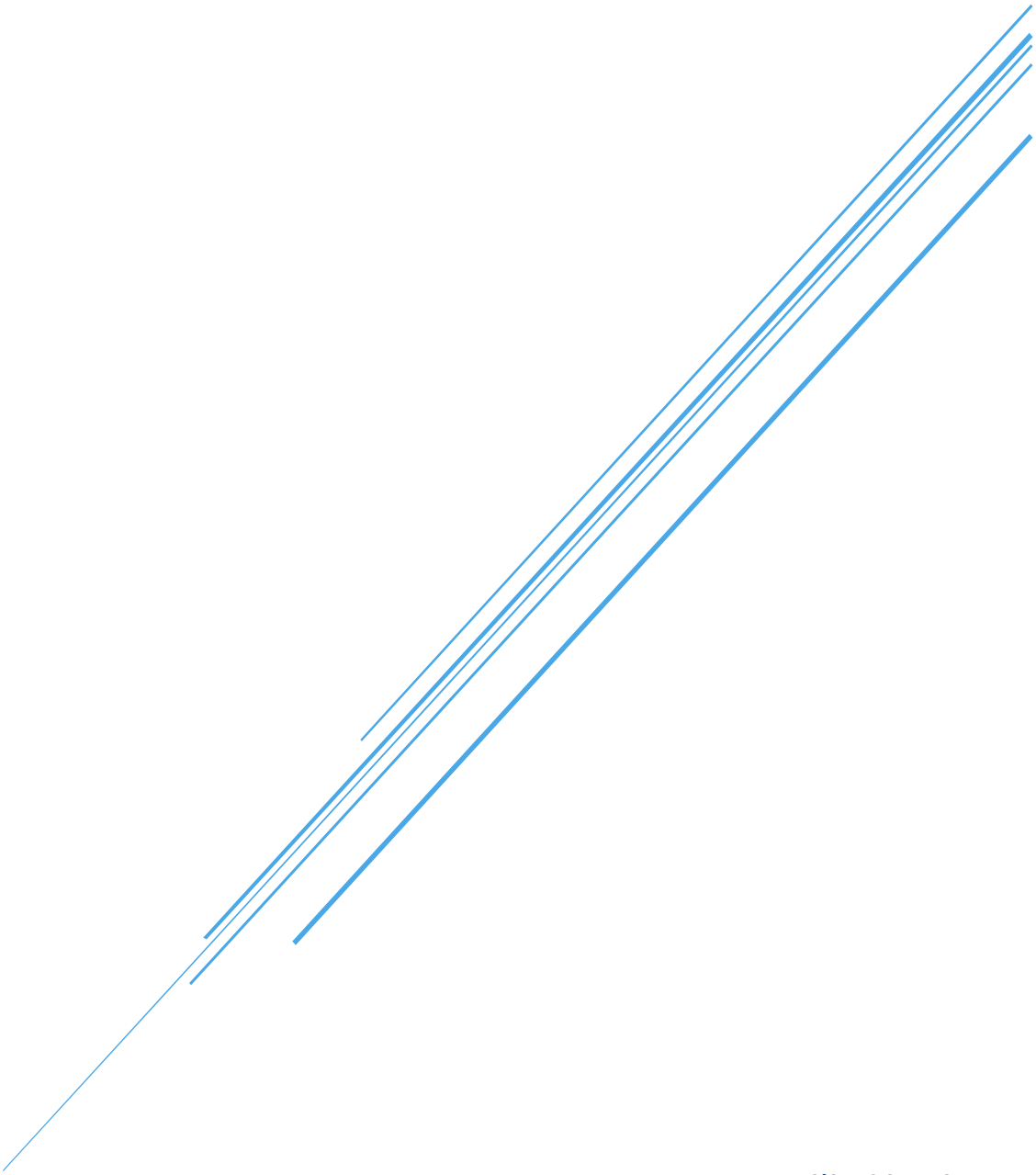
2048: THE GAME

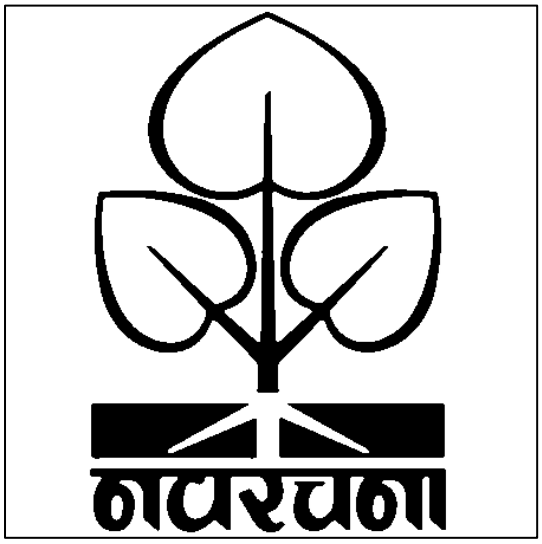
Computer Science Project 18-19



Hardik Shah
Class XII C, Roll No.15

Contents

Certificate



This is to certify that Hardik Shah of Class 12-C, Roll No. 15 has satisfactorily completed the computer science project for the academic year 2018-19 and has duly submitted the computer science project on time.

Signature of Teacher

Signature of Inviligator

Acknowledgements

It would be my utmost pleasure to express my heartfelt gratitude towards our Computer Science Teacher, Ms. Smriti Upadhyay for extending a helping hand which was quintessential in the completion of this Project. I would also like thank Mrs. Bijoya Bakshi, Director of Nacrachana School.

I, along with my able partners, Jainil Shah and Mridul Mittal would also like to offer our sincere thanks to CBSE for giving us the opportunity to learn extensively from this project.

System Requirements

Turbo C++ 3.0, Turbo C++ 4.5(Windows Vista, Windows 7; non-graphics mode).

Operating System:

Windows 98, Windows 2000, Windows XP, Windows Vista, Windows 7(in small screen, non-graphics mode).

RAM:

Minimum 256 MB and higher.

Hard Disk Drive Space:

Minimum 10 MB of space and higher.

Bibliography

1. Computer Science with C++, Class 11 by Sumita Arora
2. Computer Science with C++, Class 12 by Sumita Arora
3. https://www.google.in
4. https://www.codestacks.com
5. 2048 game by Ketchapp

Overview of C++

This overview of C++ presents the key design, programming and language –technical concepts using examples to give the reader feel of the language.

C++ is a general purpose programming language with a bias towards system programming that supports efficient low- level computation, data abstraction,object oriented programming and generic programming. It was developed by Bjarne Stroustrup starting in 1979 at Bell Labs as an enhancement to the C Language features and originally named C with classes. It was renamed C++ in 1983.

A one of the most popular programming languages ever created, C++ is widely used in the software industry. Some of its applications domains include system software, application software, device drivers, embedded software, high-performance server and client applications, and entertainment software such as video games. Several groups provide both free and propiertary C++ compiler software, include the GNU Project, Microsoft, Intel and Borland. C++ has greatly influenced many other popular programming languages, most notably C# and Java.

About the Game

2048 is a single-player sliding block puzzle game designed by Italian web developer Gabriele Cirulli. The game's objective is to slide numbered tiles on a grid to combine them to create a tile with the number 2048. However, one can continue to play the game after reaching the goal, creating tiles with larger numbers.

2048

Developer	Gabriele Cirulli
Publisher	Solebon LLC
Platform(s)	Browser, iOS, Nintendo 3DS, Android, Apple TV
Release	Web: 9 March 2014[1]
Genre	Puzzle
Modes	Single-player, Timed

Gameplay

2048 is played on a 4×4 grid, with numbered tiles that slide smoothly when a player moves them using the four arrow keys. Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4.[2] Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move. Higher-scoring tiles emit a soft glow.

A scoreboard on the upper-right keeps track of the user's score. The user's score starts at zero, and is incremented whenever two tiles combine, by the value of the new tile. As with many arcade games, the user's best score is shown alongside the current score.

