BattleShip

Pacific Warriors

Programmed by :

Gagandeep Singh

2008 - 2009

Cover Designed by : Gagandeep Singh

BattleShip

Pacific Warriors

Programmed and compiled in :

Turbo C++ version 3.0

Programmed by :

Gagandeep Singh

Year 2008 - 2009

ACKNOWLEDGMENT

I express my deep sense of gratitude and obligation to my respected teacher Mrs. Vandana Arora for her valuable guidance, interest and constant encouragement given to me for the fulfillment of this project.

I am also grateful to my parents and also my friends for providing me required material and helping in compiling the project.

Gagandeep Singh

Preamble

The project titled ' BattleShip - Pacific Warriors ' has been programmed as per the requirement of **CBSE** for the subject Computer Science (083) for AISSCE -2008-09. The source code has been programmed and Compiled in C++ version 3.0.

The Project aims at developing a event game on C++ using concepts of graphics , sounds , structure , classes , datafile handling as described in Class XI and XII text book.

Gagandeep Singh

Contents

Data.h Header File

Data.h Header Files includes all the necessary functions required by main program.
Number of functions : 56
Code Length : 2741 lines , 4805 words

Main Program Code

Number of functions : 2
Code Length : 1142 lines , 2082 words

PROGRAM OUTPUT

Game Requirements Verifier

This program search for missing file required by the game and prompts an error.
Code Length : 86 lines , 151 words
OUTPUT

Game Uninstall Wizard

BattleShip Uninstall Shield removes all the game files from computer hard disk on user choice .
Code Length : 157 lines , 238 words
OUTPUT

Inside the project

Flow Chart

OOP About the project

Bibliography

User Guide

BattleShip
Pacific Warriors

DATA.H Header File

‘ Data.h ’ Header Files includes all the necessary functions required by btlship.cpp .These functions are complete unit in itself and work in accordance with other functions.

Header File Code

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
#include<stdlib.h>
#include<fstream.h>


int i;


void game_sounds(int a)
{
 switch(a)
 {
  case -2 :          // lost game
              for(i=850;i>=500;i=i-10)
              {
               sound(i);
               delay(25);
               nosound();
              }
               sound(500);
               delay(450);
               nosound();
               break;

  case -1 :          // termination error
              sound(900);
              delay(900);
              nosound();
              break;

  case 0 :           // user error
              sound(900);
              delay(200);
              nosound();
```

```
            delay(100);

            sound(900);
            delay(200);
            nosound();
            break;

 case 1 :           // hit
            sound(120);
            delay(210);
            nosound();
            break;

 case 2 :      // WIN
            sound(300);
            delay(200);
            nosound();
            sound(500);
            delay(900);
            nosound();

            delay(500);

            sound(300);
            delay(200);
            nosound();
            sound(500);
            delay(900);
            nosound();
            break;
 }//end of switch

}
/*--------------------------------------------------------------------------*/

void system_chk(int z)
{
 if(z!=grOk)
 {cout<<"Battleship  Pacific Warriors "<<endl;
  cout<<"Developed By : Gagandeep (2008-2009)"<<endl;
  cout<<"-----------------------------------------------------\n";
  cout<<"\t\tGraphic load Error!\n\n";
  cout<<" Please make sure that the graphics driver file\n";
  cout<<" EGAVGA.BGI exist in the folder.\n";
  cout<<"-----------------------------------------------------\n";
  game_sounds(-1);
  getch();
  exit(0);
 }
 else
 {
```

```
cout<<"                    WARNING !"<<endl;
cout<<"This game requires couple of font files of extension '.CHR'";
cout<<"\nWithout these file the game may start malfunctioning!";
cout<<"\nTo avoid such problem it is advisable that you check";
cout<<"\nthe system requirements.";
cout<<"\n\nDo you want to check the requirements(y/n) ? ";
fstream f;
int ctr=0;
char a;
char file_name[32][25]={"EGAVGA.BGI",
                        "TRIP.CHR",
                        "LITT.CHR",
                        "SANS.CHR",
                        "GOTH.CHR",
                        "SCRI.CHR",
                        "SIMP.CHR",
                        "TSCR.CHR",
                        "LCOM.CHR",
                        "EURO.CHR",
                        "BOLD.CHR",
                        "levels\\level00.dat",
                        "levels\\level01.dat",
                        "levels\\level02.dat",
                        "levels\\level03.dat",
                        "levels\\level04.dat",
                        "levels\\level05.dat",
                        "levels\\level06.dat",
                        "levels\\level07.dat",
                        "levels\\level08.dat",
                        "levels\\level09.dat",
                        "levels\\level10.dat",
                        "levels\\level11.dat",
                        "levels\\level12.dat",
                        "levels\\level13.dat",
                        "levels\\level14.dat",
                        "levels\\level15.dat",
                        "levels\\level16.dat",
                        "levels\\level17.dat",
                        "levels\\level18.dat",
                        "levels\\level19.dat",
                        "levels\\level20.dat", };

cin>>a;
if((a=='y') || (a=='Y'))
{
 cleardevice();
 gotoxy(1,1);
 cout<<"                BattleShip - Pacific Warriors"<<endl;
 cout<<"                 Verifying Game Requirements "<<endl;
 cout<<"\nThe following files are under error state : ";
 cout<<"\n--------------------------------------------------------------------------"<<endl;
 cout<<"    File Name                    Status";
```

```
  cout<<"\n--------------------------------------------------------------------------------"<<endl;
  int temp=0;
  for(i=0;i<32;i++)
  {
   f.open(file_name[i],ios::in);
    if(f==0)
    {
      if(i<=11)
            {cout<<"    "<<file_name[i]<<"                        missing"<<endl; }
      else
            {cout<<"    "<<file_name[i]<<"                missing"<<endl; }
      ctr++;
      temp++;
    }
   f.close();


  if(temp>12)
  {cout<<"\n\n   Press any key to continue...."; getch(); temp=0; cout<<"\n\n";}

  }

  // comments
  if(ctr==0)
   {cout<<"\n\n                  No Error Found!"; }
  else
   {cout<<"\n\n    Please Make sure that these file are present in the folder";
     cout<<"                                or reinstall the game!";
   }
  cout<<"\n--------------------------------------------------------------------------------"<<endl;
  cout<<"\n        Enter any Key to Exit and Log in the Game !";

  getch();
  }
 }
 cleardevice();
}

/*--------------------------------------------------------------------------------*/

void logo(int x,int y)
{
 settextstyle(GOTHIC_FONT,0,8);
 setcolor(WHITE);
 outtextxy(x,y,"BattleShip");
 settextstyle(8,0,2);
 setcolor(14);
 outtextxy(x+70,y+87,"Pacific Warriors");

 line(x-10,y+100,x+60,y+100);         //left 1
 line(x,y+105,x+60,y+105);           //left 2
 line(x+10,y+110,x+60,y+110);         //left 3
```

```
 line(x+260,y+100,x+330,y+100);         //right 1
 line(x+260,y+105,x+320,y+105);         //right 2
 line(x+260,y+110,x+310,y+110);         //right 3

}

/*-------------------------------------------------------------*/

void trademark(int col)
{
 setcolor(col);
 settextstyle(0,0,1);
 outtextxy(10,335,"Programmed by : Gagandeep Singh (2008-2009)    Compiled in : Turbo C++ v3.0");
}

/*-------------------------------------------------------------*/

void intro()
{
 delay(2500);
 settextstyle(1,0,3);
 outtextxy(160,120,"BattleShip - Pacific Warriors");
 setcolor(4);
 line(150,151,500,151);
 settextstyle(5,0,1);
 setcolor(WHITE);
 delay(1200);
 outtextxy(265,160,"Programmed By :");
 delay(100);
 settextstyle(2,0,7);
 outtextxy(165,190,"G a g a n d e e p  S i n g h");
 delay(100);
 settextstyle(2,0,6);
 outtextxy(310,215,"2008");
 getch();
}

/*-------------------------------------------------------------*/

void loading_game()
{
 intro();
 cleardevice();

 // Ship Graphics
 // right guns
   int G1_R[]={395,82 , 450,65, 453,70 , 395,97 };
   int G2_R[]={410,100 , 470,82, 473,87 , 410,115 };
   setfillstyle(1,8);
   setcolor(WHITE);
   fillpoly(4,G1_R);
```

```
  fillpoly(4,G2_R);

// Left guns
  int G1_L[]={265,82 , 210,65, 207,70 , 265,97 };
  int G2_L[]={250,100 , 190,82, 187,87 , 250,115 };
  setfillstyle(1,8);
  setcolor(WHITE);
  fillpoly(4,G1_L);
  fillpoly(4,G2_L);

// tower
  setfillstyle(1,8);       // antena 1
  setcolor(0);
  bar(328,3,331,105);
  bar(310,10,350,12);

  bar(299,15,301,105);     // antena 2
  bar(288,20,312,21);

  setfillstyle(1,8);       // main tower
  setcolor(WHITE);
  bar(310,40,350,105);
  rectangle(310,40,350,105);

  setfillstyle(1,WHITE);
  setcolor(8);
  fillellipse(330,52,7,7);


// floor 3
  setfillstyle(1,8);
  setcolor(WHITE);
  bar(290,65,370,105);
  rectangle(290,65,370,105);

// floor 2
  setfillstyle(1,7);
  setcolor(WHITE);
  bar(265,75,395,105);
  rectangle(265,75,395,105);

  setfillstyle(1,0);
  setcolor(7);
  bar(280,82,380,88);
  line(300,82,300,88);
  line(320,82,320,88);
  line(340,82,340,88);
  line(360,82,360,88);


// floor 1
  setfillstyle(1,8);
```

```
   setcolor(WHITE);
   bar(250,95,410,140);
   rectangle(250,95,410,140);

 // ship base
   int ship_base[]={330,105 , 435,130 , 405,200 , 255,200 , 225,130 };
   setfillstyle(1,6);
   setcolor(WHITE);
   fillpoly(5,ship_base);
   setcolor(WHITE);
   line(330,100,330,200);

 // fence
   setcolor(WHITE);
   line(330,100,225,125); line(330,100,435,125);
   line(225,125,225,130); line(435,125,435,130);

 // logo display
   settextstyle(GOTHIC_FONT,0,8);
   setcolor(0);
   outtextxy(162,152,"BattleShip");
   logo(160,150);

 // loading

 delay(3000);
 settextstyle(SMALL_FONT,0,5);
 setcolor(WHITE);
 outtextxy(300,285,"Loading...");
 delay(500);
 setfillstyle(1,4);
 for(i=100;i<=540;i=i+10)
 {bar(100,310,i,305);
   delay(200);
 }

 delay(1000);
 setfillstyle(1,0);
 bar(295,280,380,300);

 setcolor(WHITE);
 outtextxy(250,315,"Press Enter to Continue.");
 getch();

}

/*---------------------------- Buttons --------------------------------*/

void button_play(int st)
{
 if(st==1)
 {setfillstyle(1,1);
```

```
  bar(150,150,450,180);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(270,150,"Play");
 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,150,450,180);
  rectangle(150,150,450,180);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(270,150,"Play");
 }
}
              /*---------------------------------------------*/
void button_score(int st)
{
 if(st==2)
 {setfillstyle(1,1);
  bar(150,180,450,210);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(245,180,"High Score");
 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,180,450,210);
  rectangle(150,180,450,210);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(245,180,"High Score");
 }
}
              /*---------------------------------------------*/
void button_help(int st)
{
 if(st==3)
 {setfillstyle(1,1);
  bar(150,210,450,240);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(190,210,"Instructions & Help");
 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,210,450,240);
  rectangle(150,210,450,240);
  setcolor(1);
  settextstyle(7,0,2);
```

```
  outtextxy(190,210,"Instructions & Help");
 }
}
              /*--------------------------------------------*/
void button_credits(int st)
{
 if(st==4)
 {setfillstyle(1,1);
  bar(150,240,450,270);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(260,240,"Credits");
 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,240,450,270);
  rectangle(150,240,450,270);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(260,240,"Credits");
 }
}
              /*--------------------------------------------*/
void button_exit(int st)
{
 if(st==5)
 {setfillstyle(1,1);
  bar(150,270,450,300);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(270,270,"Exit");
 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,270,450,300);
  rectangle(150,270,450,300);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(270,270,"Exit");
 }
}

     /*-------------------- game mode ------------------------*/
void button_mission(int st)
{
 if(st==2)
 {setfillstyle(1,1);
  bar(150,180,450,210);
  setcolor(WHITE);
  settextstyle(7,0,2);
```

```
  outtextxy(255,180,"Mission");

  settextstyle(2,0,4);
  setcolor(0);
  outtextxy(10,180,"Discription :");
  outtextxy(10,200,"Include 20 exciting");
  outtextxy(10,210,"levels. Complete all");
  outtextxy(10,220,"level to unlock them");
  outtextxy(10,230,"and complete the game.");
  outtextxy(10,250,"Your Game gets ");
  outtextxy(10,260,"automatically saved on");
  outtextxy(10,270,"completing each level.");
  outtextxy(10,290,"Press Enter to Play !");

 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,180,450,210);
  rectangle(150,180,450,210);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(255,180,"Mission");

  settextstyle(2,0,4);
  setcolor(7);
  outtextxy(10,180,"Discription :");
  outtextxy(10,200,"Include 20 exciting");
  outtextxy(10,210,"levels. Complete all");
  outtextxy(10,220,"level to unlock them");
  outtextxy(10,230,"and complete the game.");
  outtextxy(10,250,"Your Game gets ");
  outtextxy(10,260,"automatically saved on");
  outtextxy(10,270,"completing each level.");

  outtextxy(10,290,"Press Enter to Play !");

 }
}
              /*-------------------------------------------*/
void button_arcade(int st)
{
 if(st==3)
 {setfillstyle(1,1);
  bar(150,210,450,240);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(255,210,"Arcade");

  settextstyle(2,0,4);
  setcolor(0);
  outtextxy(10,180,"Discription :");
```

```
  outtextxy(10,200,"Play any of the");
  outtextxy(10,210,"unlocked levels.");
  outtextxy(10,220,"To play locked level ,");
  outtextxy(10,230,"finish the Mission Mode.");
  outtextxy(10,250,"Wining this mode will");
  outtextxy(10,260,"not unlock next level.");

  outtextxy(10,290,"Press Enter to Play !");

 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,210,450,240);
  rectangle(150,210,450,240);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(255,210,"Arcade");

 }
}
              /*---------------------------------------------*/
void button_training(int st)
{
 if(st==4)
 {setfillstyle(1,1);
  bar(150,240,450,270);
  setcolor(WHITE);
  settextstyle(7,0,2);
  outtextxy(250,240,"Training");

  settextstyle(2,0,4);
  setcolor(0);
  outtextxy(10,180,"Discription :");
  outtextxy(10,200,"Play a random level");
  outtextxy(10,210,"select by computer to");
  outtextxy(10,220,"test your skills.");
  outtextxy(10,290,"Press Enter to Play !");

 }
 else
 {setcolor(1);
  setfillstyle(1,WHITE);
  bar(150,240,450,270);
  rectangle(150,240,450,270);
  setcolor(1);
  settextstyle(7,0,2);
  outtextxy(250,240,"Training");

 }
}
              /*---------------------------------------------*/
```

```
void button_back(int st)
{
 if(st==5)
 {setfillstyle(1,1);
   bar(150,270,450,300);
   setcolor(WHITE);
   settextstyle(7,0,2);
   outtextxy(270,270,"Back");

   settextstyle(2,0,4);
   setcolor(0);
   outtextxy(10,190,"Press Enter to ");
   outtextxy(10,200,"return to the");
   outtextxy(10,210,"Main Menu .");

 }
 else
 {setcolor(1);
   setfillstyle(1,WHITE);
   bar(150,270,450,300);
   rectangle(150,270,450,300);
   setcolor(1);
   settextstyle(7,0,2);
   outtextxy(270,270,"Back");
 }
}


/*-----------------------------------------------------------------*/
int option_screen()
{
 cleardevice();

 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);

 setfillstyle(1,4);           // seperate bar
 bar(0,125,640,130);
 trademark(1);

                /*------------- select ----------------*/
 char temp1;
 int opt,temp2=1;

 while(temp1!=13)    // enter
 {
```

```
 button_play(temp2);
 button_score(temp2);
 button_help(temp2);
 button_credits(temp2);
 button_exit(temp2);

                        // ---------- key display
 setfillstyle(1,14);
 settextstyle(0,0,2);
 bar(550,240,590,270);
  setcolor(4);
  settextstyle(0,0,2);
  outtextxy(560,250,"W");
  setcolor(1);
  settextstyle(2,0,4);
  outtextxy(578,241,"Up");

 bar(550,270,590,300);
  setcolor(4);
  settextstyle(0,0,2);
  outtextxy(560,275,"S");
  setcolor(1);
  settextstyle(2,0,4);
  outtextxy(565,288,"Down");

 setcolor(4);
 rectangle(550,240,590,270);
 rectangle(550,270,590,300);
                        //-------------------------

            //soud for active button
               sound(40);
               delay(100);
               nosound();
 temp1=getche();
 if((temp1=='w' || temp1=='W') && temp2!=1)
 {temp2--; }
 if((temp1=='s' || temp1=='S') && temp2!=5)
 {temp2++; }
 if(temp1==13)
 {opt=temp2; }

 }
              // sound for click
               sound(900);
               delay(150);
               nosound();
 return(opt);

}

/*-----------------------------------------------------------------*/
```

```
int exit_menu()
{
 cleardevice();

 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);

 setfillstyle(1,4);            // seperate bar
 bar(0,125,640,130);
 trademark(1);
     /*------------------ select ----------------*/

 settextstyle(7,0,4);
 outtextxy(140,150,"Do you want to exit ?");

 char temp1_ex;
 int opt_ex,temp2_ex=2;

 while(temp1_ex!=13)    // enter
 {
             // ---------- key display
setfillstyle(1,14);
bar(510,280,550,310);
 setcolor(4);
 settextstyle(0,0,2);
 outtextxy(525,285,"A");
 setcolor(1);
 settextstyle(2,0,4);
 outtextxy(513,298,"Left");

bar(550,280,590,310);
 setcolor(4);
 settextstyle(0,0,2);
 outtextxy(565,285,"D");
 setcolor(1);
 settextstyle(2,0,4);
 outtextxy(560,298,"Right");

setcolor(4);
rectangle(510,280,550,310);
rectangle(550,280,590,310);
             //------------------------------

settextstyle(7,0,3);
```

```
/*----------------- yes -------------*/
if(temp2_ex==1)
{setcolor(1);
 setfillstyle(1,1);
 bar(200,215,270,250);                  // yes
 rectangle(200,215,270,250);
 setcolor(WHITE);
 outtextxy(215,217,"Yes");
}
else
{
 setcolor(1);
 setfillstyle(1,WHITE);
 bar(200,215,270,250);                  // yes
 rectangle(200,215,270,250);
 outtextxy(215,217,"Yes");
}


/*----------------- no ------------------------------*/
if(temp2_ex==2)
{
 setfillstyle(1,1);
 bar(350,215,420,250);      // no
 rectangle(350,215,420,250);
 setcolor(WHITE);
 outtextxy(370,217,"No");
}
else
{
 setfillstyle(1,WHITE);
 bar(350,215,420,250);      // no
 setcolor(1);
 rectangle(350,215,420,250);
 outtextxy(370,217,"No");
}
/*---------- button end -----------*/

temp1_ex=getche();

   // active button sound
            sound(40);
            delay(100);
            nosound();

 if((temp1_ex=='a' || temp1_ex=='A') && temp2_ex!=1)
 {temp2_ex--; }
 if((temp1_ex=='d' || temp1_ex=='D') && temp2_ex!=2)
 {temp2_ex++; }
 if(temp1_ex==13)
 {opt_ex=temp2_ex; }
```

```
 }// end of while

            // sound for click
              sound(900);
              delay(150);
              nosound();
 return(opt_ex);
}

/*-----------------------------------------------------------------*/
void tutorial()
{
              setbkcolor(2);

              setfillstyle(1,0);        // logo bg
              bar(0,0,640,60);
              setfillstyle(1,YELLOW);
              bar(0,56,640,60);

              settextstyle(4,0,5);
              setcolor(YELLOW);
              outtextxy(121,1," BattleShip");
              setcolor(4);
              outtextxy(120,0," BattleShip");

              settextstyle(7,0,5);
              setcolor(WHITE);
              outtextxy(350,0,"Tutorial");


              setfillstyle(1,14);        // board border
              bar(0,60,340,338);

              setfillstyle(1,9);        // board
              bar(30,85,330,335);
              setcolor(WHITE);              // grid border
              rectangle(30,85,330,335);

              for(int a=30;a<330;a=a+30)   //grid
              {
               for(int b=85;b<335;b=b+25)
               {
                setcolor(WHITE);
                rectangle(a,b,a+30,b+25);
               }
              }

              // grid identity

              setcolor(4);
              settextstyle(0,0,1);
                outtextxy(40,75,"0");
```

