ACKNOWLEDGEMENT

I would like to owe the sense of gratitude to our principal ma'am Mrs. Shanthy Asokan for providing me with this wonderful opportunity and our computer science teacher Mrs.L.Hemalatha for her consistent guidance, support and encouragement without which our project would not exist.

I would also like to thank our lab assistant Mr.S.Sasikumar, my parents for their blessings and my friends for their help in order to complete this project.

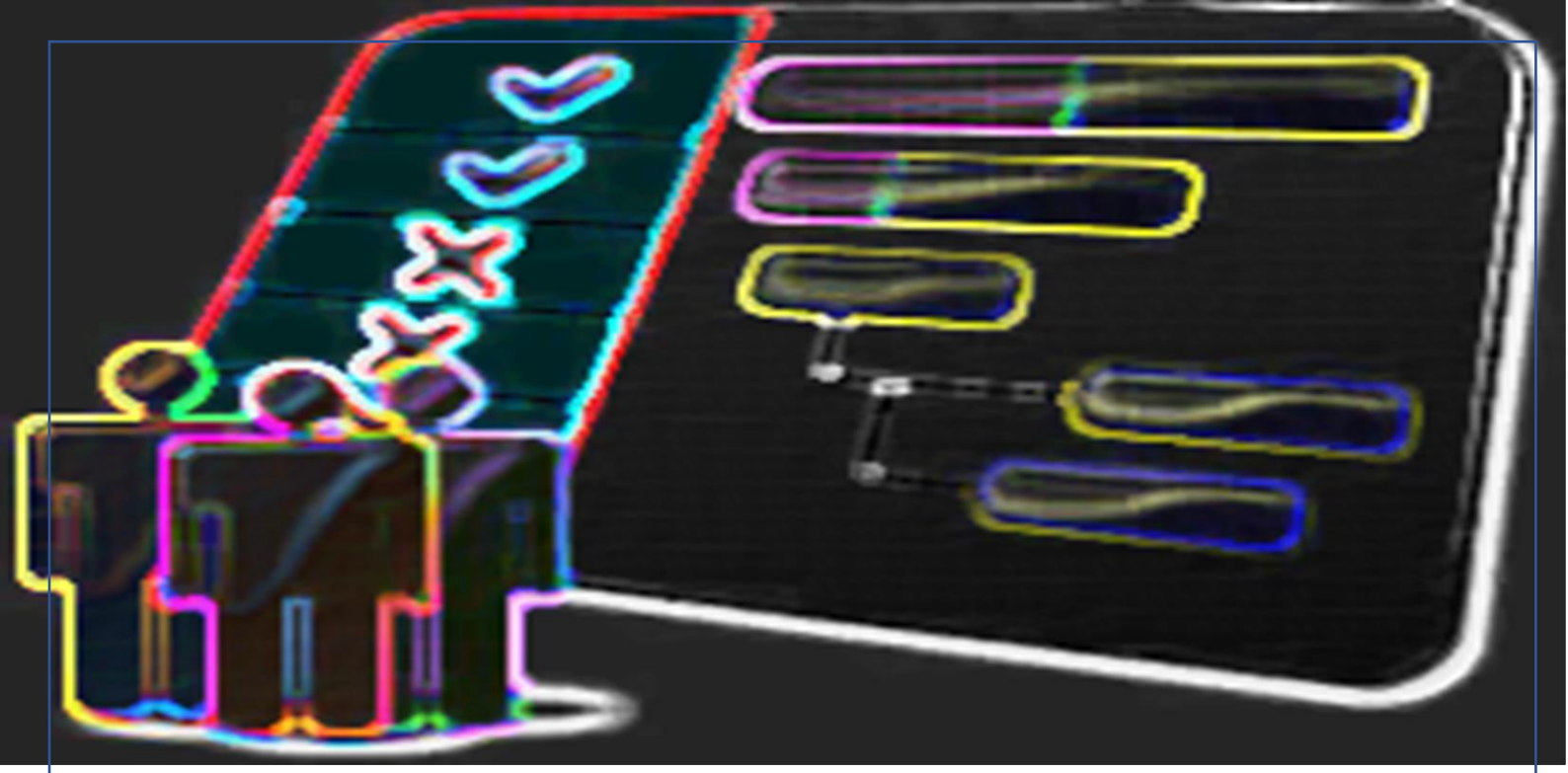
TABLE OF CONTENTS

INTRODUCTION

The project we have done is a multiplayer trading game which is similar to the board game Monopoly or commonly known as Business.

The reason I took up this project is that I always loved playing board games such as this with my friends and family and I wanted everyone to also experience this but the problem with board games was that they used cards and boards that get damaged easily and needed to be properly sorted each time, which wasted a lot of time but playing in a computer will help to overcome these difficulties.

Secondly, I wanted to do a project that was challenging, unique and reflected upon my passion for coding and love for computers in general.

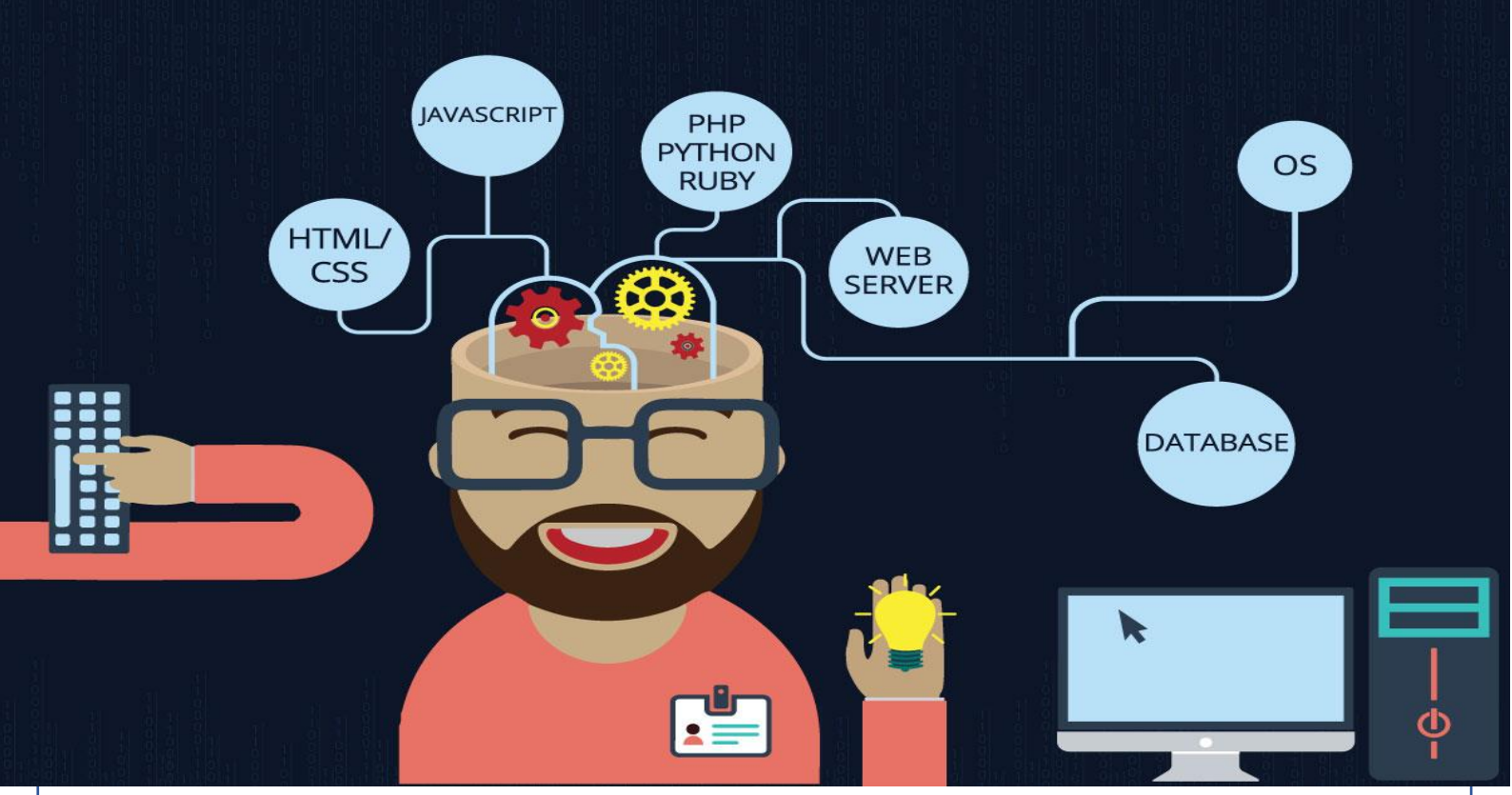


PROGRAM BRIEFING

The game we created is for 2-6 players where the players can roll a die to move around the board buying various sites to become the richest player without losing all their lives.

This system of lives for players is our own creative idea where the player loses one of his life whenever he is unable to pay any debt to another player or to the computer.

We have not incorporated some features such as building houses from the original Monopoly game and instead added a few of our own innovative touches to make the game fast, fun and friendly.



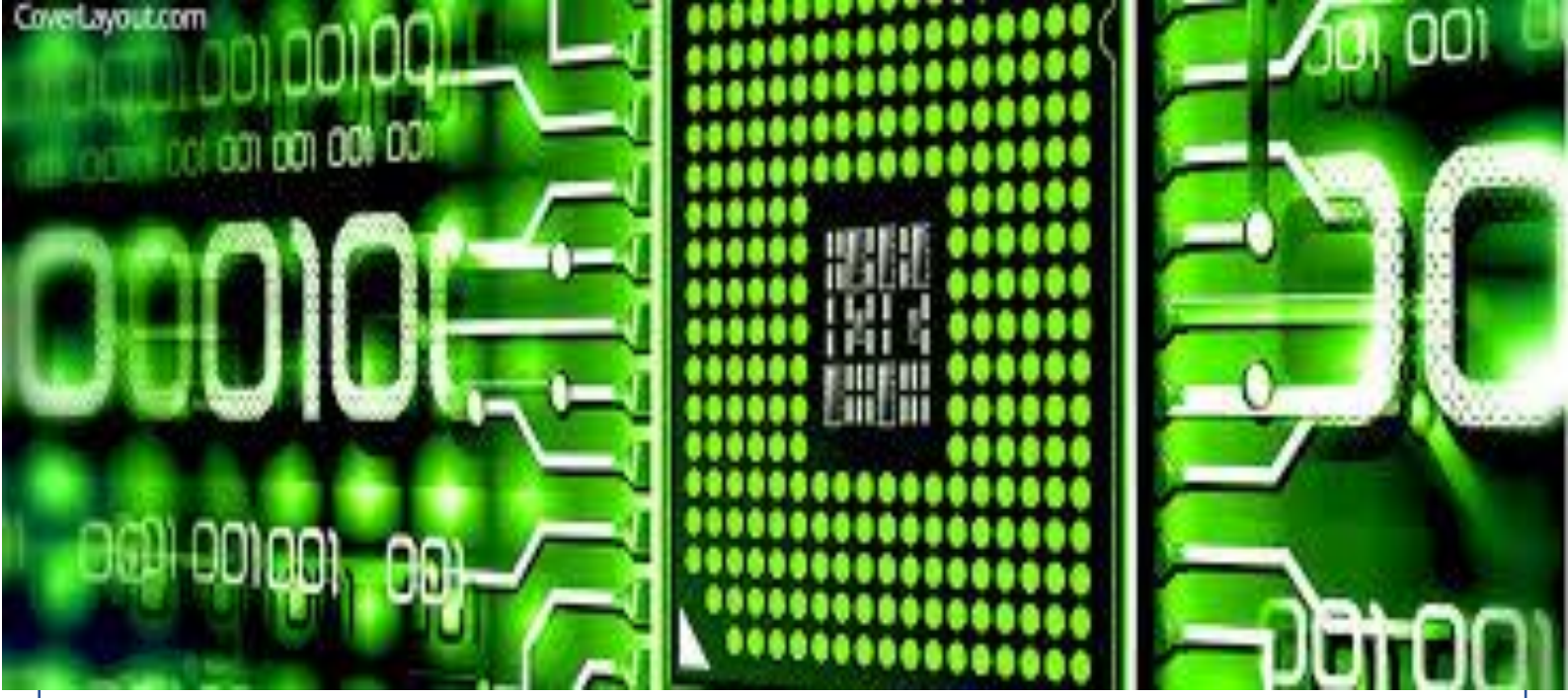
PRE-REQUISITE KNOWLEDGE

To be able to understand this program, one needs to have prior basic knowledge about the game mechanics and general rules of a similar trading game.

The coding uses C++ language, basic concepts, creating and using user-defined functions, implementation and application of classes and objects, complex looping mechanisms, file handling and extensive knowledge about graphical functions in c++ (graphics.h).

PROGRAMMING METHODOLOGY

Accept the no. of players (2-6) and accept their respective names and color of the mipmap indicator, Sort the players based on alphabetical order of their names and let them roll the die by pressing any key in that same order, Move the player accordingly and if the player lands on an unowned site, ask if the player wants to buy it and perform accordingly, if the site is owned by another player, he has to pay rent to the owner to continue or if the player lands on a mystery chest or special event, perform a random command or corner site perform accordingly, display menu check if the player wants to sell one of his sites, trade with another player, quit the game or end his turn and perform accordingly and then pass the turn to the next player and keep on going till all the sites are sold or only one player remains. The player with the highest score which is calibrated based on the no. of rounds they have completed, no. of sites they own and the gold they have is declared as the winner.