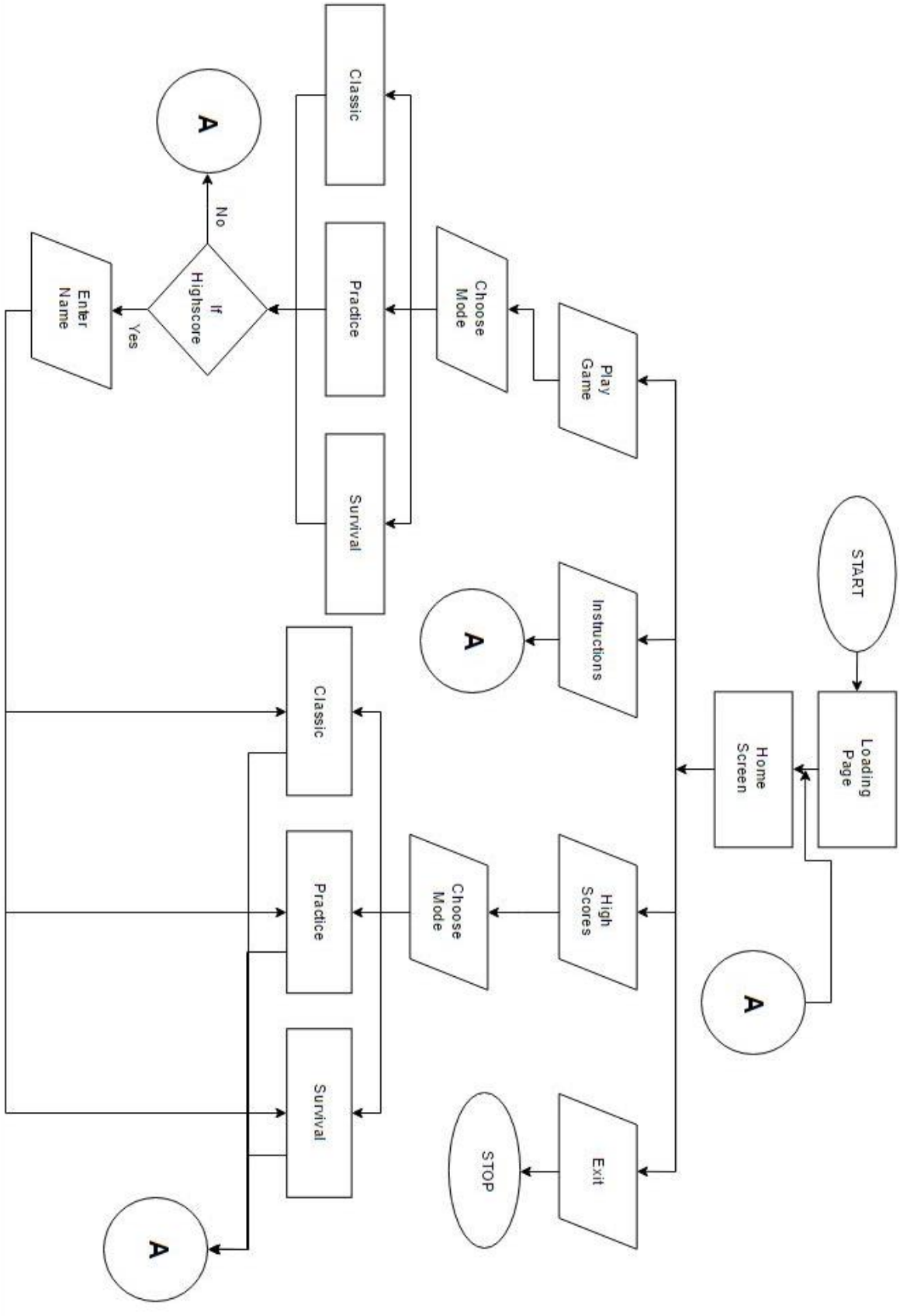
A game of 2048 in progress

The game is won when a tile with a value of 2048 appears on the board, hence the name of the game. After reaching the 2048 tile, players can continue to play (beyond the 2048 tile) to reach higher scores. When the

player has no legal moves (there are no empty spaces and no adjacent tiles with the same value), the game ends.

A completed game of 2048. The 2048 tile can be seen in the bottom-right corner.

Flowchart: Working of the Game



Header Files Used

- #include <fstream.h>
- #include <conio.h>
- #include <string.h>
- #include <stdio.h>
- #include <math.h>
- #include <graphics.h>
- #include <process.h>
- #include <dos.h>
- #include <stdlib.h>
- #include <bios.h>
- #include <time.h>

Classes Used

Name of Class	**Members**

class backup C[2]	double B[6][6];
	long bscore();
	backup();

class newposn INSERT[16];	int I; int J; newposn();

class score_card obj;	char name[20][20];
	long score_save;
	int name_len;
	void add();
	void display();
	score_card();

class virtualkeyboard G[26];	char let[3]; int bkey[2];
class Hall_Of_Fame N[20];	char HOF_Name[20][20]; long HOF; int HOF_Name_Len;

User Defined functions:

- void storekey()
- void cover()
- void enter_name(int p)
- void couldnt()
- void OVER(int wol)
- void score_card:: display()
- void score_card:: add()
- void i_basic(int A[4][4],int slide)
- void instructions()
- void quit(int d)
- void convert_score()
- void convert_time()
- void print()
- void left()
- void right()
- void up()
- void down()
- void loadscreen()
- void initialize()
- void UNDO()
- void mode(int dop)
- void game()

