Panjab University

Swami Sarvanand Giri Regional Centre

Computer Graphics

Project: *Don't Cross the Line*

Submitted to: Ms. Prabha Ma'am

Submitted by: Puneet – SG18337

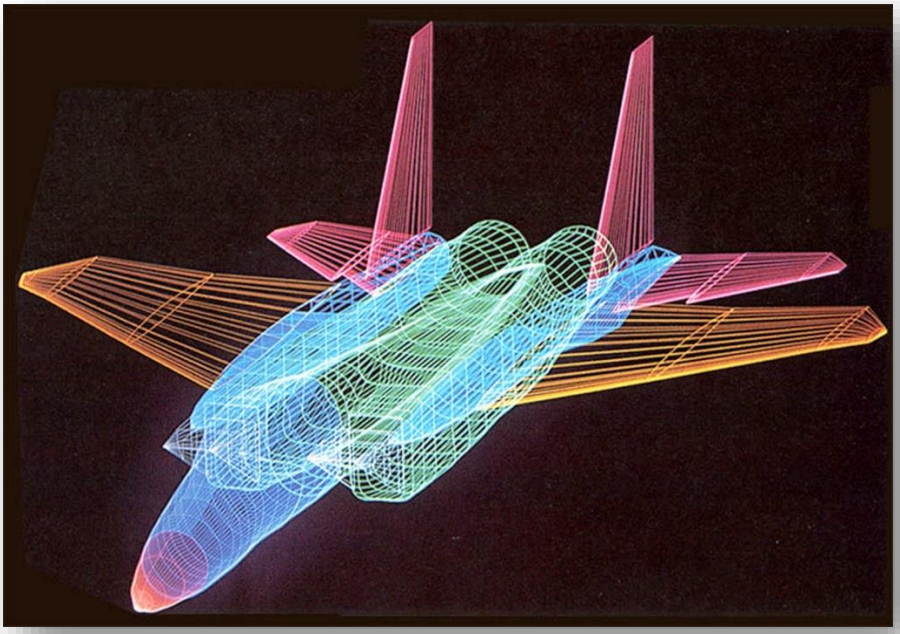
Saundarya – SG18344

Saurabh – SG18345

BATCH: B.E. CSE 5TH SEMESTER

Content

Introduction to Computer Graphics:



Computer Graphics involves technology to access. The Process transforms and presents information in a visual form. The role of computer graphics insensible. In today life, computer graphics has now become a common element in user interfaces, T.V. commercial motion pictures.

Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering.

In computer graphics, two or three-dimensional pictures can be created that are used for research. Many hardware devices algorithm has been developing for improving the speed of picture generation with the passes of time. It includes the creation storage of models and image of objects. These models for various fields like engineering, mathematical and so on.

Today computer graphics is entirely different from the earlier one. It is not possible. It is an interactive user can control the structure of an object of various input devices.

Some Important Functions used:

- **<graphics.h>**: This header file contains all prototype and the definition of all graphics functions. Before starting any Drawing action we need to initialize graphics mode.

- **<dos.h>**: This DOS Disc operating system, it is also a header file in C program for DOS type screen.
- **initgraph( )** : This function initialize the graphics system. Prototype is defined in 'graphics.h'.
- **setcolor( )**: This function sets the current drawing color . This is declared as under *void for set color (int color);*
- **setcolor( )**: The header file graphics.h contains setcolor() function which is used to set the current drawing color to the new color.
- **setbkcolor( ):** setbkcolor function changes current background color e.g. setbkcolor(YELLLOW) changes the current background color to YELLOW. Remember that default drawing color is WHITE and background color is BLACK.
- **outtextxy( ):** outtextxy function display text or string at a specified point(x,y) on the screen.
- **settextstyle( ):** This function is used to change the way in which text appears, using it we can modify the size of text, change the direction of text, and change the font of the text.
- **rectangle( ):** Declaration: void rectangle(int left, int top, int right, int bottom); It is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.
- **circle( ):** This function Draw a circle at (x,y) of the given radius. This function is declared as under.void far circle*(int x, int y, int radius); .*
- **getbkcolor( ):**Function getbkcolor returns the current background-color.
- **getcolor( )**: getcolor function returns the current drawing color.
- **setlinestyle( )***:* The header file graphics.h contains **setlinestyle()** function which sets the style for all lines drawn by line, lineto, rectangle, drawpoly, and so on.

PROJECT SRS

Introduction:

Don't cross the line is a two player mini game GTA 5 developed using C programming. It also has some basic graphics added to it so as to make the game more intriguing. The purpose of this document is that all the functional and non-functional requirements mentioned in it should be utilized and understood by software developer to implement the game.