```
 outtextxy(70,75,"1");
 outtextxy(100,75,"2");
 outtextxy(130,75,"3");
 outtextxy(160,75,"4");
 outtextxy(190,75,"5");
 outtextxy(220,75,"6");
 outtextxy(250,75,"7");
 outtextxy(280,75,"8");
 outtextxy(310,75,"9");

 outtextxy(17,94,"0");
 outtextxy(17,119,"1");
 outtextxy(17,144,"2");
 outtextxy(17,169,"3");
 outtextxy(17,194,"4");
 outtextxy(17,219,"5");
 outtextxy(17,244,"6");
 outtextxy(17,269,"7");
 outtextxy(17,294,"8");
 outtextxy(17,319,"9");



setfillstyle(1,14);    // key box
bar(0,338,640,350);


setfillstyle(1,7);    // objective box
bar(340,60,640,337);
setfillstyle(1,1);
bar(340,60,640,75);
setcolor(WHITE);     // box Border
rectangle(340,60,639,337);

setcolor(WHITE);
settextstyle(0,0,1);
outtextxy(345,65,"Game Tutorial ...");

setcolor(1);
outtextxy(355,85,"The game consist of 10x10 Grid");
outtextxy(355,95,"containing 5 hidden enemy ships as");
outtextxy(355,105,"shown : ");
setcolor(6);
outtextxy(355,115,"    Aircraft Carrier - 5 shots ");
outtextxy(355,125,"    BattleShip       - 4 shots ");
outtextxy(355,135,"    Destroyer        - 3 shots ");
outtextxy(355,145,"    Submarine         - 3 shots ");
outtextxy(355,155,"    Patrol Boat      - 2 shots ");


setcolor(WHITE);
outtextxy(356,201,"WARNING :");
```

```
setcolor(4);
outtextxy(355,200,"WARNING :");
outtextxy(355,210,"* Do not press any key unnecessarily");
outtextxy(355,220,"* Enter the coordinate only if");
outtextxy(355,230,"  prompt.");
outtextxy(355,240,"* Enter the coordinate only once.");
outtextxy(355,250,"  Do not use BACKSPACE to delete");
outtextxy(355,260,"  your entry.In such a case enter");
outtextxy(355,270,"  any alphabet and then re-enter");
outtextxy(355,280,"  the coordinates.");


settextstyle(6,0,1);
setcolor(1);
outtextxy(360,300,"Press any key to play The Game !");

// To exit
settextstyle(2,0,4);
setcolor(4);
outtextxy(33,337," THIS is a Dummy View of the game ! To Play the game press any key !");


// 44541020301617184252627248586878888
  int dummy_cod[34]={4,4,5,4,1,0,2,0,3,0,1,6,1,7,1,8,4,2,5,2,6,2,7,2,4,8,5,8,6,8,7,8,8,8};
  char D_temp_x,D_temp_y;
  int c,d;

  for(i=0;i<=34;i=i+2)
  {
        D_temp_x=dummy_cod[i];
        D_temp_y=dummy_cod[i+1];
         c=30;
         d=85;

         c=c+D_temp_x*30;
         d=d+D_temp_y*25;

         setfillstyle(1,4);
         bar(c,d,c+30,d+25);
         setcolor(WHITE);
         rectangle(c,d,c+30,d+25);
  }

  setfillstyle(1,7);
  setcolor(7);
  fillellipse(105,97,35,7);    // 3-1
  fillellipse(210,147,45,7);   // 4
  fillellipse(225,297,65,7);   // 5
  fillellipse(180,197,25,7);   // 2
  fillellipse(75,272,7,30);    // 3-2

getch();
```

```
            cleardevice();
            setbkcolor(BLACK);
}
/*--------------------------------------------------------------------*/

void help()
{
          /*------------------ intructions ----------------*/
 cleardevice();
 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);

 setfillstyle(1,4);           // seperate bar
 bar(0,125,640,130);
 trademark(1);

 settextstyle(7,0,3);
 outtextxy(185,135,"Help and Instructions");

 settextstyle(2,0,5);
 setcolor(4);
 outtextxy(50,180,"The Game include a 10x10 Grid/matrix coordinate system. Your objective");
 outtextxy(50,195,"is to locate five hidden enemy ships and destroy them in specified ammos");
 outtextxy(50,210,"according to the level. You require 17 shots to destroy all ships.");
 outtextxy(50,225,"For every hit,the respective block becomes red and white for a false shot.");
 outtextxy(50,240,"You need to finish your game before you are out of your ammo.");
 outtextxy(50,255,"   Aircraft Carrier-5 shots , Battleship-4 shots , Destroyer-3 shots");
 outtextxy(50,270,"   SubMarine-3 Shots and Patrol Boat-2 Shots");


 settextstyle(8,0,1);
 setcolor(5);
 outtextxy(140,300,"Press Enter to see sample Screenshot... ");
 getch();
 tutorial();

          /*------------------ trouble shootings ----------------*/
 cleardevice();
 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);
```

```
setfillstyle(1,4);          // seperate bar
bar(0,125,640,130);
trademark(1);

settextstyle(7,0,3);
outtextxy(205,135,"Trouble Shootings");

settextstyle(2,0,5);
setcolor(4);
outtextxy(50,180,"This game requires number of graphic driver and font files.");
outtextxy(50,195,"If these files are missing then the game would not run or may ");
outtextxy(50,210,"start malfunctioning.To avoid this problem make sure that the following ");
outtextxy(50,225,"file on next page are present in the folder : ");
outtextxy(50,255,"If Any problem still exist restart the game again.");

settextstyle(8,0,1);
setcolor(5);
outtextxy(180,300,"Press Enter to View Files... ");
getch();
cleardevice();
setfillstyle(1,1);   // logo BG
bar(0,0,640,127);
logo(160,0);

setcolor(4);

setfillstyle(1,7);       // BG
bar(0,127,640,350);

setfillstyle(1,4);          // seperate bar
bar(0,125,640,130);
trademark(1);

settextstyle(2,0,5);
setcolor(1);
line(110,152,560,152);
line(110,165,560,165);
outtextxy(120,150,"File Name                  File Information");
setcolor(6);
settextstyle(2,0,4);
outtextxy(120,170,"EGAVGA.BGI                    EGA Graphic driver File");
outtextxy(120,190,"TRIP.CHR                   Triplex Font File");
outtextxy(120,200,"LITT.CHR                  Small Font File");
outtextxy(120,210,"SANS.CHR                    Sans Serif Font File");
outtextxy(120,220,"GOTH.CHR                    Gothic Font File");
outtextxy(120,230,"SCRI.CHR                   Script Font File");
outtextxy(120,240,"SIMP.CHR                    Simplex Font File");
outtextxy(120,250,"TSCR.CHR                    Triplex Script Font File");
outtextxy(120,260,"LCOM.CHR                    Complex Font File");
outtextxy(120,270,"EURO.CHR                    European Font File");
outtextxy(120,280,"BOLD.CHR                    Bold Font File");
```

```
setcolor(1);
line(110,300,560,300);

getch();

            /*------------------ About ----------------*/
cleardevice();
setfillstyle(1,1);   // logo BG
bar(0,0,640,127);
logo(160,0);

setcolor(4);

setfillstyle(1,7);        // BG
bar(0,127,640,350);

setfillstyle(1,4);           // seperate bar
bar(0,125,640,130);

settextstyle(1,0,3);
setcolor(1);
outtextxy(225,135,"About the Game");

settextstyle(4,0,4);
setcolor(4);
outtextxy(140,170,"BattleShip - ");
settextstyle(8,0,2);
outtextxy(320,175,"Pacific Warriors");

setcolor(0);
settextstyle(2,0,5);
outtextxy(200,220,"Programmed By : Gagandeep Singh");
outtextxy(195,235,"Compiler : Turbo C++ , Version 3.0");
outtextxy(200,265,"  2008-2009 Gagandeep Creations");


settextstyle(8,0,1);
setcolor(5);
outtextxy(140,300,"Press Enter to return to the Menu !");

setfillstyle(1,1);
bar(0,340,650,350);
setfillstyle(1,4);
bar(0,340,650,342);
getch();

}
/*-------------------------------------------------------------*/
void box()
{

  setfillstyle(SOLID_FILL, 7 );  //Main Box
```

```
   bar(190,125,430,205);

   setfillstyle(SOLID_FILL, 1 );   //Title bar
   bar(190,125,430,139);

   setcolor(WHITE);
   rectangle(190,125,430,205);     //Boder

   setfillstyle(SOLID_FILL, WHITE );   // Exit Box
   bar(414,128,427,137);
   setcolor(8);
   settextstyle(0,0,1);
   outtextxy(418,128, "x");
}
/*-------------------------------------------------------------*/
void error_box()
{
      cleardevice();
      setbkcolor(4);
      box();
      setcolor(WHITE);
      outtextxy(200,129, "Game Error!");
      setcolor(EGA_RED);
      setcolor(1);
      outtextxy(200,155, "File missing or corrupted.");
      outtextxy(200,165, "Please reinstall the game!");
      setcolor(4);
      outtextxy(200,185, "   Press any key to exit");
}
/*-----------------------------------------------------------------*/

void pass_enter()
{
   setfillstyle(SOLID_FILL, 7 );  //Main Box
   bar(140,70,480,260);

   setfillstyle(SOLID_FILL, 1 );   //Title bar
   bar(140,70,480,84);

   setcolor(WHITE);
   rectangle(140,70,480,260);       //Boder

   setfillstyle(SOLID_FILL, WHITE );   // Exit Box
   bar(463,73,476,82);
   setcolor(8);
   outtextxy(467,74, "x");

   /*-------------------------------------------------------------*/

    setcolor(WHITE);
    outtextxy(145,74, "Game Key Verification...");
```

```
    setcolor(4);
    outtextxy(150,94, "This Game is not activated on this");
    outtextxy(150,106, "computer.To activate this game , Please");
    outtextxy(150,118, "enter a  8-digit key.");

    outtextxy(150,142, "Warning : ");
    outtextxy(150,154, "The game will not run until you enter");
    outtextxy(150,166, "the correct serial number.");

    setfillstyle(0,0);            // pass input box
    bar(180,220,440,240);
    setcolor(WHITE);
    rectangle(180,220,440,240);
}
/*-------------------------------------------------------------*/
void loading_key()
{
  box();
  setcolor(EGA_WHITE);
  outtextxy(200,129, "Registering Key...");
  setcolor(EGA_RED);
  outtextxy(255,145, "Key Accepted!");
  setcolor(1);
  outtextxy(270,165, "Saving...");

  setcolor(4);
  rectangle(220,179,400,195);

  setfillstyle(SOLID_FILL, GREEN );
  for(int i=224;i<=396;i=i+4)
  {
   delay(100);
   bar(223,182,i,192);
  }
}
/*-------------------------------------------------------------*/

void loading_main_game(int ch,int x)
{
 int l;
                                  // background
   setfillstyle(1,YELLOW);
   bar(10,10,630,335);         //30,30,610,325
   setcolor(6);
   rectangle(10,10,630,335);

                                  // holes
   for(l=25;l<625;l+=15)
   {setfillstyle(1,0);
    fillellipse(l,17,3,3);    //fillellipse(l,37,3,3);
   }
```

```
                        // rings
for(l=25;l<625;l+=15)
{setcolor(0);
  ellipse(l+1,10,0,260,4,7);   //ellipse(l+1,30,0,260,4,7);
  setcolor(WHITE);
  ellipse(l,10,0,260,4,7);     //ellipse(l,30,0,260,4,7);
}

                       //lines
setcolor(4);
line(11,53,629,53);   // 68
line(11,55,629,55);   // 70

setcolor(7);
for(l=70;l<300;l=l+17)
{line(11,l,629,l); }

setcolor(0);
settextstyle(4,0,3);
outtextxy(180,20,"BattleShip - Pacific Warriors");
//------------------------------------------------------------

switch(ch)
{
 case 1 :    // Mission
             setcolor(0);
             gotoxy(35,5); cout<<" Level : "<<x<<" ";
             settextstyle(8,0,1);
             outtextxy(250,63,"\"Mission Mode\"");

             setcolor(4);
             settextstyle(8,0,1);
             outtextxy(30,97,"Objectives : ");
              setcolor(0);
              settextstyle(5,0,1);
              outtextxy(170,97,"Find 5 enemy ship in 10x10 matrix and destroy");
              outtextxy(170,114,"them before you run out of ammos .");
              outtextxy(170,131,"You will require 17 shot to complete your objective !");

             setcolor(4);
             settextstyle(8,0,1);
             outtextxy(30,165,"Total Ammos : ");
              setcolor(0);
              settextstyle(5,0,1);
              if(x>=0 && x<=5)
               {outtextxy(190,165,"50 Missiles ."); }
              else
              if(x>=6 && x<=10)
               {outtextxy(190,165,"45 Missiles ."); }
              else
              if(x>=11 && x<=15)
```

```
 {outtextxy(190,165,"40 Missiles ."); }
 else
 if(x>=16 && x<=20)
 {outtextxy(190,165,"34 Missiles ."); }

setcolor(4);
settextstyle(8,0,1);
outtextxy(30,199,"Enemy Ships :");
 setcolor(0);
 settextstyle(5,0,1);
 outtextxy(180,199,"Aircraft Carrier - USS Nimitz CUN-68 |  5 shots");
 outtextxy(180,216,"Battleship - USS Lawa BB-61       |  4 shots");
 outtextxy(180,233,"Destroyer - US Fletcher DD-445     |  3 shots");
 outtextxy(180,250,"Submarine - USS Ohio ISSBN-726   |  3 shots");
 outtextxy(180,267,"Patrol Boat - OSS Seahawk PT-813  |  2 shots");
break;

case 2 :   // Arcade
setcolor(0);
gotoxy(35,5); cout<<" Level : "<<x<<" ";
settextstyle(8,0,1);
outtextxy(250,63,"\"Arcade Mode\"");

setcolor(4);
settextstyle(8,0,1);
outtextxy(30,97,"Objectives : ");
 setcolor(0);
 settextstyle(5,0,1);
 outtextxy(170,97,"Find 5 enemy ship in selected 10x10 matrix and");
 outtextxy(170,114,"destroy them before you run out of ammos .");
 outtextxy(170,131,"You will require 17 shot to complete your objective !");

setcolor(4);
settextstyle(8,0,1);
outtextxy(30,165,"Total Ammos : ");
 setcolor(0);
 settextstyle(5,0,1);
 if(x>=0 && x<=5)
 {outtextxy(190,165,"50 Missiles ."); }
 else
 if(x>=6 && x<=10)
 {outtextxy(190,165,"45 Missiles ."); }
 else
 if(x>=11 && x<=15)
 {outtextxy(190,165,"40 Missiles ."); }
 else
 if(x>=16 && x<=20)
 {outtextxy(190,165,"34 Missiles ."); }

setcolor(4);
settextstyle(8,0,1);
outtextxy(30,199,"Enemy Ships :");
```

```
 setcolor(0);
 settextstyle(5,0,1);
 outtextxy(180,199,"Aircraft Carrier - USS Nimitz CUN-68 |  5 shots");
 outtextxy(180,216,"Battleship - USS Lawa BB-61       |  4 shots");
 outtextxy(180,233,"Destroyer - US Fletcher DD-445      |  3 shots");
 outtextxy(180,250,"Submarine - USS Ohio ISSBN-726   |  3 shots");
 outtextxy(180,267,"Patrol Boat - OSS Seahawk PT-813  |  2 shots");
 break;

case 3 :    // Training
 setcolor(0);
 settextstyle(8,0,1);
 outtextxy(200,63,"\"Training\\Random Mode\"");

 setcolor(4);
 settextstyle(8,0,1);
 outtextxy(30,97,"Objectives : ");
 setcolor(0);
 settextstyle(5,0,1);
 outtextxy(170,97,"Find 5 enemy ship in randomly selected 10x10");
 outtextxy(170,114,"matrix and distroy them before you run out of ammos .");
 outtextxy(170,131,"You will require 17 shot to complete your objective !");

 setcolor(4);
 settextstyle(8,0,1);
 outtextxy(30,165,"Total Ammos : ");
 setcolor(0);
 settextstyle(5,0,1);
 if(x>=0 && x<=5)
  {outtextxy(190,165,"50 Missiles ."); }
 else
 if(x>=6 && x<=10)
  {outtextxy(190,165,"45 Missiles ."); }
 else
 if(x>=11 && x<=15)
  {outtextxy(190,165,"40 Missiles ."); }
 else
 if(x>=16 && x<=20)
  {outtextxy(190,165,"34 Missiles ."); }


 setcolor(4);
 settextstyle(8,0,1);
 outtextxy(30,199,"Enemy Ships :");
 setcolor(0);
 settextstyle(5,0,1);
 outtextxy(180,199,"Aircraft Carrier - USS Nimitz CUN-68 |  5 shots");
 outtextxy(180,216,"Battleship - USS Lawa BB-61       |  4 shots");
 outtextxy(180,233,"Destroyer - US Fletcher DD-445      |  3 shots");
 outtextxy(180,250,"Submarine - USS Ohio ISSBN-726   |  3 shots");
 outtextxy(180,267,"Patrol Boat - OSS Seahawk PT-813  |  2 shots");
 break;
```

```
   default : cleardevice();
                break;
  }

  //--------------------

  delay(4000);
  {              // loading bar
    setcolor(4);
    rectangle(220,300,420,315);
    delay(1000);
    for(l=222;l<=418;l=l+4)
    {setfillstyle(1,2);
          bar(222,302,l,313);
          delay(250);
    }
    settextstyle(0,0,1);
    setcolor(YELLOW);
    outtextxy(260,304,"Press any key !");
    getch();
  }

}
/*-------------------------------------------------------------*/

int mode_select()
{
 cleardevice();

 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);

 settextstyle(7,0,3);     // game mode
 setcolor(1);
 outtextxy(210,140,"Select your Mode");

 setfillstyle(1,4);           // seperate bar
 bar(0,125,640,130);
 trademark(1);

               /*------------- select ----------------*/
 char temp1;
 int opt,temp2=2;

 while(temp1!=13)    // enter
```

```
{
 button_mission(temp2);
 button_arcade(temp2);
 button_training(temp2);
 button_back(temp2);

                        // ---------- key display
 setfillstyle(1,14);
 settextstyle(0,0,2);
 bar(550,240,590,270);
  setcolor(4);
  settextstyle(0,0,2);
  outtextxy(560,250,"W");
  setcolor(1);
  settextstyle(2,0,4);
  outtextxy(578,241,"Up");

 bar(550,270,590,300);
  setcolor(4);
  settextstyle(0,0,2);
  outtextxy(560,275,"S");
  setcolor(1);
  settextstyle(2,0,4);
  outtextxy(565,288,"Down");

 setcolor(4);
 rectangle(550,240,590,270);
 rectangle(550,270,590,300);
                        //-------------------------

 temp1=getche();

    // active button sound
              sound(40);
              delay(100);
              nosound();

 if((temp1=='w' || temp1=='W') && temp2!=2)
 {temp2--; }
 if((temp1=='s' || temp1=='S') && temp2!=5)
 {temp2++; }
 if(temp1==13)
 {opt=temp2; }

 // ersae discription text
 setfillstyle(1,7);
 bar(0,180,149,300);

}

           // sound for click
              sound(900);
```

```
                delay(150);
                nosound();

 return(opt-1);

}

/*-----------------------------------------------------------------*/

int level_file_chk(int lno)
{
 fstream f_l;
 char l_path[19]={"levels\\level00.dat"};
 int flag,l_temp;
 int rt;
   if(lno>=21)
   {lno=20; }

 if(lno==10)                              // level 10
  {l_path[12]=49; }
 else
 if(lno>10 && lno<20)                          // level 11 - 19
  {l_path[12]=49;
   l_temp=lno;
   l_temp=l_temp-10; l_path[13]=l_temp+48;
  }
 else
 if(lno==20)                              // level 20
  {l_path[12]=50; }
 else                                    // level 1 - 9
  {l_path[13]=lno+48; }

 f_l.open(l_path,ios::in);
  if(f_l==0)
   {rt=0; }
  else
   {rt=1; }

 return(rt);

}
/*-----------------------------------------------------------------*/
int level_randomize()
{
                fstream f_l;
                char l_path[19]={"levels\\level00.dat"};
                int lno,flag,l_temp,l_err=0,rp=0;

               do
               {
                randomize();
                lno=random(21);
```

```
 if(lno==10)                          // level 10
  {l_path[12]=49; }
 else
 if(lno>10 && lno<20)                      // level 11 - 19
  {l_path[12]=49;
        l_temp=lno;
        l_temp=l_temp-10; l_path[13]=l_temp+48;
  }
 else

 if(lno==20)                          // level 20
  {l_path[12]=50; }
 else                                 // level 1 - 9
  {l_path[13]=lno+48; }

 f_l.open(l_path,ios::in);                  // check file
  if(f_l==0)
   {flag=0;
         if(lno>10)          // if level>10 and file does not exist random again
         {rp=1;
           randomize();
           lno=random(11);
         }
   }
  else
   {flag=1; l_err=1; }
  f_l.close();


   if( (lno>0 && lno<=10) && (flag==0) )
    {l_err=0; }

}while(flag==0 && lno>10);
    // ----------------- end searching --------------------

 //-------------------- read file --------------------
 // (1) Reprocess file

 if(rp==1)
 {
  l_path[12]=48; l_path[13]=48;
  if(lno==10)                  // level 10
   {l_path[12]=49; }
  else                         // level 1 - 9
   {l_path[13]=lno+48; }
 }

 //(2) Verify file
 f_l.open(l_path,ios::in);          // check file
 if(f_l==0)
  {l_err=0; }
```