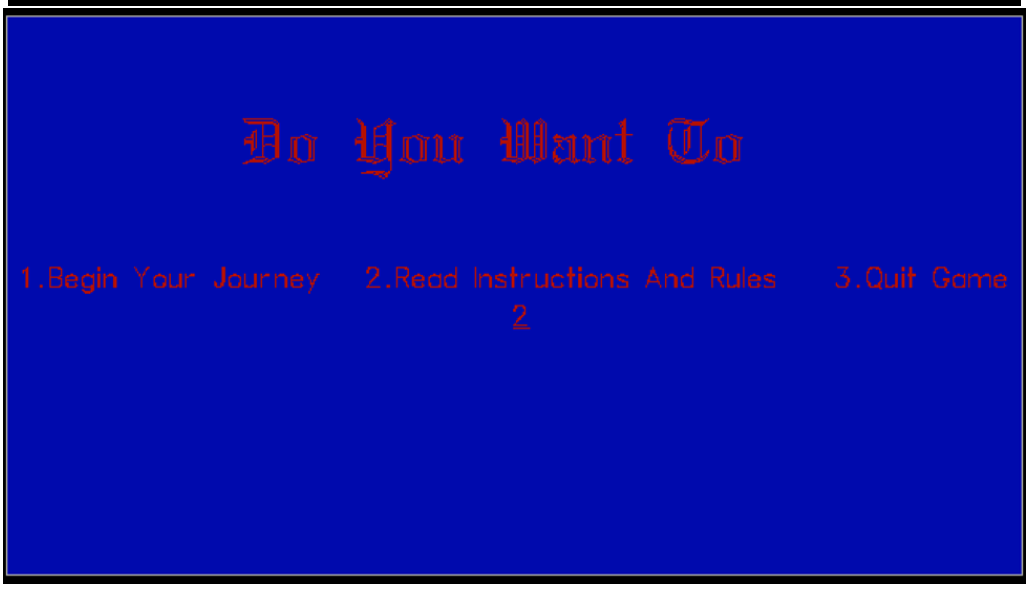


ALGORITHM

1. Start
2. Accept No. of players
3. Using loop accept their details.
4. Sort the players acc. to alphabetical order.
5. Within a loop the following is executed till no. of players is 1 or no. of available sites is 0:
   - Roll the die by pressing any key
   - Based on which site the player lands, perform action accordingly
   - Display menu and if choice is 1-call function sell
     2-call function trade
     3-call function quit
     4-end turn and pass it to another player
   - If no. of lives =0 then call the function quit.
6. Sort the players based on their scores and display the one with the highest score as winner
7. Stop

ADDITIONAL SCREENS

1. Title Screen and Intro Menu



2. Instructions Screen (2 Pages)

Instructions:

1.Enter the no. of players.
2.Enter the name and colour of the mipmap indicator for each player.
3.Press any key other than Esc to roll.
4.If you land on a unowned site, choose if you want to buy it, If owned you will have to play the rent to the owner.
5.On the menu choose if you would like to sell one of your sites for gold, trade with other player to get one of his site for gold, quit to delete only your assets without ending the game, or end your turn and the next player will play.
6.Landing on Mystery chest or special event will trigger a random command.
7.If you land on one of the portals you can use it to travel to the next portal by using fast travel even if you dont own the site.
8.Landing on Guild or Tavern will trigger no event.
9.Landing on Jail directly will not trigger any event but landing on forbidden forest will teleport you automatically to jail and you will have to pay a price for your release.
10.Focus on getting more sites and completing more rounds but gold is also important as they are the deciding factors for the winner of the game.

Rules:

1.This game can be played only by 2-6 players.
2.All inputs must be made after '=>' symbol.
3.No 2 players can use exactly identical names.
4.Same mipmap indicator colors can be used by more 1 player.
5.Pressing Esc key allows you to exit the game anytime but it won't display the winner.
6.For the player to make their choice enter only the respective serial no. of the choice in the list (integer value), Entering other values might cause errors.
7.Inputs should not be repeatedly spammed or entered more than required,
This will lead to that input being accepted for suceeding input statements.
8.Each player begins with 3 lives, these lives are reduced whenever that player is unable to pay rent to other player or to the bank and when all the lives are lost that player automatically loses.
9.The game will end only when all the sites are sold and winner is chosen based the on of sites owned, gold and no of rounds completed or if only one player remains.
10.Trading must be done with mutual understanding only,please try to make the trades equally profitable for both sides.
11.Selling your site would give you only the 75% of its cost.

(Press any key to change the page)
(Press Esc to return to title screen)

Game Entities


- Player Mipmap Indicator


- Guild


- Jail


- Tavern


- Forbidden Forest


- Ownable sites


- Mystery Chest


- Special Event


- Portal

CREDITS

THANK YOU FOR PLAYING, THIS GAME WAS DONE BY
S.BALAVIGNESH AND T.K.SIVARAAM

FOR COMPUTERSCIENCE BOARD PROJECT OF YEAR 2019 AKA
THE FINAL YEAR OF C++ BEING A PART OF CBSE SYLLABUS
THIS PROJECT WOULD HAVE NOT BEEN POSSIBLE WITHOUT THE
SUPPORT FROM OUR COMPUTERSCIENCE FACULTY
MRS L.HEMALATHA

IF YOU FIND ANY BUG(s) PLEASE CONTACT US THROUGH =>
balviky1@gmail.com
sivaraamtk@hotmail.com

(Press any key to change the page)
(Press Esc to return to title screen)

3. Winner Screen and Credits Screen

OBJECTS AND THEIR USE

Classes Used

1. Class name➔ player
   - Members in private scope
     - Player name – character array
     - Player position – integer
     - Player coin – integer
     - Round number – integer
     - Number of sites owned – integer
     - Number of lives – integer
     - Player gold – floating point value
   - Members in public scope
     - *Void create(integer n)* – inputs all necessary details form players like name player color for n players
     - *Void buy()* – this function allows the user(player) to buy any unowned site in the game by checking the sites price from the file named SPACE.dat and the players remaining gold stash this function changes the site owner in the file(SPACE.dat)
     - *Void sell()* – this function allows the user(player) to sell his site for 75% of the cost price of the site and change the owner of the site in file back to the null terminator(\0)

- *Void roll()* – this function enables the user(player)to roll the die and it generates a random value in the range of 1-12(inclusive), the generated value is stored in global variable and calls the function coind() .
- *Void trade()* – this function displays all the players name to the current player except his name to choose whose site he wants to buy a player will be selected by the user(player). The function also checks if the selected player owns any site or not. If the player owns any site name of it is displayed for selection by the current player. The price of the site is proposed by the current player. The proposed price for the site is requested as a deal between the current player and the selected player in which the selected player can choose whether he accepts the deal or not. If the deal is accepted the site has been sold to the current player by the selected player for the decided price.
- *Void rent(space sr,player p)* – This function runs a for loop from I=0 to I=number of players -1 and stops if the player name given as an argument is same as the name returned in the loop. The variable I represents the player number in memory. The function checks if the selected players gold stash is greater than the rent, which in turn calculated by sum of rent of site which is passed as argument and (round number of the player ×10×level of the site owned). If the difference of player gold stash and rent falls below or equal to 0, appropriate message is displayed and a life is deducted else the rent is subtracted from the players gold stash and the player is allowed to stay in the site and continue the game.
- *Void mystch()* – This function opens the text file MYSTCH.txt in input mode(read only mode) for reading from the file the flavored text and the action is performed accordingly the action is being chosen randomly from the available five choices from the file.
- *Void splevt()* – This function opens the text file SPLEVT.txt in input mode(read only mode) for reading from the file the flavored text and the action is performed accordingly the action is being chosen randomly from the available five choices from the file.
- *Void coind()* – This function stores the current player position in a variable current position and reads the random value which was

generated from the function roll stored in the global variable and adds it to the player position. The function then runs a for loop form the current position to the player positon the animation of movement of player coin across the board is rendered accordingly

- *Void chspace()* - This function is used to retrieve the sites information from the file SPACE.dat and check the details weather the site is a unowned site ,a visiting site, a special event place, a mystery chest place, a place owned by any 1 of the players , the players site itself or a portal and invokes the appropriate function and tasks. If the player visits his own site, his site will be renovated and his rents will be increased.
- *Void ftravel()* – This function is used by *chspace()* function and this teleports the player to the next portal across the board telepotatin costs 150×round number.
- *Void dispmenu()* – This function displays the player the available options like sell, trade, quit, end turn. The choice is accepted and appropriate functions are called.
- *Void disppos()* – This function is used to display a static player position on the board until his turn ends.
- *Int ret_nos()* – This function returns number of sites of the player.
- *Void pay (float f)* – This function is used to reduce f amount of gold from a given player.
- *Int ret _ life()* – This function is used to return the number of lifes of the player.
- *Void collect (float f)* – This function is used add f amount of gold to the given player.
- *Char* ret _ pname()* – This function is used to return the name of the player.
- *Int ret _ rndno()* – This function is used to return the round number of the player.
- *Void ssite()* – This function is used after selling the site to reduce the number of sites owned by the player inorder to maintain the points in balance.
- *Float points()* – This function is used to calculate the points for the player by number sites×1000 + player gold×100 + round number×500.

2. Class name ➔ space

- Members in protected scope
  - Site cost – floating point value
  - Site rent – floating point value
  - Site name – character array
  - Site owner – character array
- Members in public scope
  - *Void chgowner(char nonwer[25])* – This function is used to change the name of the site owner.
  - *Char** ret_sowner() – This function is used to return the name of the site owner.
  - *Char* ret_sname()* – This function is used to return the name of the site.
  - *Float ret_scost()* – This function is used to return the cost of the site.
  - *Float ret_srent()* – This function is used to return the rent of the site.

3. Class name ➔ non_site

   Parent class ➔ public space

- Members in public scope
  - non_site() – This is a constructer used to initialize all the data members of class space to 0's and null's respectively.

Global Functions Used

1. *void ptr(int on*) – this function turns the little underscore cursor when we are typing for the input in the graphic mode in c++
2. *void iip(char inputstring[9999])* – this function is used as a input function in the program this function gets the character using getch() dynamically and writes it to a sting character by character and displays it simultaneously on the graphical interface
3. *void mainscr()* – this function is used to create the brief into secreen on our game where the user(s) can choose what they want to do
4. *void instr()* – this function creates 2 active graphical pages in the memory and draws the instruction, rules, game entities and credits on to the screen and cycles them until the user presses "*Esc*" key
5. *void intro()* – this function uses one of graphical screen to print the welcome message at the start of the game
6. *void clrp()* – this function is used to clear the entire screen along with the monopoly board and redraws it and splits the console into 2 parts (i.e.)active part and the static part in which the active part involves in all the input actions in the game and the static part is just for the movement of the coin and displaying the board
7. *void dip(char t[9999])* – this function use the method of traversing a character array by each character and prints it on the graphical console with all the text styles retained this also makes sure that the text does not overflow the range of the active part and tuncates the sentence and displays it in the next line
8. *void gbord()* – this is the main function of the program sice this function helps making the whole monopoly bord be displayed on the screen by splitting left half for the board and the right half for the input purpose

9. *void sitesown(int n)* – this function is used to assign 1's and 0's to a 2d integer array o[6][36] which holds the detail weather the site is owned by that player or not this also assigns the value of 1 and 0 to the level of the site for each player
10. *void quit()* – this function enables the player to quit the game at any point of time during his chance this function sells all the sites of the player and removes him from the player list by sorting him to the last position in the array of players
11. *void endg()* – this function is used for Urgent all together EXIT of the game it rewrites all the site owners to null when "*Esc*" key is pressed during an input
12. *void end()* – this function uses an global variable pass to check if the game was quitted from the main screen or at the end of the game if the function is called at the end of the game it calculates the point and displays the winner along with the credit screen else only credit screen is printed

Files Used

There are 3 files used in this program they are as follows ➔

1. SPACE.dat – This is the binary file containing all the details of the all the sites in the monopoly game board this file is used for reading site details and writing the sites owners