Purpose:

Don't cross the line is a game designed for users to play and release their stress. They can choose levels according to their will: Easy, Medium and Difficult. Before starting the game they should go through the help section provided for all the instructions. After they are done with playing, they can exit it.

The purpose for creating this game is to learn and practice C programming thoroughly, along with all the different graphics and functions. The best part about this game is that it is not complex and is compatible for people of about any age.

Scope:

With this idea of a simple game, one can create many such games with broader ideas and complexities. By developing a game on your own provides a learning as well as entertaining experience. Developing such a less data consuming and efficient gaming system also helps achieving the user's needs and satisfaction.

User Perspective:

The user shall get a clean and stable interface on which only needs to select the pre-defined options in form of numbers and the operation shall begin to perform.

Functional Specifications:

The product requires programming knowledge from the developer and a basic knowledge of computer system from the user to use the product. The functions involved in this game are:

1. New game
2. Levels
3. About
4. Help
5. Exit

1) **New game:** One can start a new game under this option. In this, the two players involved should try and not to touch each other's snake.
    a. If Player1 touches the Player2's snake first, player1 wins.
    b. If Player2 touches the Player1's snake first, player2 wins.
    c. If both touch at the same time, it's a draw.

2) **Levels:** There are three levels defined in accordance of their difficulty. Speed is increased and dimensions of the box are reduced, as the level increase.
    a. Easy
    b. Medium
    c. Difficult

3) **Help:** It contains the rules and control keys of the game, about how to play it. The user needs to read this before starting the new game.

- **Rules:**
    1) DO NOT TOUCH THE BOUNDARY LINE
    2) DO NOT TOUCH THE OPPONENT
    3) DO NOT OVERLAP ON YOUR OWN PATH

- **Control keys:**
    1) W,A,S,D KEYS FOR PLAYER 1
    2) I ,J, K, L KEYS FOR PLAYER 2

4) **About:** This has some basic information about the game. It also contains the information about the creators of the game.

5) **Exit:** This is used to quit the game after you are done playing with it and it will stop the program.

Hardware Requirements:

1. Monitor
2. Keyboard
3. Continuous battery backup
4. Computer System

Software Requirements:

1. Any windows operating system.
2. To be able to play this game your machine should have any C graphical Environment like Turbo- C, Codeblocks, etc.

Performance Requirements:

The system should be compatible enough to do the basic operations. It should not get hang or show some other problems arising out due to large number of concurrent users. The system should be fast enough to play the game without any lags. The high and low temperature should not affect the performance of the device.

Design Constraints:

- **Software Language Used**: The language that is used for coding "**Don't cross the line**" is **C**.
- The **graphics.h** header file is being used so as to draw and color lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window.

Performance:

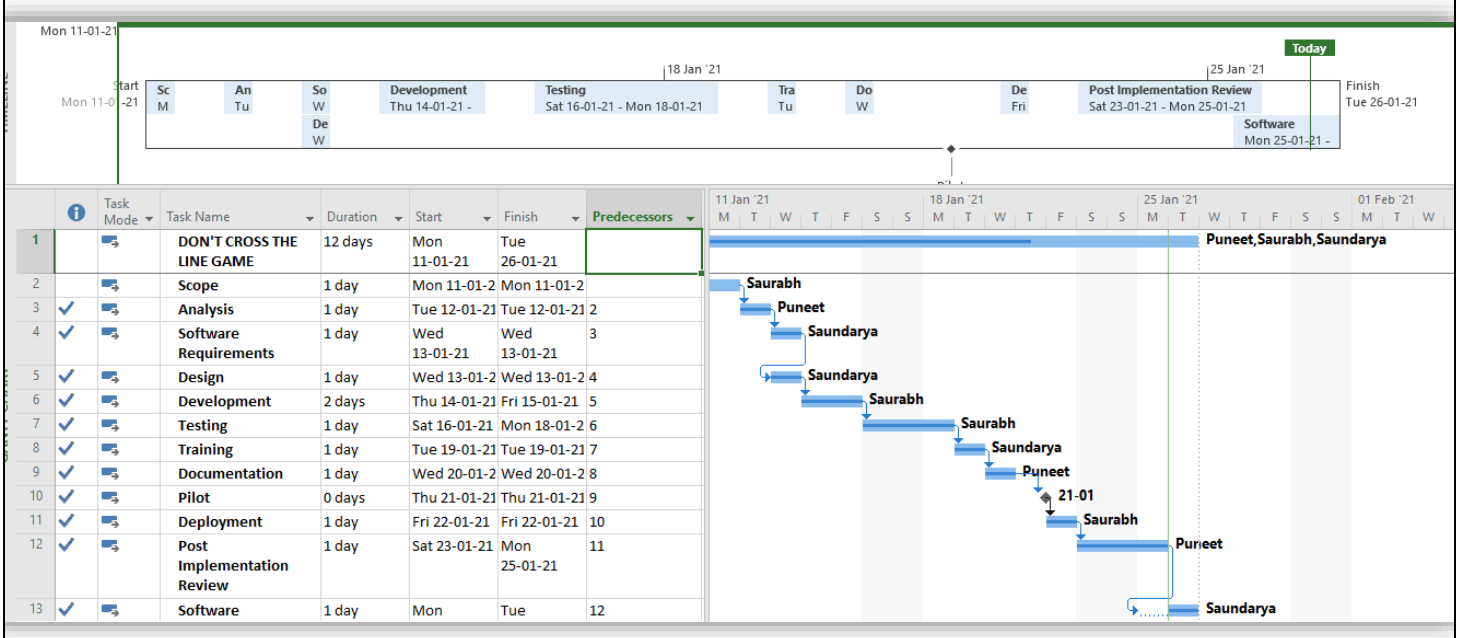
- **Reliability:** The game should be highly reliable and it should generate all the updated information in correct order.
- **Easy to use:** It should have simple interface for the user and not the complex options for the user to get confused.