```
                else
                 {l_err=1; }
                f_l.close();

               if(l_err==0)                    // prompt error and exit
                {error_box();
                  game_sounds(-1);
                  return(lno);
                }

      return(lno);
}

/*-----------------------------------------------------------------*/
void board_comment(int com)
{
 int temp;

 switch(com)
 {
  case -3 : setfillstyle(1,2);
                bar(350,292,630,327);         // bg
                gotoxy(46,22); cout<<"DO NOT ENTER 'BACKSAPCE' KEY !";
                game_sounds(0);
                break;
  case -2 : setfillstyle(1,2);
                bar(350,292,630,327);         // bg
                game_sounds(0);
                gotoxy(46,22); cout<<"Do You Want quit ? (Y/N)";
                int e_temp;
                e_temp=getch();
                if(e_temp=='y'||e_temp=='Y')
                 { exit(0);    }
                else
                 {
                  bar(350,292,630,327);         // bg
                  gotoxy(46,22); cout<<"Please Enter the coordinates!";
                 }
                break;

  case -1 : setfillstyle(1,2);
                bar(350,292,630,327);         // bg
                gotoxy(46,22); cout<<"Wrong Coordinates! Enter Again.";
                game_sounds(0);
                break;

  case 0 :  setfillstyle(1,2);
                bar(350,292,630,327);         // bg
                randomize();
                temp=random(5);
                switch(temp)
                {
```

```
            case 0 : gotoxy(46,22); cout<<"You are wasting your ammo!"; break;
            case 1 : gotoxy(46,22); cout<<"You missed it !"; break;
            case 2 : gotoxy(46,22); cout<<"Better luck next time!"; break;
            case 3 : gotoxy(46,22); cout<<"Just don't try anything!"; break;
            case 4 : gotoxy(46,22); cout<<"Target still exist!"; break;
            }
            break;

case 1 : setfillstyle(1,2);
            bar(350,292,630,327);        // bg
            randomize();
            temp=random(4);
            switch(temp)
            {
            case 0 : gotoxy(46,22); cout<<"Well done !"; break;
            case 1 : gotoxy(46,22); cout<<"You spotted a ship !"; break;
            case 2 : gotoxy(46,22); cout<<"Target Identified !"; break;
            case 3 : gotoxy(46,22); cout<<"That was a hit !"; break;
            }
            game_sounds(1);
            break;

case 5 : setfillstyle(1,2);
            bar(350,292,630,327);        // bg
            gotoxy(46,22); cout<<"Please wait!";
            gotoxy(46,23); cout<<"Processing.....";
            break;

case 6 : setfillstyle(1,2);
            bar(350,292,630,327);        // bg
            gotoxy(46,22); cout<<"Enter the coordinates!";
            break;

case 7 : setfillstyle(1,2);
            bar(350,292,630,327);        // bg
            gotoxy(46,22); cout<<"Repeated Coordinates !";
            gotoxy(46,23); cout<<"Please enter New coordinates.";
            game_sounds(0);
            break;

}// end of switch com

}
/*-------------------------------------------------------------*/

void game_board(int lg)
{
            setbkcolor(2);

            setfillstyle(1,4);       // logo bg
            bar(0,0,640,60);
            setfillstyle(1,4);
```

```
bar(0,56,640,60);

settextstyle(4,0,5);
setcolor(WHITE);
outtextxy(60,0,"BattleShip - Pacific Warriors");


setfillstyle(1,14);        // board border
bar(0,60,340,338);

setfillstyle(1,9);        // board
bar(30,85,330,335);
setcolor(WHITE);               // grid border
rectangle(30,85,330,335);

for(int a=30;a<330;a=a+30)   //grid
{
 for(int b=85;b<335;b=b+25)
 {
  setcolor(WHITE);
  rectangle(a,b,a+30,b+25);
 }
}

// grid identity

setcolor(4);
settextstyle(0,0,1);
  outtextxy(40,75,"0");
  outtextxy(70,75,"1");
  outtextxy(100,75,"2");
  outtextxy(130,75,"3");
  outtextxy(160,75,"4");
  outtextxy(190,75,"5");
  outtextxy(220,75,"6");
  outtextxy(250,75,"7");
  outtextxy(280,75,"8");
  outtextxy(310,75,"9");

  outtextxy(17,94,"0");
  outtextxy(17,119,"1");
  outtextxy(17,144,"2");
  outtextxy(17,169,"3");
  outtextxy(17,194,"4");
  outtextxy(17,219,"5");
  outtextxy(17,244,"6");
  outtextxy(17,269,"7");
  outtextxy(17,294,"8");
  outtextxy(17,319,"9");
```

```
setfillstyle(1,14);    // key box
bar(0,338,640,350);


setfillstyle(1,7);    // objective box
bar(340,60,640,129);
setfillstyle(1,1);
bar(340,60,640,75);
setcolor(WHITE);
settextstyle(0,0,1);
outtextxy(345,65,"Objective");
setcolor(4);
outtextxy(355,85,"Locate 5 ships in 10x10 matrix and");
outtextxy(355,95,"distroy them before you run out of");
outtextxy(355,105,"ammo's. You will require 17 shots");
outtextxy(355,115,"to complete your mission.");


setfillstyle(1,7);    // entry & Ammo box
bar(340,129,640,198);
setfillstyle(1,1);
bar(340,129,640,144);
settextstyle(0,0,1);
setcolor(WHITE);
outtextxy(345,134,"Enter Coordinates");
setcolor(1);
outtextxy(355,159,"X-Axis : ");
outtextxy(355,179,"Y-Axis : ");

setfillstyle(1,0);
bar(425,150,465,170);
setcolor(WHITE);
rectangle(425,150,465,170);

bar(425,173,465,193);
rectangle(425,173,465,193);



outtextxy(516,134,"Ammo");     // ammo box
setcolor(1);
outtextxy(540,179,"Ammo Left");

setcolor(WHITE);
setfillstyle(1,0);
bar(540,150,570,170);
rectangle(540,150,570,170);
gotoxy(69,12); cout<<"00";

settextstyle(0,0,2);
setcolor(4);
if(lg>=0 && lg<=5)
```

```
 {outtextxy(575,153,"/50"); }
else
if(lg>=6 && lg<=10)
 {outtextxy(575,153,"/45"); }
else
if(lg>=11 && lg<=15)
 {outtextxy(575,153,"/40"); }
else
if(lg>=16 && lg<=20)
 {outtextxy(575,153,"/34"); }



setfillstyle(1,7);    // status box
bar(340,198,640,267);
setfillstyle(1,1);
bar(340,198,640,213);
settextstyle(0,0,1);
setcolor(WHITE);
outtextxy(345,203,"Game Status");

setcolor(1);
settextstyle(0,0,1);
outtextxy(380,220,"No. of Hits");
setfillstyle(1,0);
bar(370,235,410,255);
setcolor(WHITE);
rectangle(370,235,410,255);
settextstyle(0,0,2);
setcolor(4);
outtextxy(417,238,"/17");

setcolor(1);
settextstyle(0,0,1);
outtextxy(518,220,"Game Progress");
setfillstyle(1,7);
bar(510,238,631,251);
setcolor(WHITE);
rectangle(510,238,631,251);

settextstyle(0,0,1);
setcolor(1);
switch(lg)
{
 case 0 : outtextxy(540,255,"Level 0"); break;
 case 1 : outtextxy(540,255,"Level 1"); break;
 case 2 : outtextxy(540,255,"Level 2"); break;
 case 3 : outtextxy(540,255,"Level 3"); break;
 case 4 : outtextxy(540,255,"Level 4"); break;
 case 5 : outtextxy(540,255,"Level 5"); break;
 case 6 : outtextxy(540,255,"Level 6"); break;
```

```
                case 7 : outtextxy(540,255,"Level 7"); break;
                case 8 : outtextxy(540,255,"Level 8"); break;
                case 9 : outtextxy(540,255,"Level 9"); break;
                case 10 : outtextxy(540,255,"Level 10"); break;
                case 11 : outtextxy(540,255,"Level 11"); break;
                case 12 : outtextxy(540,255,"Level 12"); break;
                case 13 : outtextxy(540,255,"Level 13"); break;
                case 14 : outtextxy(540,255,"Level 14"); break;
                case 15 : outtextxy(540,255,"Level 15"); break;
                case 16 : outtextxy(540,255,"Level 16"); break;
                case 17 : outtextxy(540,255,"Level 17"); break;
                case 18 : outtextxy(540,255,"Level 18"); break;
                case 19 : outtextxy(540,255,"Level 19"); break;
                case 20 : outtextxy(540,255,"Level 20"); break;
               }


               setfillstyle(1,7);    // comment box
               bar(340,267,640,337);
               setfillstyle(1,1);
               bar(340,267,640,282);
               settextstyle(0,0,1);
               setcolor(WHITE);
               outtextxy(345,272,"Comments");
               setfillstyle(1,0);

               // To exit
               settextstyle(2,0,4);
               setcolor(4);
               outtextxy(30,337,"To Exit enter 'e' or 'E' or 'Esc'in x-axis coordinate box. ( Note : This will terminate game. )");


               // white border of boxes
               setcolor(WHITE);
               rectangle(340,60,639,129);   // obj
               rectangle(340,129,639,198);  //entry
               line(510,129,510,198);
               line(511,129,511,198);
               rectangle(340,198,639,267);  //status
               rectangle(340,267,639,337);

               //-------------------- default display ends

}
/*-------------------------------------------------------------------------*/

int play_mission()
{
 fstream save;
 int L,l_chk,x=0;
                        //check for saved game
 save.open("C:\\Windows\\btl_save.dat",ios::in);
```

```
 if(save==0)
 {save.open("C:\\Windows\\btl_save.dat",ios::out);
   save<<x;
 }
save.close();

save.open("C:\\Windows\\btl_save.dat",ios::in);
save>>L;
save.close();

                                  // check level file

if(L==0)
 {l_chk=level_file_chk(L+1); }
else
 {l_chk=level_file_chk(L); }

if(l_chk==0)
 {error_box();  game_sounds(-1); getch(); exit(0); }

if(L==0)
 {return(L+1); }
else
 {return(L); }

}

/*---------------------------------------------------------------------------*/
                                  //Buttons for arcade
 void button_level1(int x,int y)
 {
  settextstyle(3,0,1);

  if(y<1)
  {
   setcolor(8);
   outtextxy(50,160,"Level 1");
  }
  else
  if(x==1)
  {setcolor(4);
    outtextxy(50,160,"Level 1");
    setcolor(1);
    rectangle(40,160,130,160+25);
  }
  else
  {setcolor(4);
    outtextxy(50,160,"Level 1");
    setcolor(7);
    rectangle(40,160,130,160+25);
  }
 }
```

```
//-------------------------------------------

void button_level2(int x,int y)
{
 settextstyle(3,0,1);

 if(y<2)
 {
  setcolor(8);
  outtextxy(50,190,"Level 2");
 }
 else
 if(x==2)
 {setcolor(4);
   outtextxy(50,190,"Level 2");
   setcolor(1);
   rectangle(40,190,130,190+25);
 }
 else
 {setcolor(4);
   outtextxy(50,190,"Level 2");
   setcolor(7);
   rectangle(40,190,130,190+25);
 }
}
//-------------------------------------------

void button_level3(int x,int y)
{
 settextstyle(3,0,1);

 if(y<3)
 {
  setcolor(8);
  outtextxy(50,220,"Level 3");
 }
 else
 if(x==3)
 {setcolor(4);
   outtextxy(50,220,"Level 3");
   setcolor(1);
   rectangle(40,220,130,220+25);
 }
 else
 {setcolor(4);
   outtextxy(50,220,"Level 3");
   setcolor(7);
   rectangle(40,220,130,220+25);
 }
}
//-------------------------------------------
```

```
void button_level4(int x,int y)
{
 settextstyle(3,0,1);

 if(y<4)
 {
  setcolor(8);
  outtextxy(50,250,"Level 4");
 }
 else
 if(x==4)
 {setcolor(4);
   outtextxy(50,250,"Level 4");
   setcolor(1);
   rectangle(40,250,130,250+25);
 }
 else
 {setcolor(4);
   outtextxy(50,250,"Level 4");
   setcolor(7);
   rectangle(40,250,130,250+25);
 }
}
//-----------------------------------------

void button_level5(int x,int y)
{
 settextstyle(3,0,1);

 if(y<5)
 {
  setcolor(8);
  outtextxy(50,280,"Level 5");
 }
 else
 if(x==5)
 {setcolor(4);
   outtextxy(50,280,"Level 5");
   setcolor(1);
   rectangle(40,280,130,280+25);
 }
 else
 {setcolor(4);
   outtextxy(50,280,"Level 5");
   setcolor(7);
   rectangle(40,280,130,280+25);
 }
}
//-----------------------------------------
//-----------------------------------------
void button_level6(int x,int y)
{
```

```
 settextstyle(3,0,1);

 if(y<6)
 {
  setcolor(8);
  outtextxy(200,160,"Level 6");
 }
 else
 if(x==6)
 {setcolor(4);
   outtextxy(200,160,"Level 6");
   setcolor(1);
   rectangle(190,160,280,160+25);
 }
 else
 {setcolor(4);
   outtextxy(200,160,"Level 6");
   setcolor(7);
   rectangle(190,160,280,160+25);
 }
}
//-----------------------------------------

void button_level7(int x,int y)
{
 settextstyle(3,0,1);

 if(y<7)
 {
  setcolor(8);
  outtextxy(200,190,"Level 7");
 }
 else
 if(x==7)
 {setcolor(4);
   outtextxy(200,190,"Level 7");
   setcolor(1);
   rectangle(190,190,280,190+25);
 }
 else
 {setcolor(4);
   outtextxy(200,190,"Level 7");
   setcolor(7);
   rectangle(190,190,280,190+25);
 }
}
//-----------------------------------------

void button_level8(int x,int y)
{
 settextstyle(3,0,1);
```

```
 if(y<8)
 {
  setcolor(8);
  outtextxy(200,220,"Level 8");
 }
 else
 if(x==8)
 {setcolor(4);
   outtextxy(200,220,"Level 8");
   setcolor(1);
   rectangle(190,220,280,220+25);
 }
 else
 {setcolor(4);
   outtextxy(200,220,"Level 8");
   setcolor(7);
   rectangle(190,220,280,220+25);
 }
}
//-----------------------------------------

void button_level9(int x,int y)
{
 settextstyle(3,0,1);

 if(y<9)
 {
  setcolor(8);
  outtextxy(200,250,"Level 9");
 }
 else
 if(x==9)
 {setcolor(4);
   outtextxy(200,250,"Level 9");
   setcolor(1);
   rectangle(190,250,280,250+25);
 }
 else
 {setcolor(4);
   outtextxy(200,250,"Level 9");
   setcolor(7);
   rectangle(190,250,280,250+25);
 }
}
//-----------------------------------------

void button_level10(int x,int y)
{
 settextstyle(3,0,1);

 if(y<10)
 {
```

```
 setcolor(8);
 outtextxy(200,280,"Level 10");
 }
 else
 if(x==10)
 {setcolor(4);
  outtextxy(200,280,"Level 10");
  setcolor(1);
  rectangle(190,280,280,280+25);
 }
 else
 {setcolor(4);
  outtextxy(200,280,"Level 10");
  setcolor(7);
  rectangle(190,280,280,280+25);
 }
}
//-------------------------------------------

void button_level11(int x,int y)
{
 settextstyle(3,0,1);

 if(y<11)
 {
  setcolor(8);
  outtextxy(350,160,"Level 11");
 }
 else
 if(x==11)
 {setcolor(4);
  outtextxy(350,160,"Level 11");
  setcolor(1);
  rectangle(340,160,450,160+25);
 }
 else
 {setcolor(4);
  outtextxy(350,160,"Level 11");
  setcolor(7);
  rectangle(340,160,450,160+25);
 }
}
//-------------------------------------------

void button_level12(int x,int y)
{
 settextstyle(3,0,1);

 if(y<12)
 {
  setcolor(8);
  outtextxy(350,190,"Level 12");
```

```
 }
 else
 if(x==12)
 {setcolor(4);
  outtextxy(350,190,"Level 12");
  setcolor(1);
  rectangle(340,190,450,190+25);
 }
 else
 {setcolor(4);
  outtextxy(350,190,"Level 12");
  setcolor(7);
  rectangle(340,190,450,190+25);
 }
}
//----------------------------------------

void button_level13(int x,int y)
{
 settextstyle(3,0,1);

 if(y<13)
 {
  setcolor(8);
  outtextxy(350,220,"Level 13");
 }
 else
 if(x==13)
 {setcolor(4);
  outtextxy(350,220,"Level 13");
  setcolor(1);
  rectangle(340,220,450,220+25);
 }
 else
 {setcolor(4);
  outtextxy(350,220,"Level 13");
  setcolor(7);
  rectangle(340,220,450,220+25);
 }
}
//----------------------------------------

void button_level14(int x,int y)
{
 settextstyle(3,0,1);

 if(y<14)
 {
  setcolor(8);
  outtextxy(350,250,"Level 14");
 }
 else
```

```
 if(x==14)
 {setcolor(4);
   outtextxy(350,250,"Level 14");
   setcolor(1);
   rectangle(340,250,450,250+25);
 }
 else
 {setcolor(4);
   outtextxy(350,250,"Level 14");
   setcolor(7);
   rectangle(340,250,450,250+25);
 }
}
//-----------------------------------------

void button_level15(int x,int y)
{
 settextstyle(3,0,1);

 if(y<15)
 {
  setcolor(8);
  outtextxy(350,280,"Level 15");
 }
 else
 if(x==15)
 {setcolor(4);
   outtextxy(350,280,"Level 15");
   setcolor(1);
   rectangle(340,280,450,280+25);
 }
 else
 {setcolor(4);
   outtextxy(350,280,"Level 15");
   setcolor(7);
   rectangle(340,280,450,280+25);
 }
}
//-----------------------------------------
//-----------------------------------------

void button_level16(int x,int y)
{
 settextstyle(3,0,1);

 if(y<16)
 {
  setcolor(8);
  outtextxy(500,160,"Level 16");
 }
 else
 if(x==16)
```

```
 {setcolor(4);
  outtextxy(500,160,"Level 16");
  setcolor(1);
  rectangle(490,160,580,160+25);
 }
 else
 {setcolor(4);
  outtextxy(500,160,"Level 16");
  setcolor(7);
  rectangle(490,160,580,160+25);
 }
}
//-----------------------------------------

void button_level17(int x,int y)
{
 settextstyle(3,0,1);

 if(y<17)
 {
  setcolor(8);
  outtextxy(500,190,"Level 17");
 }
 else
 if(x==17)
 {setcolor(4);
  outtextxy(500,190,"Level 17");
  setcolor(1);
  rectangle(490,190,580,190+25);
 }
 else
 {setcolor(4);
  outtextxy(500,190,"Level 17");
  setcolor(7);
  rectangle(490,190,580,190+25);
 }
}
//-----------------------------------------

void button_level18(int x,int y)
{
 settextstyle(3,0,1);

 if(y<18)
 {
  setcolor(8);
  outtextxy(500,220,"Level 18");
 }
 else
 if(x==18)
 {setcolor(4);
  outtextxy(500,220,"Level 18");
```

```
  setcolor(1);
  rectangle(490,220,580,220+25);
 }
 else
 {setcolor(4);
  outtextxy(500,220,"Level 18");
  setcolor(7);
  rectangle(490,220,580,220+25);
 }
}
//-----------------------------------------

void button_level19(int x,int y)
{
 settextstyle(3,0,1);

 if(y<19)
 {
  setcolor(8);
  outtextxy(500,250,"Level 19");
 }
 else
 if(x==19)
 {setcolor(4);
  outtextxy(500,250,"Level 19");
  setcolor(1);
  rectangle(490,250,580,250+25);
 }
 else
 {setcolor(4);
  outtextxy(500,250,"Level 19");
  setcolor(7);
  rectangle(490,250,580,250+25);
 }
}
//-----------------------------------------

void button_level20(int x,int y)
{
 settextstyle(3,0,1);

 if(y<20)
 {
  setcolor(8);
  outtextxy(500,280,"Level 20");
 }
 else
 if(x==20)
 {setcolor(4);
  outtextxy(500,280,"Level 20");
  setcolor(1);
  rectangle(490,280,580,280+25);
```

```
 }
 else
 {setcolor(4);
  outtextxy(500,280,"Level 20");
  setcolor(7);
  rectangle(490,280,580,280+25);
 }
}
//-----------------------------------------


/*-------------------------------------------------------------------------*/

int play_arcade()
{
 int L;
 L=play_mission();
 if(L==0)
 {L=1; }

 if(L>1)
 {L=L-1; }

 //--------------------------- display ----------------------------

 cleardevice();

 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,7);        // BG
 bar(0,127,640,350);

 settextstyle(7,0,3);     // game mode
 setcolor(1);
 outtextxy(230,130,"Arcade Mode");

 setfillstyle(1,4);           // seperate bar
 bar(0,125,640,130);
 trademark(1);

                /*------------- select -----------------*/


 char temp1;
 int opt,temp2=1;

 while(temp1!=13)    // enter
 {
```

```
 // Display level

    button_level1(temp2,L);
    button_level2(temp2,L);
    button_level3(temp2,L);
    button_level4(temp2,L);
    button_level5(temp2,L);

    button_level6(temp2,L);
    button_level7(temp2,L);
    button_level8(temp2,L);
    button_level9(temp2,L);
    button_level10(temp2,L);

    button_level11(temp2,L);
    button_level12(temp2,L);
    button_level13(temp2,L);
    button_level14(temp2,L);
    button_level15(temp2,L);

    button_level16(temp2,L);
    button_level17(temp2,L);
    button_level18(temp2,L);
    button_level19(temp2,L);
    button_level20(temp2,L);

  temp1=getche();
  if((temp1=='w' || temp1=='W') && temp2!=1)
  {temp2--; }
  if((temp1=='s' || temp1=='S') && temp2!=L)
  {temp2++; }
  if(temp1==13)
  {opt=temp2; }