Source Code:

```
//Project 2018_19: 2048 v5.0
//Source Code:

#include <fstream.h>
#include <conio.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include <graphics.h>
#include <process.h>
#include <dos.h>
#include <stdlib.h>
#include <bios.h>
#include <time.h>

class backup  // for use in undo function
{
      public:
      double B[6][6];
      long bscore;
      backup()
      {
            for(i=0;i<4;i++)                        // to clear the grid
            {
                  for(j=0;j<4;j++)
                  {
                        B[i][j]=0;
                        bscore=0;
                  }
            }
      }

}C[2];

class  newposn    //insert random number every turn
{
      public:
      int I;
      int J;

      newposn()
      {
            I=0;
            J=0;
      }
}INSERT[16];

class score_card
{
      char name[20][20];
      long score_save;
      int name_len;

      public:
```

```
      void add();
      void display();
      score_card()
      {
            name_len=3;
      }
} obj;

class virtualkeyboard
{
      public:

      char let[3];
      int bkey[2];
}G[26];

class Hall_Of_Fame
{
       public:

       char HOF_Name[20][20];
       long HOF;
       int HOF_Name_Len;
}N[20];

void storekey()
{
      int k1=0;
      G[k1].bkey[0]=4209;G[k1].bkey[1]=4177;strcpy(G[k1].let,"Q");k1++;
      G[k1].bkey[0]=4471;G[k1].bkey[1]=4439;strcpy(G[k1].let,"W");k1++;
      G[k1].bkey[0]=4709;G[k1].bkey[1]=4677;strcpy(G[k1].let,"E");k1++;
      G[k1].bkey[0]=4978;G[k1].bkey[1]=4946;strcpy(G[k1].let,"R");k1++;
      G[k1].bkey[0]=5236;G[k1].bkey[1]=5204;strcpy(G[k1].let,"T");k1++;
      G[k1].bkey[0]=5497;G[k1].bkey[1]=5465;strcpy(G[k1].let,"Y");k1++;
      G[k1].bkey[0]=5749;G[k1].bkey[1]=5717;strcpy(G[k1].let,"U");k1++;
      G[k1].bkey[0]=5993;G[k1].bkey[1]=5961;strcpy(G[k1].let,"I");k1++;
      G[k1].bkey[0]=6255;G[k1].bkey[1]=6223;strcpy(G[k1].let,"O");k1++;
      G[k1].bkey[0]=6512;G[k1].bkey[1]=6480;strcpy(G[k1].let,"P");k1++;
      G[k1].bkey[0]=7777;G[k1].bkey[1]=7745;strcpy(G[k1].let,"A");k1++;
      G[k1].bkey[0]=8051;G[k1].bkey[1]=8019;strcpy(G[k1].let,"S");k1++;
      G[k1].bkey[0]=8292;G[k1].bkey[1]=8260;strcpy(G[k1].let,"D");k1++;
      G[k1].bkey[0]=8550;G[k1].bkey[1]=8518;strcpy(G[k1].let,"F");k1++;
      G[k1].bkey[0]=8807;G[k1].bkey[1]=8775;strcpy(G[k1].let,"G");k1++;
      G[k1].bkey[0]=9064;G[k1].bkey[1]=9032;strcpy(G[k1].let,"H");k1++;
      G[k1].bkey[0]=9322;G[k1].bkey[1]=9290;strcpy(G[k1].let,"J");k1++;
      G[k1].bkey[0]=9579;G[k1].bkey[1]=9547;strcpy(G[k1].let,"K");k1++;
      G[k1].bkey[0]=9836;G[k1].bkey[1]=9804;strcpy(G[k1].let,"L");k1++;
      G[k1].bkey[0]=11386;G[k1].bkey[1]=11354;strcpy(G[k1].let,"Z");k1+
+;
      G[k1].bkey[0]=11640;G[k1].bkey[1]=11608;strcpy(G[k1].let,"X");k1+
+;
      G[k1].bkey[0]=11875;G[k1].bkey[1]=11843;strcpy(G[k1].let,"C");k1+
+;
      G[k1].bkey[0]=12150;G[k1].bkey[1]=12118;strcpy(G[k1].let,"V");k1+
+;
      G[k1].bkey[0]=12386;G[k1].bkey[1]=12354;strcpy(G[k1].let,"B");k1+
+;
```

```
      G[k1].bkey[0]=12654;G[k1].bkey[1]=12622;strcpy(G[k1].let,"N");k1+
+;
      G[k1].bkey[0]=12909;G[k1].bkey[1]=12877;strcpy(G[k1].let,"M");
}

//global declaration of variables
int maxx,maxy,midx,midy,midy1;
      //graphic variables
int i,j,k,m,m1[10],m2=0,n,c;                                    //loop
variables
int undo,tp=0,flag=1,over=1,key,a,b,new1,new2,new3,decider;
//logical variables
int T,DELAY;
int ft;

long score=0;

int const y=0;

double A[6][6];

// A character array to use in outtextxy function
char
NUM[][5]={"2","4","8","16","32","64","128","256","512","1024","2048"};
char basearray[][10]={"0","1","2","3","4","5","6","7","8","9"};
char fname[11];

// Array to insert random numbers after each move
// Notice that probability of 2 is more than 4
int X[]={2,2,4,2,2,2,2,2,2,4};

void cover()
{
      void mode(int);
      void quit(int);
      void instructions();

      int n=0,position=0;
      score=0;
      char opt[][20]={"PLAY
GAME","INSTRUCTIONS","LEADERBOARDS","CREDITS","EXIT"};

      cleardevice();

      if(ft==0)
      {
            ft++;
            settextjustify(CENTER_TEXT,CENTER_TEXT);
            for(i=0; i<11; i++)
            {
                  settextstyle(0,0,i+1);
                  setcolor(i+5);
                  outtextxy(midx, midy-125, NUM[i]);
                  delay(50 + 50*i);

                  if(i!=10)
                  {
                        setcolor(0);
```

```
                        outtextxy(midx, midy-125, NUM[i]);
                }

                else
                        break;
        }
    }

    else
    {
        setcolor(ft);
        settextstyle(0,0,11);
        settextjustify(1,1);
        outtextxy(midx, midy-125, NUM[10]);
        ft++;

        if(ft==15)
            ft=1;
    }

    int key,xco;

    setcolor(WHITE);
    settextstyle(0,HORIZ_DIR,8);
    settextjustify(1,1);

    setfillstyle(1,0);
    settextjustify(CENTER_TEXT,CENTER_TEXT);

    xco=midx;
    settextstyle(0,0,3);
    setcolor(0);
    setfillstyle(1,15);
    bar(xco-150,midy+(35*position)-18,xco+150,midy+15+(35*position));
    outtextxy(xco,midy+(35*position),opt[position]);

    settextstyle(0,HORIZ_DIR,2);
    setcolor(15);
    outtextxy(xco,midy+35,"INSTRUCTIONS");
    outtextxy(xco,midy+70,"LEADERBOARDS");
    outtextxy(xco,midy+105,"CREDITS");
    outtextxy(xco,midy+140,"EXIT");

    settextstyle(2,0,4);
    settextjustify(0,1);
    outtextxy(2,maxy-20,"Use arrow keys to choose. Press ENTER key to
continue.");

    settextjustify(CENTER_TEXT,CENTER_TEXT);
    while(n==0)
    {
        while (bioskey(1) == 0);
            key = bioskey(0);

        if(key==18432 && position>=0)
        {
            settextstyle(0,0,2);
            setcolor(15);
```

```
                setfillstyle(1,0);
                bar(xco-150,midy+(35*position)-
18,xco+150,midy+15+(35*position));
                outtextxy(xco,midy+(35*position),opt[position]);

                if(position==0)
                      position=4;
                else
                      position--;

                settextstyle(0,0,3);
                setcolor(0);
                setfillstyle(1,15);
                bar(xco-150,midy+(35*position)-
18,xco+150,midy+15+(35*position));
                outtextxy(xco,midy+(35*position),opt[position]);
           }

           else if(key==20480 && position<=4)
           {
                settextstyle(0,0,2);
                setcolor(15);
                setfillstyle(1,0);
                bar(xco-150,midy+(35*position)-
18,xco+150,midy+15+(35*position));
                outtextxy(xco,midy+(35*position),opt[position]);

                if(position==4)
                      position=0;
                else
                      position++;

                settextstyle(0,0,3);
                setcolor(0);
                setfillstyle(1,15);
                bar(xco-150,midy+(35*position)-
18,xco+150,midy+15+(35*position));
                outtextxy(xco,midy+(35*position),opt[position]);
           }

           else if(key==7181)
                break;
      }

     if(position==0)
           mode(0);

     else if(position==1)
           instructions();

     else if(position==2)
           mode(2);

     else if(position==3)
            quit(0);

     else if(position==4)
            quit(1);
```

```
}

void enter_name(int p)
{
      int ifloop=0,sft=0,t=0,bac[20];

      storekey();
      setfillstyle(1,15);
      bar(midx-150,midy1-10,midx+150,midy1+25);
      bar(midx-150,midy1-10,midx-140,midy1+70);
      bar(midx+150,midy1-10,midx+140,midy1+70);
      bar(midx-150,midy1+60,midx+150,midy1+70);

      if(p==0)
      {
            settextjustify(1,1);
            settextstyle(3,0,4);
            setcolor(9);
            outtextxy(midx,midy1-90,"NEW HIGHSCORE!!");

            settextstyle(3,0,3);
            outtextxy(midx,midy1-60,"Congratulations");
      }

      else
      {
            settextjustify(1,1);
            settextstyle(3,0,4);
            setcolor(9);
            outtextxy(midx,midy1-90,"CONGRATULATIONS");

            settextstyle(3,0,3);
            outtextxy(midx,midy1-60,"You made it to the HALL OF FAME!");
      }

      settextstyle(2,0,4);
      settextjustify(0,1);
      setcolor(15);
      outtextxy(2,maxy-20,"Type your name and press ENTER key to
submit.");

      setcolor(0);
      settextstyle(2,0,7);
      settextjustify(1,1);

      for(i=0;i<2;i++)
            outtextxy(midx-45+i,midy1+7,"Enter Your Name:");

      settextstyle(3,0,3);
      setcolor(15);

      while(ifloop==0)
      {
            while(kbhit()==0)
            {
                  setcolor(WHITE);
                  line(190+sft,272,190+sft,293);
                  delay(150);
```

```
                setcolor(BLACK);
                line(190+sft,272,190+sft,293);
                delay(100);
        }

        while (bioskey(1) == 0);
        key = bioskey(0);

        if(key==3592 && t>0)          //backspace
        {
              t--;
              setcolor(0);
              sft-=15;
              outtextxy(midx-125+sft,midy1+40,G[bac[t]].let);
        }

        else if(key==14624)  //space
        {
              strcpy(N[p].HOF_Name[t]," ");
              sft+=15;
              t++;
        }

        else if(key==7181)    //enter
        {
        ifloop=1;
        N[p].HOF_Name_Len=t-1;
        }

        else
        {
              for(i=0;i<26;i++)
              {
                    if(key==G[i].bkey[0] || key==G[i].bkey[1])
                    {
                          bac[t]=i;
                          strcpy(N[p].HOF_Name[t],G[i].let);
                          setcolor(15);
                          outtextxy(midx-125+sft,midy1+40,G[i].let);
                          sft+=15;
                          t++;
                    }
              }
        }

  }
}

void couldnt()
{
      cleardevice();

      settextjustify(1,1);
      settextstyle(3,0,4);
      setcolor(9);
      outtextxy(midx,midy1-15,"SORRY!");

      settextstyle(3,0,3);
```

```
	outtextxy(midx,midy1+15,"You could not make it to the HALL OF
FAME!");

	settextstyle(2,0,4);
	settextjustify(0,1);
	setcolor(15);
	outtextxy(2,maxy-20,"Press ENTER key to continue to the
homescreen.");
		settextjustify(1,1);

	while(y==0)
	{
		while (bioskey(1) == 0);
			key = bioskey(0);

		if(key==7181)
		cover();
	}
}

void OVER(int wol)
{
	void cover();
	cleardevice();

	int y=0;

	setcolor(9);
	settextstyle(0,0,7);
	settextjustify(1,1);

	if(wol==0)
	outtextxy(midx,midy1-50,"GAME OVER");

	else if(wol==1)
	outtextxy(midx,midy1-50,"YOU WIN!");

	setcolor(15);
	settextstyle(3,0,2);
	outtextxy(midx-35,midy1,"Your Score: ");

	int r,p1=0,AA=0,number[12];
	long x;
	x=score;

	if (x==0)
	{
		setcolor(15);
		outtextxy(midx+35+AA,midy1,"00");
	}

	else
	{
		while (x > 0)
		{
			r=x%10;
			x=x/10;
			number[p1]=r;
```

```
                p1++;
          }

          for(j=0; j<p1; j++)
          {
                setcolor(15);
                outtextxy(midx+35+AA,midy1,basearray[number[p1-j-1]]);
                AA+=15;
          }
    }

    settextstyle(2,0,4);
    settextjustify(0,1);
    setcolor(15);
    outtextxy(2,maxy-20,"Press ENTER key to continue.");

    while(y==0)
    {
          while (bioskey(1) == 0);
                key = bioskey(0);

          if(key==7181)
          {
                cleardevice();
                obj.add();
                obj.display();
          }
    }
}

void score_card:: display()
{
    cleardevice();
    void mode(int);

    int r,p1,number[12],AA,sft=0,y=0,d;
    long x;

    d=decider;

    int color[3]={14,7,6};

    for(j=0;j<3;j++)
    {
          setcolor(color[j]);
          setfillstyle(1,color[j]);
          pieslice(midx-200,midy-75+(25*j),0,360,7);

          setcolor(0);
          settextstyle(0,0,1);
          settextjustify(1,1);
          outtextxy(midx-200,midy-75+(25*j),basearray[j+1]);
    }

    settextjustify(1,1);
    settextstyle(3,0,7);
    setcolor(9);
```

```
outtextxy(midx,midy1-209,"HALL OF FAME");
outtextxy(midx,midy1-210,"HALL OF FAME");
outtextxy(midx,midy1-211,"HALL OF FAME");
outtextxy(midx-1,midy1-210,"HALL OF FAME");
outtextxy(midx+1,midy1-210,"HALL OF FAME");

setcolor(0);
outtextxy(midx,midy1-210,"HALL OF FAME");

setfillstyle(1,9);
bar(100,75,530,80);

setfillstyle(1,0);
bar(105,77,525,78);

setfillstyle(1,12);
bar(250,140,385,142);

settextstyle(3,0,4);
setcolor(12);

if(d==0)
      outtextxy(midx,120,"PRACTICE");

else if(d==1)
      outtextxy(midx,120,"CLASSIC");

else if(d==2)
      outtextxy(midx,120,"SURVIVAL");

fstream fio;
fio.open(fname,ios::in);//open file in input mode
fio.seekg(0);//set file pointer to first object

fio.read((char*)&obj,sizeof(obj));

i=0;

while(!fio.eof())
{
      setcolor(15);
      settextstyle(3,0,2);
      x=obj.score_save;

      p1=0;
      AA=0;
      sft=0;

      while (x > 0)
      {
            r=x%10;
            x=x/10;
            number[p1]=r;
            p1++;
      }

      for(j=0; j<p1; j++)
      {
```

```
                  setcolor(15);
                  outtextxy(midx+140+AA,175+(25*i),basearray[number[p1-
j-1]]);
                  AA+=15;
            }

            settextstyle(3,0,3);
            setcolor(15);

            for(j=0;j<=obj.name_len;j++)
            {
                  outtextxy(midx-175+sft,175+(25*i),obj.name[j]);
                  sft+=15;
            }

            i++;
            fio.read((char*)&obj,sizeof(obj));
      }

      fio.close();

      settextstyle(2,0,4);
      settextjustify(0,1);
      setcolor(15);
      outtextxy(2,maxy-20,"Press ENTER key to continue.");
      outtextxy(2,maxy-40,"Press BACKSPACE key to go back.");

      while(y==0)
      {
            while (bioskey(1) == 0);
                  key = bioskey(0);

            if(key==7181)                 //enter
                  cover();

            else if(key==3592)            //backspace
                  mode(2);
      }

}

void score_card:: add()
{
      fstream fin,fout;
      fin.open(fname,ios::in);//open file in input mode
      fin.seekg(0);//set file pointer to first object

      int y;

      i=0;

      fin.read((char*)&obj,sizeof(obj));

      while(!fin.eof())
      {
            N[i].HOF=obj.score_save;
            N[i].HOF_Name_Len=obj.name_len;
```

```
			for(j=0;j<=N[i].HOF_Name_Len;j++)
				strcpy(N[i].HOF_Name[j],obj.name[j]);

			i++;
			fin.read((char*)&obj,sizeof(obj));
		}

		for(i=9; i>=0; i--)
		{
			if(score<=N[i].HOF)
				break;

		}

		if(i==9)
			couldnt();

		else
		{
			for(j=9; j>i+1; j--)
			{
				N[j].HOF=N[j-1].HOF;
				N[j].HOF_Name_Len=N[j-1].HOF_Name_Len;
				for(y=0;y<=N[j-1].HOF_Name_Len;y++)
					strcpy(N[j].HOF_Name[y],N[j-1].HOF_Name[y]);
			}

			N[i+1].HOF=score;
			enter_name(i+1);

			fout.open("temp.txt",ios::app);//open temporary file

			for(i=0; i<10; i++)
			{
				obj.score_save=N[i].HOF;
				obj.name_len=N[i].HOF_Name_Len;

				for(y=0;y<=N[i].HOF_Name_Len;y++)
					strcpy(name[y],N[i].HOF_Name[y]);

				fout.write((char*)&obj,sizeof(obj));
			}

			fin.close();
			fout.close();
			remove(fname);//delete old file
			rename("temp.txt",fname);//rename temporary file as master
file
		}
}

void i_basic(int Ai[4][4],int slide)
{
		cleardevice();

		int nmidy;
		nmidy=midy+35;
```

```
	settextstyle(0,0,4);
	settextjustify(1,1);
	setcolor(9);
	outtextxy(midx,midy-200,"INSTRUCTIONS");

	setfillstyle(1,15);
	// for grid
	for(i=-2;i<=2;i++)
//columns
		bar(midx+(60*i)-1,nmidy-121,midx+(60*i)+1,nmidy+121);
	for(i=-2;i<=2;i++)                                          // rows
		bar(midx-121,nmidy+(60*i)-1,midx+121,nmidy+(60*i)+1);

	settextstyle(0,0,2);

	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
		{
			m=Ai[i][j];

			if(m!=0)
			{
				n=log10(m)/log10(2);
				setfillstyle(1,0);
				bar(midx-116+(60*j),nmidy-116+(60*i),midx-
64+(60*j),nmidy-64+(60*i));

				setcolor(n);
				setfillstyle(1, n);
				bar(midx-116+(60*j),nmidy-116+(60*i),midx-
64+(60*j),nmidy-64+(60*i));

				if(m==2048)
					settextstyle(0,0,1);

				else
					settextstyle(0,0,2);

				setcolor(0);
				outtextxy(midx-90+(60*j),nmidy-90+(60*i),NUM[n-
1]);

			}
		}
	}

	settextstyle(2,0,4);
	settextjustify(0,1);
	setcolor(15);

	if(slide!=3)
		outtextxy(2,maxy-15,"Press ENTER key to go to next page");
	else
		outtextxy(2,maxy-15,"Press ENTER key to continue to
homescreen.");

	settextjustify(1,1);
```

```
	setcolor(15);
	setfillstyle(1,15);

	for(i=0;i<4;i++)