- **Low Storage:** It should not be so heavy that it lags or hangs, low storage consumed will lead to fast speed.
- **Portability:** The game should be portable on any windows based system. It should not be machine specific.

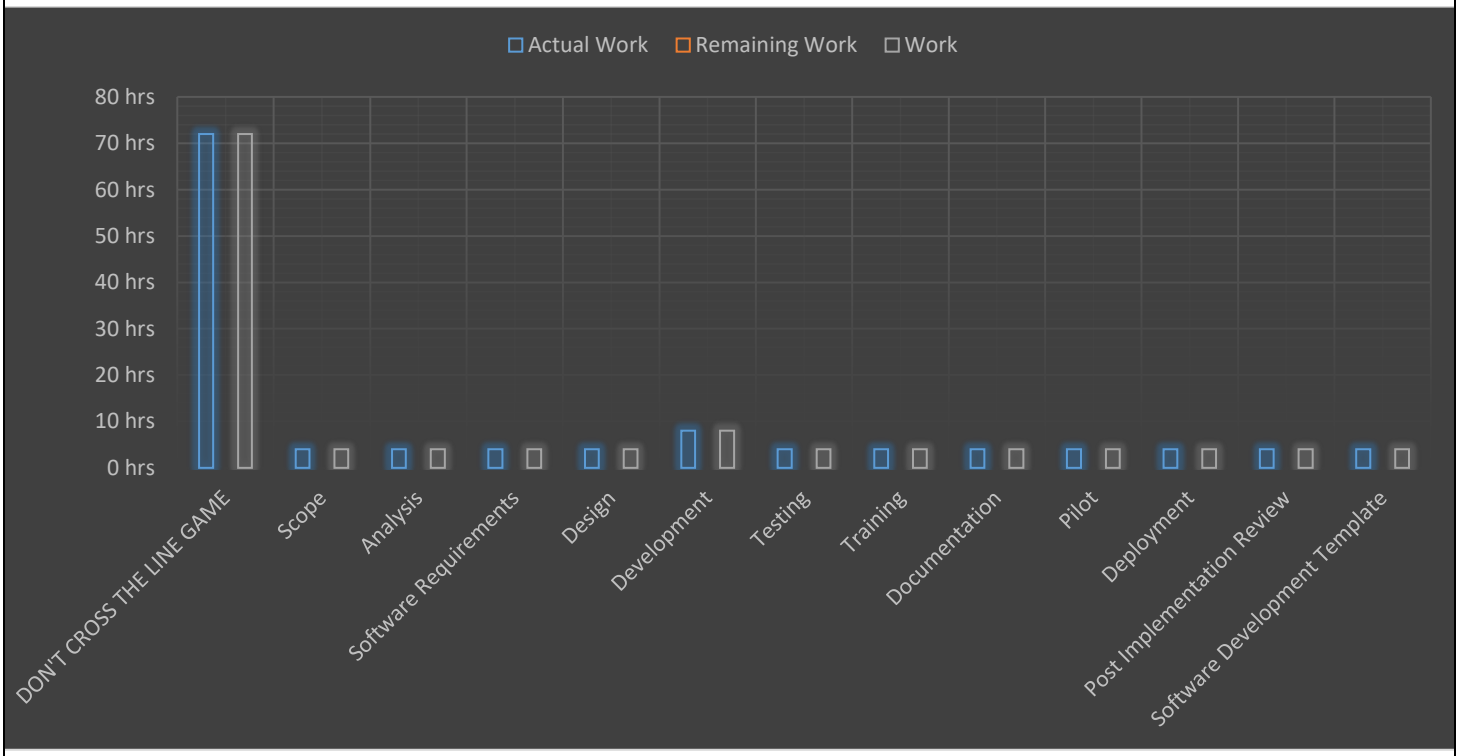
END OF SRS PROJECT

Gantt Chart

A Gantt chart is a project management tool that can be used to display a project and its milestones over time. It provides a visual guide of the start and end dates of different activities, the statuses of those activities, and whether or not those activities have been completed.