2. MYSTCH.txt - This text file contains all the flavored text necessary for the mystery chest site in the game

3. SPLEVT.txt – This text file contains all the flavored text necessary for the Special event site in the game

These files contain the necessary details for the games full functioning

PROGRAM CODE

```
#include<fstream.h>
#include<conio.h>
#include<dos.h>
#include<string.h>
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#define svpi setviewport(402,2,634,346,1);setcolor(0)
#define svp setviewport(0,0,getmaxx(),getmaxy(),1)
int l=-1,nop,turn=0,dice,avs=24,o[6][36],ht,ch,x=50,y=304,pass,endt=0,r=-
1,lvl[6][36];
char ch1[20];
void end();
void endg();
void dip(char t[999]);
void ptr(int on)
{
 int curX;
 char uBarStr[2] = { '_',0 };
 if (!on)
 setcolor(63);
 curX=getx();
 outtext(uBarStr);
 moveto(curX,gety());
 if (!on) setcolor(0);
}
void iip(char inputString[9999])
{
 int stringIndex=0,xVal[9999];
 char inputChar,outString[2];
 outString[1]='\0';
 xVal[0]=getx();
 do
 {
```

```
  ptr(1);
  inputChar=getch();
  ptr(0);
  if (inputChar==0) getch();
  else if(inputChar==27) { endg(); exit(0); }
  else
  {
   if (inputChar==8)
   {                  // backspace
    --stringIndex;
    if (stringIndex<0) stringIndex=0;
     moveto(xVal[stringIndex],gety());
    setcolor(63);
    outString[0]=inputString[stringIndex];
    outtext(outString);
    moveto(xVal[stringIndex],gety());
    setcolor(0);
   }
   else
   {
    inputString[stringIndex]=inputChar;
    outString[0]=inputChar;
    outtext(outString);
    ++stringIndex;
    xVal[stringIndex]=getx();
   }
  }
 }while(inputChar!=13 && inputChar!=10);
 inputString[stringIndex]=0;
}
void mainscr();
void instr()
{
 svp;
 char c;
 cleardevice();
 setactivepage(0);
 setfillstyle(1,63);
 setlinestyle(0,0,1);
 rectangle(2,0,getmaxx()-2,getmaxy());
 floodfill(5,5,63);
 setcolor(0);
 settextstyle(2,0,4);
 moveto(getmaxx()-39*textwidth("W"),getmaxy()-textheight("W")-1);
 outtext("(Press Esc to return to title screen)");
 moveto(getmaxx()-36*textwidth("W"),getmaxy()-2*textheight("W")-1);
 outtext("(Press any key to change the page)");
 moveto(getmaxx()/2,(++r)*ht+2);
 outtext("Instructions:");
 moveto(2,(++r)*ht+2);
 outtext("1.Enter the no. of players.");
 moveto(2,(++r)*ht+2);
 outtext("2.Enter the name and colour of the mipmap indicator for each
player.");
 moveto(2,(++r)*ht+2);
 outtext("3.Press any key other than Esc to roll.");
 moveto(2,(++r)*ht+2);
 outtext("4.If you land on a unowned site, choose if you want to buy it, If
owned you will have to pay the ");
```

```
 moveto(2,(++r)*ht+2);
 outtext("  rent to the owner.");
 moveto(2,(++r)*ht+2);
 outtext("5.On the menu choose if you would like to sell one of your sites for
gold, trade with other player to get one ");
 moveto(2,(++r)*ht+2);
 outtext("  of his site for gold, quit to delete only your assets without
ending the game, or end your turn and the next ");
 moveto(2,(++r)*ht+2);
 outtext("  player will play. ");
 moveto(2,(++r)*ht+2);
 outtext("6.Landing on Mystery chest or special event will trigger a random
command.");
 moveto(2,(++r)*ht+2);
 outtext("7.If you land on one of the portals you can use it to travel to the
next portal by using fast travel even if ");
 moveto(2,(++r)*ht+2);
 outtext("  you dont own the site. ");
 moveto(2,(++r)*ht+2);
 outtext("8.Landing on Guild or Tavern will trigger no event.");
 moveto(2,(++r)*ht+2);
 outtext("9.Landing on Jail directly will not trigger any event but landing on
forbidden forest will teleport you ");
 moveto(2,(++r)*ht+2);
 outtext("  automatically to jail and you will have to pay a price for your
release.");
 moveto(2,(++r)*ht+2);
 outtext("10.Focus on getting more sites and completing more rounds but gold is
also important as they are the deciding ");
 moveto(2,(++r)*ht+2);
 outtext("  factors for the winner of the game.");
 r+=2;
 setlinestyle(0,0,3);
 line(2,r*ht+2,getmaxx()-2,r*ht+2);
 moveto(getmaxx()/2,r*ht+2);
 outtext("Rules:");
 moveto(2,(++r)*ht+2);
 outtext("1.This game can be played only by 2-6 players.");
 moveto(2,(++r)*ht+2);
 outtext("2.All inputs must be made after '=>' symbol.");
 moveto(2,(++r)*ht+2);
 outtext("3.No 2 players can use exactly identical names.");
 moveto(2,(++r)*ht+2);
 outtext("4.Same mipmap indicator colors can be used by more 1 player.");
 moveto(2,(++r)*ht+2);
 outtext("5.Pressing Esc key allows you to exit the game anytime but it won't
display the winner.");
 moveto(2,(++r)*ht+2);
 outtext("6.For the player to make their choice enter only  the respective
serial no. of the choice in the list");
 moveto(2,(++r)*ht+2);
 outtext("  (integer value), Entering other values might cause errors.");
 moveto(2,(++r)*ht+2);
 outtext("7.Inputs should not be repeatedly spammed or entered more than
required,");
 moveto(2,(++r)*ht+2);
 outtext("  This will lead to that input being accepted for suceeding input
statements.");
 moveto(2,(++r)*ht+2);
```

```
 outtext("8.Each player begins with 3 lives, these lives are reduced whenever
that player is unable to pay rent to");
 moveto(2,(++r)*ht+2);
 outtext("  other player or to the bank and when all the lives are lost that
player automatically loses.");
 moveto(2,(++r)*ht+2);
 outtext("9.The game will end only when all the sites are sold and winner is
chosen based the on of sites owned,");
 moveto(2,(++r)*ht+2);
 outtext("  gold and no of rounds completed or if only one player remains.");
 moveto(2,(++r)*ht+2);
 outtext("10.Trading must be done with mutual understanding only,please try to
make the trades equally profitable for ");
 moveto(2,(++r)*ht+2);
 outtext("  both sides.");
 moveto(2,(++r)*ht+2);
 outtext("11.Selling your site would give you only the 75% of its cost.");
 r=-1;
 setactivepage(1);
 setfillstyle(1,63);
 setlinestyle(0,0,1);
 rectangle(2,0,getmaxx()-2,getmaxy());
 floodfill(5,5,63);
 setcolor(0);
 settextstyle(2,0,4);
 moveto(getmaxx()-39*textwidth("W"),getmaxy()-textheight("W")-1);
 outtext("(Press Esc to return to title screen)");
 moveto(getmaxx()-36*textwidth("W"),getmaxy()-2*textheight("W")-1);
 outtext("(Press any key to change the page)");
 moveto(getmaxx()/4,(++r)*ht+2);
 outtext("Game Entities");
 setlinestyle(0,0,3);
 line(getmaxx()/2,0,getmaxx()/2,getmaxy());
 int o=getmaxx()/2+3;
 moveto((getmaxx()/4)*3,r*ht+2);
 outtext("CREDITS");
 moveto(o,(++r)*ht+2);
 outtext("THANK YOU FOR PLAYING, THIS GAME WAS DONE BY");
 moveto(o,(++r)*ht+2);
 outtext("S.BALAVIGNESH AND T.K.SIVARAAM");
 moveto(o,(++r)*ht+2);
 moveto(o,(++r)*ht+2);
 outtext("FOR COMPUTERSCIENCE BOARD PROJECT OF YEAR 2019 AKA");
 moveto(o,(++r)*ht+2);
 outtext("THE FINAL YEAR OF C++ BEING A PART OF CBSE SYLLABUS");
 moveto(o,(++r)*ht+2);
 outtext("THIS PROJECT WOULD HAVE NOT BEEN POSSIBLE WITHOUT THE");
 moveto(o,(++r)*ht+2);
 outtext("SUPPORT FROM OUR COMPUTERSCIENCE FACULTY ");
 moveto(o,(++r)*ht+2);
 outtext("MRS L.HEMALATHA");
 moveto(o,(++r)*ht+2);
 moveto(o,(++r)*ht+2);
 outtext("IF YOU FIND ANY BUG(s) PLEASE CONTACT US THROUGH =>");
 moveto(o,(++r)*ht+2);
 outtext("balviky1@gmail.com");
 moveto(o,(++r)*ht+2);
 outtext("sivaraamtk@hotmail.com");
 setfillstyle(SOLID_FILL,58);
```

```
setcolor(0);
r=0;
setlinestyle(0,0,3);
setaspectratio(getmaxx()/2,getmaxy());
circle(30,30,10);
floodfill(30,30,0);
setcolor(56);
setcolor(0);
r+=2;
moveto(60,r*ht+2);
outtext("- Player Mipmap Indicator");
setfillstyle(1,63);
 r+=3;
for(int i=0;i<8;i++)
{
 setcolor(0);
 moveto(60,r*ht+2);
 if(i==0)
 {
   setfillstyle(1,1);
   outtext("- Guild");
 }
 else if(i==1)
 {
  setfillstyle(1,4);
   outtext("- Jail");
 }
 else if(i==2)
 {
  setfillstyle(1,5);
   outtext("- Tavern");
 }
 else if(i==3)
 {
  setfillstyle(1,2);
  outtext("- Forbidden Forest");
 }
 else if(i==4)
  {
   setfillstyle(1,60);
   outtext("- Ownable sites");
  }
 else if(i==5)
 {
  setfillstyle(1,59);
  outtext("- Mystery Chest");
 }
 else if(i==6)
 {
  setfillstyle(1,57);
   outtext("- Special Event");
 }
 else if(i==7)
 {
  setfillstyle(1,61);
   outtext("- Portal");
 }
 rectangle(20,50+(i*35),40,70+(i*35));
 floodfill(22,52+(i*35),0);
```

```
  r+=4;
 }

 r=-1;
 s:
 setvisualpage(0);
 c=getch();
 if(c==27)
 goto e;
 else
 goto b;
 b:
 setvisualpage(1);
 c=getch();
 if(c==27)
 goto e;
 else
 goto s;
 e:
 mainscr();
}
void mainscr()
{
 setactivepage(0);
 setvisualpage(0);
 cleardevice();
 setlinestyle(0,0,2);
 int ply[10];
 ply[0]=2;       ply[1]=0;
 ply[2]=2;       ply[3]=getmaxy();
 ply[4]=getmaxx()-2;  ply[5]=getmaxy();
 ply[6]=getmaxx()-2;  ply[7]=0;
 ply[8]=2;       ply[9]=0;
 setcolor(7);
 setfillstyle(1,1);
 drawpoly(5,ply);
 fillpoly(5,ply);
 int t=0;
 setcolor(RED);
 settextstyle(4,0,5);
 moveto(150,50);
 outtext("Do You Want To ");
 settextstyle(3,0,1);
 moveto(10,150);
 outtext("1.Begin Your Journey   2.Read Instructions And Rules    3.Quit
Game");
 char ch;
 moveto(getmaxx()/2,getmaxy()/2+2);
 outtext("_");
 a:
 moveto(getmaxx()/2,getmaxy()/2);
 ch=getch();
 if( ch!=27 && (ch!=10 || ch!=13) && (ch>48 && ch<52) )
 {
  if(ch==49)
  {
   t=1;
   moveto(getmaxx()/2,getmaxy()/2);
   outtext("1");
```

```
   }
   else if(ch==50)
   {
    t=2;
    moveto(getmaxx()/2,getmaxy()/2);
    outtext("2");
   }
   else if(ch==51)
   {
    t=3;
    moveto(getmaxx()/2,getmaxy()/2);
    outtext("3");
   }
  }
  else
  goto a;
  char c[2];
  c[1]=0;
  ch=getch();
  int oc=getcolor();
  if(ch==8)
  {
   oc=getcolor();
   moveto(getmaxx()/2,getmaxy()/2);
   setcolor(1);
   sprintf(c,"%i",t);
   outtext(c);
   setcolor(oc);
   goto a;
  }
  else if(ch==10 || ch==13)
  goto c;
  else
  goto a;
  c:
  setbkcolor(0);
  cleardevice();
  if(t==1)
  return;
  else if(t==2)
  instr();
  else if(t==3)
  {
   pass++;
   end();
  }
 }
 void intro()
 {
  int ob[10],i,j;
  ob[0]=2;         ob[1]=0;
  ob[2]=2;         ob[3]=349;
  ob[4]=636;       ob[5]=349;
  ob[6]=636;       ob[7]=0;
  ob[8]=2;   ob[9]=0;
  for(i=0;i<5 ;i++)
  {
   setlinestyle(i,0,2);
   setcolor(i+9);
```

```
  for(j=0;j<10;j++)
   if(j>=3 &&  j<=6)
    ob[j]-=4;
   else ob[j]+=4;
  drawpoly(5,ob);
 }
 char str[6][50];
  strcpy(str[5],"FANTASY MONOPOLY");
  strcpy(str[4],"                          ");
  strcpy(str[3],"          Press Any Key To Begin The...");
  for (i=3; i<=5; i++)
  {
   if(i==3)
   setcolor(i);
   else if(i==5)
   setcolor(i-1);
   settextstyle(i+3,0,i);
   outtextxy(100,20*i,str[i]);
   delay(500);
  }
 getch();
 cleardevice();
}
void clrp();
void dip(char t[9999])
{
 char str[9999],c[2];
 c[1]='\0';
 strcpy(str,t);
 int len=strlen(str);
 ++l;
 if(l>33)
  {
   delay(1000);
   clrp();
  }
 int i,j=0,ps=1;
 moveto(1,l*ht);
 if( len*textwidth("W") >39*textwidth("W") )