		circle(midx-18+(12*i),midy+180,3);

	floodfill(midx-18+(12*slide),midy+180,15);
}

void instructions()
{
	int slide,nmidy;
	int A1[][4]={0,0,0,0,0,4,0,0,0,0,0,8,0,16,0,0};
	int A2[][4]={0,0,0,0,0,2,2,4,0,0,0,0,0,4,4,8};
	int A3[][4]={16,128,32,16,4,512,16,8,8,64,8,4,2,8,4,2};
	int A4[][4]={0,2,4,0,8,2048,32,2,4,32,128,4,16,2,4,2};


	nmidy=midy+35;
	slide=0;
	i_basic(A1,slide);

	setcolor(15);
	settextstyle(6,0,2);
	outtextxy(midx,midy-160,"Use Arrow Keys to move all tiles.");
	outtextxy(midx,midy-140,"For eg. On pressing right arrow key all
tiles ");
	outtextxy(midx,midy-120," move to the extreme right of the
grid.");


	setfillstyle(1,6);
	setcolor(6);
	bar(midx-65,midy+2,midx-100,midy+8);
	line(midx-100,midy-5,midx-100,midy+15);
	line(midx-110,midy+5,midx-100,midy-5);
	line(midx-110,midy+5,midx-100,midy+15);
	setfillstyle(1,6);
	floodfill(midx-105,midy+5,6);

	bar(midx-65,midy+122,midx-100,midy+128);
	line(midx-100,midy+115,midx-100,midy+135);
	line(midx-110,midy+125,midx-100,midy+115);
	line(midx-110,midy+125,midx-100,midy+135);
	floodfill(midx-105,midy+125,6);

	bar(midx+55,midy+62,midx-100,midy+68);
	line(midx-100,midy+55,midx-100,midy+75);
	line(midx-110,midy+65,midx-100,midy+55);
	line(midx-110,midy+65,midx-100,midy+75);
	floodfill(midx-105,midy+65,6);

	while(y==0)
	{
		while (bioskey(1) == 0);
			key = bioskey(0);
```

```
            if(key==7181)
                  break;
      }

      slide++;
      i_basic(A2,slide);

      setcolor(15);
      settextstyle(6,0,2);
      outtextxy(midx,midy-160,"When two tiles with the same");
      outtextxy(midx,midy-140,"number touch,they merge into one!");
      outtextxy(midx,midy-120,"It's over when the board fills up...");

      setfillstyle(1,4);
      bar(midx-4,nmidy-12,midx+4,nmidy-48);
      bar(midx-18,nmidy-26,midx+18,nmidy-34);

      bar(midx-4,nmidy-12+120,midx+4,nmidy-48+120);
      bar(midx-18,nmidy+94,midx+18,nmidy+86);

      bar(midx-16+60,nmidy-26-6,midx+16+60,nmidy-34-6);
      bar(midx-16+60,nmidy-26+6,midx+16+60,nmidy-34+6);

      bar(midx-16+60,nmidy-26-6+120,midx+16+60,nmidy-34-6+120);
      bar(midx-16+60,nmidy-26+6+120,midx+16+60,nmidy-34+6+120);

      while(y==0)
      {
            while (bioskey(1) == 0);
                  key = bioskey(0);

            if(key==7181)
                  break;
      }

      slide++;
      i_basic(A3,slide);

      setcolor(15);
      settextstyle(6,0,2);
      outtextxy(midx,midy-160,"Join the numbers and get to the 2048
tile to win!");
      outtextxy(midx,midy-140,"You lose when the grid is completely
filled");
      outtextxy(midx,midy-120,"and there is no possibility of merging
of any two tiles....");

      while(y==0)
      {
            while (bioskey(1) == 0);
                  key = bioskey(0);

            if(key==7181)
                  break;
      }


      slide++;
```

```
	i_basic(A4,slide);

	setcolor(15);
	settextstyle(6,0,2);
	outtextxy(midx,midy-170,"Practice Mode: You can undo one move
backwards.");
	outtextxy(midx,midy-150,"Classic Mode: You cannot undo your
move...");
	outtextxy(midx,midy-130,"Survival Mode: Start with a countdown of
5 sec.");
	outtextxy(midx,midy-110,"Get one sec each time you create a tile
greater than or equal to 8.");

	while(y==0)
	{
		while (bioskey(1) == 0);
			key = bioskey(0);

		if(key==7181)
			break;
	}

	cover();
}

void quit(int d)
{
	cleardevice();

	int nmidy;
	nmidy=midy+190;

	settextstyle(9,HORIZ_DIR,5);
	settextjustify(CENTER_TEXT,CENTER_TEXT);
	setcolor(15);

	for(i=0;i<=4;i++)
		outtextxy(midx-2+i,50,"Credits") ;

	settextstyle(6,HORIZ_DIR,3);
	outtextxy(midx,120,"A very sincere and heartfelt thank you to");
	settextstyle(2,HORIZ_DIR,18);
	outtextxy(midx,150,"SMRITI MAAM");
	settextstyle(6,HORIZ_DIR,3);
	outtextxy(midx,200,"EFFORTS BY:");

	settextstyle(6,HORIZ_DIR,4);
	setcolor(15);

	for(i=0;i<=4;i++)
		outtextxy(midx,240,"Mridul Mittal") ;

	for(i=0;i<=4;i++)
		outtextxy(midx,270,"Jainil Shah") ;

	for(i=0;i<=4;i++)
		outtextxy(midx,300,"Hardik Shah") ;
```

```
	setcolor(LIGHTCYAN);
	setlinestyle(0,0,3);

	line(midx-50,nmidy-10,midx,nmidy-100);
	line(midx,nmidy-100,midx+50, nmidy-10);
	line(midx-50,nmidy-10,midx+50, nmidy-10);
	line(midx, nmidy-100, midx, nmidy-10);
	circle(midx, nmidy-39, 29);

	setlinestyle(0,0,1);

	setcolor(RED);
	settextstyle(6,HORIZ_DIR,3);
	outtextxy(midx+1,nmidy,"PRODUCTIONS");
	outtextxy(midx+2,nmidy,"PRODUCTIONS");
	outtextxy(midx+3,nmidy,"PRODUCTIONS");

	settextstyle(2,0,4);
	settextjustify(0,1);
	setcolor(15);

	if(d==0)
		outtextxy(2,maxy-20,"Press ENTER key to return");
	else if(d==1)
		outtextxy(2,maxy-20,"Press ENTER key to exit");

	while(y==0)
	{
		while (bioskey(1) == 0);
			key = bioskey(0);

		if(key==7181)
			break;
	}

	if(d==0)
		cover();
	else if(d==1)
	{
		cleardevice();
		exit(0);
	}
}

void convert_score()
{
	int Z=1;
	int number[12],r, p1=0;
	int NUMBER[12],R,P1=0;

	long x,X;

	setcolor(WHITE);
	settextstyle(3,0,1);
	outtextxy(70,10,"Your Score: ");

	x=score;
	X=C[0].bscore;
```

```
	setcolor(WHITE);
	settextstyle(3,0,2);

	if(x==0)
		outtextxy(138,10,"00");

	if(x!=0 && Z!=0)
	{
		setcolor(BLACK);
		outtextxy(138,10,"00");
		Z=0;
	}

	while (x > 0)
	{
		r=x%10;
		x=x/10;
		number[p1]=r;
		p1++;
	}

	while (X > 0)
	{
		R=X%10;
		X=X/10;
		NUMBER[P1]=R;
		P1++;
	}

	int AA=0,MM=0;

	for(i=0; i<P1; i++)
	{
		setcolor(BLACK);
		outtextxy(130+MM,10,basearray[NUMBER[P1-i-1]]);
		MM+=15;
	}

	for(i=0; i<p1; i++)
	{
		setcolor(WHITE);
		outtextxy(130+AA,10,basearray[number[p1-i-1]]);
		AA+=15;
	}

	settextstyle(0,0,2);
}

void convert_time()
{
	int x;
	int number[12],r, p1=0;

	setcolor(WHITE);
	settextstyle(3,0,1);

	outtextxy(470,10,"Time: ");
	x=T;
```

```
	setcolor(WHITE);
	settextstyle(3,0,2);

	while (x > 0)
	{
		r=x%10;
		x=x/10;
		number[p1]=r;
		p1++;
	}

	int AA=0;

	for(i=0; i<p1; i++)
	{
		setcolor(WHITE);
		outtextxy(500+AA,10,basearray[number[p1-i-1]]);
		AA+=15;
	}
	settextstyle(0,0,2);
}

void print()
{
	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
		{
			m=A[i][j];
			n=0;

			if(m!=0)
			{
				n=log10(m)/log10(2);

				setfillstyle(1,0);
				bar(midx-156+(80*j),midy-156+(80*i),midx-
84+(80*j),midy-84+(80*i));

				setcolor(n);
				setfillstyle(1, n);
				bar(midx-156+(80*j),midy-156+(80*i),midx-
84+(80*j),midy-84+(80*i));

				setcolor(0);
				outtextxy(midx-120+(80*j), midy-
120+(80*i),NUM[n-1]);
			}

			else
			{
				setfillstyle(1,0);
				setcolor(0);
				bar(midx-156+(80*j),midy-156+(80*i),midx-
84+(80*j),midy-84+(80*i));
			}
		}
```

```
	}
}

void left()	//left arrow key
{
	for(c=0;c<2;c++)
	{
		for(i=0;i<4;i++)
		{
			for(j=0;j<4;j++)
			{
				if(A[i][j]==0)
				{
					for(k=j;k<4;k++)
						A[i][k]=A[i][k+1];

					A[i][3]=0;
				}
			}
		}
	}

	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
		{
			if(A[i][j]==A[i][j+1])
			{
				A[i][j]*=2;
				if(A[i][j]!=0)
				{
					m1[m2]=A[i][j];
					m2++;
				}
				score+=A[i][j];

				for(k=j+1;k<4;k++)
					A[i][k]=A[i][k+1];

				A[i][3]=0;
			}
		}
	}
}

void right()	///right arrow key
{
	for(c=0;c<2;c++)

	{
		for(i=0;i<4;i++)
		{
			for(j=3;j>=0;j--)
			{
				if(A[i][j]==0)
				{
					for(k=j;k>=0;k--)
					A[i][k]=A[i][k-1];
```

```
                        A[i][0]=0;
                        }
                  }
            }
      }

      for(i=0;i<4;i++)
      {
            for(j=3;j>=0;j--)
            {
                  if(A[i][j]==A[i][j-1])
                  {
                  A[i][j]*=2;
                  if(A[i][j]!=0)
                        {
                              m1[m2]=A[i][j];
                              m2++;
                        }
                  score+=A[i][j];

                  for(k=j-1;k>=0;k--)
                        A[i][k]=A[i][k-1];
                  A[i][0]=0;
                  }
            }
      }
}

void up()   //up arrow key
{
      for(c=0;c<2;c++)
      {
            for(i=0;i<4;i++)
            {
                  for(j=0;j<4;j++)
                  {
                        if(A[j][i]==0)
                        {
                              for(k=j;k<4;k++)
                                    A[k][i]=A[k+1][i];
                        A[3][i]=0;
                        }
                  }
            }
      }

      for(i=0;i<4;i++)
      {
            for(j=0;j<4;j++)
            {
                  if(A[j][i]==A[j+1][i])
                  {
                  A[j][i]*=2;

                  if(A[j][i]!=0)
                        {
                              m1[m2]=A[j][i];
                              m2++;
```

```
                        }
                  score+=A[j][i];
                        for(k=j+1;k<4;k++)
                              A[k][i]=A[k+1][i];
                  A[3][i]=0;
                  }
            }
      }
}

void down() //down arrow key
{
      for(c=0;c<2;c++)
      {
            for(i=0;i<4;i++)
            {
                  for(j=3;j>=0;j--)
                  {
                        if(A[j][i]==0)
                        {
                              for(k=j;k>=0;k--)
                                    A[k][i]=A[k-1][i];
                        A[0][i]=0;
                        }
                  }
            }
      }

      for(i=0;i<4;i++)
      {
            for(j=3;j>=0;j--)
            {
                  if(A[j][i]==A[j-1][i])
                  {
                  A[j][i]*=2;

                  if(A[j][i]!=0)
                        {
                              m1[m2]=A[j][i];
                              m2++;
                        }
                  score+=A[j][i];
                        for(k=j-1;k>=0;k--)
                              A[k][i]=A[k-1][i];
                  A[0][i]=0;
                  }
            }
      }
}

void loadscreen()
{
      cleardevice();
      int endangle,radius=110,t=0,x;
      setcolor(2);
      settextjustify(CENTER_TEXT,CENTER_TEXT);

      char h[][40]={"defining variables","loading header files...",
```

```
"#include<iostream.h>","#include<conio.h>","#include<string.h>",

"#include<stdio.h>","#include<math.h>","#include<graphics.h>",

"#include<process.h>","#include<dos.h>","#include<stdlib.h>",
                "creating functions","loading 2048","defining
database","extracting directory",
                "printing virtual grid","loading homescreen"};

      setcolor(GREEN);
      settextstyle(6,HORIZ_DIR,6);

      for(i=0;i<=4;i++)
            outtextxy(midx-2+i,midy+170,"...LOADING...");

      settextstyle(0,HORIZ_DIR,4);
      outtextxy(midx,50,"A VENTURE BY");

      setcolor(LIGHTCYAN);
      setlinestyle(0,0,3);

      line(midx-50,midy-10,midx,midy-100);
      line(midx,midy-100,midx+50, midy-10);
      line(midx-50,midy-10,midx+50, midy-10);
      line(midx, midy-100, midx, midy-10);
      circle(midx, midy-39, 29);

      setlinestyle(0,0,1);

      setcolor(RED);
      settextstyle(6,HORIZ_DIR,3);
      outtextxy(midx+1,midy,"PRODUCTIONS");
      outtextxy(midx+2,midy,"PRODUCTIONS");
      outtextxy(midx+3,midy,"PRODUCTIONS");

      for(endangle=90;endangle<=450;endangle++)
      {
            delay(8);
            setcolor(15);
            arc(midx, midy-30,90, endangle, radius);

            for(i=0;i<=10;i++)
                  arc(midx, midy-30,90, endangle, radius-i);

            x=endangle%20;

            if(x==0)
            {
                  setfillstyle(1,0);
                  bar(midx-150,midy+100,midx+150,midy+140);
                  setcolor(YELLOW);
                  settextstyle(6,HORIZ_DIR,3);
                  outtextxy(midx,midy+120,h[t]);
                  t++;x++;
            }

      }
```

```
	setcolor(CYAN);
	settextstyle(6,HORIZ_DIR,5);

	for(i=0;i<=4;i++)
		outtextxy(midx-2+i,midy+120,"READY!");

	delay(800);
	cleardevice();
}

void initialize()
{
	void game();
	score=0;
	T=5;
	for(i=0;i<4;i++)                        // to clear the grid
	{
		for(j=0;j<4;j++)
		{
			C[0].B[i][j]=0;
			C[0].bscore=0;
			C[1].B[i][j]=0;

			C[1].bscore=0;
		}
	}

	for(i=0; i<16; i++)
	{
		INSERT[i].I=0;
		INSERT[j].J=0;
	}

	for(i=0; i<4; i++)
	{
		for(j=0; j<4; j++)
			A[i][j]=0;
	}

	setfillstyle(1,15);

	cleardevice();

	// for grid
	for(i=-2;i<=2;i++)
//columns
		bar(midx+(80*i)-1,midy-161,midx+(80*i)+1,midy+161);
	for(i=-2;i<=2;i++)                                          // rows
		bar(midx-161,midy+(80*i)-1,midx+161,midy+(80*i)+1);

	settextjustify(1,1);
	setcolor(6);
	settextstyle(0,0,5);
	outtextxy(midx+5,midy-205,"2048");

	settextstyle(2,0,4);
	settextjustify(0,1);
```

```
	setcolor(15);
	outtextxy(2,maxy-20,"Press ENTER/ESC key to quit");

	settextjustify(1,1);
	settextstyle(0,0,2);

	for(i=0;i<2;i++)
	{
		new1=random(4);
		new2=random(4);
		new3=random(9);

		A[new1][new2]=X[new3];
		C[1].B[new1][new2]=X[new3];
		C[1].bscore=score;

		print();
	}
	game();

}

void UNDO()
{

	undo=0;

	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
			A[i][j]=C[0].B[i][j];
	}

	C[1].bscore=C[0].bscore;
	C[0].bscore=score;
	score=C[1].bscore;

	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
			C[1].B[i][j]=C[0].B[i][j];
	}
	print();
}

void mode(int dop)
{
	cleardevice();

	settextstyle(9,HORIZ_DIR,5);
	settextjustify(CENTER_TEXT,CENTER_TEXT);
	setcolor(15);

	for(int i=0;i<=4;i++)
		outtextxy(midx-2+i,50,"Choose Mode") ;

	int n=0,position=-1;
	char opt[][20]={"PRACTICE","CLASSIC","SURVIVAL"};
```

```
	setfillstyle(1,0);
	settextjustify(CENTER_TEXT,CENTER_TEXT);

	settextstyle(0,0,5);
	setfillstyle(1,15);
	setcolor(0);

	bar(midx-200,midy+(50*position)-
30,midx+200,midy+20+(50*position));
	outtextxy(midx,midy+(50*position),opt[position+1]);

	settextstyle(0,0,3);
	setcolor(15);

	outtextxy(midx,midy,"CLASSIC");
	outtextxy(midx,midy+50,"SURVIVAL");

	settextstyle(2,0,4);
	settextjustify(0,1);
	outtextxy(2,maxy-20,"Use arrow keys to choose.Press ENTER key to
continue.");
	outtextxy(2,maxy-40,"Press BACKSPACE key to go back.");

	settextjustify(CENTER_TEXT,CENTER_TEXT);

	while(n==0)
	{
		while (bioskey(1) == 0);
			key = bioskey(0);

		if(key==18432 && position>=-1)
		{
			settextstyle(0,0,3);
			setfillstyle(1,0);
			setcolor(15);
			bar(midx-200,midy+(50*position)-
30,midx+200,midy+20+(50*position));
			outtextxy(midx,midy+(50*position),opt[position+1]);

			if(position==-1)
				position=1;
			else
				position--;

			settextstyle(0,0,5);
			setfillstyle(1,15);
			setcolor(0);
			bar(midx-200,midy+(50*position)-
30,midx+200,midy+20+(50*position));
			outtextxy(midx,midy+(50*position),opt[position+1]);
		}

		else if(key==20480 && position<=1)
		{
			settextstyle(0,0,3);
			setfillstyle(1,0);
			setcolor(15);
```

```
                  bar(midx-200,midy+(50*position)-
30,midx+200,midy+20+(50*position));
                  outtextxy(midx,midy+(50*position),opt[position+1]);

                  if(position==1)
                        position=-1;
                  else
                        position++;

                  settextstyle(0,0,5);
                  setfillstyle(1,15);
                  setcolor(0);
                  bar(midx-200,midy+(50*position)-
30,midx+200,midy+20+(50*position));
                  outtextxy(midx,midy+(50*position),opt[position+1]);
            }

            else if(key==3592)          //backspace
                  cover();

            else if(key==7181)
                  break;
      }

      if(position==-1)
      {
            decider=0;
            strcpy(fname,"scorep.txt");
      }

      else if(position==0)
      {
            decider=1;
            strcpy(fname,"scorec.txt");
      }

      else if(position==1)
      {
            decider=2;
            strcpy(fname,"scores.txt");
      }

      if(dop==0)
            initialize();

      else if(dop==2)
            obj.display();
}

void game()
{
      T=5;
      DELAY=0;
      score=0;
      tp=0;

      while(tp==0)
```

```
{
	timer:;
	convert_score();

	flag=0;
	undo=1;

	if(decider==2)
	{
		while(kbhit()==0)
		{

			convert_time();

			delay(1);
			DELAY++;

			if(DELAY==450)
			{
				T--;
				setfillstyle(1,0);
				bar(450,0,560,50);

				if(T<0)
				{
					over=0;
					goto b;
				}
				DELAY=0;
			}
		}
	}

	while (bioskey(1) == 0);
		key = bioskey(0);

	if(key==19200)		//left arrow
		left();

	else if(key==19712)		//right arrow key
		right();

	else if(key==18432)		//up arrow key
		up();

	else if(key==20480)		//down arrow key
		down();

	else if((key==5749||key==5717) && decider==0) //undo: U or u
		UNDO();

	else if(key==7181 || key==283)	//exit: enter key or esc
		OVER(0);

	else
		goto end;

	if(undo!=0)
```