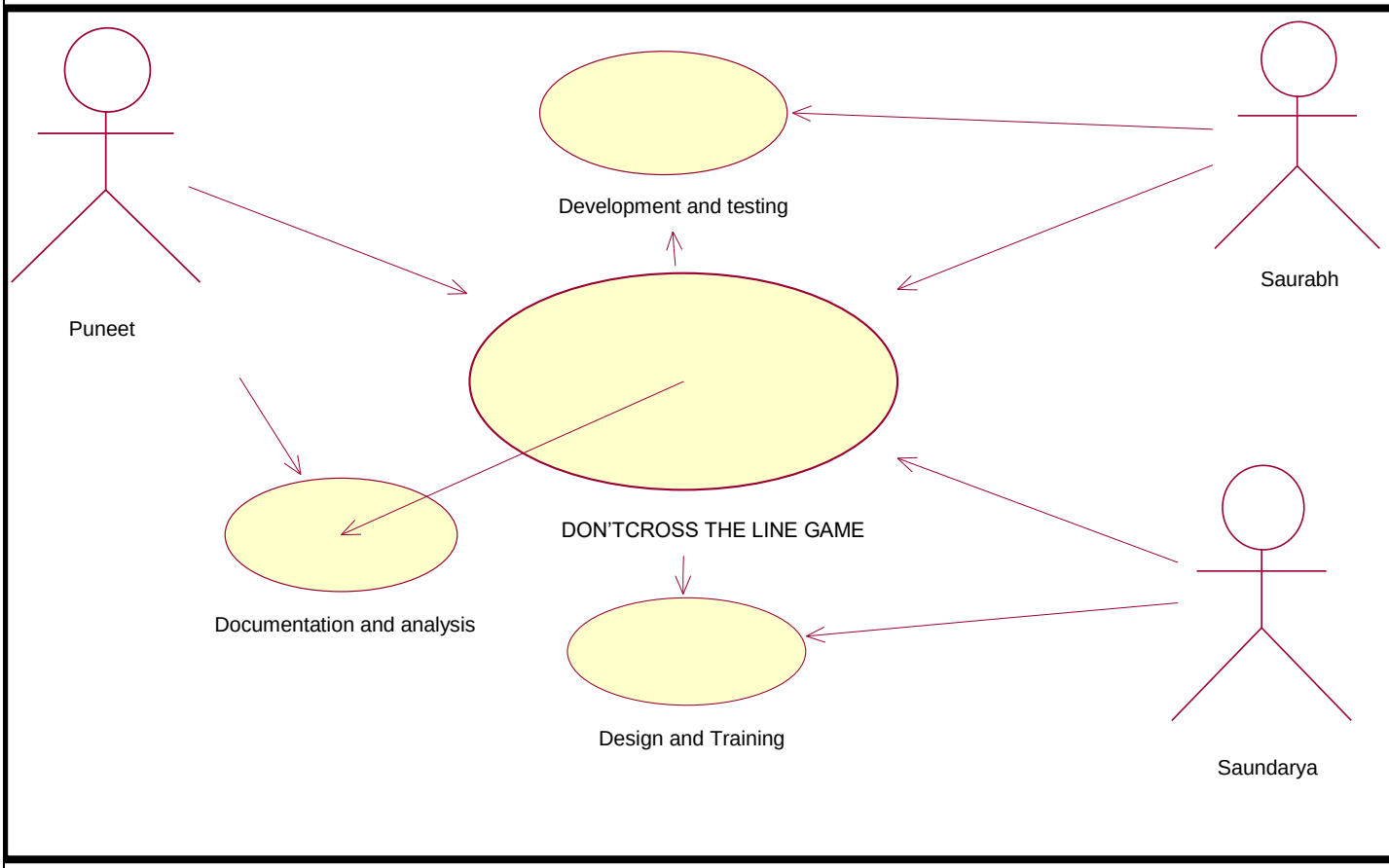


	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		DON'T CROSS THE LINE GAME	12 days	Mon 11-01-21	Tue 26-01-21	
2		Scope	1 day	Mon 11-01-2	Mon 11-01-2	
3	✓	Analysis	1 day	Tue 12-01-21	Tue 12-01-21	2
4	✓	Software Requirements	1 day	Wed 13-01-21	Wed 13-01-21	3
5	✓	Design	1 day	Wed 13-01-2	Wed 13-01-2	4
6	✓	Development	2 days	Thu 14-01-21	Fri 15-01-21	5
7	✓	Testing	1 day	Sat 16-01-21	Mon 18-01-2	6
8	✓	Training	1 day	Tue 19-01-21	Tue 19-01-21	7
9	✓	Documentation	1 day	Wed 20-01-2	Wed 20-01-2	8
10	✓	Pilot	0 days	Thu 21-01-21	Thu 21-01-21	9
11	✓	Deployment	1 day	Fri 22-01-21	Fri 22-01-21	10
12	✓	Post Implementation Review	1 day	Sat 23-01-21	Mon 25-01-21	11
13	✓	Software	1 day	Mon	Tue	12



Use Case Diagram

Use Case Diagrams are used to depict the functionality of a system or a part of a system. They are widely used to illustrate the functional requirements of the system and its interaction with external agents (actors). A use case is basically a diagram representing different scenarios where the system can be used. A use case diagram gives us a high level view of what the system or a part of the system does without going into implementation details.



Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<graphics.h>
#include<dos.h>

char ch1,ch2,ch='a';
int m,n;
int gta=35;
void direc();
void right1();
void right2();
void down1();
void down2();
void left1();
void left2();
void up1();
void up2();
void draw();
void check(int op);
void win1();
void win2();
void name();
void menu();
void help();
void about();
void lvl();
struct cood
{
  int x;
  int y;
}head1,tail1,head2,tail2;
char a[200][200];
char p1[20],p2[20];
void main()
{
  int gd=DETECT,gm,i,j;
  char cho;
  initgraph(&gd,&gm,"C:\\Turboc3\\BGI");
  head1.x=129;
  head1.y=239;
  tail1.x=129;
  tail1.y=239;
```