 {
  while(len>0)
  {
   for(i=j;i<39*ps;i++)
   {
    c[0]=str[i];
    outtext(c);
   }
   len-=39;
   j+=39;
   ps++;
   moveto(1,(++l)*ht);
  }
 }
 else if(len<=39)
 outtext(str);
 for(i=0;i<999;i++)
 str[i]=' ';
}
void gboard()
```

```
{
 setlinestyle(0,0,3);
 setcolor(56);
 int ob[10];
 ob[0]=2;         ob[1]=0;
 ob[6]=636;       ob[7]=0;
 ob[2]=2;         ob[3]=349;
 ob[4]=636;       ob[5]=349;
 ob[8]=0;   ob[9]=0;
 drawpoly(5,ob);
 setfillstyle(1,63);
 fillpoly(5,ob);
 setcolor(56);
 line(400,0,400,349);
 int ib[10];
 ib[0]=20;        ib[1]=15;
 ib[2]=20;        ib[3]=334;
 ib[4]=380;       ib[5]=334;
 ib[6]=380;       ib[7]=15;
 ib[8]=20;  ib[9]=15;
 drawpoly(5,ib);
 setfillstyle(11,3);
 fillpoly(5,ib);
 setcolor(56);
 int pc1[10],pc2[10],pc3[10],pc4[10];
 pc1[0]=20;       pc1[1]=15;    //c1p1
 pc1[2]=20;       pc1[3]=75;   //c1p2
 pc1[4]=80;       pc1[5]=75;   //c1p3
 pc1[6]=80;       pc1[7]=15;    //c1p4
 pc1[8]=20;       pc1[9]=15;    //c1p1==c1p5
 drawpoly(5,pc1);
 pc2[0]=20;       pc2[1]=334;   //c2p1
 pc2[2]=20;       pc2[3]=274;   //c2p2
 pc2[4]=80;       pc2[5]=274;   //c2p3
 pc2[6]=80;       pc2[7]=334;   //c2p4
 pc2[8]=20;       pc2[9]=334;   //c2p1==c2p5
 drawpoly(5,pc2);
 pc3[0]=380;      pc3[1]=334;   //c3p1
 pc3[2]=320;      pc3[3]=334;   //c3p2
 pc3[4]=320;      pc3[5]=274;   //c3p3
 pc3[6]=380;      pc3[7]=274;   //c3p4
 pc3[8]=380;      pc3[9]=334;   //c3p1==c3p5
 drawpoly(5,pc3);
 pc4[0]=380;      pc4[1]=15;    //c4p1
 pc4[2]=320;      pc4[3]=15;    //c4p2
 pc4[4]=320;      pc4[5]=75;   //c4p3
 pc4[6]=380;      pc4[7]=75;   //c4p4
 pc4[8]=380;      pc4[9]=15;    //c4p1==c4p5
 drawpoly(5,pc4);
 int sbl[10],k=2,s=0,c=0;
 for(int j=0;j<2;j++)
 for(int i=0;i<200;i+=25)      //for(i times)
 {
  if(i/25==3) //myst chest
   setfillstyle(1,57);
  else if(i/25==5)//spl evt
   setfillstyle(1,59);
  else if(i/25==4)//portal
   setfillstyle(1,61);
```

```
 else
  {
   setfillstyle(k,60);
   s++;
  }
 if(j==0)
 {
  sbl[0]=20;    sbl[1]=75+i; //cp1
  sbl[2]=20;    sbl[3]=100+i; //cp2
  sbl[4]=70;    sbl[5]=100+i; //cp3
  sbl[6]=70;    sbl[7]=75+i; //cp4
  sbl[8]=25;    sbl[9]=75+i; //cp1==cp5
  drawpoly(5,sbl);
  fillpoly(5,sbl);
 }
 else
 {
  sbl[0]=380;    sbl[1]=75+i; //cp1
  sbl[2]=330;    sbl[3]=75+i; //cp2
  sbl[4]=330;    sbl[5]=100+i; //cp3
  sbl[6]=380;    sbl[7]=100+i; //cp4
  sbl[8]=380;    sbl[9]=75+i; //cp1==cp5
  drawpoly(5,sbl);
  fillpoly(5,sbl);
 }
if((c%2) && (s==2))
 {
  k++;
  s=0;
  c++;
 }
else if(!(c%2) && (s==3))
 {
  k++;
  s=0;
  c++;
 }
}
int sbb[10];
for(j=0;j<2;j++)
for(i=0;i<240;i+=30)          //for(i times)
{
 if(i/30==3)    //mc
  setfillstyle(1,57);
 else if(i/30==5)   //se
  setfillstyle(1,59);
 else if(i/30==4)     //p
  setfillstyle(1,61);
 else
  {
   setfillstyle(k,60);
   s++;
  }
 if(j==0)
 {
  sbb[0]=80+i;   sbb[1]=334; //cp1
  sbb[2]=80+i;   sbb[3]=284; //cp2
  sbb[4]=110+i;   sbb[5]=284; //cp3
  sbb[6]=110+i;   sbb[7]=334; //cp4
```

```
   sbb[8]=80+i;   sbb[9]=334; //cp1==cp5
   drawpoly(5,sbb);
   fillpoly(5,sbb);
  }
  else
  {
   sbb[0]=80+i;   sbb[1]=15;   //cp1
   sbb[2]=80+i;   sbb[3]=65;   //cp2
   sbb[4]=110+i;   sbb[5]=65;   //cp3
   sbb[6]=110+i;   sbb[7]=15;   //cp4
   sbb[8]=80+i;   sbb[9]=15;   //cp1==cp5
   drawpoly(5,sbb);
   fillpoly(5,sbb);
  }
 if((c%2) && (s==2))
  {
   k++;
   s=0;
   c++;
  }
 else if(!(c%2) && (s==3))
  {
   k++;
   s=0;
   c++;
  }
 }
 //Filling colours may start ...
 setfillstyle(1,4);
 fillpoly(5,pc1);
 setfillstyle(1,1);
 fillpoly(5,pc2);
 setfillstyle(1,2);
 fillpoly(5,pc3);
 setfillstyle(1,MAGENTA);
 fillpoly(5,pc4);
 setcolor(0);
 moveto(getmaxx()-20*textwidth("W"),getmaxy()-12);
 outtext("(Press Esc To Exit)");
}
void clrp()
{
 svp;
 cleardevice();
 gboard();
 svpi;
 ::l=-1;
 dip(" ");
}

class space
{
 protected:
  float scost,srent;
  char sname[15],sowner[25];
 public:
  void chgowner(char nowner[25])  {   strcpy(sowner,nowner);        }
  char* ret_sname()                {   return sname       ;     }
  char* ret_sowner()              {   return sowner      ;     }
```

```
  float ret_scost()               {   return scost       ;     }
  float ret_srent()               {   return srent       ;     }
};

class non_site : public space
{
 public:
  non_site()
  {
   scost=0;
   srent=0;
   strcpy(sowner,'\0');
   strcpy(sname,'\0');
  }
};

class player
{
 char pname[25],sowned[][24];
 int ppos,pcoin,rndno,nos,life;
 float pgld;
 public:
 void create(int n);
 void roll();
 void buy();
 void sell();
 void trade();
 void rent(space s,player p);
 void mystch();
 void splevt();
 void coind();
 void chspace();
 void ftravel();
 void dispmenu();
 void disppos();
 int ret_nos()          { return nos;                        }
 void pay(float f)    { pgld-=f;                          }
 int ret_life()       { return life;                       }
 void collect(float f)      { pgld+=f;                     }
 char*  ret_pname()   { return pname;                      }
 int ret_rndno() { return rndno;                          }
 void ssite()         { nos--;                            }
 float points()         { return (nos*1000)+(pgld*100)+(rndno*500);    }
}p[6];
void player::create(int n)
{
 char t[15];
 m:
 clrp();
 dip("Enter Player Name=>");
 iip(t);
 strcpy(pname,t);
 ppos=0;
 pgld=1500;
 rndno=0;
 life=3;
 nos=0;
 for(int i=0;i<n;i++)
 if(strcmpi(pname,p[i].ret_pname())==0)
```

```
 {
   dip("Name Already In Use Choose Another Name!");
   delay(1000);
   goto m;
 }
 dip("Enter Your Mipmap Indicator Colour:>");
 dip("1.BLUE");
 dip("2.GREEN");
 dip("3.CYAN");
 dip("4.RED");
 dip("5.MAGENTA");
 dip("6.BROWN");
 int h=0;
 char pc[20];
 a:
 dip("=>");
 iip(pc);
 pcoin=atoi(pc);
 if(pcoin>6 || pcoin<1)
  {
   if(!h)
   {
    dip("Invalid Enter Your Choice Again!");
    h=1;
    goto a;
   }
   else if(h)
   {
    dip("You Exceeded Your Limit");
    dip("Default Colour(LIGHT GREEN) Has Been Set!");
    pcoin=58;
   }
 }
}
void sitesown(int n)
{
 ifstream f("SPACE.dat",ios::binary);
  if(!f)
  {
   dip("\nERROR!Space File not found");
   getch();
   exit(0);
  }
  int i=0;
  space s;
  for(i=0;i<36;i++)
  {
   f.read((char*)&s,sizeof(s));
   if(strcmpi(s.ret_sowner(),p[n].ret_pname())==0)
     {
      o[n][i]=1;
      lvl[n][i]=1;
     }
   else
    {
     o[n][i]=0;
     lvl[n][i]=0;
    }
```

```
if(i==0||i==18||i==9||i==27||i==3||i==15||i==24||i==30||i==5||i==13||i==22||i==
32)
     {
      o[n][i]=0;
      lvl[n][i]=0;
     }
  }
  f.close();
}
void quit()
{
 clrp();
 dip(p[turn].ret_pname());
 outtext(",Thank You For Playing");
 fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
 fstream t("temp.dat",ios::app|ios::binary);
  if(!f)
  {
   dip("\nERROR!Space File not found");
   getch();
   exit(0);
  }
  space s;
  int i=0;
  while(f.read((char*)&s,sizeof(s)))
  {
   if(strcmpi(s.ret_sowner(),p[turn].ret_pname())==0)
   {
    s.chgowner('\0');
    avs++;
   }
   t.write((char*)&s,sizeof(s));
  }
 f.close();
 t.close();
 remove("SPACE.dat");
 rename("temp.dat","SPACE.dat");
 if(turn==nop-1)
 {
  turn=-1;
  goto m;
 }
 else
 {
  for(i=turn;i<nop;i++)
  {
   p[i]=p[i+1];
  }
  for(i=turn;i<nop;i++)
   for(int j=0;j<36;j++)
   {
    o[i][j]=o[i+1][j];
    lvl[i][j]=lvl[i+1][j];
   }
 }
 turn--;
 m:
 nop--;
```

```
 endt=1;
 getch();
}
void player::roll()
{
 char rld[3];
 dice=random(12)+1;
 dip("You Have Rolled: ");
 sprintf(rld,"%i",dice);
 outtext(rld);
 delay(500);
 coind();
}

void player::buy()
{
 fstream fb("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
 ofstream temp("TEMP.dat",ios::binary|ios::app);
 if(!fb)
  {
   dip("\nERROR!Space File not found");
   getch();
   exit(0);
  }
 space sb,s;
 int r=0;
 fb.seekg(ppos*sizeof(sb));
 fb.read((char*)&sb,sizeof(sb));
 if(pgld-sb.ret_scost()<=0)
  {
   dip("\nSorry, Insuffient  Gold !");
   fb.close();
   temp.close();
  }
 else
 {
  sb.chgowner(pname);
  pgld-=sb.ret_scost();
  fb.seekg(0);
  while(fb.read((char*)&s,sizeof(s)))
  {
   if(r==ppos)
    temp.write((char*)&sb,sizeof(sb));
   else
    temp.write((char*)&s,sizeof(s));
   r++;
  }
  fb.close();
  temp.close();
  remove("SPACE.dat");
  rename("TEMP.dat","SPACE.dat");
  nos++;
  avs--;
  sitesown(turn);
 }
}
void player::sell()
{
 fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
```

```
 if(!f)
 {
  dip("\nERROR!Space File not found");
  getch();
  exit(0);
 }
space s,ss;
int i=0,q[36][1],n=0;
char in[9];
for(i=0;i<36;i++)
{
 f.read((char*)&s,sizeof(s));
 if(o[turn][i]==1)
 {
  q[n][0]=i;
  sprintf(in,"%i",++n);
  dip(in);
  outtext(".");
  outtext(s.ret_sname());
 }
}
if(n==0)
{
 dip("You Dont Have Any Site's To Sell");
 getch();
 return;
}
dip("Which Site Would You Like To Sell?");
z:
dip("=>");
iip(::ch1);
int max=n+1;
n=atoi(ch1);
if(n>max||n<0)
goto z;
n--;
f.seekg(q[n][0]*sizeof(s));
f.read((char*)&s,sizeof(s));
dip("Would You Really Like To Sell The Site For ");
char pri[60];
float  pr=0.75*s.ret_scost();
sprintf(pri,"%g",pr);
outtext(pri);
outtext(" Gold?");
dip("1.Yes             2.No");
z7:
dip("=>");
iip(::ch1);
::ch=atoi(::ch1);
if(ch!=1&&ch!=2)
goto z7;

ofstream temp("temp.dat",ios::app|ios::binary);
if(::ch==1)
 {
   pgld+=0.75*s.ret_scost();
   s.chgowner('\0');
   nos--;
   avs++;
```

```
  }
  int r=0;
  f.seekg(0);
  while(f.read((char*)&ss,sizeof(ss)))
  {
   if(r==q[n][0])
    temp.write((char*)&s,sizeof(s));
   else
    temp.write((char*)&ss,sizeof(ss));
   r++;
  }
  f.close();
  temp.close();
  remove("SPACE.dat");
  rename("temp.dat","SPACE.dat");
}

void player::trade()
{
 char num[19];
 fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
  if(!f)
  {
   dip("\nERROR!Space File not found");
   getch();
   exit(0);
  }
 space s,st;
 int ara[7];
 for(int i=0;i<nop;i++)
 if(i!=turn)
  {
    ara[i]=i+1;
    sprintf(num,"%u",i+1);
    dip(num);
    outtext(".");
    outtext(p[i].ret_pname());
  }
 else
 ara[i]=0;
  dip("Which Player Would You Like To Trade With?");
 z6:
 dip("=>");
 int t;
 char tt[60];
 iip(tt);
 t=atoi(tt);
 int maxx=ara[0],minn=ara[0],ios;
 for(int pl=0;pl<nop;pl++)
 {
  if(maxx<ara[pl])
   maxx=ara[pl];
  if(minn>ara[pl]&&ara[pl]!=0)
   minn=ara[pl];
  if(ara[pl]==0)
   ios=pl+1;
 }
 for(pl=0;pl<nop;pl++)
 if(t>maxx||t<minn||t==ios||t==0)
```

```
goto z6;
t--;
char in[60];
int n=0,q[36][1];
for(i=0;i<36;i++)
{
 if(o[t][i]==1)
 {
  q[n][0]=i;
  f.seekg(i*sizeof(s));
  f.read((char*)&s,sizeof(s));
  sprintf(in,"%i",++n);
  dip(in);
  outtext(".");
  outtext(s.ret_sname());
 }
}
if(n==0)
{
 dip("The Choosen Player Does Not Own Any Site's To Trade");
 getch();
 return;
}
dip("Which Site Would You Like To Trade?");