 }

 return(opt);
}
/*--------------------------------------------------------------------------*/
void saving_game_mission(int l)
{
  cleardevice();
  setbkcolor(BLACK);
  box();
  setcolor(EGA_WHITE);
  outtextxy(200,129, "Saving Game...");
  setcolor(EGA_RED);
  outtextxy(255,145, "Please Wait!");
  setcolor(1);
  outtextxy(270,165, "Saving...");
```

```
 setcolor(4);
 rectangle(220,179,400,195);

 setfillstyle(SOLID_FILL, GREEN );
 for(int i=224;i<=396;i=i+4)
 {
  delay(100);
  bar(223,182,i,192);
 }

 //-----------------------------------------------------

 fstream save;

 save.open("C:\\Windows\\btl_save.dat",ios::out);
 save.seekp(0,ios::beg);
 save<<(l+1);
 save.close();

 //----------done--------------------------------
 cleardevice();
 setbkcolor(BLACK);
 box();
 setcolor(EGA_WHITE);
 outtextxy(200,129, "Saving Game...");
 setcolor(EGA_RED);
 outtextxy(255,145, "Press any Key!");
 setcolor(1);
 outtextxy(270,165, "  Done...");

 setcolor(4);
 rectangle(220,179,400,195);

 setfillstyle(SOLID_FILL, GREEN );
 bar(223,182,396,192);

 getch();

}
/*-------------------------------------------------------------*/
void shots(int x,int y,int z)
{
 int j;

 setcolor(WHITE);

 i=30;
 j=85;

 i=i+x*30;
 j=j+y*25;
```

```
 if(z==0)
  {setfillstyle(1,7); }
 else
  {setfillstyle(1,4); }

 bar(i,j,i+30,j+25);
 rectangle(i,j,i+30,j+25);

}
/*-------------------------------------------------------------*/
void game_over(int w)
{
  switch(w)
  {
  case 0 : box();
              setcolor(EGA_WHITE);
              outtextxy(200,129, "Game Over");

              settextstyle(0,0,2);
              setcolor(EGA_RED);
              outtextxy(235,155, "Game Over");

              settextstyle(0,0,1);
              setcolor(1);
              outtextxy(210,190, "Press any key to continue!");
              game_sounds(-2);
              getch();

              cleardevice();
              setbkcolor(4);
              logo(160,0);
              setfillstyle(1,YELLOW);          // seperate bar
              bar(0,125,640,150);
              settextstyle(0,0,2);
              setcolor(4);
              outtextxy(140,130,"You have Lost the Game");

              settextstyle(8,0,1);
              setcolor(YELLOW);
              outtextxy(140,320,"Press Enter to return to the Menu !");

              break;

  case 1 : box();
              setcolor(EGA_WHITE);
              outtextxy(200,129, "Game Over");

              settextstyle(0,0,2);
              setcolor(EGA_RED);
              outtextxy(235,145, "Objective");
              outtextxy(225,167, "Completed !");
```

```
            settextstyle(0,0,1);
            setcolor(1);
            outtextxy(210,190, "Press any key to continue!");
            game_sounds(2);
            getch();

            cleardevice();
            setbkcolor(1);
            logo(160,0);
            setfillstyle(1,7);          // seperate bar
            bar(0,125,640,150);
            settextstyle(0,0,2);
            setcolor(1);
            outtextxy(140,130,"You have Won this Level");

            settextstyle(8,0,1);
            setcolor(YELLOW);
            outtextxy(140,320,"Press Enter to return to the Menu !");

            break;
 }
}
/*-------------------------------------------------------------*/
void Credits(int d)
{
  cleardevice();
  if(d==1)
  {
  setcolor(WHITE);
  settextstyle(1,0,3);
  outtextxy(160,120,"BattleShip - Pacific Warriors");
  setcolor(4);
  line(150,151,500,151);
  settextstyle(5,0,1);
  setcolor(WHITE);
  delay(1200);
  outtextxy(265,160,"Programmed By :");
  delay(100);
  settextstyle(2,0,7);
  outtextxy(165,190,"G a g a n d e e p  S i n g h");
  delay(100);
  settextstyle(2,0,6);
  outtextxy(310,215,"2008");
  delay(2500);
  cleardevice();
  }

 int pos_y=350;
 while(!kbhit())    // play till no key is pressed
 {
     // Display
  setcolor(WHITE);
```

```
settextstyle(1,0,3);
outtextxy(160,pos_y,"BattleShip - Pacific Warriors");
setcolor(4);
line(150,pos_y+31,500,pos_y+31);

setcolor(WHITE);
settextstyle(8,0,2);
outtextxy(290,pos_y+31,"Credits");
setcolor(4);
line(285,pos_y+60,375,pos_y+60);

// preamble
setcolor(WHITE);
settextstyle(2,0,5);

outtextxy(70,pos_y+80,"'BattleShip - Pacific Warriors' has been programmed as per the");
outtextxy(70,pos_y+95,"requirement of CBSE for the subject Computer Science (083)");
outtextxy(70,pos_y+110,"for AISSCE -2008-09. The source code has been programmed and");
outtextxy(70,pos_y+125,"Compiled in C++ version 3.0 .");

outtextxy(70,pos_y+155,"The Project aims at developing an event game on turbo C++ using");
outtextxy(70,pos_y+170,"the concept of graphics , sounds , structure , classes , datafile");
outtextxy(70,pos_y+185,"handling as described in Class XI and XII text book.");


outtextxy(70,pos_y+215,"It is a multilevel game including a 10x10 grid which contains 5 enemy");
outtextxy(70,pos_y+230,"ship. The player's aim is to locate all ships and destroy them in");
outtextxy(70,pos_y+245,"specified numbers of missile. The game includes 3 different levels.");


//acknowlegment
outtextxy(70,pos_y+305,"I express my deep sense of gratitude and obligation to my respected");
outtextxy(70,pos_y+320,"teacher Mrs. Vandana Arora  for her valuable guidance, interest and");
outtextxy(70,pos_y+335,"constant encouragement given to me for the fulfillment of this project.");
outtextxy(70,pos_y+350,"I am also grateful to my friends for providing me required material and");
outtextxy(70,pos_y+365,"helping in compiling the project.");


setcolor(WHITE);
settextstyle(8,0,2);
outtextxy(220,pos_y+420,"Gagandeep Singh");
setcolor(4);
line(210,pos_y+451,420,pos_y+451);



   // timer
delay(80);

   // Erase
cleardevice();
```

```
      // Move up
   pos_y=pos_y-2;

     // replay
   if(pos_y<=(-490))
   {pos_y=350; }

 }//end of for loop

 getch();
}

/*-------------------------------------------------------------*/
void high_score()
{
 cleardevice();
 setbkcolor(4);
 setfillstyle(1,1);   // logo BG
 bar(0,0,640,127);
 logo(160,0);

 setcolor(4);

 setfillstyle(1,0);        // BG
 bar(0,127,640,350);

 setfillstyle(1,WHITE);           // seperate bar
 bar(0,125,640,130);

 setcolor(WHITE);
 settextstyle(7,0,3);
 outtextxy(245,135,"High Scores");

 //table
 setcolor(YELLOW);
 line(150,175,500,175);
 line(150,190,500,190);
 line(150,300,500,300);
 line(350,175,350,300);

 settextstyle(2,0,4);
 outtextxy(210,176,"PLAYER'S NAME");
 outtextxy(380,176,"ACCURACY");


 settextstyle(8,0,1);
 setcolor(1);
 outtextxy(140,320,"Press Enter to return to the Menu !");

}
/*-------------------------------------------------------------*/
```

Data.h Library

```
void system_chk(int);
void logo(int x, int y);
void trademark(int col);
void into();
void loading_game();
void button_play(int st);
void button_score(int st);
void button_help(int st);
void button_credits(int st);
void button_exit(int st);
void button_mission(int st);
void button_arcade(int st);
void button_training(int st);
void button_back(int st);
int option_screen();
int exit_menu();
void help();
void box();
void error_box();
void pass_enter();
void loading_key();
void loading_main_game();
int mode_select();
int level_file_chk(int lno);
int level_randomize();
void game_board();
int play_mission();
void button_level1(int x, int y);
void button_level2(int x, int y);
void button_level3(int x, int y);
void button_level4(int x, int y);
void button_level5(int x, int y);
void button_level6(int x, int y);
void button_level7(int x, int y);
void button_level8(int x, int y);
void button_level9(int x, int y);
void button_level10(int x, int y);
void button_level11(int x, int y);
void button_level12(int x, int y);
void button_level13(int x, int y);
void button_level14(int x, int y);
void button_level15(int x, int y);
void button_level16(int x, int y);
void button_level17(int x, int y);
void button_level18(int x, int y);
void button_level19(int x, int y);
void button_level20(int x, int y);
int play_arcade();
void saving_game_mission(int);
void board_comment(int);
void shots(int x, int y , int z);
void game_over(int win);
void Credits();
void high_score()
void tutorial();
void game_sounds(int a);
```

BattleShip

Pacific Warriors

Main Program Code

```
#include<fstream.h>
#include<graphics.h>
#include<conio.h>
#include<process.h>
#include<string.h>
#include<stdio.h>
#include<SHIP\data.h>

struct l_file_content        // structure for file content
{char a,b; };

struct ship                  // structure for enemy ship
{int x,y;

 ship()
 {x=-1; y=-1; }
};

struct file_score            // Structure for high score
{char name[10];
 int acc;
};
//---------------------------------------------------------------------

void convert_level_to_path(int lno,char *l_path)
{
 int l_temp;
 if(lno==10)                          // level 10
 {l_path[12]=49; }
 else
 if(lno>10 && lno<20)                 // level 11 - 19
 {l_path[12]=49;
   l_temp=lno;
   l_temp=l_temp-10; l_path[13]=l_temp+48;
 }
 else
 if(lno==20)                          // level 20
 {l_path[12]=50; }
 else                                 // level 1 - 9
 {l_path[13]=lno+48; }
```

```
}
//---------------------------------------------------------------

void display_hit_ship(ship *f,ship *us,int lm)
{
 int a;
 int x1,x2,x3,x4; int y1,y2,y3,y4;
 int x0=30,y0=85;

 //---------------- 2 POINT SHIP ---------------------------
  int chk2_1=0,chk2_2=0;
  int axis_2;
  x1=x2=y1=y2=0;

  for(a=0;a<=lm;a++)                    // check 1 coordinate
  {if( us[a].x==f[0].x && us[a].y==f[0].y)
    {chk2_1=1; break; }
   else
    {chk2_1=0; }
  }
  for(a=0;a<=lm;a++)                    // check 2 coordinate
  {if( us[a].x==f[1].x && us[a].y==f[1].y)
    {chk2_2=1; break; }
   else
    {chk2_2=0; }
  }

  if(chk2_1==1 && chk2_2==1)
  {
   x1=f[0].x; y1=f[0].y;
   x2=f[1].x; y2=f[1].y;

  // convert to graphic coordinates
   x1=x0+x1*30;
   y1=y0+y1*25;
   x2=x0+x2*30;
   y2=y0+y2*25;

  if(x1==x2)
   {axis_2=1; }  // along y
  else
   {axis_2=0; }  // along x


  setfillstyle(1,WHITE);
  switch(axis_2)
  {
  case 0 : fillellipse(x1+30,y1+12,25,6); // x axis
           break;
  case 1 : fillellipse(x1+15,y1+25,8,20);
           break;
 }//end of switch
```

```
 }// end of if

//---------------- 3 (1) POINT SHIP ---------------------------
  int chk31_1=0,chk31_2=0,chk31_3=0;
  x1=x2=x3=y1=y2=y3=0;
  x0=30,y0=85;
  int axis_31;

  for(a=0;a<=lm;a++)                    // check 1 coordinate
  {if( us[a].x==f[2].x && us[a].y==f[2].y)
    {chk31_1=1; break; }
   else
    {chk31_1=0; }
  }
  for(a=0;a<=lm;a++)                    // check 2 coordinate
  {if( us[a].x==f[3].x && us[a].y==f[3].y)
    {chk31_2=1; break; }
   else
    {chk31_2=0; }
  }
  for(a=0;a<=lm;a++)                    // check 3 coordinate
  {if( us[a].x==f[4].x && us[a].y==f[4].y)
    {chk31_3=1; break; }
   else
    {chk31_3=0; }
  }

  if(chk31_1==1 && chk31_2==1 && chk31_3==1)
  {
   x2=f[3].x; y2=f[3].y;
   x3=f[4].x; y3=f[4].y;

   // convert to graphic coordinates
   x2=x0+x2*30;
   y2=y0+y2*25;
   x3=x0+x3*30;
   y3=y0+y3*25;

  if(x2==x3)
   {axis_31=1; }  // along y
  else
   {axis_31=0; }  // along x


  setfillstyle(1,WHITE);
  switch(axis_31)
  {
  case 0 : fillellipse(x2+15,y2+12,40,6); // along x axis
           break;
  case 1 : fillellipse(x2+15,y2+12,8,31); // along y axis
           break;
 }//end of switch
```

```
 }// end of if

//---------------- 3 (2) POINT SHIP ---------------------------
  int chk32_1=0,chk32_2=0,chk32_3=0;
  x1=x2=x3=y1=y2=y3=0;
  x0=30,y0=85;
  int axis_32;

  for(a=0;a<=lm;a++)                    // check 1 coordinate
  {if( us[a].x==f[5].x && us[a].y==f[5].y)
    {chk32_1=1; break; }
   else
    {chk32_1=0; }
  }
  for(a=0;a<=lm;a++)                    // check 2 coordinate
  {if( us[a].x==f[6].x && us[a].y==f[6].y)
    {chk32_2=1; break; }
   else
    {chk32_2=0; }
  }
  for(a=0;a<=lm;a++)                    // check 3 coordinate
  {if( us[a].x==f[7].x && us[a].y==f[7].y)
    {chk32_3=1; break; }
   else
    {chk32_3=0; }
  }

  if(chk32_1==1 && chk32_2==1 && chk32_3==1)
  {
   x2=f[6].x; y2=f[6].y;
   x3=f[7].x; y3=f[7].y;

   // convert to graphic coordinates
   x2=x0+x2*30;
   y2=y0+y2*25;
   x3=x0+x3*30;
   y3=y0+y3*25;

  if(x2==x3)
   {axis_32=1; }  // along y
  else
   {axis_32=0; }  // along x


  setfillstyle(1,WHITE);
  switch(axis_32)
  {
  case 0 : fillellipse(x2+15,y2+12,40,6); // along x axis
           break;
  case 1 : fillellipse(x2+15,y2+12,8,31); // along y axis
           break;
 }//end of switch
```

```
 }// end of if

//---------------- 4 POINT SHIP ---------------------------
  int chk4_1=0,chk4_2=0,chk4_3=0,chk4_4=0;

  x1=x2=x3=x4=y1=y2=y3=y4=0;
  x0=30,y0=85;
  int axis_4;

  for(a=0;a<=lm;a++)                    // check 1 coordinate
  {if( us[a].x==f[8].x && us[a].y==f[8].y)
    {chk4_1=1; break; }
   else
    {chk4_1=0; }
  }
  for(a=0;a<=lm;a++)                    // check 2 coordinate
  {if( us[a].x==f[9].x && us[a].y==f[9].y)
    {chk4_2=1; break; }
   else
    {chk4_2=0; }
  }
  for(a=0;a<=lm;a++)                    // check 3 coordinate
  {if( us[a].x==f[10].x && us[a].y==f[10].y)
    {chk4_3=1; break; }
   else
    {chk4_3=0; }
  }
  for(a=0;a<=lm;a++)                    // check 4 coordinate
  {if( us[a].x==f[11].x && us[a].y==f[11].y)
    {chk4_4=1; break; }
   else
    {chk4_4=0; }
  }

  if(chk4_1==1 && chk4_2==1 && chk4_3==1 && chk4_4==1)
  {
   x2=f[9].x; y2=f[9].y;
   x3=f[10].x; y3=f[10].y;

   // convert to graphic coordinates
   x2=x0+x2*30;
   y2=y0+y2*25;
   x3=x0+x3*30;
   y3=y0+y3*25;

  if(x2==x3)
   {axis_4=1; }  // along y
  else
   {axis_4=0; }  // along x


  setfillstyle(1,WHITE);
```

```
  switch(axis_4)
  {
  case 0 : fillellipse(x2+30,y2+12,50,6); // along x axis
             break;
  case 1 : fillellipse(x3+15,y3,8,40); // along y axis
             break;
 }//end of switch
}// end of if

//---------------- 5 POINT SHIP --------------------------
  int chk5_1=0,chk5_2=0,chk5_3=0,chk5_4=0,chk5_5=0;
  x1=x2=x3=x4=y1=y2=y3=y4=0;
  x0=30,y0=85;
  int axis_5;

  for(a=0;a<=lm;a++)                    // check 1 coordinate
  {if( us[a].x==f[12].x && us[a].y==f[12].y)
    {chk5_1=1; break; }
   else
    {chk5_1=0; }
  }
  for(a=0;a<=lm;a++)                    // check 2 coordinate
  {if( us[a].x==f[13].x && us[a].y==f[13].y)
    {chk5_2=1; break; }
   else
    {chk5_2=0; }
  }
  for(a=0;a<=lm;a++)                    // check 3 coordinate
  {if( us[a].x==f[14].x && us[a].y==f[14].y)
    {chk5_3=1; break; }
   else
    {chk5_3=0; }
  }
  for(a=0;a<=lm;a++)                    // check 4 coordinate
  {if( us[a].x==f[15].x && us[a].y==f[15].y)
    {chk5_4=1; break; }
   else
    {chk5_4=0; }
  }
  for(a=0;a<=lm;a++)                    // check 5 coordinate
  {if( us[a].x==f[16].x && us[a].y==f[16].y)
    {chk5_5=1; break; }
   else
    {chk5_5=0; }
  }

  if(chk5_1==1 && chk5_2==1 && chk5_3==1 && chk5_4==1 && chk5_5==1)
  {
   x3=f[14].x; y3=f[14].y;
   x4=f[15].x; y4=f[15].y;

   // convert to graphic coordinates
```

```
   x3=x0+x3*30;
   y3=y0+y3*25;
   x4=x0+x4*30;
   y4=y0+y4*25;

  if(x3==x4)
   {axis_5=1; }  // along y
  else
   {axis_5=0; }  // along x


  setfillstyle(1,2);
  switch(axis_5)
  {
  case 0 : fillellipse(x3+15,y3+12,65,6); // along x axis
              break;
  case 1 : fillellipse(x3+15,y3+12,8,53); // along y axis

              break;
 }//end of switch
}// end of if

}
//---------------------------------------------------------------------
//---------------------------------------------------------------------


main()
{
clrscr();
 /*------------------- initialising graphic system ----------------*/
 int gdriver=EGA,gmode=EGAHI,gr_chk;
 initgraph(&gdriver,&gmode,"");
 gr_chk=graphresult();

 /*----------------- graphics check ------------------------------*/
  system_chk(gr_chk);

 /*----------------------- Game Key Check -------------------------*/
  // search for file

  int chk_f=0;
  fstream file_chk;
  file_chk.open("C:\\WINDOWS\\pass.dat",ios::in);
  if(!file_chk)
  {chk_f=1; }
  file_chk.close();

  if(chk_f==1)
  {
   char init='0';                          // make file
   file_chk.open("C:\\WINDOWS\\pass.dat",ios::out);
```

```
 file_chk.seekp(0,ios::beg);
 file_chk.write((char *)&init,1);
 file_chk.seekp(0,ios::beg);
 file_chk.close();
}
//-------------------------------------------------------

int p=0;
const char pass[8]={"75199108"};
char filep[8];
fstream file_pass;

file_pass.open("C:\\WINDOWS\\pass.dat",ios::in);     // get for old pass
p=0;
file_pass.seekg(0,ios::beg);
while(!file_pass.eof())
{
 file_pass.seekg(p,ios::beg);
 file_pass.read((char *)&filep[p],1);
 p++;
}
file_pass.close();


  //--------------------  checking the key in file ----------------------
int flag1=0,flag2=0;
for(p=0;p<8;p++)
{if(filep[p]==pass[p])
   {flag1=1; }
  else
   {flag1=0; break; }
}

  //--------------------------- get pass if wrong----------------------
char pin[8];
if(flag1==0)
{
  cleardevice();
  pass_enter();         // get new pass if wrong
  gotoxy(30,17);
  for(p=0;p<8;p++)
  {pin[p]=getche();
    cout<<pin[p];
  }
  gotoxy(1,1);

  for(p=0;p<8;p++)       // check new pass
  {if(pin[p]==pass[p])
         {flag2=1; }
    else
         {flag2=0; break; }
  }
```

```
    if(flag2==1)
    {
     file_pass.open("C:\\WINDOWS\\pass.dat",ios::out);                    // save new pass
     p=0;
     while(p<8)
     {
      file_pass.seekp(p,ios::beg);
      file_pass.write((char *)&pin[p],1);
      p++;
     }
     file_pass.close();
     cleardevice();
     loading_key();
     delay(200);
     cleardevice();
    }
    else
    {cleardevice();
      box();
      setcolor(WHITE);
      outtextxy(200,129, "Activation Failed...");
      setcolor(EGA_RED);
      outtextxy(240,145, "ERROR! Invalid Key");
      setcolor(0);
      outtextxy(200,165, "Please Restart the Game and");
      outtextxy(200,175, "enter correct serial number.");
      game_sounds(-1);
      getch();
      exit(0);
    }
    setcolor(WHITE);
}
     //------------------- end of checking