```
 head2.x=509;
 head2.y=239;
 tail2.x=509;
 tail2.y=239;
 for(i=0;i<200;i++)
 {
  for(j=0;j<200;j++)
  {
    a[i][j]='a';
  }
 }
 setbkcolor(BLACK);
 setcolor(YELLOW);
 rectangle(5,5,634,474);
 rectangle(6,6,633,473);
 settextstyle(TRIPLEX_SCR_FONT,HORIZ_DIR,7);
 outtextxy(200,250,"DON'T CROSS THE LINE");
 delay(2500);
 cleardevice();
 menu();
 cleardevice();
 rectangle(5,5,634,474);
 rectangle(6,6,633,473);
 rectangle(119,64,519,414);
 setlinestyle(SOLID_LINE, 0, 2);
 right1();
 left2();
 draw();
 getch();
 closegraph();
}


void direc(char ele)
{
if(ele=='d' && ch1!='a'){
        right1();
        ch1='d';}
else if(ele=='s' && ch1!='w'){
        down1();
        ch1='s'; }
else if(ele=='a' && ch1!='d'){
        left1();
        ch1='a';}
else if(ele=='w' && ch1!='s'){
```

```
            up1();
            ch1='w'; }
else if(ele=='j' && ch2!='l'){
             left2();
             ch2='j';}
else if(ele=='k'&& ch2!='i'){
            down2();
            ch2='k';}
else if(ele=='l' && ch2!='j'){
            right2();
            ch2='l'; }
else if(ele=='i'&& ch2!='k'){
            up2();
            ch2='i';}
else if(ele=='y')
            exit(0);
}


void right1()
{
 head1.x=head1.x+1;
 check(2);
 m=4;
}

void right2()
{
 head2.x=head2.x+1;
 check(6);
 n=4;
}

void left1()
{
  head1.x=head1.x-1;
  check(4);
  m=2;
 }

void left2()
{
 head2.x=head2.x-1;
 check(8);
 n=2;
```

```
}

void down1()
{
 head1.y=head1.y+1;
 check(1);
 m=3;
}

void down2()
{
 head2.y=head2.y+1;
 check(5);
  n=3;
}
void up1()
{
 head1.y=head1.y-1;
 check(3);
 m=1;
}

void up2()
{
 head2.y=head2.y-1;
 check(7);
 n=1;
}


void draw()
{
char ch;
ch1='d';
ch2='j';
while(1)
  {
  setcolor(CGA_YELLOW);
  circle(head1.x,head1.y,0.5);
  setcolor(CGA_RED);
  circle(head2.x,head2.y,0.5);
  if(kbhit())
  {
     ch=getch();
     direc(ch);
```