int max=n;
z5:
dip("=>");
char ii[300];
iip(ii);
n=atoi(ii);
if(n>max||n<=0)
goto z5;
n--;
f.seekg(q[n][0]*sizeof(s));
f.read((char*)&s,sizeof(s));
float offer;
dip(pname);
dip(",What Is The Price You Propose For The Site ?");
dip("=>");
char ofer[15];
iip(ofer);
offer=atof(ofer);
clrp();
dip(p[t].ret_pname());
dip(",Would You Like To Trade ");
dip(s.ret_sname());
outtext(" With ");
outtext(pname);
dip("For ");
outtext(ofer);
dip("1.Yes              2.No");
z4:
dip("=>");
iip(::ch1);
::ch=atoi(::ch1);
if(ch!=1&&ch!=2)
goto z4;
if(::ch==1)
{
```

```
    s.chgowner(ret_pname());
    pay(offer);
    p[t].collect(offer);
    nos--;
    p[t].ssite();
  }
  else
  {
    dip(p[t].ret_pname());
    outtext(" Refused To Trade.");
  }
  ofstream temp("temp.dat",ios::app|ios::binary);
  int r=0;
  f.seekg(0);
  while(f.read((char*)&st,sizeof(st)))
   {
    if(r==q[n][0])
     temp.write((char*)&s,sizeof(s));
    else
     temp.write((char*)&st,sizeof(st));
    r++;
   }
   f.close();
   temp.close();
   remove("SPACE.dat");
   rename("temp.dat","SPACE.dat");
   sitesown(turn);
   sitesown(t);
}
void player::rent(space sr,player p)
{
 for(int i=0;i<nop;i++)
 if(!(strcmpi(sr.ret_sowner(),p.ret_pname())))
  break;
 if(pgld-(sr.ret_srent()+(p.ret_rndno()*10*lvl[i][ppos]))<0)
 {
  dip("Insufficient Gold! You Lost A Life");
  life-=1;
 }
 else
 {
  pay(sr.ret_srent()+(p.ret_rndno()*10*lvl[i][ppos]));
  p.collect(sr.ret_srent()+(p.ret_rndno()*10*lvl[i][ppos]));
 }
}
void player::coind()
{
 int cp;
 ppos+=dice;
 svp;
 for(cp=ppos-dice;cp<=ppos;cp++)
 {
  if(cp==36)
  {
   svpi;
   rndno++;
   dip("\nYou Have Completed The Round, Collect ");
   char f[9];
   sprintf(f,"%i",(rndno*25)+250);
```

```
 dip(f);
 outtext(" Gold");
 pgld+=(rndno*25)+250;
 getch();
 ppos-=36;
 cp=0;
 svp;
}
setlinestyle(0,0,1);
setfillstyle(SOLID_FILL,pcoin);
setcolor(62);
if( cp==0 )
{
 circle(50,304,7);
 floodfill(50,304,62);
 delay(500);
 gboard();
}
else if( cp>0 && cp<9 )
{
 x=45;y=261-((cp-1)*25);
 circle(x,y,7);
 floodfill(x,y,62);
 delay(500);
 gboard();
}
else if( cp==9 )
{
 x=50;y=45;
 circle(x,y,7);
 floodfill(x,y,62);
 delay(500);
 gboard();
}
else if( cp>9 && cp<18 )
{
 x=95+((cp-10)*30),y=40;
 circle(x,y,7);
 floodfill(x,y,62);
 delay(500);
 gboard();
}
else if( cp==18 )
{
 x=350;y=45;
 circle(x,y,7);
 floodfill(x,y,62);
 delay(500);
 gboard();
}
else if( cp>18 && cp<27)
{
 x=355;y=87+((cp-19)*25);
 circle(x,y,7);
 floodfill(x,y,62);
 delay(500);
 gboard();
}
else if( cp==27)
```

```
   {
    x=350;y=304;
    circle(x,y,7);
    floodfill(x,y,62);
    delay(500);
    gboard();
   }
   else if( cp>27 && cp<36 )
   {
    x=305-((cp-28)*30);y=309;
    circle(x,y,7);
    floodfill(x,y,62);
    delay(500);
    gboard();
   }
  }
  setcolor(56);
  setlinestyle(0,0,3);
  setcolor(0);
  l=-1;
  svpi;
}
void player :: disppos()
{
 svp;
 setlinestyle(0,0,1);
 setfillstyle(SOLID_FILL,pcoin);
 setcolor(62);
 if( ppos==0)          { x=50;           y=304;              }
 else if( ppos>0 && ppos<9 )     { x=45;                y=261-((ppos-1)*25);}
 else if( ppos==9 )         { x=50;           y=45;               }
 else if( ppos>9 && ppos<18 )    { x=95+((ppos-10)*30);   y=40;             }
 else if( ppos==18 )        { x=350;          y=45;               }
 else if( ppos>18 && ppos<27)    { x=355;          y=87+((ppos-19)*25);}
 else if( ppos==27)         { x=350;          y=304;              }
 else if( ppos>27 && ppos<36 )   { x=305-((ppos-28)*30);  y=309;             }
 circle(x,y,7);
 floodfill(x,y,62);
 setcolor(56);
 setlinestyle(0,0,3);
 setcolor(0);
 svpi;
}
void player :: ftravel()
{
 if(pgld<150+(rndno*15))
  dip("Sorry Insuffficient Gold");
 else
  {
   if(ppos==4)
    ppos+=10;
   else if(ppos==14||ppos==31)
    ppos+=9;
   else if(ppos==23)
    ppos+=8;
   pgld-=150+(rndno*15);
  }
}
void player :: mystch()
```

```
{
 ifstream fm("MYSTCH.TXT");
 if(!fm)
 {
  cout<<"\nERROR!MYSTCH File not found";
  getch();
  exit(0);
 }
 int x=random(5),ps=-1;
 char st[5000];
 while(fm.getline(st,500,'.'))
 {
  ps++;
  if(ps==x)
  dip(st);
 }
 if(x==0)
  pgld+=100;
 else if(x==1)
  if(pgld>=200)
   pgld-=200;
  else
  {
   dip("Insufficient Gold, Lose A Life");
   life-=1;
  }
 else if(x==2)
  {
   ppos-=3;
  }
 else if(x==3)
  {
   ppos=18;
  }
 else if(x==4)
  life+=1;
 fm.close();
}
void player :: splevt()
{
 ifstream fm("SPLEVT.TXT");
 if(!fm)
 {
  dip("ERROR!SPLEVT File not found");
  getch();
  exit(0);
 }
 int x=random(5),ps=-1;
 char st[500];
 while(fm.getline(st,500,'.'))
 {
  ps++;
  if(ps==x)
  dip(st);
 }
 if(x==0)
  if(pgld>=100)
    pgld-=100;
  else
```

```
  {
   dip("Since You Had No Gold, He Harvested Your Organs , Lose A Life");
   life-=1;
   }
 else if(x==1)
   pgld+=200;
 else if(x==2)
  {
   ppos+=3;
  }
 else if(x==3)
  {
   ppos=0;
  }
 else if(x==4)
  life-=1;
 fm.close();
}

void player :: chspace()
{
 Q:
  ifstream f("SPACE.dat",ios::nocreate|ios::binary);
  if(!f)
  {
   dip("ERROR!Space File not found");
   getch();
   exit(0);
  }
  space s;
  if(ppos==3 || ppos==15 ||  ppos==24 || ppos==30  || ppos==5 || ppos==13 ||
ppos==22 || ppos==32 || ppos==0 || ppos==9  || ppos==18 || ppos==27 )
   s=non_site();
  f.seekg(ppos*sizeof(s));
  f.read((char*)&s,sizeof(s));
  dip("You Have Reached ");
  outtext(s.ret_sname());
  if(!strcmpi(s.ret_sowner(),'\0'))
  {
   if(ppos==3 || ppos==15 ||  ppos==24 || ppos==30 )
     {
      dip("Lets See What Your Curiosity Gets You In This Mystery Chest");
      dip("Press Any Key To Open It");
      getch();
      int cpos=ppos;
      mystch();
      if(cpos!=ppos)
      {
       disppos();
       getch();
       goto Q;

      }
     }
   else if(ppos==5 || ppos==13 ||  ppos==22 || ppos==32 )
     {
      dip("Good or Bad, Rich or Rags ,Lets See What Fate Leads You In This
Special Event");
      dip("Press Any Key To Unveil Your Fate");
```

```
       getch();
       int cpos=ppos;
       splevt();
       if(cpos!=ppos)
       {
        disppos();
        getch();
        goto Q;
       }
      }
   else if(ppos==0)
     dip("Welcome To Guild, You're Free To Stay");
   else if(ppos==9)
     dip("The Jail Doesn't Welcome Anyone, You're Free To Go As Long As You
Haven't Done Anything Wrong");
   else if(ppos==18)
     dip("Welcome To The Tavern, Enjoy Yourself, Everything Is On Our Tab");
   else if(ppos==27)
     {
      dip("You're Caught Infringing Upon The Forbidden Forest, GO TO JAIL!!");
      dip("Pay 150 Gold or Lose A Life");
      dip("1.Pay                      2.Lose Life");
      z3:
      dip("=>");
      iip(::ch1);
      ::ch=atoi(ch1);
      ppos=9;
      disppos();
      if(ch!=1&&ch!=2)
       goto z3;
      if(::ch==1)
      {
       if(pgld<=150)
       {
     dip("You Have Insufficient Gold");
     life-=1;
       }
       else
       pgld-=150;
      }
      else
      life-=1;
     }
   else
    {
     dip("The Site Is Unowned,It Costs ");
     char c[60];
     sprintf(c,"%g",s.ret_scost());
     outtext(c);
     dip(" Would You Like To Buy It?");
     dip("1.YES                2.NO ");
     z2:
     dip("=>");
     iip(::ch1);
     ::ch=atoi(::ch1);
    if(ch!=1&&ch!=2)
    goto z2;
     if(::ch==1)
    {
```

```
      f.close();
      buy();
      fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
     }
     }
    }
   else if(!(strcmpi(s.ret_sowner(),pname)))
   {
    dip("Welcome Back To Your Domain");
    dip("Your Site Has Been Renovated, Rents Are Incresed");
    lvl[turn][ppos]++;
   }
   else
    {
     for(int i=0;i<nop;i++)
     if(!(strcmpi(s.ret_sowner(),p[i].ret_pname())))
     {
       dip("You Are in The Domain of ");
       outtext(p[i].ret_pname());
       dip("To Pass, Pay A Rent Of ");
       float ret=s.ret_srent();
       char re[200];
       sprintf(re,"%g",ret);
       outtext(re);
       rent(s,p[i]);
       getch();
       break;
     }
    }
   if(ppos==4 || ppos==14 || ppos==23 || ppos==31)
    {
     dip("Would You Like To Fast Travel?");
     dip("It Costs :");
     int c=150+(rndno*15);
     char c1[30];
     sprintf(c1,"%i",c);
     outtext(c1);
     dip("1.Yes      2.No");
     z1:
     dip("=>");
     iip(::ch1);
     ::ch=atoi(::ch1);
     if(ch!=1&&ch!=2)
     goto z1;
     if(::ch==1)
     {
      ftravel();
      disppos();
     }
    }
   f.close();
}
void player :: dispmenu()
{
 if(!p[turn].ret_life())
  quit();
 lab:
 for(int i=0;i<nop;i++)
  sitesown(i);
```

```
 dip("Gold Stash=");
 char s[20];
 sprintf(s,"%g",pgld);
 outtext(s);
 dip("No Of Sites Owned=");
 sprintf(s,"%i",nos);
 outtext(s);
 dip("No Of Lives Left=");
 sprintf(s,"%i",life);
 outtext(s);
 dip("What Would You Like To Do?");
 dip("1.Sell");
 dip("2.Trade");
 dip("3.Quit");
 dip("4.End Turn");
 z:
 dip("=>");
 iip(::ch1);
 ::ch=atoi(::ch1);
 if(ch!=1&&ch!=2&&ch!=3&&ch!=4)
 goto z;
 switch(::ch)
 {
 case 1:sell();
      break;
 case 2:trade();
      break;
 case 3:clrp();
      quit();
      break;
 case 4: ::endt=1;
       clrp();
       break;
 default:dip("Enter A Valid Option");
       break;
 }
 disppos();
 if(endt==0)
 goto lab;
}
void endg()
{
 fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
 ofstream temp("temp.dat",ios::app|ios::binary);
 if(!f)
 {
  dip("\nERROR!Space File not found");
  getch();
  exit(0);
 }
 space s;
 while(f.read((char*)&s,sizeof(s)))
 {
  s.chgowner('\0');
  temp.write((char*)&s,sizeof(s));
 }
 f.close();
 temp.close();
 remove("SPACE.dat");
```

```
 rename("temp.dat","SPACE.dat");
}
void end()
{
 for(int i=0;i<nop;i++)
  for(int j=1;j<nop-1-i;j++)
   if(p[j].points()>p[j+1].points())
    {
     player t=p[j];
     p[j]=p[j+1];
     p[j+1]=t;
    }
 cleardevice();
 svp;
 int ob[10];
 ob[0]=2;         ob[1]=0;
 ob[6]=636;       ob[7]=0;
 ob[2]=2;         ob[3]=349;
 ob[4]=636;       ob[5]=349;
 ob[8]=0;   ob[9]=0;
 drawpoly(5,ob);
 char str[6][999];
 if(::pass==0)
 {
  setcolor(63);
  setlinestyle(0,0,3);
  settextjustify(1,1);
  strcpy(str[0],"  The Winner Is");
  moveto(getmaxx()/2,25);
  setcolor(63);
  settextstyle(4,0,6);
  outtext(str[0]);
  for(int i=0;i<63;i++)
  {
   setcolor(i);
   settextstyle(10,0,8);
   moveto(getmaxx()/2,200);
   outtext(p[0].ret_pname());
   delay(50);
  }
  getch();
  fstream f("SPACE.dat",ios::in|ios::app|ios::nocreate|ios::binary);
  ofstream temp("temp.dat",ios::app|ios::binary);
  if(!f)
  {
   dip("\nERROR!Space File not found");
   getch();
   exit(0);
  }
  space s;
  while(f.read((char*)&s,sizeof(s)))
  {
   s.chgowner('\0');
   temp.write((char*)&s,sizeof(s));
  }
  f.close();
  temp.close();
  remove("SPACE.dat");
  rename("temp.dat","SPACE.dat");
```