 /*----------------------------- start ----------------------------------*/

 int choice;
 int EX;
 int ch_mode=0;

 loading_game();
 do
 {
 choice=0;
 ch_mode=0;
 choice=option_screen();
 switch(choice)
 {
  case 1 : cleardevice();                          // PLAY
              setbkcolor(0);
```

```
{
 ch_mode=0;
 ch_mode=mode_select();

 char PATH[19]={"levels\\level00.dat"};
 int GAME_LEVEL;
 fstream f_l;
 l_file_content lvl[17];
 ship enemy[17];
 ship user[50];

 int l,i;


  //------------------------------ MISISON mode---------------------------

  if(ch_mode==1)
  {
        cleardevice();
        int mission;
        mission=play_mission();         // load mission
        GAME_LEVEL=mission;
        convert_level_to_path(mission,PATH);   // convert

        cout<<"mission : "<<mission;
        cout<<"\nFINAL PATH : "<<PATH;

  }

  //------------------------------ arcade mode---------------------------

  if(ch_mode==2)
  {
        cleardevice();

         for(i=0;i<17;i++)                //reset
         { enemy[i].x=0;  enemy[i].y=0;  }

        int a_level;
        int al_chk;

        a_level=play_arcade();
        cout<<"Level : "<<a_level;
        GAME_LEVEL=a_level;

        al_chk=level_file_chk(a_level);  // check level file

        if(al_chk==0)
         {error_box();  game_sounds(-1); getch(); exit(0); }

        convert_level_to_path(a_level,PATH);   // convert
```

```
        cout<<"\nFINAL PATH : "<<PATH;

 }


 //----------------------------- training mode--------------------------

 if(ch_mode==3)
 {
        cleardevice();

         for(i=0;i<17;i++)                  //reset
         { enemy[i].x=0;  enemy[i].y=0;  }

        int r_level;
        int rl_chk;

        r_level=level_randomize();              // randomize level
        cout<<"Level : "<<r_level;
        GAME_LEVEL=r_level;

        rl_chk=level_file_chk(r_level);         // check level file

        if(rl_chk==0)
         {error_box();  game_sounds(-1); getch(); exit(0); }

        convert_level_to_path(r_level,PATH);      // convert

        cout<<"\nFINAL PATH : "<<PATH;


        }
        //---------------------------------------------------------------


if(ch_mode!=4)            //if not back
{
  cleardevice();
          //-------------- extract_level -------------
            l=0;
            f_l.open(PATH,ios::in);
            l=0; i=0;
            while(!f_l.eof())
            {
             f_l.seekg(i,ios::beg);
             f_l.read((char *)&lvl[l],2);
             i=i+2;
             l++;
            }
            f_l.close();
```

```
                    //  convertion
          for(i=0;i<17;i++)
          {
           enemy[i].x=lvl[i].a-48;
           enemy[i].y=lvl[i].b-48;
          }

     //---------- LOADING ------------------------
cleardevice();
loading_main_game(ch_mode,GAME_LEVEL);
tutorial();

     //---------- PLAY GAME ------------------------


char usr_x,usr_y;
int temp_x,temp_y;

int usr_chk=0;
int AMMO=0;
int WIN=0;
int HITS=0;
int SHOOTS=0;
int ACCURACY=1;
int j,ux_temp,limit;

if(GAME_LEVEL>=0 && GAME_LEVEL<=5)
     {AMMO=50; }
else
if(GAME_LEVEL>=6 && GAME_LEVEL<=10)
     {AMMO=45; }
else
if(GAME_LEVEL>=11 && GAME_LEVEL<=15)
     {AMMO=40; }
else
if(GAME_LEVEL>=16 && GAME_LEVEL<=20)
     {AMMO=34; }

limit=AMMO;

ship user[50];
for(j=0;j<17;j++)
{user[j].x=-1;  user[j].y=-1; }

game_board(GAME_LEVEL);
for(i=0;i<limit;i++)          // loop for no of shots
{
      //---------- display ammo ------------------
      setcolor(WHITE);
      setfillstyle(1,0);
      bar(540,150,570,170);
      rectangle(540,150,570,170);
```

```
 gotoxy(69,12); cout<<AMMO;

do
{
 //------------ erase coordinates --------------

 board_comment(6);

 setfillstyle(1,0);
 bar(425,150,465,170);
 setcolor(WHITE);
 rectangle(425,150,465,170);

 bar(425,173,465,193);
 rectangle(425,173,465,193);

 // -------------- get and display entry ---------------
 gotoxy(55,12); usr_x=getch(); cout<<usr_x;
 gotoxy(55,14); usr_y=getch(); cout<<usr_y;
 ux_temp=usr_x;

 if( (usr_x>=48 && usr_x<=57)&&(usr_y>=48 && usr_y<=57) )
  {usr_chk=1;}
 else
 if(usr_x=='e'||usr_x=='E'|| usr_x==27 || usr_y==27)
  {
    board_comment(-2);  // exit
    usr_chk=0;
  }
 else
  {
    if(usr_x==8 || usr_y==8)  // Backspace
     {board_comment(-3); delay(1500); usr_chk=0; }
    else
    {board_comment(-1);    //wront entry
      usr_chk=0;
      delay(1500);
    }
  }

   // ------ check for repeated entry ------

  temp_x=usr_x-48;
  temp_y=usr_y-48;

  for(j=0;j<50;j++)
  {
   if((temp_x==user[j].x)&&(temp_y==user[j].y))
    {usr_chk=0; board_comment(7); delay(1500); }
  }

}while(usr_chk==0);
```

```
//----------- processing.....  ----------------
board_comment(5);
delay(1500);

temp_x=usr_x-48;
temp_y=usr_y-48;

user[i].x=temp_x; user[i].y=temp_y;

// ------------ comparing hits
int hit_chk=0;
for(j=0;j<17;j++)
{
      if((temp_x==enemy[j].x)&&(temp_y==enemy[j].y))
       {shots(temp_x,temp_y,1);
        board_comment(1); delay(2000);
        hit_chk=1;
        display_hit_ship(enemy,user,SHOOTS);
        HITS++;
        break;
       }
      else
       {shots(temp_x,temp_y,0);
        hit_chk=0;
       }
}

//------------ Game Status
setfillstyle(1,0);
bar(370,235,410,255);
setcolor(WHITE);
rectangle(370,235,410,255);
gotoxy(48,18); cout<<HITS;
//Progress Bar
setfillstyle(1,4);
bar(511,239,511+(HITS*7),250);

if(hit_chk==0)
{ board_comment(0); delay(2000); }

AMMO--;
SHOOTS++;

// GAME OVER CHECK
       if(HITS==17)
       {WIN=1; break; }
}// end of for lop till 50
```

```
//------------------- end of game--------------------
//------------------- Display Results ---------------

game_over(WIN);

ACCURACY=(HITS*100)/SHOOTS;

gotoxy(50,13);  cout<<limit;
gotoxy(50,14);  cout<<SHOOTS;
gotoxy(50,15);  cout<<HITS;
gotoxy(50,16);  cout<<(SHOOTS-HITS);
gotoxy(43,19);  cout<<ACCURACY<<" %";

settextstyle(0,0,1);
setcolor(WHITE);
outtextxy(200,171,"Total AMMO :");
outtextxy(200,186,"Shoots Taken :");
outtextxy(200,201,"No. of Hits :");
outtextxy(200,216,"No. of Fake Shoots :");
outtextxy(230,256,"ACCURACY : ");

if( (ch_mode==1)&&(WIN==1)&&(GAME_LEVEL!=20) )
{settextstyle(0,0,2);
      setcolor(4);
      outtextxy(150,290,"Next Level Unlocked !");
}

setcolor(15);
line(0,247,640,247);
line(0,270,640,270);

getch();
//------------------- end of results
                  // save game
if( (ch_mode==1)&&(WIN==1)&&(GAME_LEVEL!=20) )
{saving_game_mission(GAME_LEVEL); }

                 // end of game
if( (ch_mode==1)&&(WIN==1)&&(GAME_LEVEL==20) )
{
      cleardevice();
      setbkcolor(0);

       // Ship Graphics
       // right guns
       int G1_R[]={395,82 , 450,65, 453,70 , 395,97 };
       int G2_R[]={410,100 , 470,82, 473,87 , 410,115 };
       setfillstyle(1,8);
       setcolor(WHITE);
       fillpoly(4,G1_R);
       fillpoly(4,G2_R);
```

```
 // Left guns
int G1_L[]={265,82 , 210,65, 207,70 , 265,97 };
int G2_L[]={250,100 , 190,82, 187,87 , 250,115 };
setfillstyle(1,8);
setcolor(WHITE);
fillpoly(4,G1_L);
fillpoly(4,G2_L);

 // tower
setfillstyle(1,8);                          // antenna 1
setcolor(0);
bar(328,3,331,105);
bar(310,10,350,12);

bar(299,15,301,105);                      // antenna 2
bar(288,20,312,21);

setfillstyle(1,8);                         // main tower
setcolor(WHITE);
bar(310,40,350,105);
rectangle(310,40,350,105);

setfillstyle(1,WHITE);
setcolor(8);
fillellipse(330,52,7,7);


 // floor 3
setfillstyle(1,8);
setcolor(WHITE);
bar(290,65,370,105);
rectangle(290,65,370,105);

 // floor 2
setfillstyle(1,7);
setcolor(WHITE);
bar(265,75,395,105);
rectangle(265,75,395,105);

setfillstyle(1,0);
setcolor(7);
bar(280,82,380,88);
line(300,82,300,88);
line(320,82,320,88);
line(340,82,340,88);
line(360,82,360,88);


 // floor 1
setfillstyle(1,8);
setcolor(WHITE);
bar(250,95,410,140);
```

```
 rectangle(250,95,410,140);

  // ship base
 int ship_base[]={330,105 , 435,130 , 405,200 , 255,200 , 225,130 };
 setfillstyle(1,6);
 setcolor(WHITE);
 fillpoly(5,ship_base);
 setcolor(WHITE);
 line(330,100,330,200);

  // fence
 setcolor(WHITE);
 line(330,100,225,125); line(330,100,435,125);
 line(225,125,225,130); line(435,125,435,130);

 // logo display
 settextstyle(GOTHIC_FONT,0,8);
 setcolor(0);
 outtextxy(162,152,"BattleShip");
 logo(160,150);
 // end of ship



settextstyle(8,0,5);
setcolor(WHITE);
outtextxy(110,265,"CONGRATULATIONS!");
setcolor(4);
line(110,318,530,318);

settextstyle(7,0,1);
setcolor(WHITE);
outtextxy(50,325,"You Have Completed All Levels!  Thank You For Playing!");

game_sounds(2);
delay(500);
sound(300);
delay(200);
nosound();
sound(500);
delay(900);
nosound();

getch();
cleardevice();
Credits(0);

cleardevice();
char RESET;
box();
setcolor(EGA_WHITE);
outtextxy(200,129, "Reset Game");
```

```
          settextstyle(0,0,1);
          setcolor(EGA_RED);
          outtextxy(235,155, "Do You want to Reset");
          outtextxy(235,165,"  the Game ? (Y/N)");
          RESET=getch();

          if(RESET=='y' || RESET=='Y')
          {
           fstream f_l_temp;
           int ret_temp=0;

           box();
           setcolor(EGA_WHITE);
           outtextxy(200,129, "Reset Game");

           settextstyle(0,0,1);
           setcolor(EGA_RED);
           outtextxy(260,155, "Please Wait !");

           f_l_temp.open("C:\\windows\\btl_save.dat",ios::out);
           f_l_temp.seekp(0,ios::beg);
           f_l_temp<<ret_temp;
           f_l_temp.close();
           delay(1500);

           outtextxy(260,165,"    Done!");

           setcolor(1);
           outtextxy(210,190, "Press any key to continue!");
           getch();
          }
  }//end of congratulation if


  //-------------------- High Score ----------------------
 fstream f_score;

 file_score sc_data[7];

//------------------- check and create file on fail ------------
 file_score def_sc={"Gagandeep",50}, defl={"default",0};
 f_score.open("hsdata.dat",ios::in);
 if(f_score==0)
 {f_score.open("hsdata.dat",ios::out);
  f_score.seekp(0,ios::beg);
  f_score.write((char *)&def_sc,sizeof(def_sc));
  f_score.close();

  for(i=0;i<6;i++)
  { f_score.open("hsdata.dat",ios::out | ios::app);
          f_score.seekg(0,ios::end);
```

```
          f_score.write((char *)&defl,sizeof(defl));
         f_score.close();
  }

}
f_score.close();


//------------------------read file ----------------------------
 f_score.open("hsdata.dat",ios::in);
  f_score.seekg(0,ios::beg);
  for(i=0;i<7;i++)
  {f_score.read((char *)&sc_data[i],sizeof(sc_data[i])); }
 f_score.close();

 //---------------------- Save New Entry --------------------
 int hs_chk=0;
 int max_hs=sc_data[0].acc;
 int min_hs=sc_data[0].acc;
 int min_pos=0;

 for(i=0;i<7;i++)
 { if(sc_data[i].acc>max_hs)   // max value
        {max_hs=sc_data[i].acc;}

        if(sc_data[i].acc<min_hs)   // min value
        {min_hs=sc_data[i].acc;}
 }

 for(i=0;i<7;i++)             // find min position
 {
  if(sc_data[i].acc==min_hs)
  {min_pos=i; break; }
 }

 if(ACCURACY>max_hs)           // check for high score
   hs_chk=1;
 else
   hs_chk=0;

 //---------------- add high score
 char usrname[10];
 if(hs_chk==1)
 {
  //----------------- graphic display ------------------
 cleardevice();
 setbkcolor(6);
 logo(160,0);
 setfillstyle(1,14);           // separate bar
 bar(0,125,640,180);
 settextstyle(0,0,2);
 setcolor(4);
```

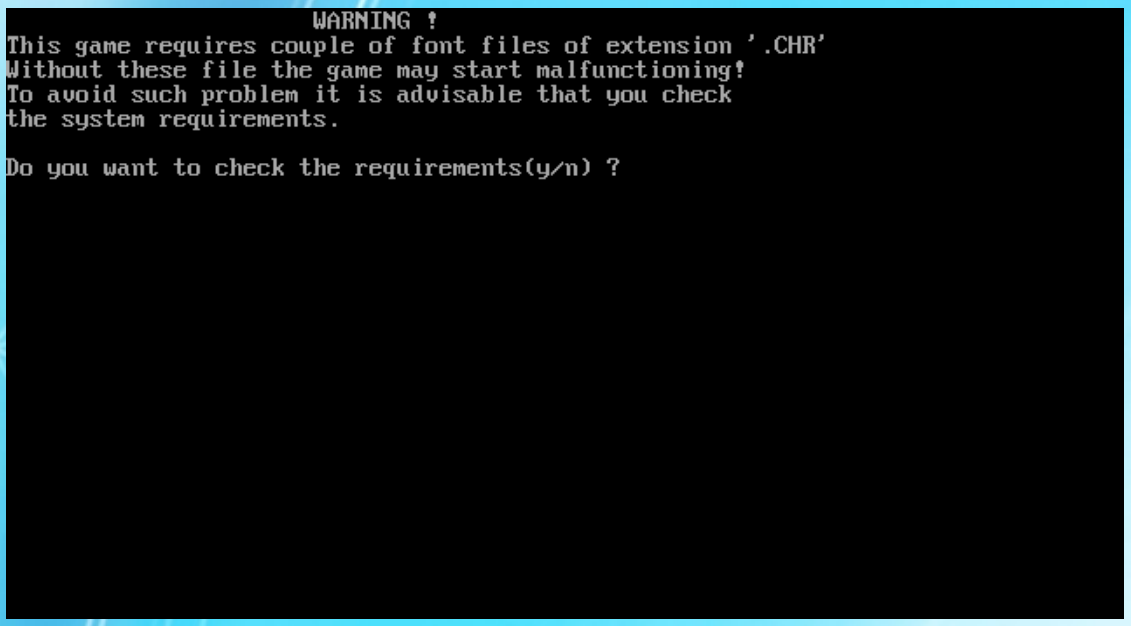
```
outtextxy(190,130,"Congratulations!");
setcolor(0);
outtextxy(110,155,"You Have The Highest Score.");

trademark(14);

settextstyle(3,0,3);
setcolor(WHITE);
outtextxy(220,197,"Accuracy : ");
gotoxy(45,16); cout<<ACCURACY;

outtextxy(150,242,"Enter your Name : ");
rectangle(365,250,470,275);

sound(300);
delay(200);
nosound();
sound(500);
delay(900);
nosound();

gotoxy(48,19); gets(usrname);

cleardevice();
setbkcolor(BLACK);
box();
setcolor(EGA_WHITE);
outtextxy(200,129, "Saving...");
setcolor(EGA_RED);
outtextxy(255,145, "Please Wait!");
setcolor(1);
outtextxy(270,165, "Saving...");

setcolor(4);
rectangle(220,179,400,195);

setfillstyle(SOLID_FILL, GREEN );
for(int i=224;i<=396;i=i+4)
{
 delay(100);
 bar(223,182,i,192);
}
//-----------------------------------------------------

 for(i=0;i<10;i++)
 {sc_data[min_pos].name[i]=usrname[i]; }
 sc_data[min_pos].acc=ACCURACY;

//------------- swapping before saving ----------------------
file_score temp;
int k;
for(i=0;i<7;i++)
```

```
            {
             for(k=0;k<6;k++)
             {
              if(sc_data[k+1].acc>sc_data[k].acc)
              {temp=sc_data[k];
                    sc_data[k]=sc_data[k+1];
                    sc_data[k+1]=temp;
              }
             }

            }

             //------------------ save file ---------------------------
             f_score.open("hsdata.dat",ios::out);
             f_score.seekg(0,ios::beg);
             f_score.write((char *)&sc_data[0],sizeof(sc_data[0]));
             f_score.close();

             for(i=1;i<7;i++)
              { f_score.open("hsdata.dat",ios::out | ios::app);
                       f_score.seekg(0,ios::end);
                       f_score.write((char *)&sc_data[i],sizeof(sc_data[i]));
                      f_score.close();
              }
             }//end of if(hs_chk==1) condition



             } //end of 'if(ch_mode!=4)'

             cleardevice();
             setbkcolor(0);

            }//end of main compuond case 1

           break;

case 2 : cleardevice();                              // HIGH SCORE
           high_score();

           fstream f_score;
           file_score sc_data[7];

           //------------------ check and create file------------------------
             file_score def_sc={"Gagandeep",50}, defl={"default",0};
             f_score.open("hsdata.dat",ios::in);
             if(f_score==0)
             {f_score.open("hsdata.dat",ios::out);
               f_score.seekp(0,ios::beg);
               f_score.write((char *)&def_sc,sizeof(def_sc));
               f_score.close();
```

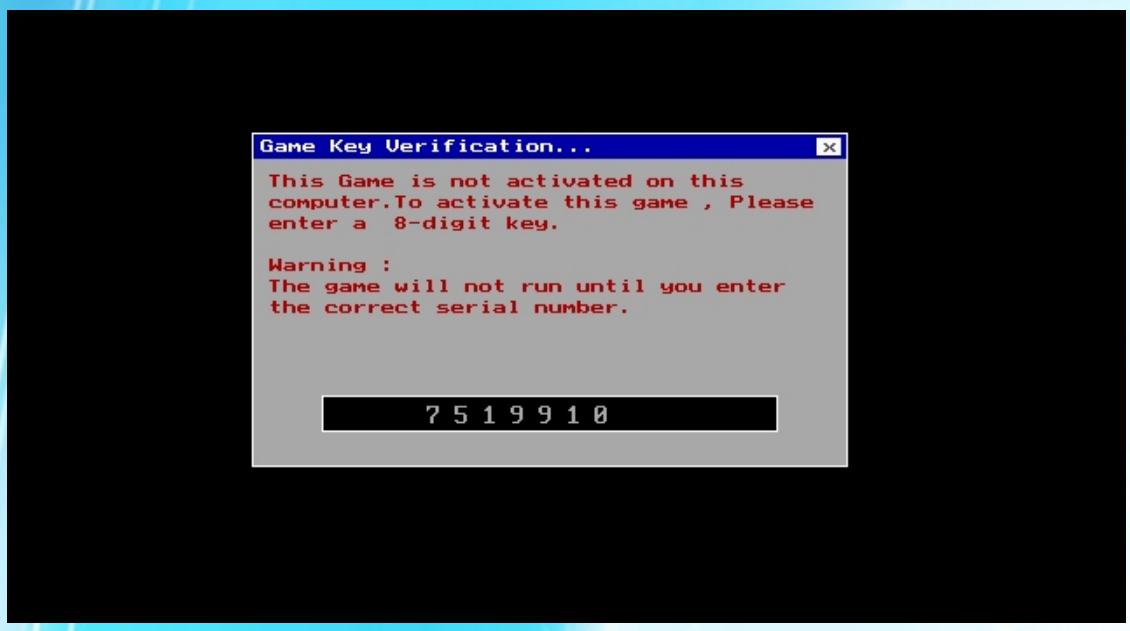
```
            for(i=0;i<6;i++)
            { f_score.open("hsdata.dat",ios::out | ios::app);
                    f_score.seekg(0,ios::end);
                    f_score.write((char *)&defl,sizeof(defl));
                   f_score.close();
            }