```
  }
  direc(ch2);
  direc(ch1);
  delay(gta);
 }
 }


 void check(int op)
 {
  int i,j,k,l;
  if(head1.x<119||head1.x>519||head1.y<64||head1.y>414)
  {
   win2();
  }
  if(head2.x<119||head2.x>519||head2.y<64||head2.y>414)
  {
   win1();
  }
  i=(int)(head1.x-50)/2;
  j=(int)(head1.y-50)/2;
  k=(int)(head2.x-50)/2;
  l=(int)(head2.y-50)/2;
  switch(op)
  {
   case 1:if(a[j+1][i]=='b'){win2();}break;
   case 2:if(a[j][i+1]=='b'){win2();}break;
   case 3:if(a[j-1][i]=='b'){win2();}break;
   case 4:if(a[j][i-1]=='b'){win2();}break;
   case 5:if(a[l+1][k]=='b'){win1();}break;
   case 6:if(a[l][k+1]=='b'){win1();}break;
   case 7:if(a[l-1][k]=='b'){win1();}break;
   case 8:if(a[l][k-1]=='b'){win1();}break;
  }
  a[j][i]='b';
  a[l][k]='b';
 }

 void win1()
 {
  delay(3000);
  cleardevice();
  rectangle(5,5,634,474);
  rectangle(6,6,633,473);
  outtextxy(200,200,"PLAYER 1 WINS!!!");
```

```
  outtextxy(400,400,"press any key to return...");
  getch();
  closegraph();
  main();
 }

 void win2()
 {
  delay(3000);
  cleardevice();
  rectangle(5,5,634,474);
  rectangle(6,6,633,473);
  outtextxy(200,200,"PLAYER 2 WINS!!!");
  outtextxy(400,400,"press any key to return...");
  getch();
  closegraph();
  main();
 }


 void menu()
 {
  char f;
  setbkcolor(BLACK);
  setcolor(YELLOW);
  rectangle(5,5,634,474);
  rectangle(6,6,633,473);
  settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR,1);
  outtextxy(70,70,"DON'T CROSS THE LINE");
  outtextxy(250,200,"1) NEW GAME");
  outtextxy(250,220,"2) LEVEL");
  outtextxy(250,240,"3) HELP");
  outtextxy(250,260,"4) ABOUT");
  outtextxy(250,280,"5) EXIT");
  f=getch();
  switch(f)
  {
    case '1':break;
    case '3':help();menu();break;
    case '4':about();menu();break;
    case '2':lvl();menu();break;
    case '5':exit(0);
    default:menu();
  }
 }
```

```
void help()
{
 int FONT;
 cleardevice();
 rectangle(5,5,634,474);
 rectangle(6,6,633,473);
 gotoxy(30,2);
 settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
 outtextxy(70,70,"RULES:");
 settextstyle(FONT,HORIZ_DIR,3);
 outtextxy(205,100,"DO NOT TOUCH THE BOUNDARY LINE");
 outtextxy(205,130,"DO NOT TOUCH THE OPPONENT");
 outtextxy(205,160,"DO NOT OVERLAP ON YOUR OWN PATH");
 settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
 outtextxy(70,250,"CONTROLS:");
 settextstyle(FONT,HORIZ_DIR,3);
 outtextxy(205,280,"W,A,S,D KEYS FOR PLAYER 1");
 outtextxy(205,310,"I,J,K,L KEYS FOR PLAYER 2");
 outtextxy(400,400,"press any key to return...");
 getch();
 cleardevice();
}

void about()
{
 int FONT;
 cleardevice();
 rectangle(5,5,634,474);
 rectangle(6,6,633,473);
 settextstyle(TRIPLEX_FONT,HORIZ_DIR,3);
 outtextxy(265,50,"!!!DISCLAIMER!!!");
 settextstyle(FONT,HORIZ_DIR,3);
 outtextxy(75,100,"THE CONCEPT OF THIS GAME IS INSPIRED FROM A MINI GAME IN GTA 5");
 outtextxy(300,280,"THIS GAME IS CREATED BY,");
 outtextxy(300,300,"STUDENTS OF BE CSE");
 outtextxy(340,320,"PUNEET SG18337");
 outtextxy(340,340,"SAUNDARYA SG18344");
 outtextxy(340,360,"SAURABH SG18345");
 outtextxy(400,400,"press any key to return...");
 getch();
 cleardevice();
}