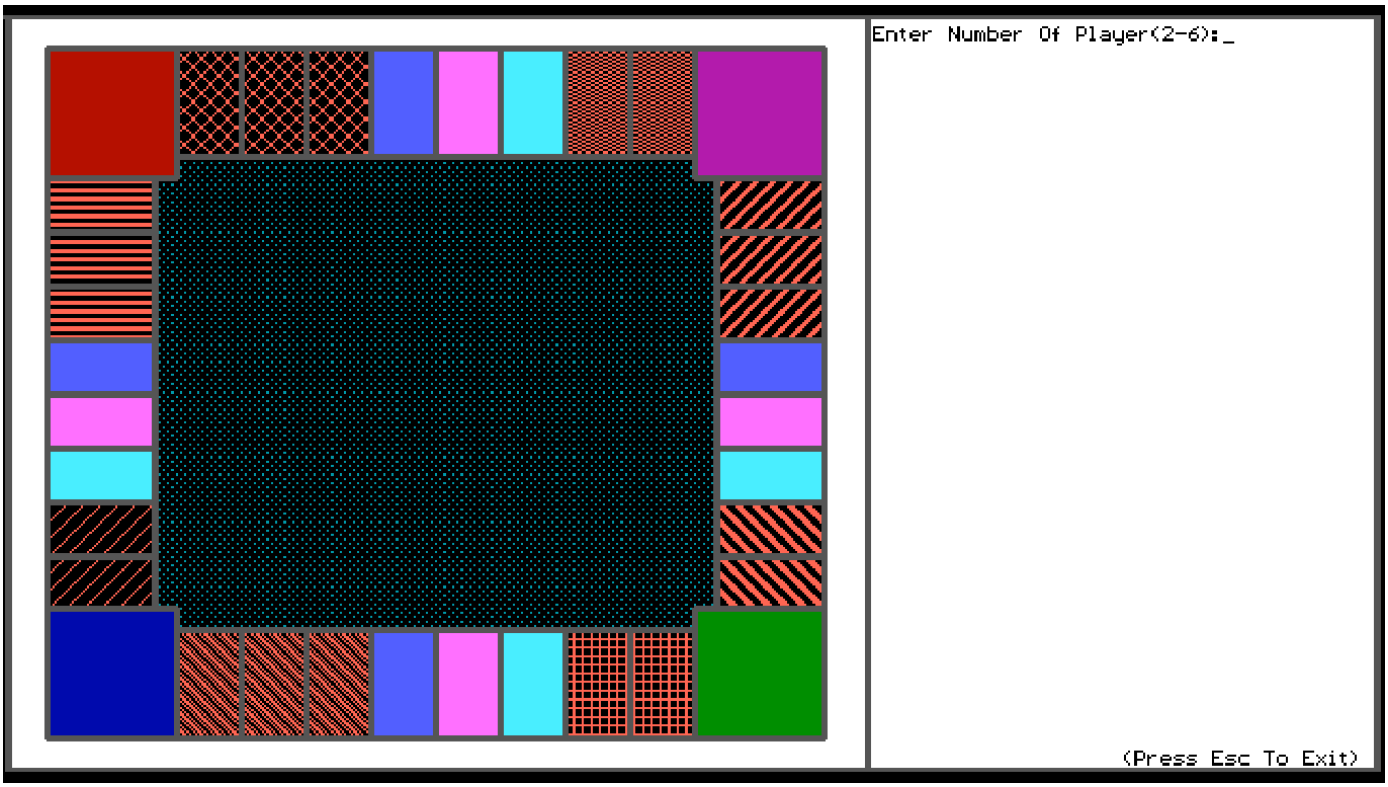
```
 }
 settextjustify(0,2);
 setcolor(63);
 cleardevice();
 drawpoly(5,ob);
 strcpy(str[1],"                                Done By");
 strcpy(str[2],"        S.Balavignesh    &     T.K.Sivaraam");
 strcpy(str[3]," THANKS        FOR        PLAYING");
 strcpy(str[4],"FANTASY MONOPOLY");
 strcpy(str[5],"Press Any Key To Exit The");

  settextstyle(4,0,4);
  setcolor(RED);
  moveto(0,0);
  outtext(str[3]);

  setcolor(RED);
  settextstyle(3,0,4);
  moveto(110,75);
  outtext(str[5]);

  settextstyle(3,0,2);
  setcolor(GREEN);
  moveto(0,250);
  outtext(str[1]);

  setcolor(GREEN);
  settextstyle(8,0,5);
  moveto(100,150);
  outtext(str[4]);

  settextstyle(6,0,4);
  setcolor(57);
  moveto(0,300);
  outtext(str[2]);

 getch();
 endg();
 closegraph();
 exit(0);
}

void main()
{
 clrscr();
 randomize();
 int gdriver =VGA , gmode=VGAMED , errorcode;
 initgraph(&gdriver, &gmode, "");
 errorcode = graphresult();
 if (errorcode != grOk)
 {
   cprintf("Graphics Error: %s \n", grapherrormsg(errorcode));
   cprintf("!Press Any Key To Halt!");
   getch();
   exit(0);
 }
 int sd885=textheight("W")+1;
 ::ht=sd885;
 intro();
```