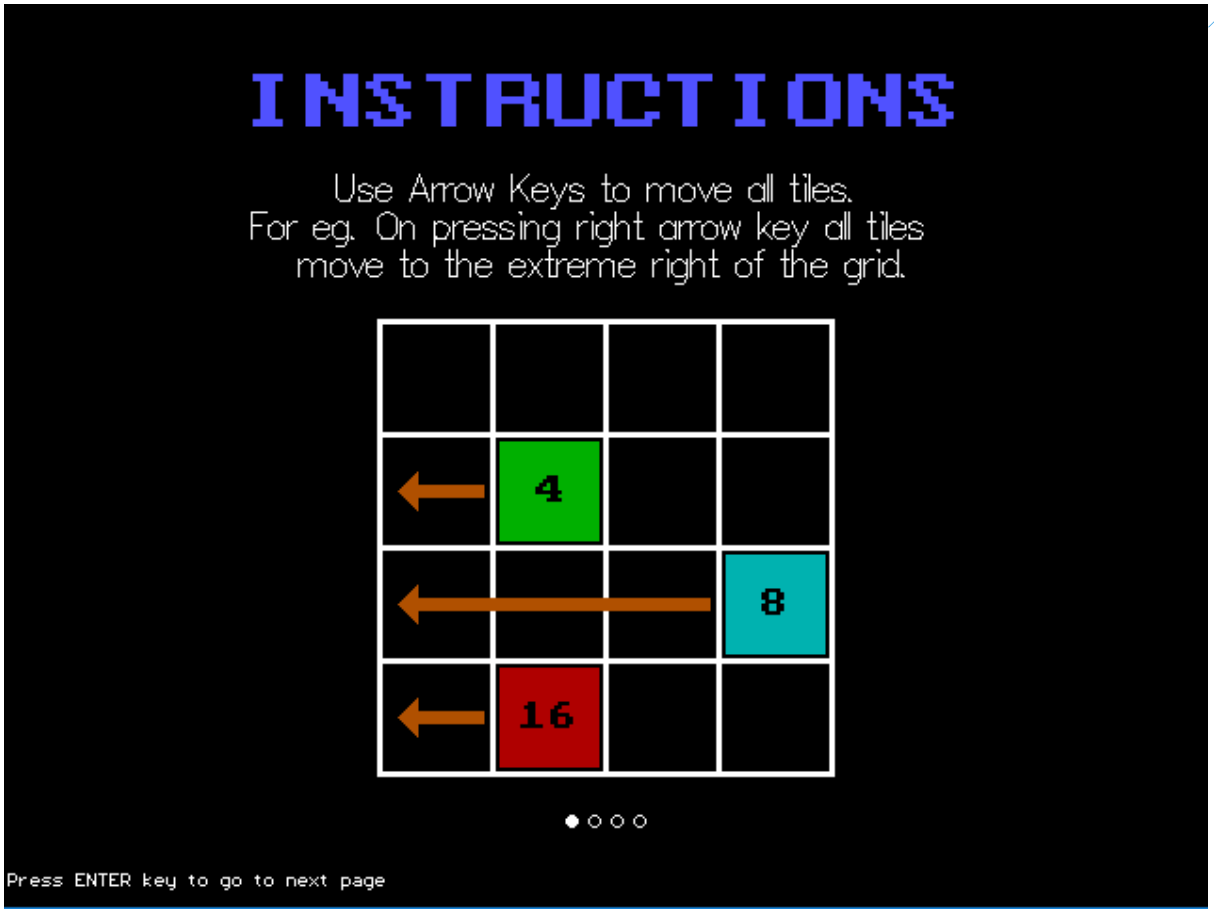
```
{
	for(i=0;i<4;i++)
	{
		for(j=0;j<4;j++)
		{
			if(C[1].B[i][j]!=A[i][j])
				flag++;

		}
	}
}

a=0;
over=0;

for(i=0;i<4;i++)
{
	for(j=0;j<4;j++)
	{
		if(A[i][j]==0)
		{
			INSERT[a].J=j;
			INSERT[a].I=i;
			a++;
		}
	}
}


if(a!=0 && undo!=0)
{
	if(flag!=0)
	{
		new1=random(9);
		new2=random(a);

		A[INSERT[new2].I][INSERT[new2].J]=X[new1];

		for(i=0;i<4;i++)
		{
			for(j=0;j<4;j++)
				C[0].B[i][j]=C[1].B[i][j];
		}

		C[0].bscore=C[1].bscore;
		C[1].bscore=score;

		for(i=0;i<4;i++)
		{
			for(j=0;j<4;j++)
				C[1].B[i][j]=A[i][j];
		}

		print();
	}
}

for(i=0;i<4;i++)  //to check end of game
```