           }
           f_score.close();

          //------------------read file--------------------
          f_score.open("hsdata.dat",ios::in);
           f_score.seekg(0,ios::beg);
           for(i=0;i<7;i++)
           {f_score.read((char *)&sc_data[i],sizeof(sc_data[i])); }
          f_score.close();

          //-------------------- Display ------------------------------
          char temp_def_chk[10]={"default"};
          for(i=0;i<7;i++)
          {
           if( strcmpi(sc_data[i].name,temp_def_chk) )
           {gotoxy(29,i+15);cout<<sc_data[i].name;    gotoxy(49,i+15); cout<<sc_data[i].acc<<" %"; }
          }


          getch();

          choice=0;
          ch_mode=0;
          break;

case 3 : cleardevice();                   // HELP
          setbkcolor(0);
          help();
          break;

case 4 : cleardevice();                        // CREDITS
          setbkcolor(0);
          Credits(1);
          cleardevice();
          break;

case 5 : cleardevice();                   // EXIT
          EX=exit_menu();
          if(EX==1)
           {exit(0); }
          break;

} // end of main switch

}while((EX==2) || (choice!=5) || (ch_mode==4));
```

```
getch();
cleardevice();
clrscr();
return 0;
}
```

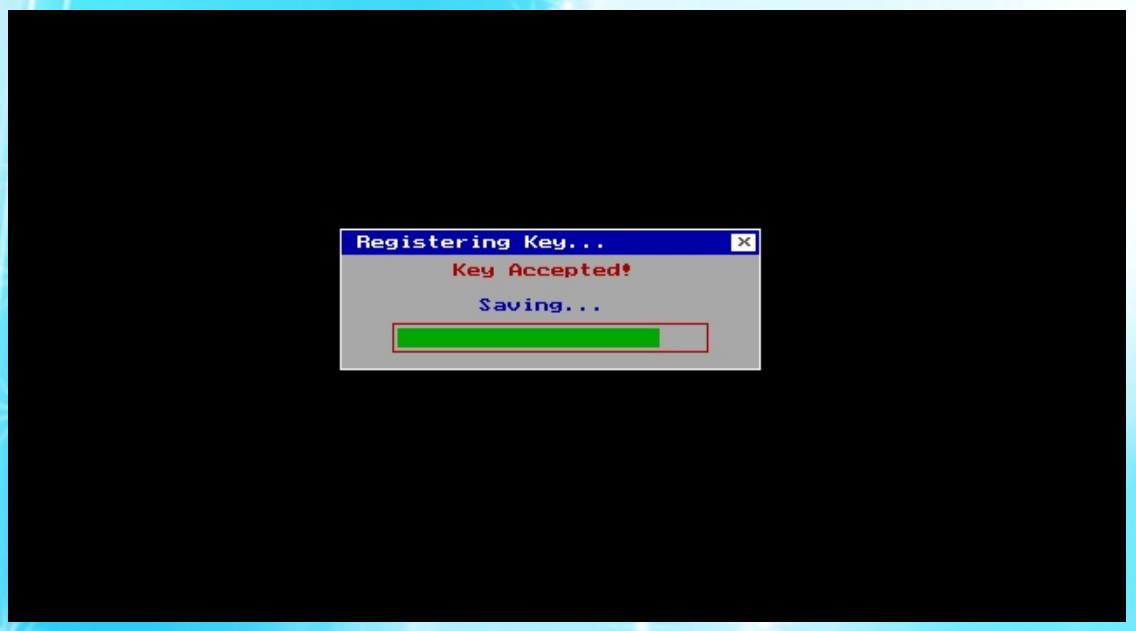
Program Output



Search for missing file required by the game and prompts an error if any.

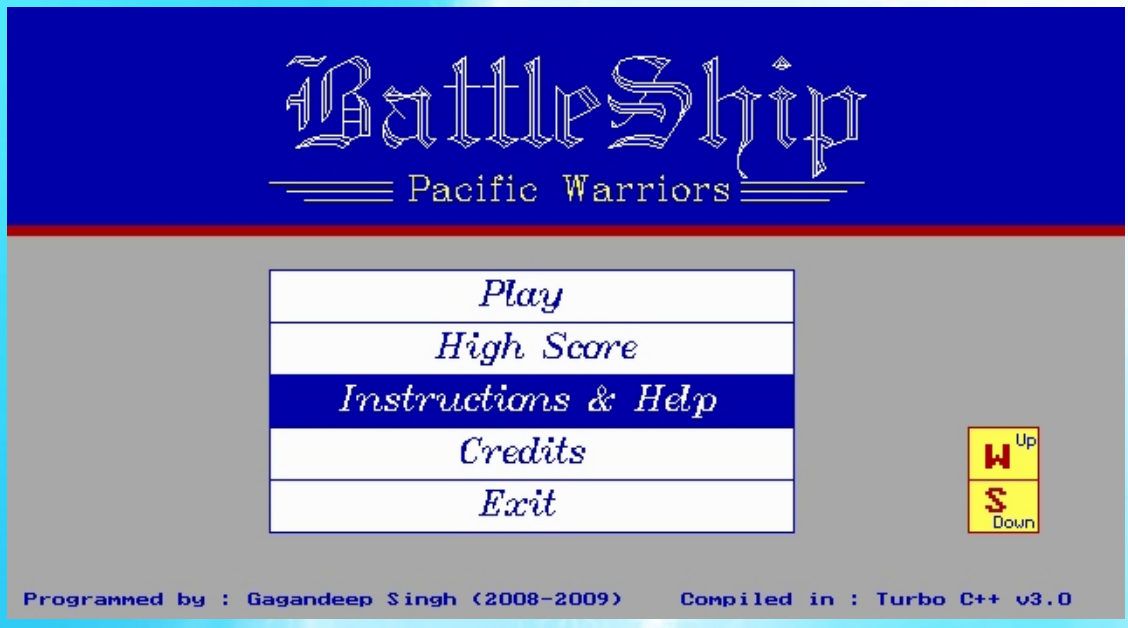


Check the activation serial key.
If user enter any invalid key then the game is terminated.



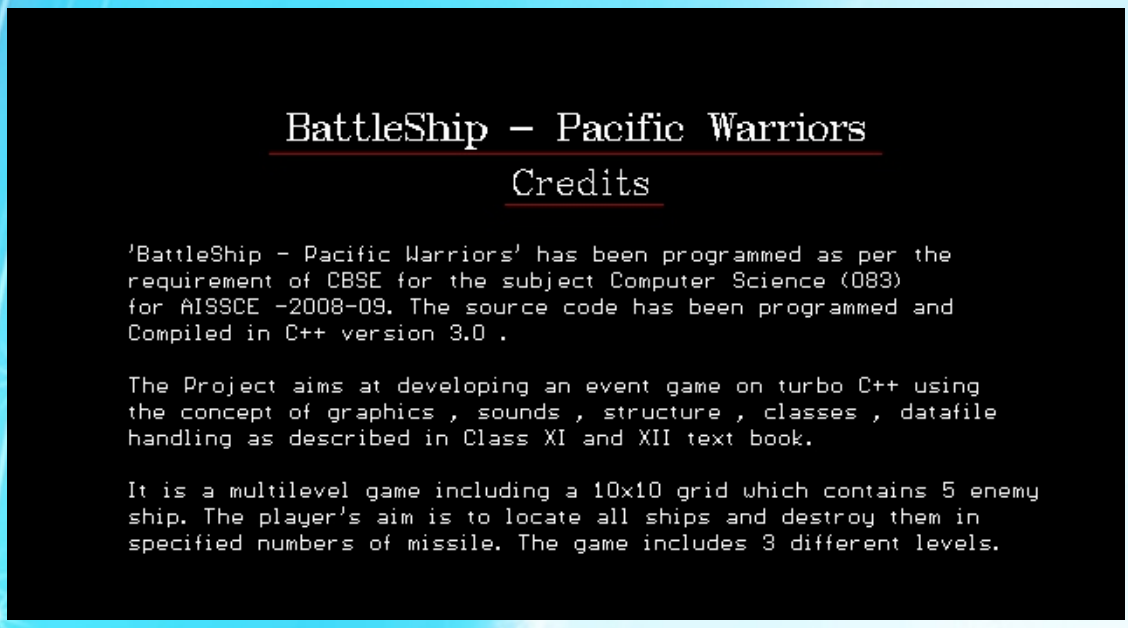
Saving the correct activation key enter by user which can be later analyze when game is run again.

Loading screen



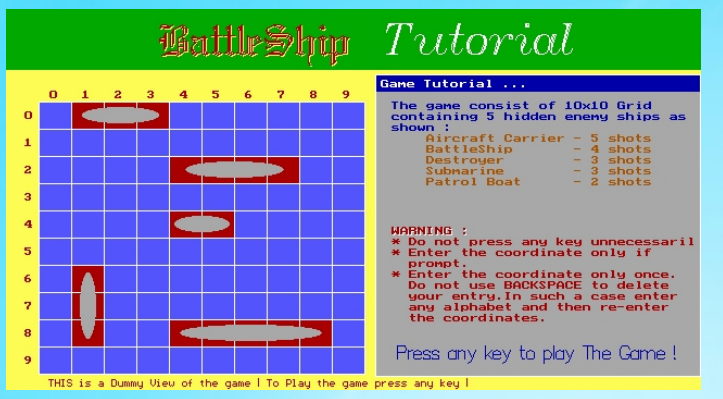
Main Menu

High Score Menu



Credits

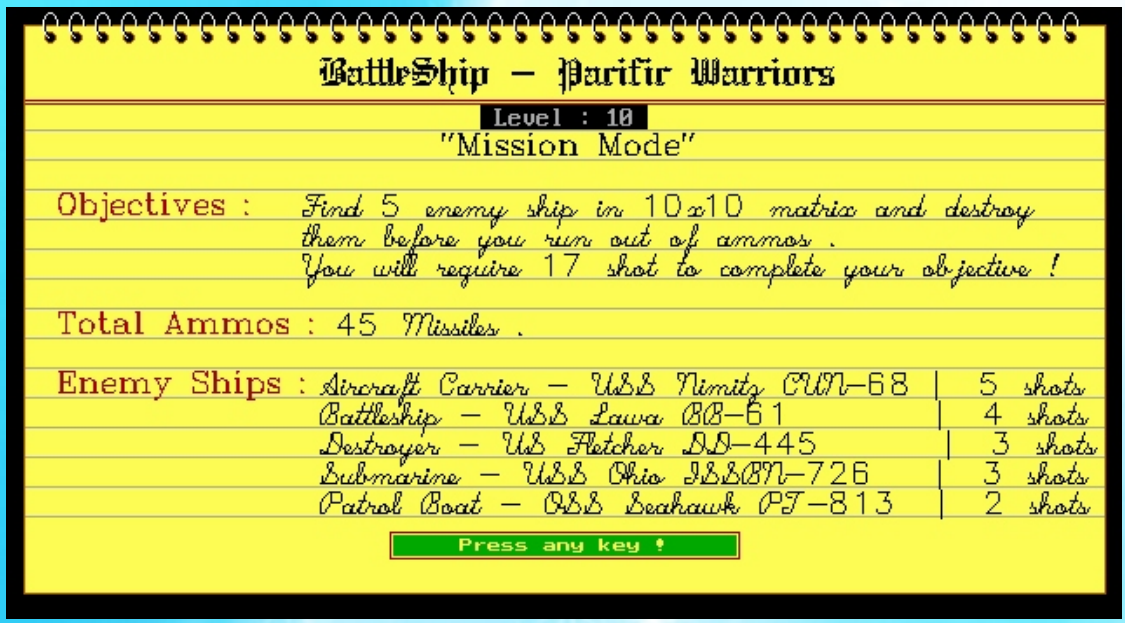
Exit Menu



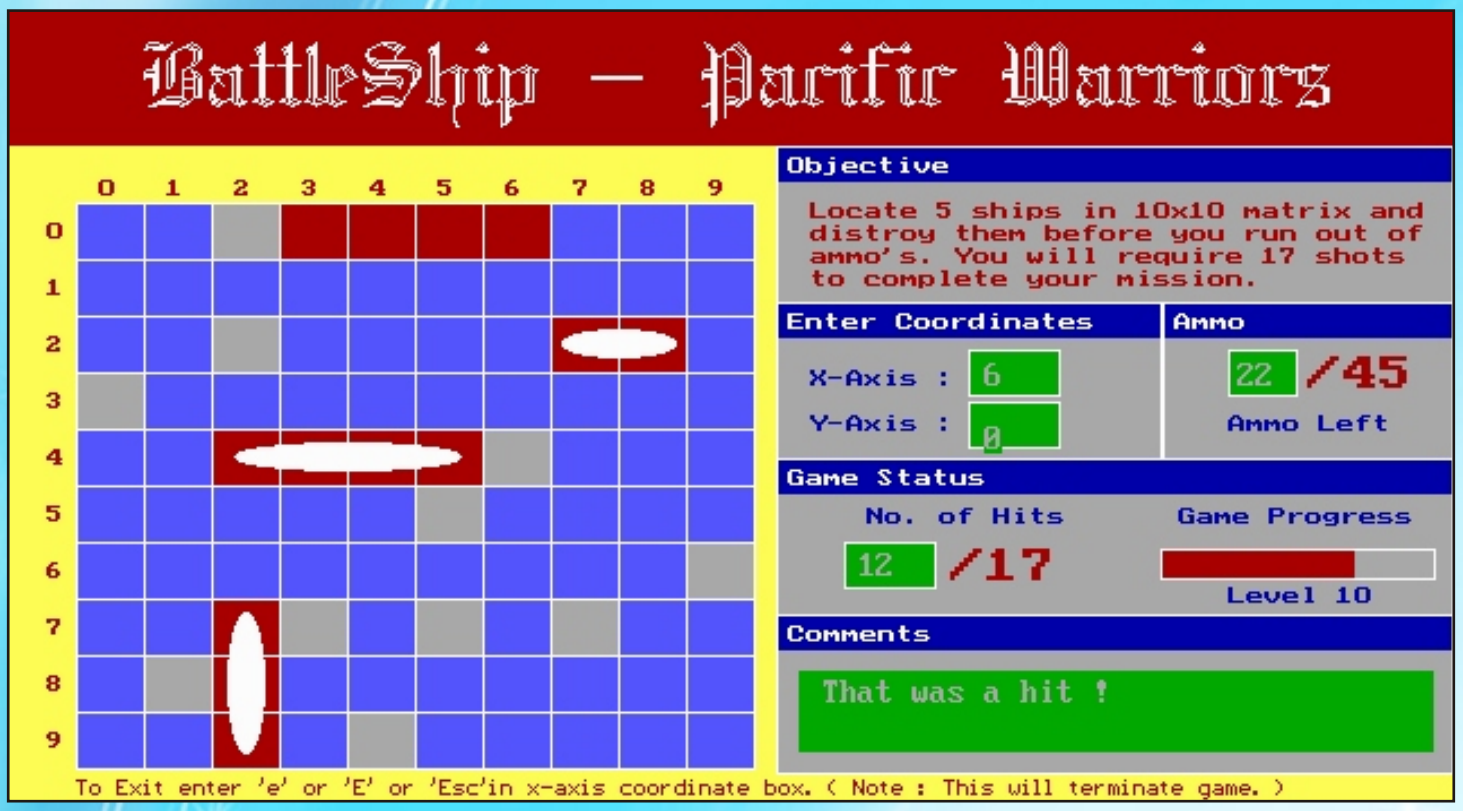
Instructions and Help screen shots

Game Mode Menu

Arcade Mode - The Levels in red text are unlocked and can be played.



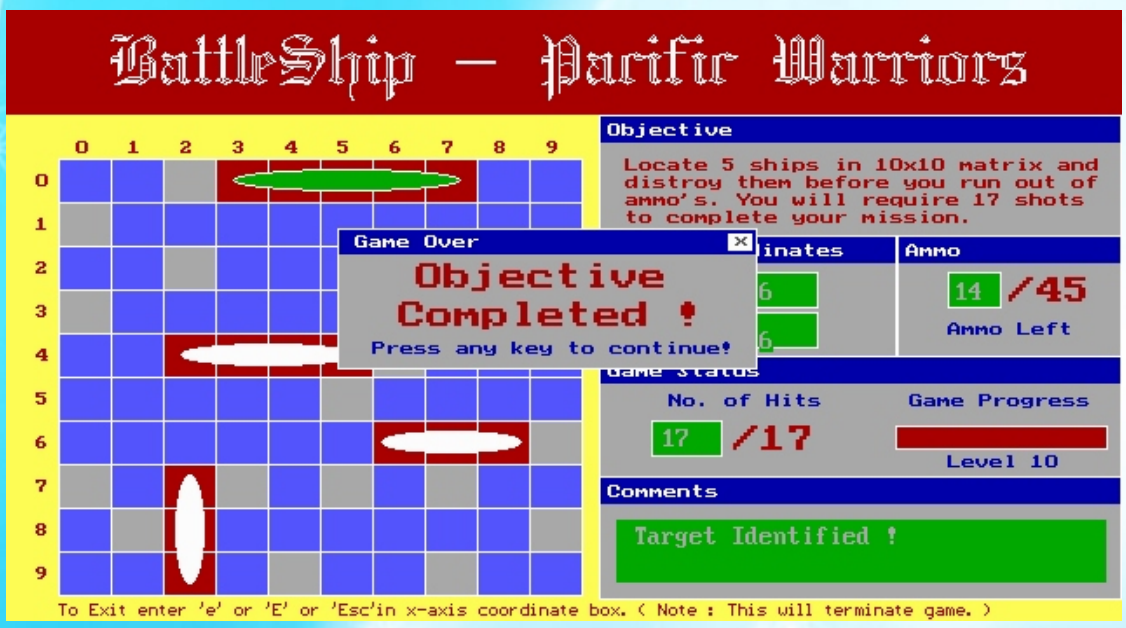
Screen for loading game board and objects.



Main Game Board

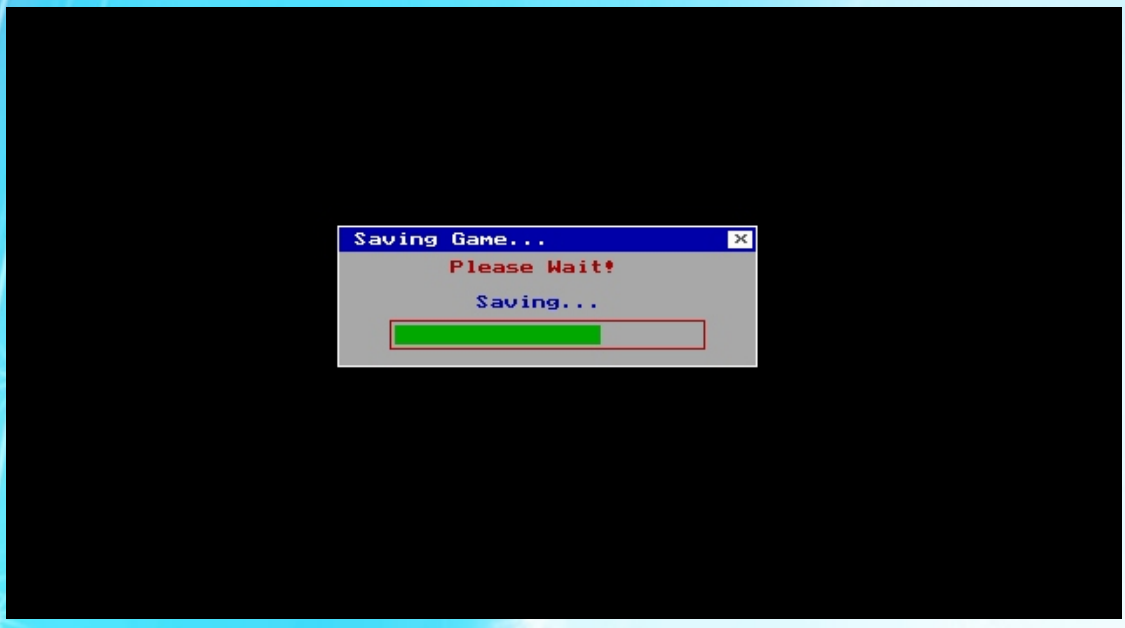
The red dots indicate a hit while the gray dots indicate a false hit.

When the complete ship has been destroyed , the board display the sunk ship.

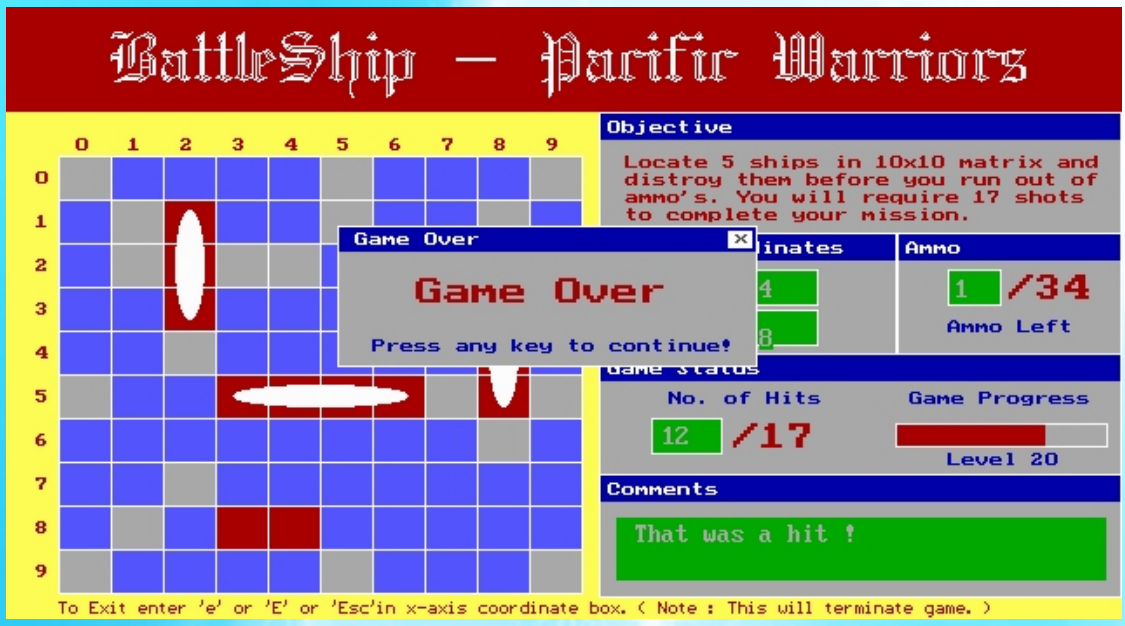


Game Over - Player has won the game

Display players performance and unlock the next level.



Saves the game so that it can be loaded again from where it ended.



Game Over - Player lost the game.

Display results of lost game.

Reads player name and stores it in high score file.

Display a ‘congratulations’ note if player has completed all levels.

BattleShip

Pacific Warriors

Game Requirements Verifier

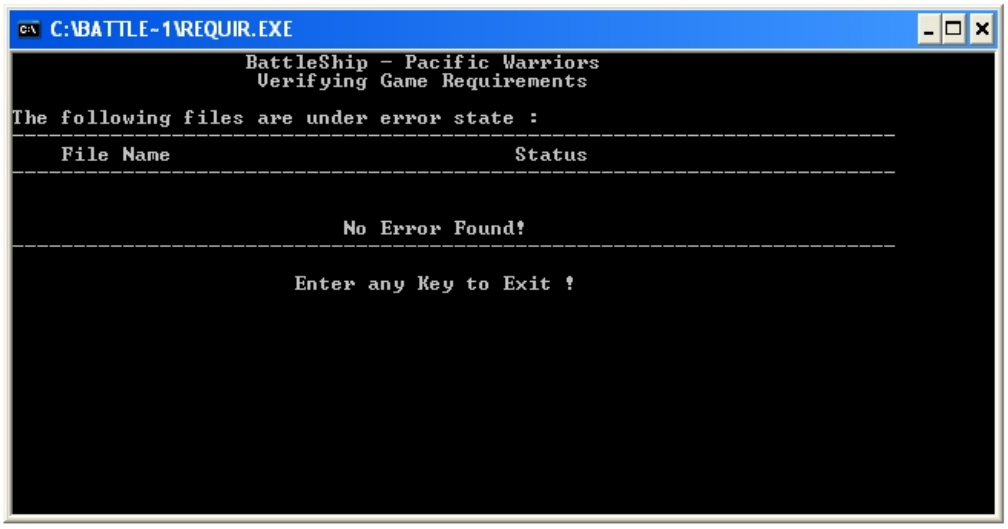
This program search for missing file required by the game and prompts an error if any of the files are missing.

Program Code

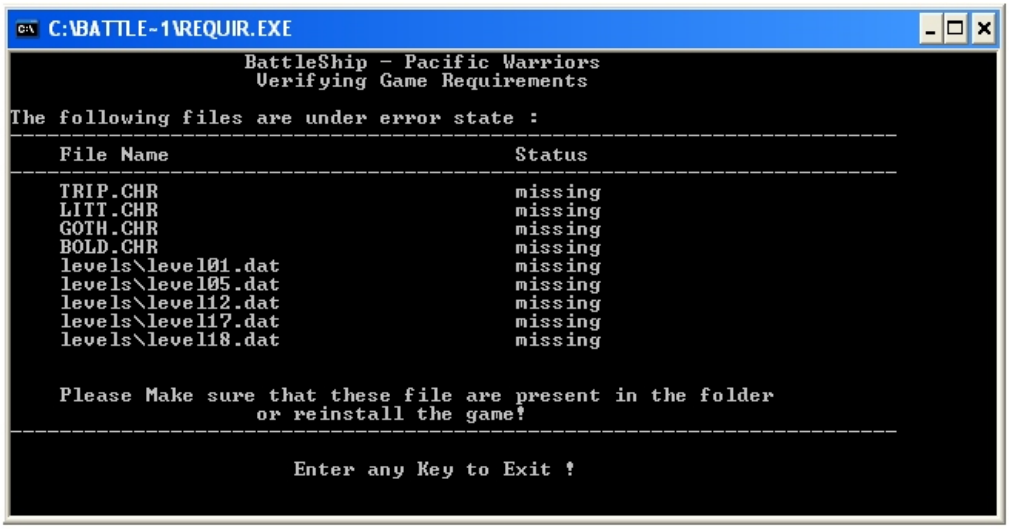
```
#include<fstream.h>
#include<conio.h>
#include<process.h>