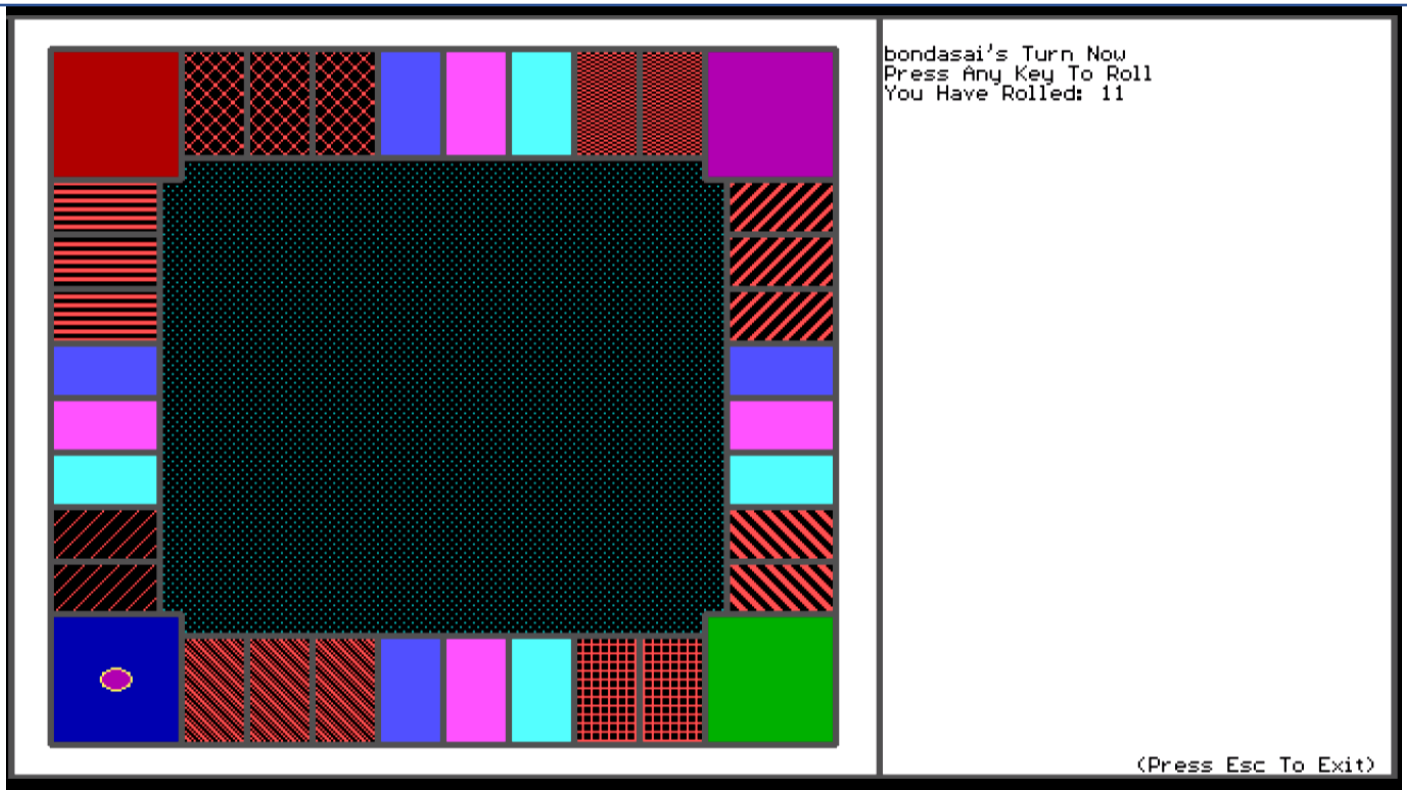
```
 mainscr();
 setactivepage(0);
 setvisualpage(0);
 settextstyle(2,0,0);
 gboard();
 svpi;
 C1:
 dip("Enter Number Of Player(2-6):");
 char nopc[2];
 iip(nopc);
 nop=atoi(nopc);
 if(nop<2 || nop>6)
  {
   dip("Invalid No Of Players,Enter Again");
   goto C1;
  }
 char num[2];
 for(int i=0;i<nop;i++)
 {
  dip("For Player No-");
  sprintf(num,"%i",i+1);
  outtext(num);
  p[i].create(i);
 }
 for(i=0;i<nop;i++)
  for(int j=0;j<nop-1-i;j++)
   if(strcmpi(p[j].ret_pname(),p[j+1].ret_pname())>0)
    {
     player temp=p[j];
     p[j]=p[j+1];
     p[j+1]=temp;
    }
  while(nop>1 && avs>0)
  {
   clrp();
   p[turn].disppos();
   dip(p[turn].ret_pname());
   outtext("'s Turn Now");
   dip("Press Any Key To Roll");
   getch();
   p[turn].roll();
   p[turn].disppos();
   p[turn].chspace();
   p[turn].dispmenu();
   clrp();
   turn++;
   if(turn==nop)
    turn=0;
  }
 end();
}
```