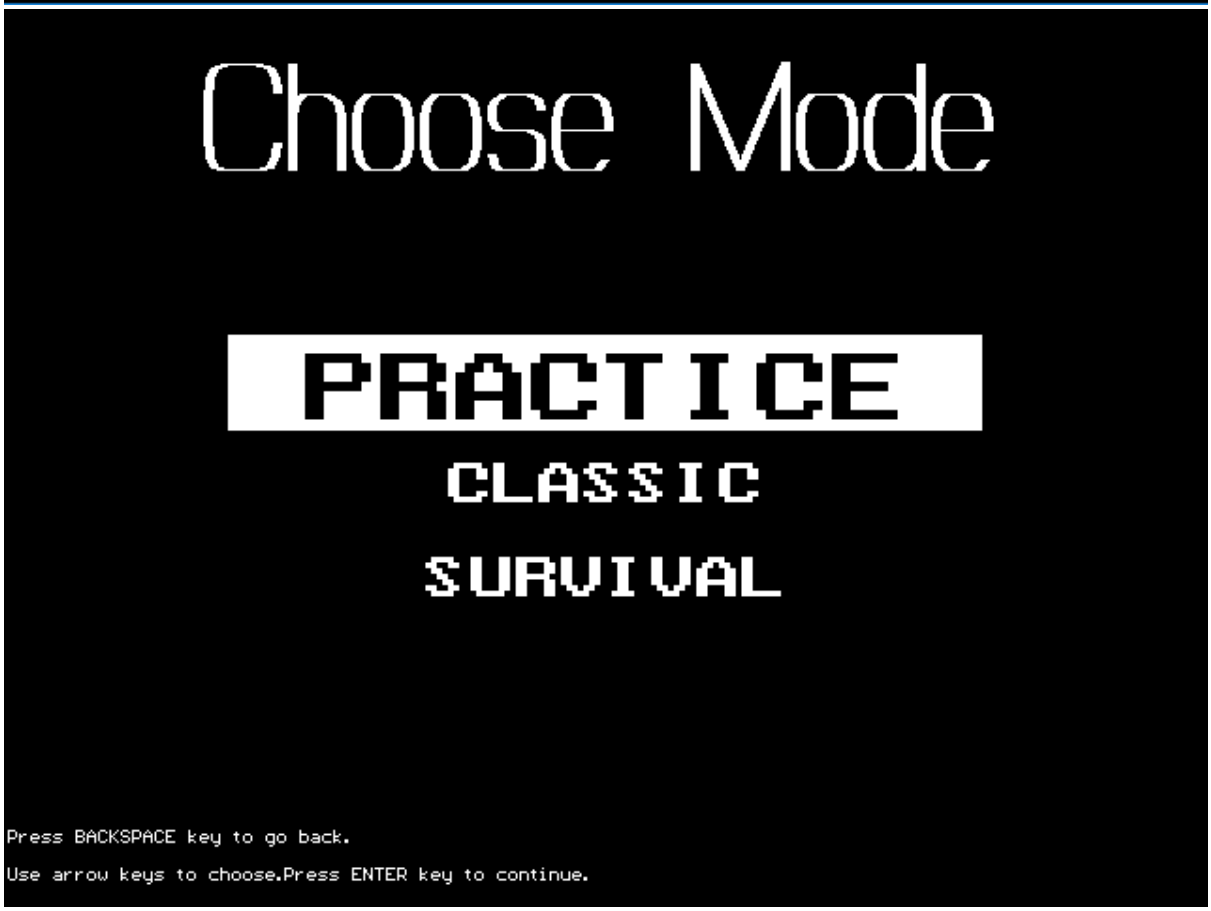
```
            {
                  for(j=0;j<4;j++)
                  {
                        if(A[i][j]==0)
                              over++;

                        if(A[i][j]==A[i][j+1] || A[i][j]==A[i+1][j])
                              over++;

                        if(A[i][j]==2048)
                        {
                              OVER(1);
                              delay(250);
                        }
                  }
            }

            b:;

            if(over==0)
            {
                   delay(1000);
                   OVER(0);
            }

            if(decider==2)
            {
                  for(i=0;i<m2;i++)
                  {
                        if(m1[i]>=8)
                        {           setfillstyle(1,0);
                              bar(450,0,560,50);
                              T++;
                              convert_time();
                        }
                  }
            }

            m2=0;
            end:
      }
}

void main()
{
      /* request auto detection */
      int gdriver = DETECT, gmode, errorcode;

      /* initialize graphics and local variables */
      initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");

      /* read result of initialization */
      errorcode = graphresult();

      if (errorcode != grOk)  /* an error occurred */
      {
               printf("Graphics error: %s\n", grapherrormsg(errorcode));
               printf("Press any key to halt:");
```