main()
{
clrscr();

 fstream f;
 int i,ctr=0;
 char a;
 char file_name[32][25]={"EGAVGA.BGI",
                         "TRIP.CHR",
                         "LITT.CHR",
                         "SANS.CHR",
                         "GOTH.CHR",
                         "SCRI.CHR",
                         "SIMP.CHR",
                         "TSCR.CHR",
                         "LCOM.CHR",
                         "EURO.CHR",
                         "BOLD.CHR",
                         "levels\\level00.dat",
                         "levels\\level01.dat",
                         "levels\\level02.dat",
                         "levels\\level03.dat",
                         "levels\\level04.dat",
                         "levels\\level05.dat",
                         "levels\\level06.dat",
                         "levels\\level07.dat",
                         "levels\\level08.dat",
                         "levels\\level09.dat",
                         "levels\\level10.dat",
                         "levels\\level11.dat",
                         "levels\\level12.dat",
```

```
                                "levels\\level13.dat",
                                "levels\\level14.dat",
                                "levels\\level15.dat",
                                "levels\\level16.dat",
                                "levels\\level17.dat",
                                "levels\\level18.dat",
                                "levels\\level19.dat",
                                "levels\\level20.dat", };

  gotoxy(1,1);
  cout<<"                BattleShip - Pacific Warriors"<<endl;
  cout<<"                 Verifying Game Requirements "<<endl;
  cout<<"\nThe following files are under error state : ";
  cout<<"\n--------------------------------------------------------------------------------"<<endl;
  cout<<"    File Name                       Status";
  cout<<"\n--------------------------------------------------------------------------------"<<endl;
  int temp=0;
  for(i=0;i<32;i++)
  {
   f.open(file_name[i],ios::in);
    if(f==0)
    {
      if(i<=11)
           {cout<<"    "<<file_name[i]<<"                        missing"<<endl; }
      else
           {cout<<"    "<<file_name[i]<<"                missing"<<endl; }
      ctr++;
      temp++;
    }
   f.close();


  if(temp>12)
  {cout<<"\n\n   Press any key to continue...."; getch(); temp=0; cout<<"\n\n";}

  }

  // comment
  if(ctr==0)
   {cout<<"\n\n                 No Error Found!"; }
  else
   {cout<<"\n\n    Please Make sure that these file are present in the folder";
     cout<<"                                or reinstall the game!";
    }
  cout<<"\n--------------------------------------------------------------------------------"<<endl;
  cout<<"\n                    Enter any Key to Exit !";


 getch();
 return 0;
}
```

Output



Display no error when no files are missing from folder.



Search for missing files and display an error.

BattleShip
Pacific Warriors

Game Uninstall Wizard

BattleShip Uninstall Shield removes all the game files from computer hard disk on user choice .

Program Code

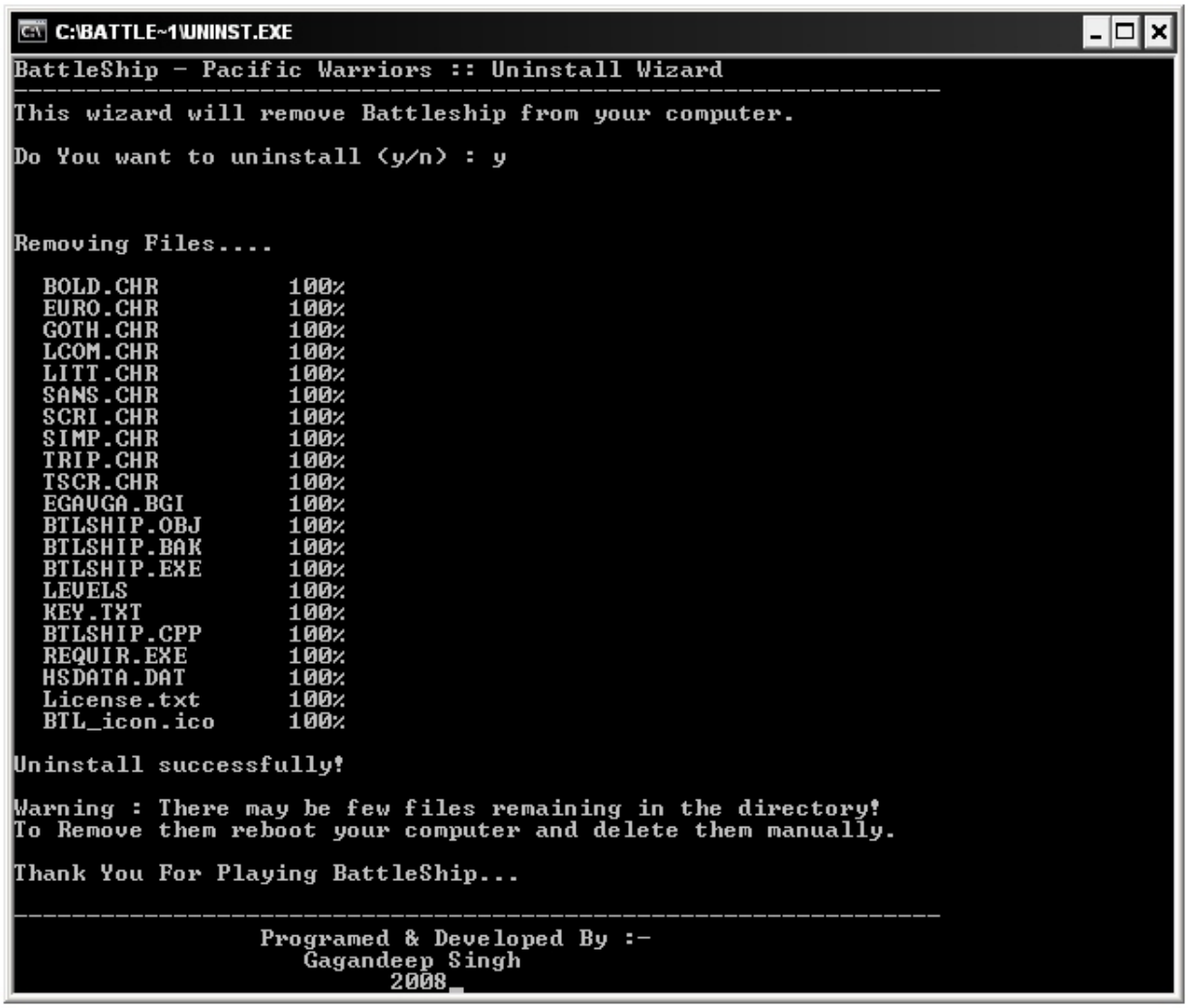
```
#Include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<dos.h>
main()
{
clrscr();
  char ch;
  char level_file[21][25]={"levels\\level00.dat",
                           "levels\\level01.dat",
                           "levels\\level02.dat",
                           "levels\\level03.dat",
                           "levels\\level04.dat",
                           "levels\\level05.dat",
                           "levels\\level06.dat",
                           "levels\\level07.dat",
                           "levels\\level08.dat",
                           "levels\\level09.dat",
                           "levels\\level10.dat",
                           "levels\\level11.dat",
                           "levels\\level12.dat",
                           "levels\\level13.dat",
                           "levels\\level14.dat",
                           "levels\\level15.dat",
                           "levels\\level16.dat",
                           "levels\\level17.dat",
                           "levels\\level18.dat",
                           "levels\\level19.dat",
                           "levels\\level20.dat", };

cout<<"BattleShip - Pacific Warriors :: Uninstall Wizard";
cout<<"\n------------------------------------------------------------";
cout<<"\nThis wizard will remove Battleship from your computer.";
cout<<"\n\nDo You want to uninstall (y/n) : ";
```

```
cin>>ch;
cout<<"\n\n";

if((ch=='Y'||ch=='y'))
 {
  cout<<"\nRemoving Files....\n\n";
  delay(1100);
    remove("bold.chr");
    cout<<"  BOLD.CHR        100%\n";

  delay(700);
    remove("euro.chr");
    cout<<"  EURO.CHR        100%\n";

  delay(700);
    remove("goth.chr");
    cout<<"  GOTH.CHR        100%\n";

  delay(700);
    remove("lcom.chr");
    cout<<"  LCOM.CHR        100%\n";

  delay(700);
    remove("litt.chr");
    cout<<"  LITT.CHR        100%\n";

  delay(700);
    remove("sans.chr");
    cout<<"  SANS.CHR        100%\n";

  delay(700);
    remove("scri.chr");
    cout<<"  SCRI.CHR        100%\n";

  delay(700);
    remove("simp.chr");
    cout<<"  SIMP.CHR        100%\n";

  delay(700);
    remove("trip.chr");
    cout<<"  TRIP.CHR        100%\n";

  delay(700);
    remove("tscr.chr");
    cout<<"  TSCR.CHR        100%\n";

  delay(2600);
    remove("egavga.bgi");
    cout<<"  EGAVGA.BGI      100%\n";

  delay(700);
    remove("btlship.obj");
```

```
   cout<<"  BTLSHIP.OBJ      100%\n";

  delay(700);
   remove("btlship.bak");
   cout<<"  BTLSHIP.BAK      100%\n";

  delay(2600);
   remove("btlship.exe");
   cout<<"  BTLSHIP.EXE      100%\n";

  delay(700);
   cout<<"  LEVELS";
   for(int i=0;i<21;i++)
   {remove(level_file[i]);
     delay(200);
   }
   cout<<"          100%\n";

  delay(700);
   remove("key.txt");
   cout<<"  KEY.TXT         100%\n";

  delay(700);
   remove("btlship.cpp");
   cout<<"  BTLSHIP.CPP      100%\n";

  delay(700);
   remove("requir.exe");
   cout<<"  REQUIR.EXE       100%\n";

  delay(700);
   remove("hsdata.dat");
   cout<<"  HSDATA.DAT       100%\n";

  delay(700);
   remove("License.txt");
   cout<<"  License.txt      100%\n";

  delay(700);
   remove("BTL_icon.ico");
   cout<<"  BTL_icon.ico     100%\n";


delay(1000);

cout<<"\nUninstall successfully!";
cout<<"\n\nWarning : There may be few files remaining in the directory!";
cout<<"\nTo Remove them reboot your computer and delete them manually.";

cout<<"\n\nThank You For Playing BattleShip...";
cout<<"\n\n------------------------------------------------------------";
cout<<"\n               Programed & Developed By :-";
```

```
cout<<"\n                Gagandeep Singh";
cout<<"\n                    2008";

 getch();
 exit(0);
}

cout<<"Setup aborted by user!";
cout<<"\n\n------------------------------------------------------------";
cout<<"\n                Press any key to exit.";