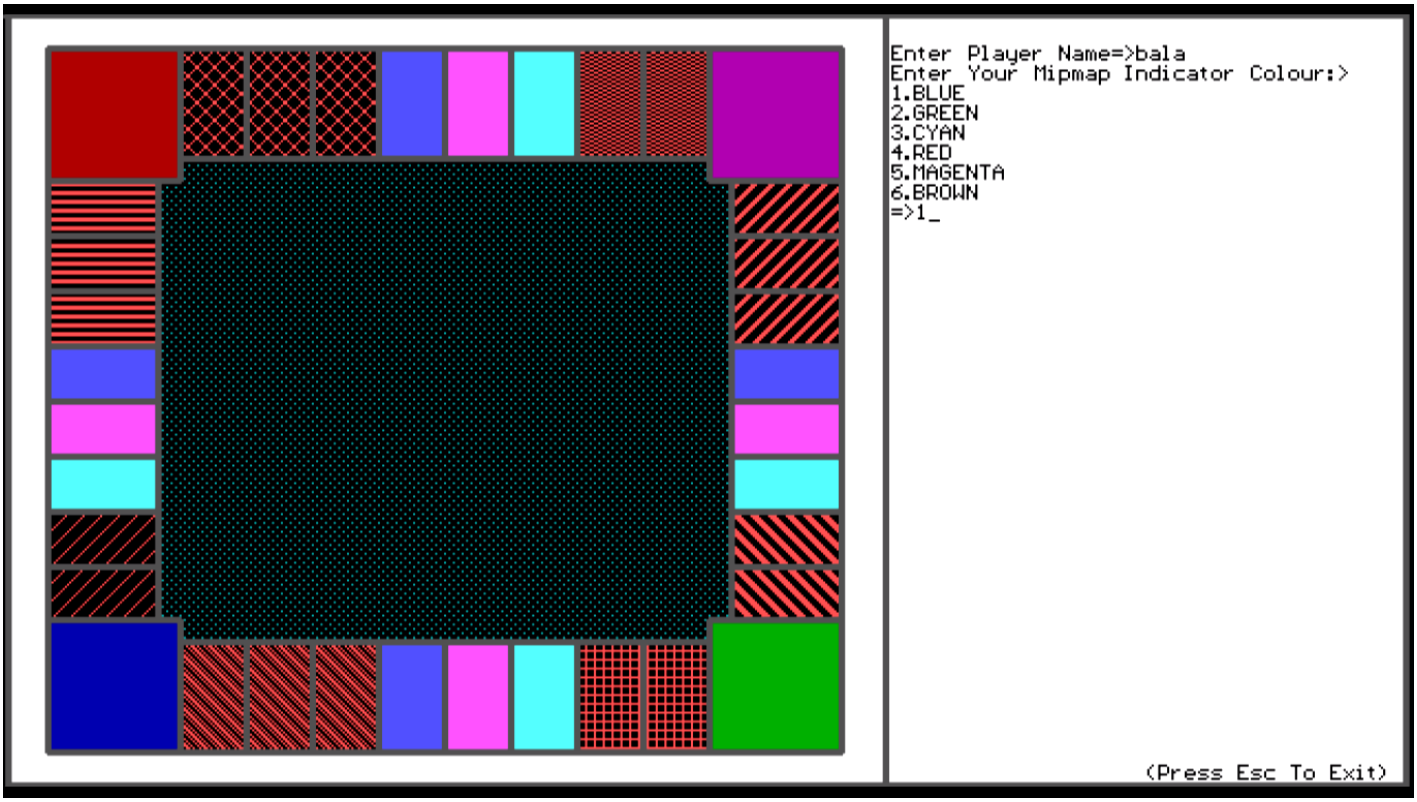
SAMPLE OUTPUT

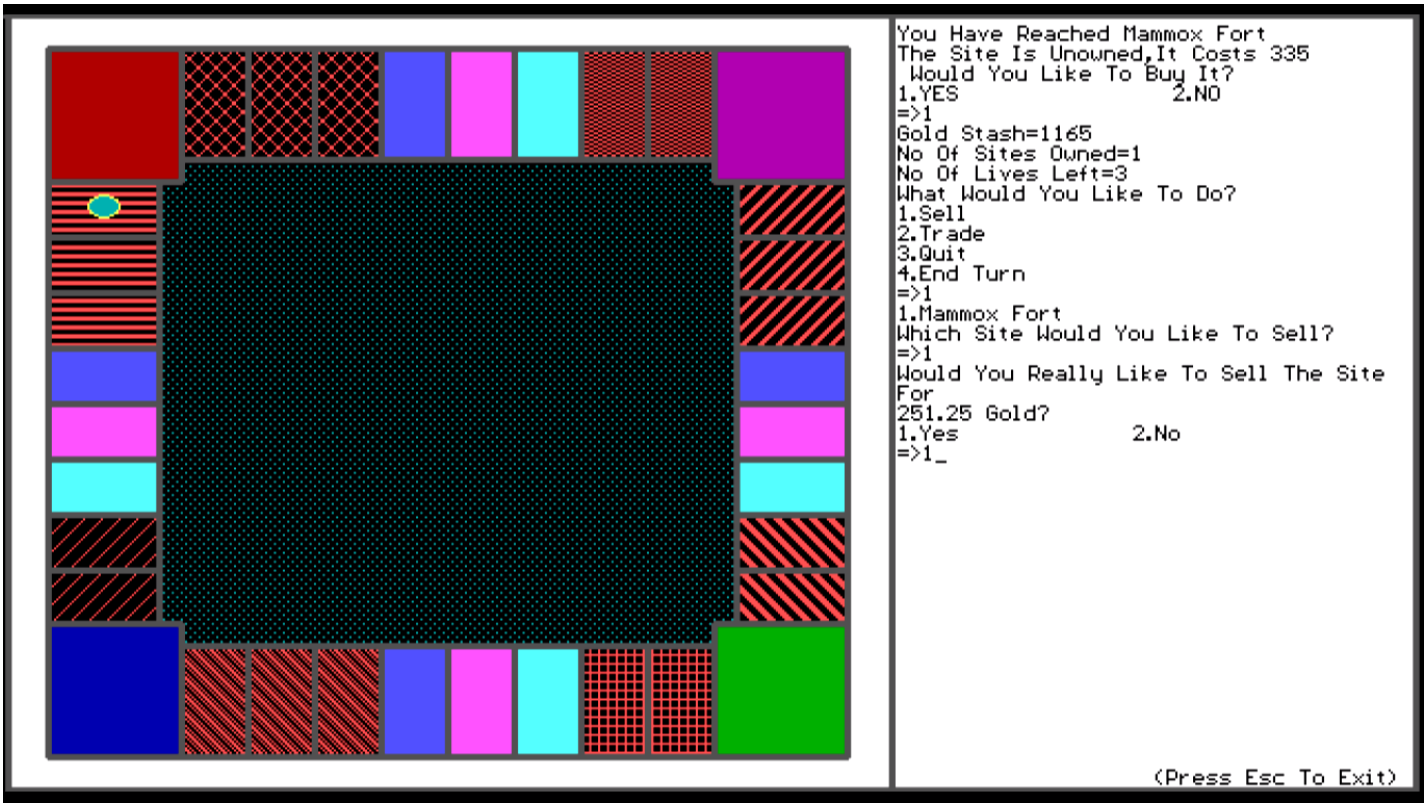
For Fantasy Monopoly

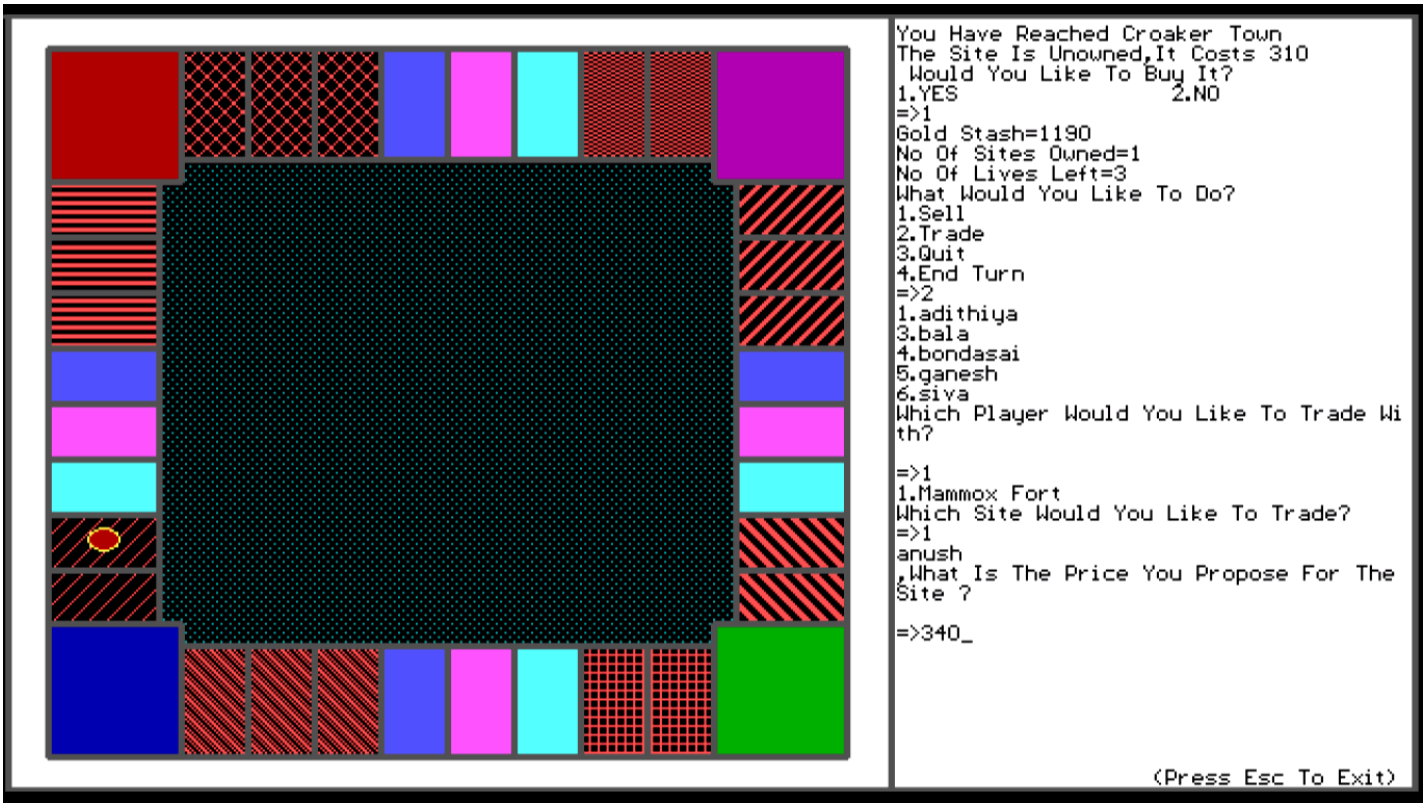
SAMPLE OUTPUT

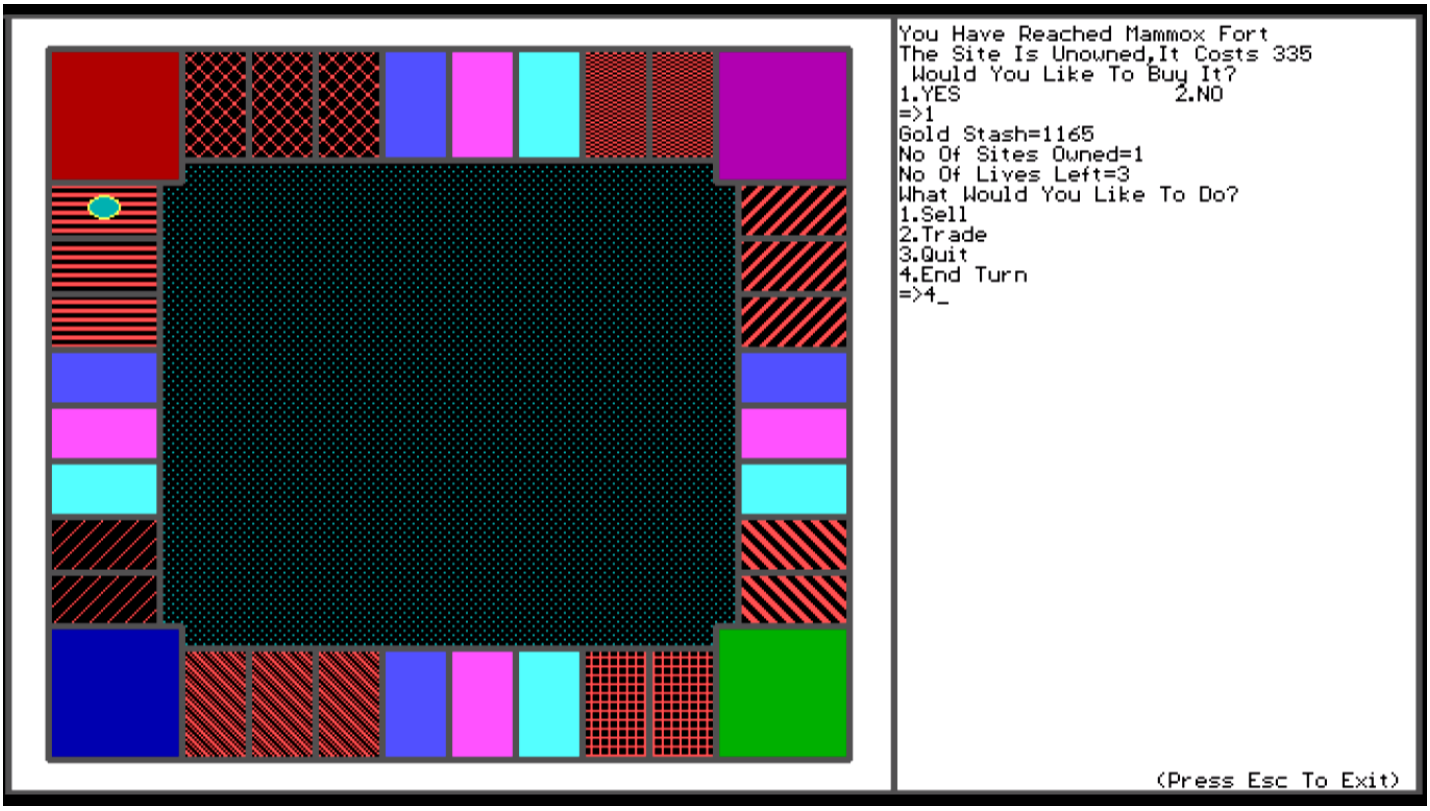


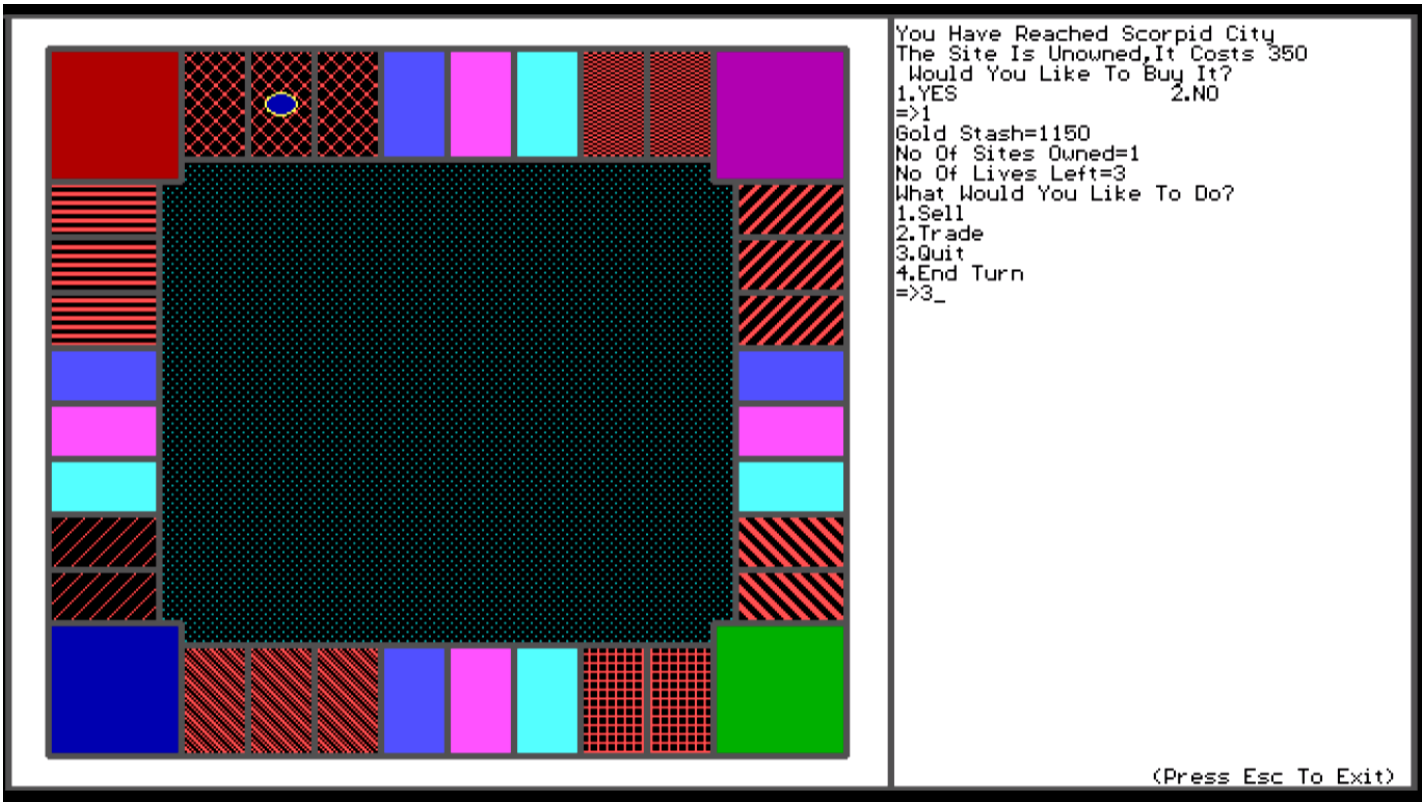

bondasai's Turn Now
Press Any Key To Roll
You Have Rolled: 11
(Press Esc To Exit)

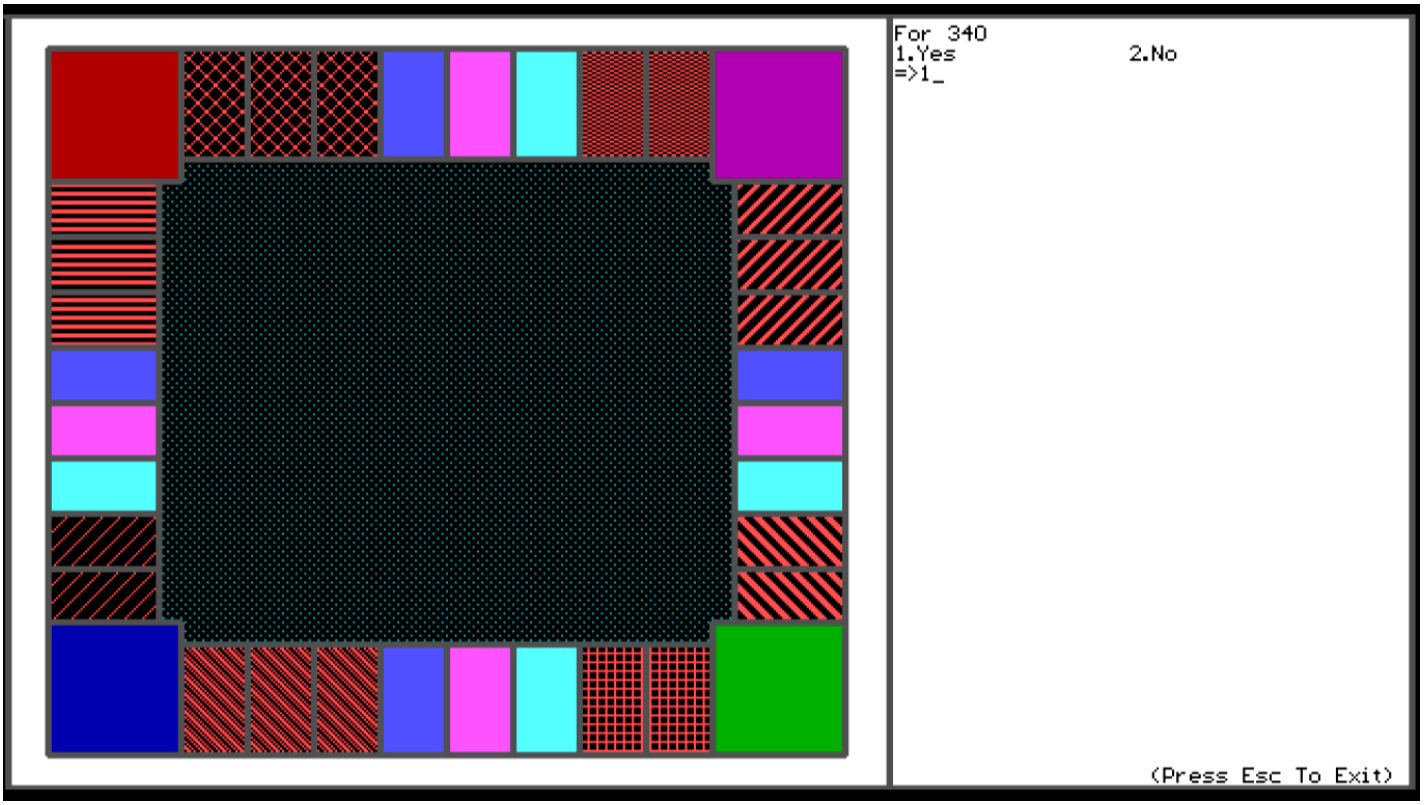

Enter Player Name=>bala
Enter Your Mipmap Indicator Colour:>
1.BLUE
2.GREEN
3.CYAN
4.RED
5.MAGENTA
6.BROWN
=>1_
(Press Esc To Exit)

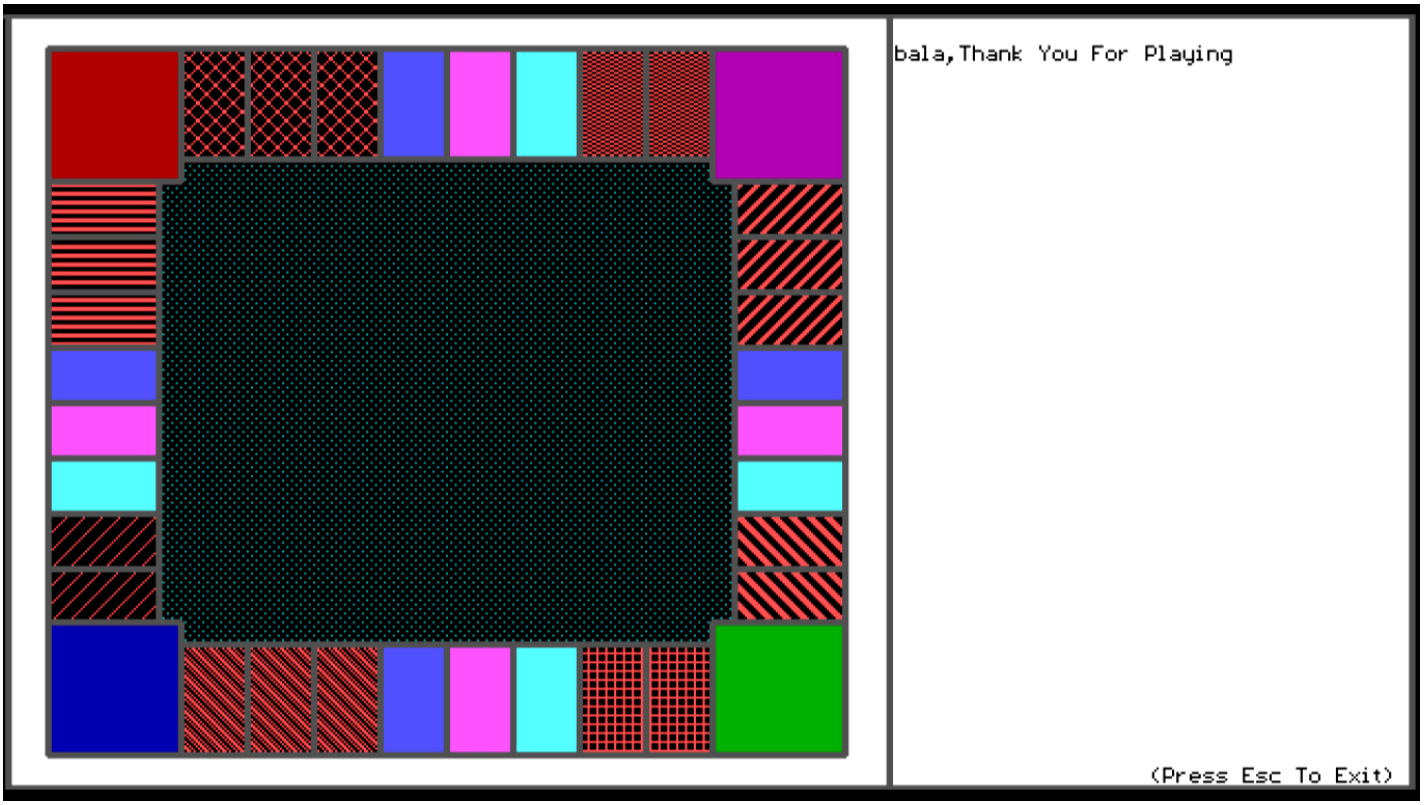

You Have Reached Mammox Fort
The Site Is Unowned,It Costs 335
Would You Like To Buy It?
1.YES 2.NO
=>1
Gold Stash=1165
No Of Sites Owned=1
No Of Lives Left=3
What Would You Like To Do?
1.Sell
2.Trade
3.Quit
4.End Turn
=>1
1.Mammox Fort
Which Site Would You Like To Sell?
=>1
Would You Really Like To Sell The Site For
251.25 Gold?
1.Yes 2.No
=>1
(Press Esc To Exit)


You Have Reached Croaker Town
The Site Is Unowned,It Costs 310
Would You Like To Buy It?
1.YES 2.NO
=>1
Gold Stash=1190
No Of Sites Owned=1
No Of Lives Left=3
What Would You Like To Do?
1.Sell
2.Trade
3.Quit
4.End Turn
=>2
1.adithiya
3.bala
4.bondasai
5.ganesh
6.siva
Which Player Would You Like To Trade With?
=>1
1.Mammox Fort
Which Site Would You Like To Trade?
=>1
anush
,What Is The Price You Propose For The Site ?
=>340
(Press Esc To Exit)


You Have Reached Mammox Fort
The Site Is Unowned,It Costs 335
Would You Like To Buy It?
1.YES 2.NO
=>1
Gold Stash=1165
No Of Sites Owned=1
No Of Lives Left=3
What Would You Like To Do?
1.Sell
2.Trade
3.Quit
4.End Turn
=>4_
(Press Esc To Exit)


You Have Reached Scorpid City
The Site Is Unowned,It Costs 350
Would You Like To Buy It?
1.YES 2.NO
=>1
Gold Stash=1150
No Of Sites Owned=1
No Of Lives Left=3
What Would You Like To Do?
1.Sell
2.Trade
3.Quit
4.End Turn
=>3_
(Press Esc To Exit)


For 340
1.Yes
2.No
=>1_
(Press Esc To Exit)


bala,Thank You For Playing
(Press Esc To Exit)

BIBILIOGRAPHY

- https://simple.wikipedia.org/wiki/Monopoly_(game)
- https://en.wikipedia.org/wiki/Template:Monopoly_board_layout
- http://www.staroceans.org/Monopoly.htm
- https://preistian.wordpress.com/2011/07/24/membuat-game-monopoli-sederhana-dengan-c/