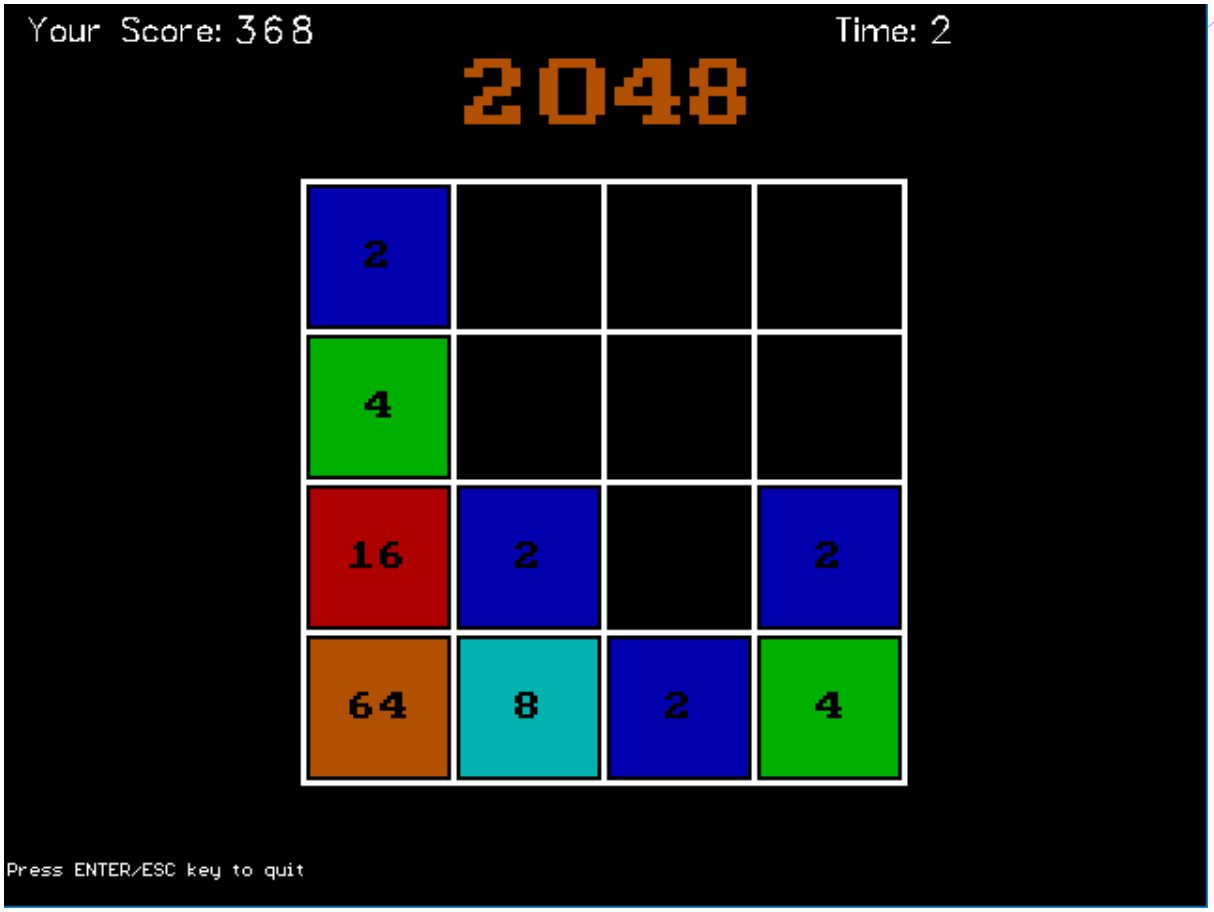
```
                getch();
                exit(1); /* terminate with an error code */
             }
      randomize();

      maxx=getmaxx();
      maxy=getmaxy();
      midx=maxx/2;
      midy1=maxy/2;
      midy=(maxy/2)+15;
      ft=0;

      storekey();
      loadscreen();
      cover();

      cleardevice();
      closegraph();
}
```