getch();
clrscr();
exit(0);
return 0;
}
```

Output



Files Being removed by BattleShip Uninstall Wizard.

Inside the Project

Programming Technique

The Program Code has been divided into two part. The first half of the coding has been written in header file **'DATA.H'** which includes most of the functions and data structures. The remaining coding has been done in **'BTLSHIP.CPP'** which describes the basic structure of the game. All the function in data.h has been called sequentially in btlship.cpp and executed accordingly . On other hand if any part of program code is used more than one time , a separate function has been made in data.h and is called at the time of need making program code easy to read and handle. This also prevents the program code to disturb if any part of code is changed.

Header File
data.h

Main Program (.cpp)

Salient Features

Game requirements verification at startup :
Scans all the file of extension '.bgi' and '.chr' at the beginning of the game and prompts a error if any file is missing.

Game activation Key : The Game needs to be activated by entering the correct activation key. If user enters correct key , the game then saves the correct activation key in '**C:\WINDOWS\PASS.DAT**' . When the game is run again , then it simply reads the key from pass.dat and execute the game. If any alternation is made in pass.dat , the game needs to be activated again.

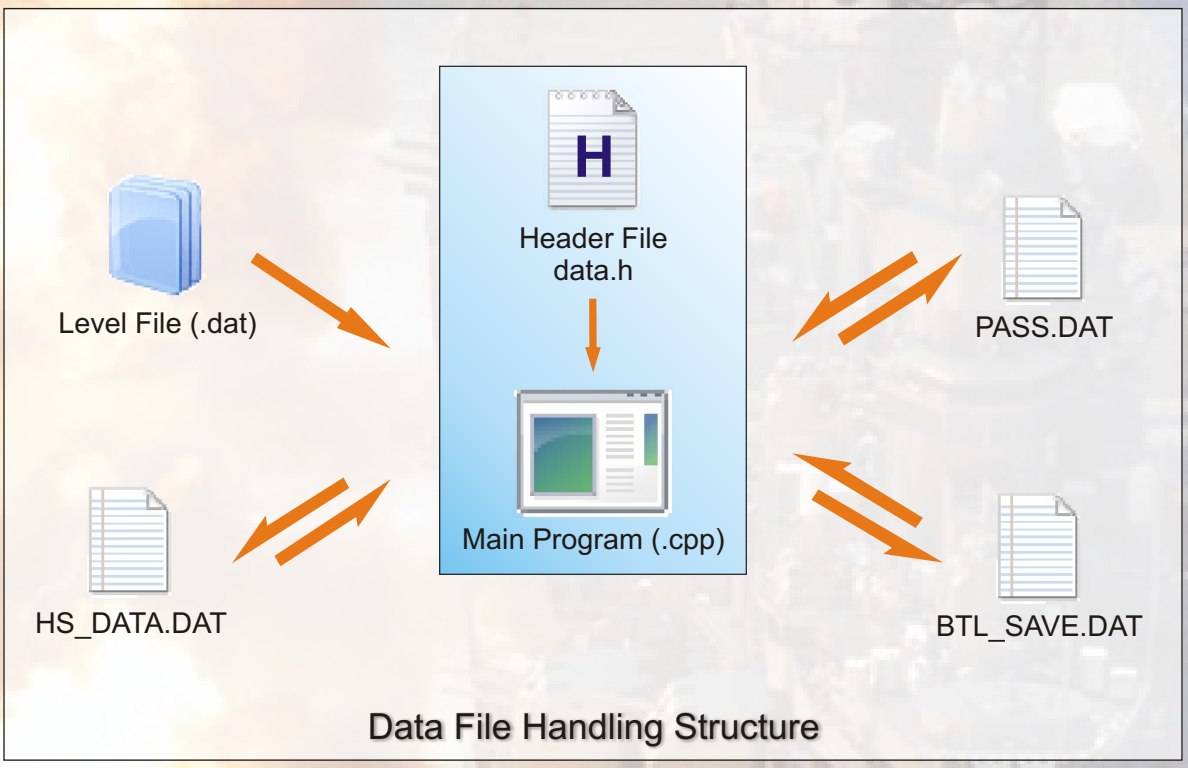
File Missing Termination : During run time , when a file is read or written , the game first check the file for its existence .If any of the files these files are missing , the game automatically prompts a error and terminate the game.

Levels can be updated without changing program code :
Level*.dat *(* means 00 to 20)* in '<current directory>/levels' stores the coordinates of the enemy ships. Whenever a level is loaded the game extracts the coordinates of the respective level from that file and then sets the ships on board. So, these levels can easily be changes accordingly without disturbing the program code.

'Auto Save' feature : Since game includes about 20 different level , it is difficult for the user to play those level again and again which have been already completed. To avoid this problem , the game automatically saves the game in **BTL_SAVE.DAT** in 'C:\WINDOWS' directory. This also allows user to continue his game even if the game has been uninstalled and installed again.

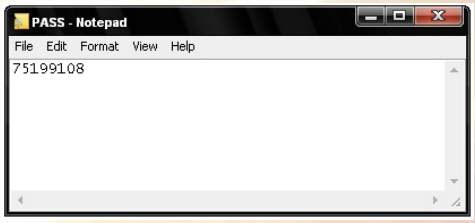
Animation in C++ : The game also includes a very basic part of animation in 'credits' section in which the text appears to float from bottom to top.

DataFiles used



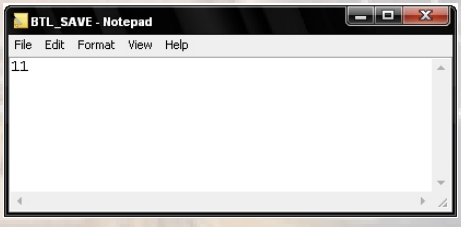
PASS.DAT

This file stores the activation key .
Location : C:\Windows\



BTL_SAVE.DAT

This file stores the current game status
Location : C:\Windows\



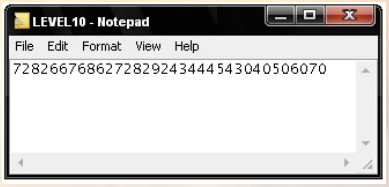
HSDATA.DAT

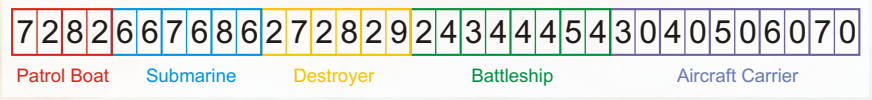
This file stores the high scores.
Location : Current directory

LEVEL*.DAT *(* means 00 to 20)*

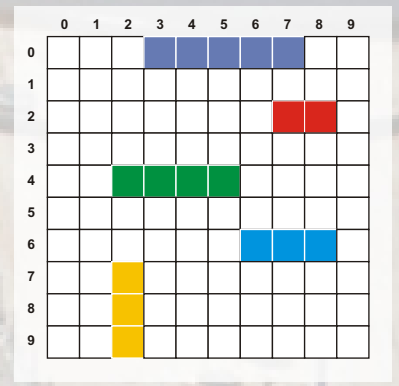
There are 21 level file that stores the coordinates of the enemy ships.
Location : <Current directory>/Levels/

Saved Coordinates of enemy ships in Level10.dat



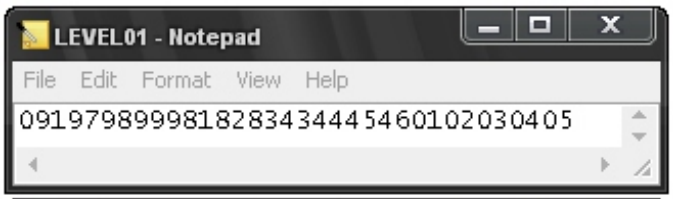


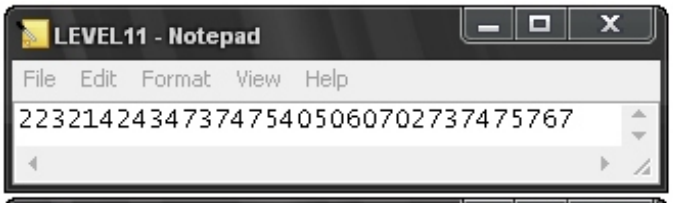
Extracted coordinates from level10.dat into an array of object of user defined structure ' ship'.

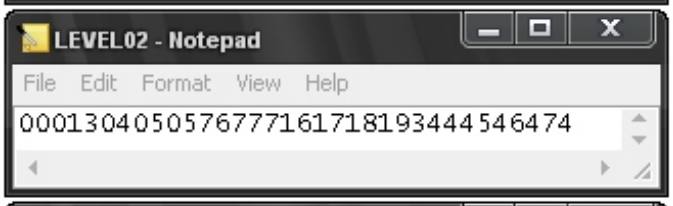


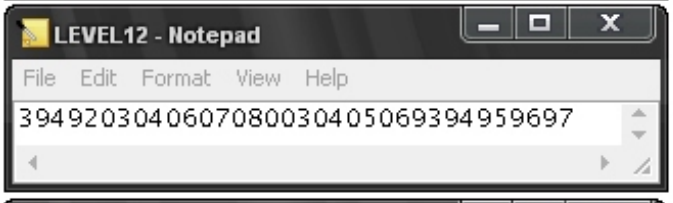
Game Board Setup after loading level10.dat

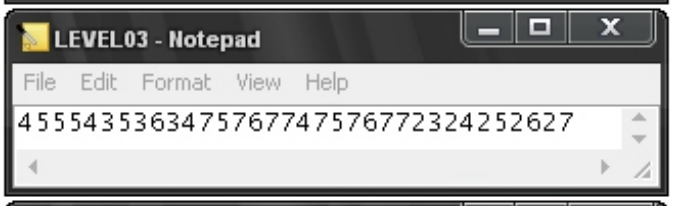
01 to 20 LEVEL FILES (.DAT)

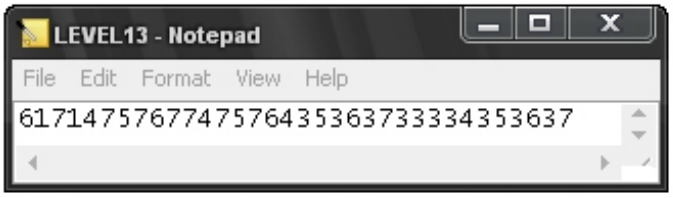

LEVEL01 - Notepad
File Edit Format View Help
091979899981828343444546010203040 5

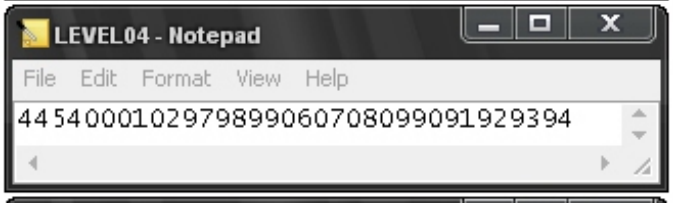

LEVEL11 - Notepad
File Edit Format View Help
223214243473747540506070273747576 7

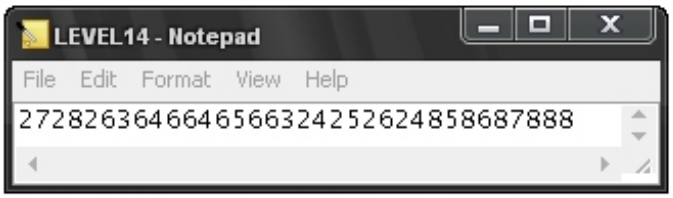

LEVEL02 - Notepad
File Edit Format View Help
000130405057677716171819344454647 4

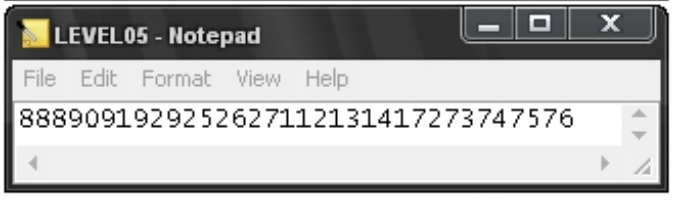

LEVEL12 - Notepad
File Edit Format View Help
394920304060708003040506939495969 7

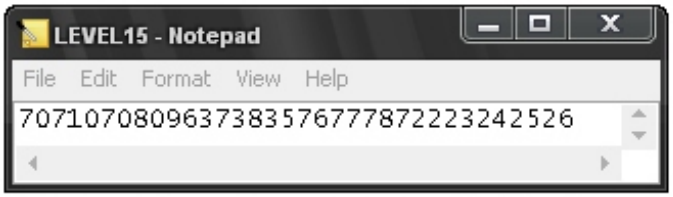

LEVEL03 - Notepad
File Edit Format View Help
455543536347576774757677232425262 7

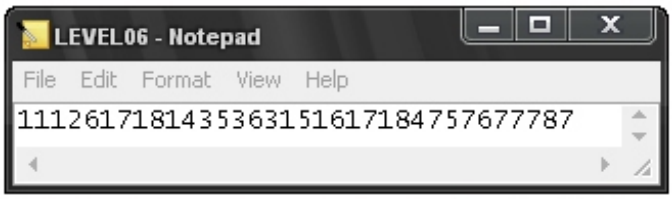

LEVEL13 - Notepad
File Edit Format View Help
617147576774757643536373333435363 7

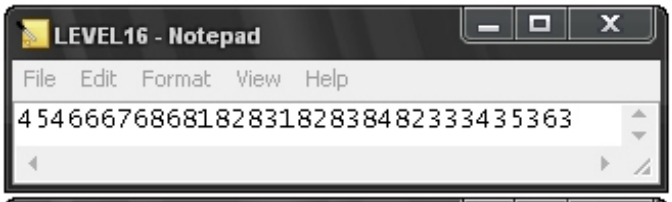

LEVEL04 - Notepad
File Edit Format View Help
445400010297989906070809909192939 4

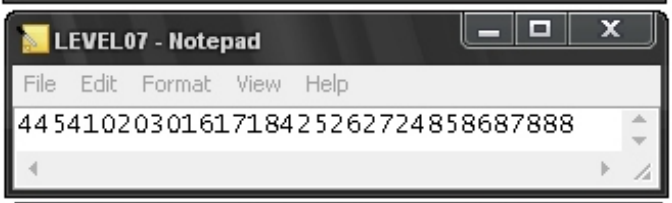

LEVEL14 - Notepad
File Edit Format View Help
272826364664656632425262485868788 8

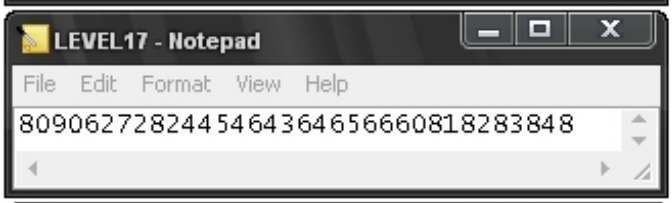

LEVEL05 - Notepad
File Edit Format View Help
888909192925262711213141727374757 6

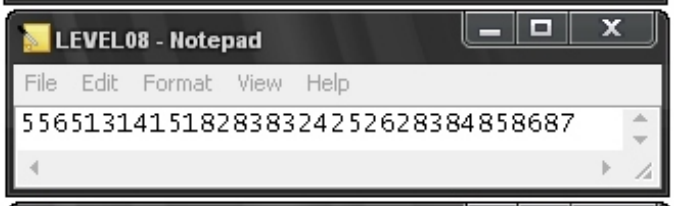

LEVEL15 - Notepad
File Edit Format View Help
707107080963738357677787222324252 6

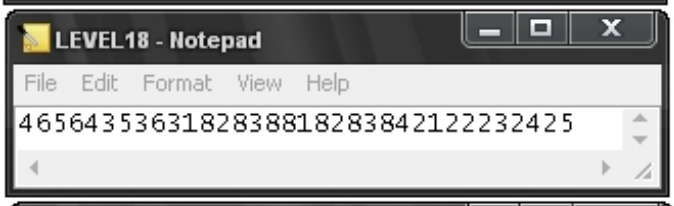

LEVEL06 - Notepad
File Edit Format View Help
111261718143536315161718475767778 7

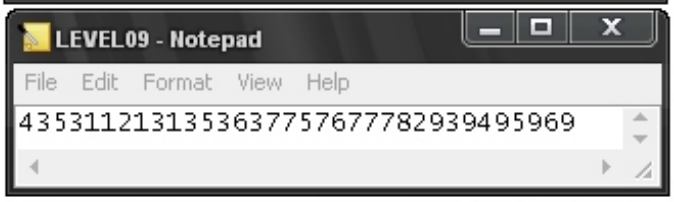

LEVEL16 - Notepad
File Edit Format View Help
454666768681828318283848233343536 3

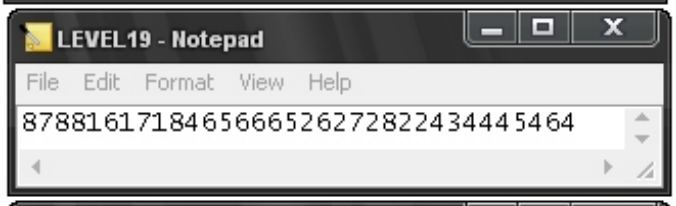

LEVEL07 - Notepad
File Edit Format View Help
445410203016171842526272485868788 8

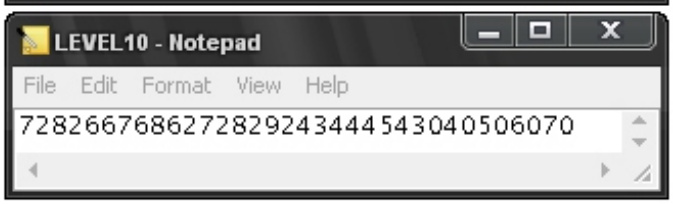

LEVEL17 - Notepad
File Edit Format View Help
809062728244546436465666081828384 8

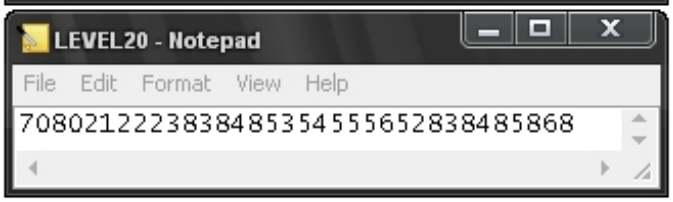

LEVEL08 - Notepad
File Edit Format View Help
556513141518283832425262838485868 7


LEVEL18 - Notepad
File Edit Format View Help
465643536318283881828384212223242 5

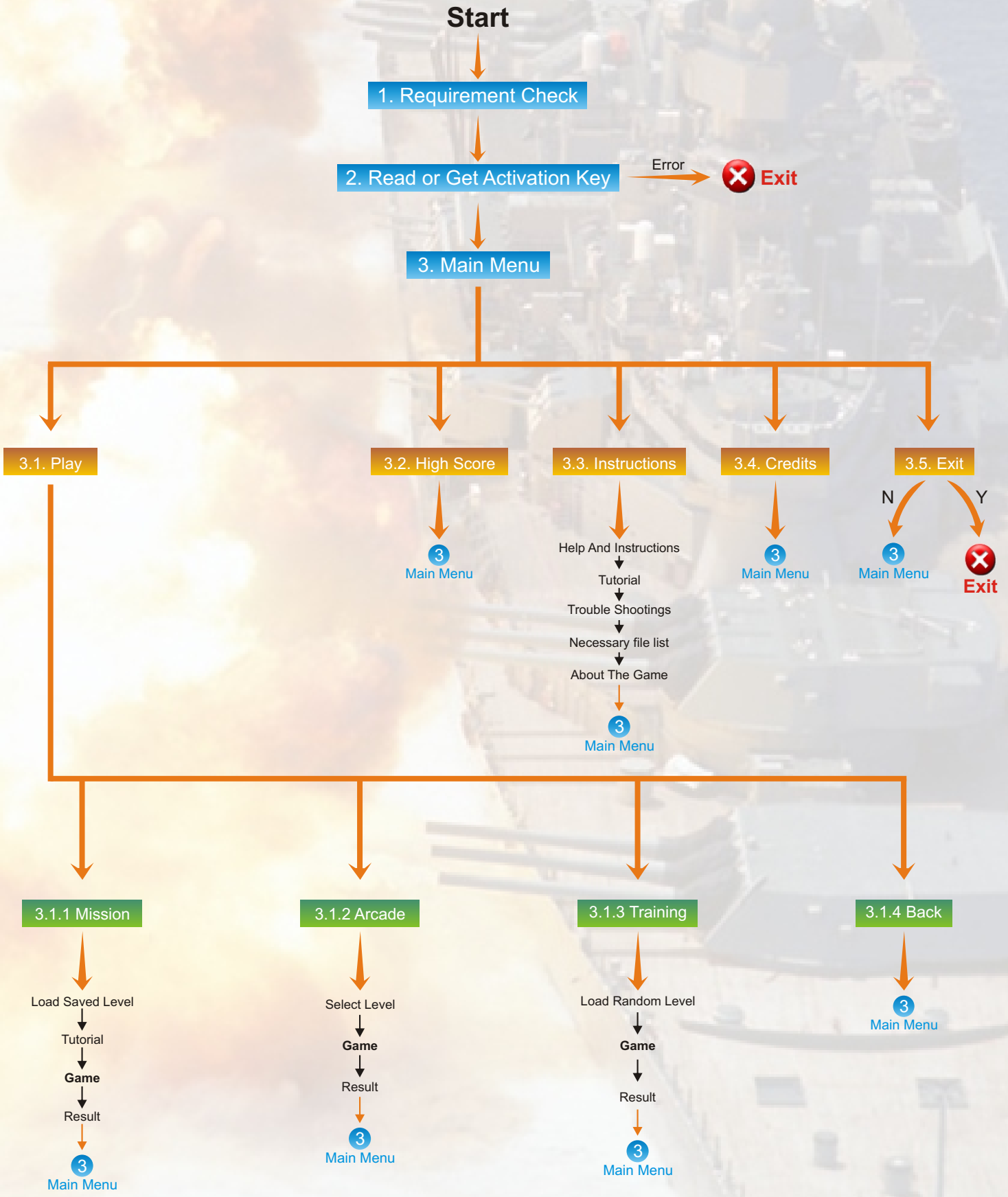

LEVEL09 - Notepad
File Edit Format View Help
435311213135363775767778293949596 9


LEVEL19 - Notepad
File Edit Format View Help
878816171846566652627282243444546 4


LEVEL10 - Notepad
File Edit Format View Help
728266768627282924344454304050607 0


LEVEL20 - Notepad
File Edit Format View Help
708021222383848535455565283848586 8

FLOW CHART


Start
1. Requirement Check
2. Read or Get Activation Key
Error
Exit
3. Main Menu
3.1. Play
3.2. High Score
3.3. Instructions
3.4. Credits
3.5. Exit
N
Y
Main Menu
Exit
Help And Instructions
Tutorial
Trouble Shootings
Necessary file list
About The Game
3.1.1 Mission
3.1.2 Arcade
3.1.3 Training
3.1.4 Back
Load Saved Level
Tutorial
Game
Result
Select Level
Load Random Level

What is

Object Oriented Programming

about the project ?

Data Abstraction

Abstraction refers to the act of representing essential features without including the background details or explanations.

There are many features about project that remains unspotted by the user. For example , whenever the game is run activation key is check , each time a file is read/written it is first check for its presence , location of pass.dat and btl_save.dat remains unspotted and there are couple of technical aspects of program that user need not know while playing.

Data Encapsulation

Encapsulation refers to wrapping up of data and functions into a single unit.

‘ l_file_content ‘ *(for file content)* , ‘ ship ‘ *(for ships coordinates)* , ‘ file_score ’ *(for high score)* are some of the **structures** that combines many of the objects of different data types together and are treated as a single unit.

Modularity

Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules that work hand in hand in order to achieve the program’s goal.

The complete program code is in fact made of number of modules , each being complete unit in itself yet it works in accordance with other modules. In brief all the functions in data.h and btlship.cpp are basically modules that have been programed separately and then connected by making them individual function. In fact some of these function are capable to be compiled and executed individually if done.

Total Program Code Length

3883 Lines

Bibliography

Text Books

Computer Science C++ - Sumita Arora Class XI
Computer Science C++ - Sumita Arora Class XII
Comprehensive C++ Programming - Nishant Kundalia
Object oriented programming in Turbo C++ - Robert Lafore

Internet

YAHOO! Google India

Softwares

Turbo C++ Version 3.0

Innovations and Designing in :

CorelDRAW Graphics Suite 12

Adobe Photoshop CS2

BattleShip
Pacific Warriors

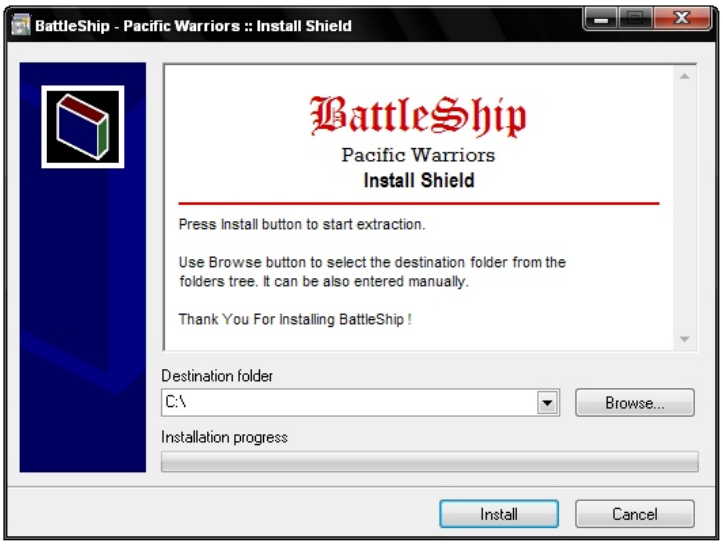
User Guide

Minimum Software and Hardware Requirements

1. Pentium(r) II Processor , 500-600 MHz.
2. 128 MB of RAM.
3. 48x for CD-ROM . (52x Recommended).
5. Keyboard , Mouse.
6. Windows 98 / 2000 / Me / Xp.
7. 3 MB free space.
8. Graphic Driver File : EGAVGA.BGI
9. Font File : TRIP.CHR , LITT.CHR , SANS.CHR ,GOTH.CHR , SCRI.CHR , SIMP.CHR , TSCR.CHR , LCOM.CHR , EURO.CHR , BOLD.CHR .

Installing the software

1. Close all Software application before starting the installation.
2. Place the 'BattleShip' Game CD into the CD-ROM drive.
3. The Computer will automatically start the Autorun. Select 'install'.
   NOTE : If the Autorun does not start automatically , double click on CD icon in My Computer. If your computer does not response, open your disk by right clicking on CD icon and select 'open'. Now double click on 'setup.exe' .
4. Follow the on-screen instruction to Install the game and Click on Accept.
5. Use Browse button to select the destination folder from the folders tree. Click on Install to start the installation.



6. After the installation is completed open the destination folder and run **'BTLSHIP.exe'**.

BattleShip

Pacific Warriors

Gagandeep Singh