void lvl()
```

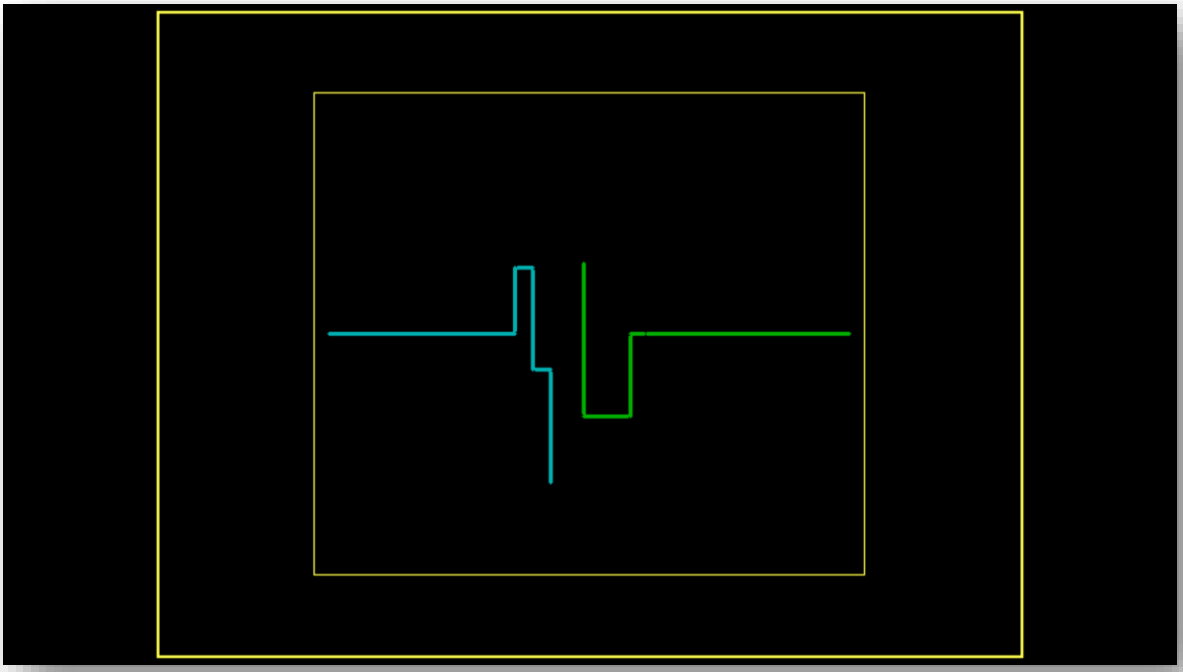
```
        {
          char GTA5;
          cleardevice();
          rectangle(5,5,634,474);
          rectangle(6,6,633,473);
          outtextxy(70,70,"SELECT YOUR LEVEL");
          outtextxy(250,200,"1) EASY");
          outtextxy(250,220,"2) MEDIUM");
          outtextxy(250,240,"3) DIFFUCULT");
          GTA5=getch();
          switch(GTA5)
          {