//------END------//

Screen Shots

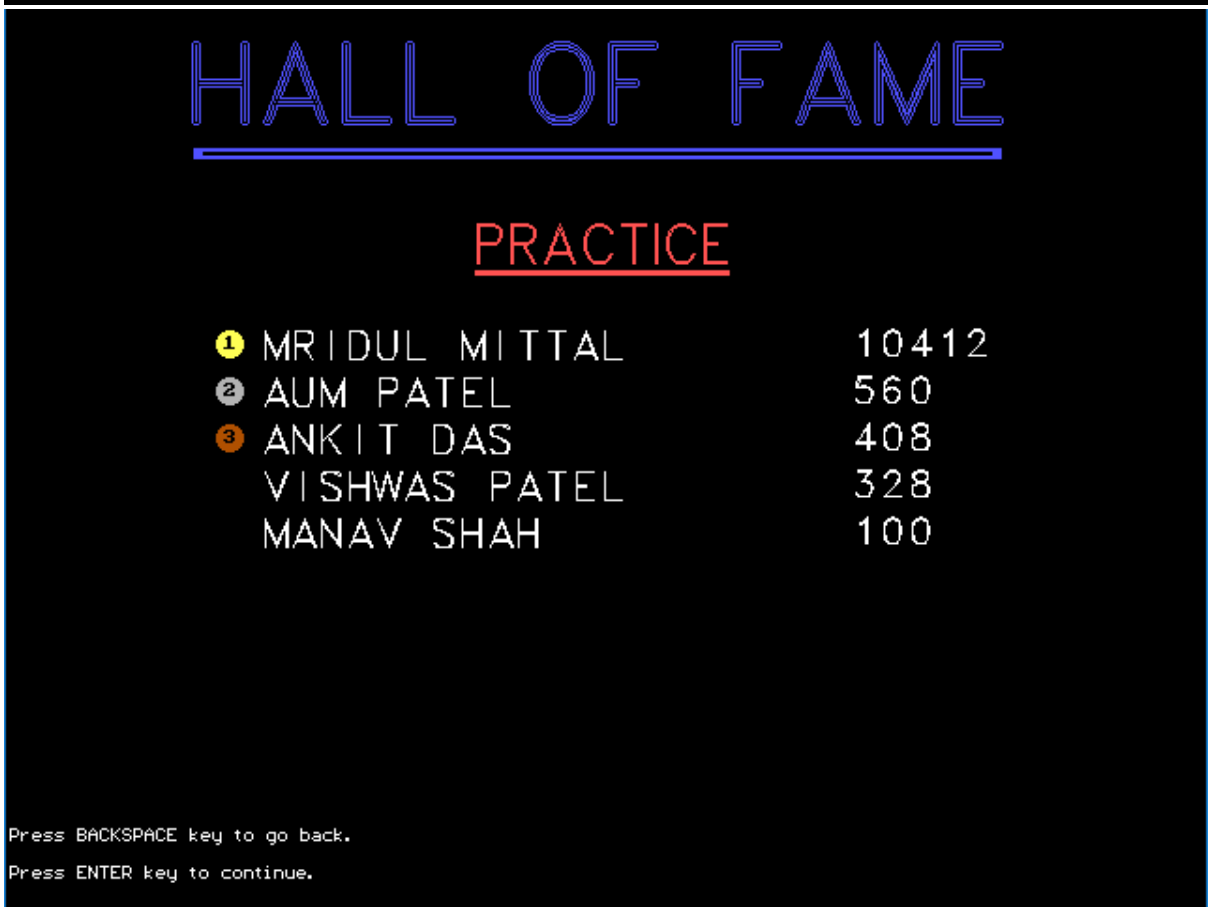

INSTRUCTIONS
Use Arrow Keys to move all tiles.
For eg. On pressing right arrow key all tiles
move to the extreme right of the grid.
4
8
16
Press ENTER key to go to next page


Choose Mode
PRACTICE
CLASSIC
SURVIVAL
Press BACKSPACE key to go back.
Use arrow keys to choose.Press ENTER key to continue.

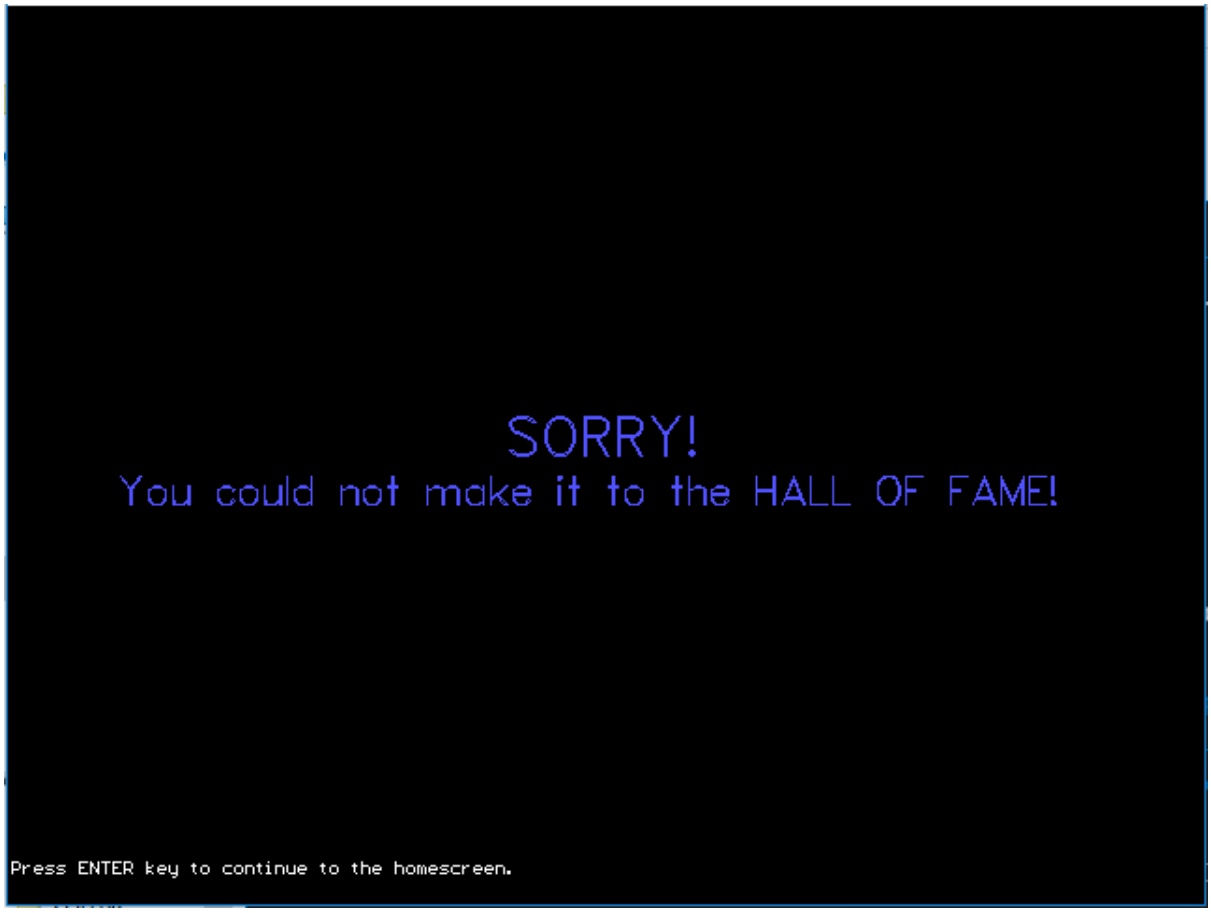

Your Score: 368
Time: 2
2048
2
4
16
2
2
64
8
2
4
Press ENTER/ESC key to quit


YOU WIN!
Your Score: 24568
Press ENTER key to continue.


NEW HIGHSCORE!!
Congratulations
Enter Your Name:
JAINIL SH
Type your name and press ENTER key to submit.


HALL OF FAME
PRACTICE
1 MRIDUL MITTAL 10412
2 AUM PATEL 560
3 ANKIT DAS 408
VISHWAS PATEL 328
MANAV SHAH 100
Press BACKSPACE key to go back.
Press ENTER key to continue.


GAME OVER
Your Score: 1848
Press ENTER key to continue.


SORRY!
You could not make it to the HALL OF FAME!
Press ENTER key to continue to the homescreen.


Credits
A very sincere and heartfelt thank you to
SMRITI MAAM
EFFORTS BY:
Mridul Mittal
Jainil Shah
Hardik Shah
PRODUCTIONS
Press ENTER key to return