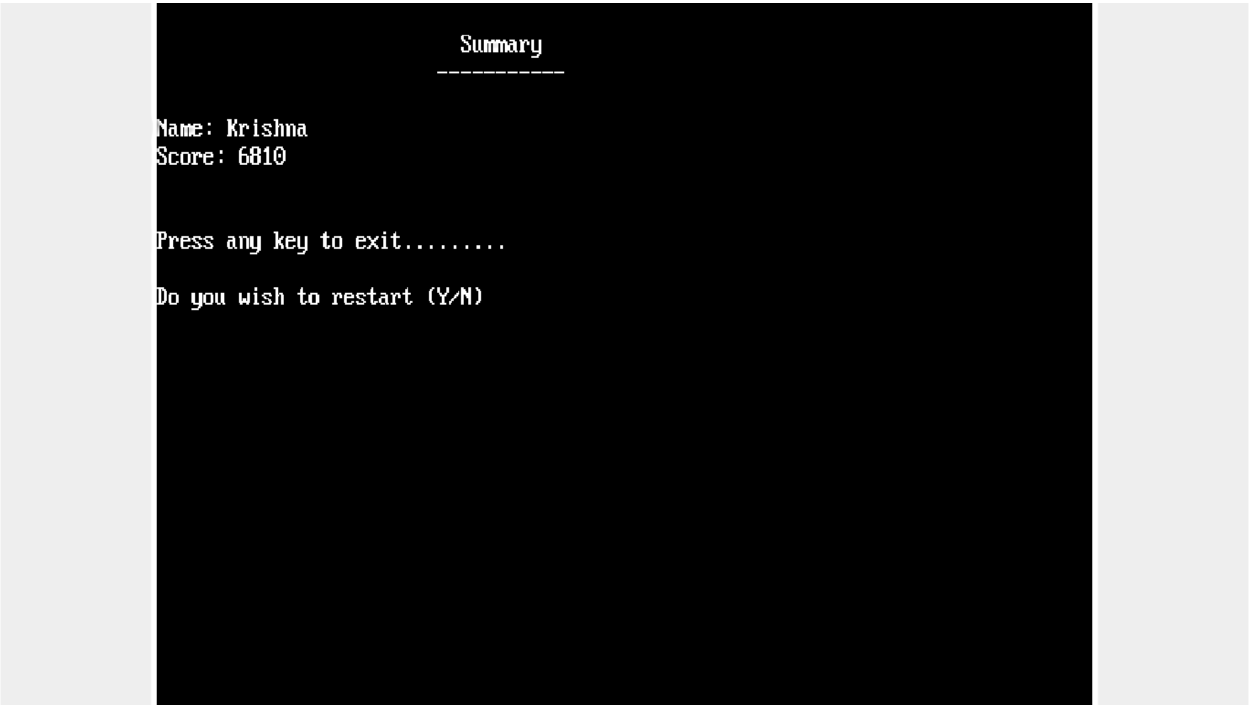


Score: 6810

Press Esc to exit...



```
Oh ! You crashed on Truck 3 !
Game Over ! :(


Press any key to proceed....
================================================================================

                        Scoreboard
                        -------------


Enter Name (Max. 20 characters): Krishna
```



```
                          Summary
                        -----------


Name: Krishna
Score: 6810


Press any key to exit.........

Do you wish to restart (Y/N)
```

BIBLIOGRAPHY

Sources that had been used as reference are:

1. COMPUTER SCIENCE WITH C++ (XI)
   Written by: Sumita Arora
   Published by: Dhanpat Rai & Co. (P) Ltd., Delhi

2. COMPUTER SCIENCE WITH C++ (XII) VOL. 1
   Written by: Sumita Arora
   Published by: Dhanpat Rai & Co. (P) Ltd., Delhi

3. BORLAND'S TURBO HELP INDEX
   Copyright © 1990, 1992 by Borland International, Inc.