            case '1':gta=gta+10;break;
            case '2':gta=gta;break;
            case '3':gta=gta-10;break;
            default:gta=gta;
          }
          cleardevice();
}
```

Screenshots: Final Project

Starting Page:

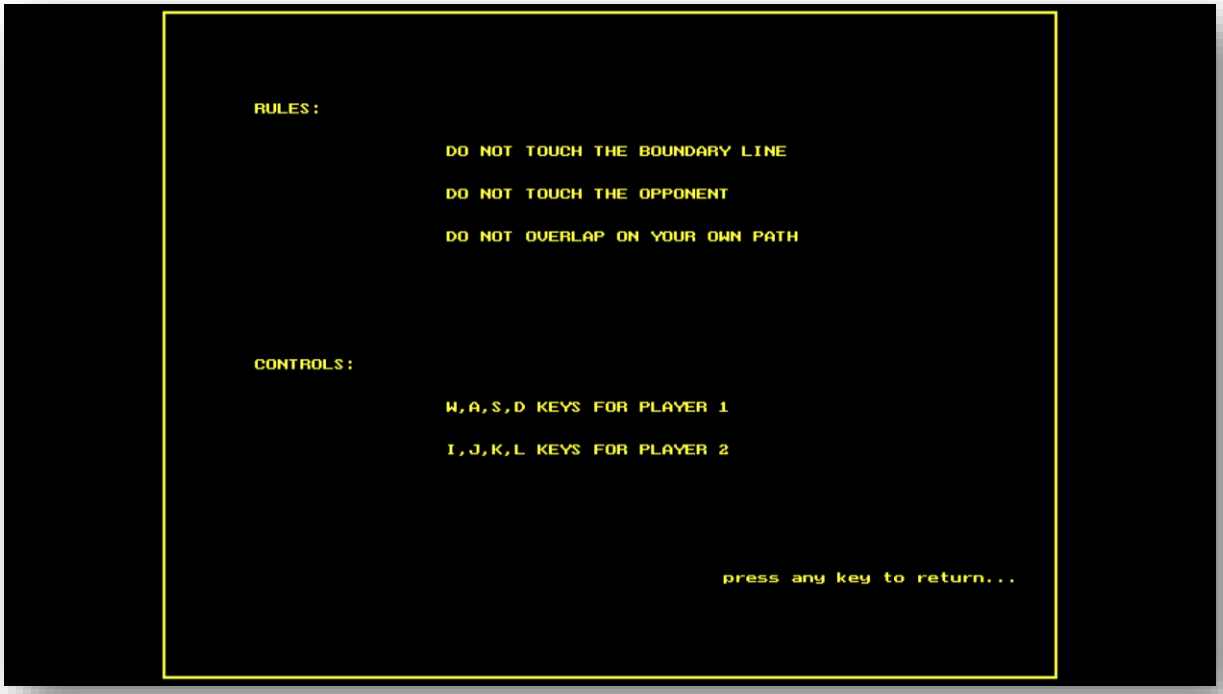
Menu Page:

In-game Page:

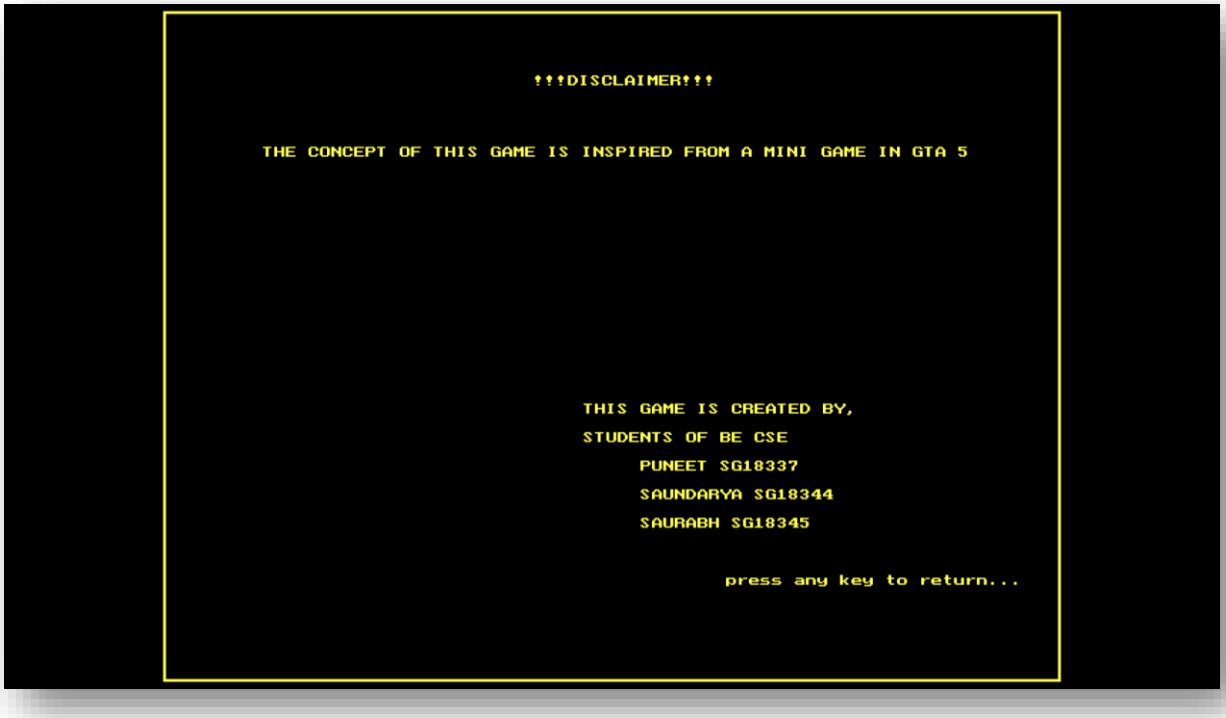


Level Page:

Help Page:



About Page:



Player 1 Wins Page:

Player 2 Wins Page: