University of Dhaka

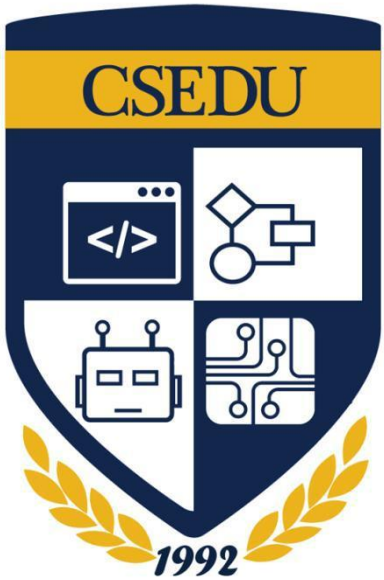
Department of Computer Science and Engineering
CSE – 1211

Lab Project: Typing Genius

Submitted to:

Md. Imran Bin Azad
Lecturer

Hasnain Heickel
Lecturer



Submitted by:

Fahim Faisal
[FH-41]
Reyad Hossain
[AE-17]
Department of Computer Science and Engineering
Session (2016-17)

Date-of-Submission: 9th November, 2017

Table of contents:

Serial	Topic	Page number
01	Introduction	4
02	Project introduction	4
03	Game outline	4
04	Game objective	5
05	Challenges for the gamer	5
06	Game features	5
07	Source code properties	8
08	Graphical interface of the game	9
09	Game structure	10
10	How the levels work	11
11	Challenges in coding	12
12	Functions used in source code	12
13	Overview	13
14	The source code	14
15	Special thanks to	27
16	Reference	27

Introduction:

The project is intended to make the students familiar with some application of basic C programming using Borland Graphics Interface (BGI). Implementation of the students' theoretical knowledge of C language in some real life scenario was the objective of this project.

Project introduction:

The project is a typing game. In the first level of the game the player has to type the green words from a given set of words. In the second level of the game the player has to type the green letters of each word from a given set of words. In both levels the player has to finish typing before time runs out. The game is named "Typing Genius". We chose this name because it goes with the theme of the game.

Game outline:

This is a simple game where the player has to type continuously and correctly as long as he can in order to continue playing.

There are two different level in this game.

In the first level of the game the player has to type the green words from a given set of words. He can also type the yellow words for bonus points. But he can also skip it if he wishes. However he has to type the words sequentially and he must finish it in time. After typing each word he has to type a space which ensures that he has submitted that word. Submitting any red word will finish the game. Scoring 100 points will make the player eligible for 2nd level.

In the second level of the game the player has to type the green letters of each word from a given set of words in a given amount of time. After typing all green letters of each word he has to type a space. In this level he is not allowed to type any letter other than the green ones.

In both levels the duration of time will be one second shorter after each round.

In the first level for each word typed correctly the point equal to the length of the world will be given.
In the second level a single point will be given for each letter of a correctly typed word.

In the first level submitting a red word will finish the game and in the second level submitting a word with a red or yellow letter in it will finish the game.

Game objective:

The Game was designed with the aim to keep the player busy always. Before starting on the project I had thought about the conception of "great games" and "boring games" and finally decided that a game that requires the player to be alert all the time is less likely to be considered as "boring".

So, the main objective of the game was to provide its user an entertaining experience, and also to make the game as addictive as possible.

High quality interactive menus, submenus as well as clean and flicker-free game play environments were created with great care and efforts to achieve that certain goal.

Challenges for the gamer:

In this game, the gamer will have to master the skill of fast typing and color detecting to achieve a high score. Three types of color have been used in this game to make the game more interesting. In addition to that as the game progresses the gamer actually have to concentrate very hard as the time gets shorter after each round. This feature makes this level incredibly hard.

Game features:

Main features:

(i) Almost zero-flicker game play:

Though there were some problems, but we tried our best and here we stand with almost zero-flicker game play.

(ii) Interactive menus:

This game has several interactive menus which are very easy to use, and also greatly eye catching. This feature adds to the charm of this game by creating great first

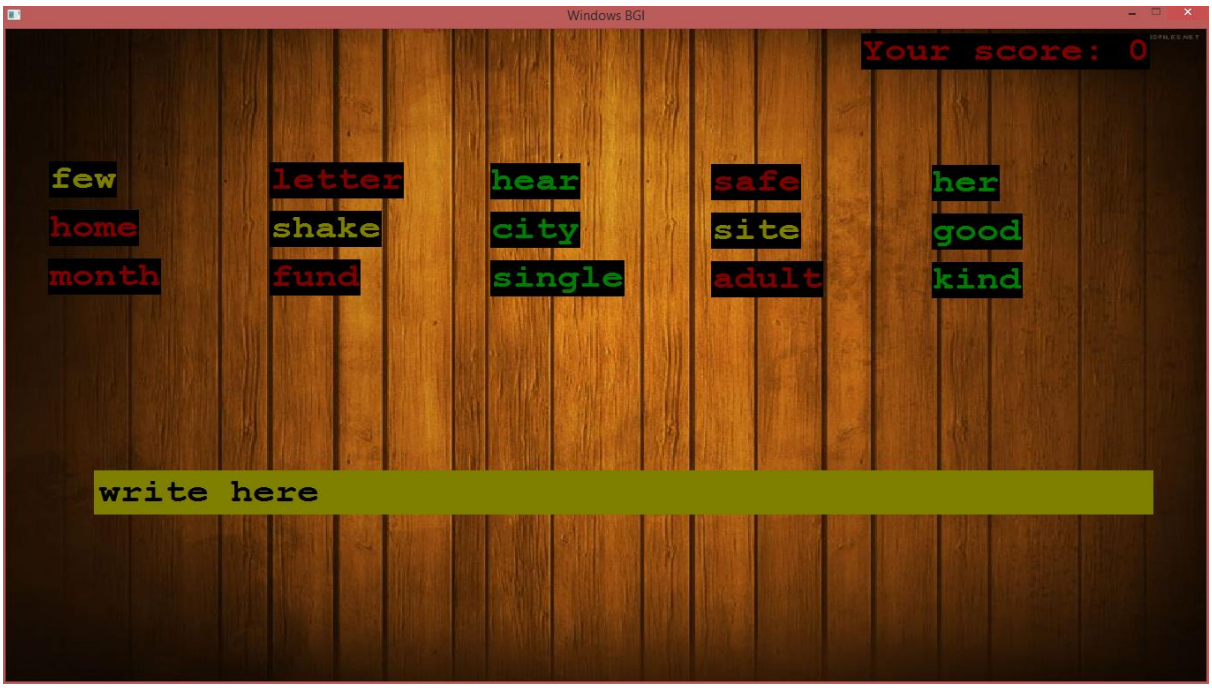
impressions. The menus and the submenus were created with high quality images [3] and appropriate keyboard and mouse functions.

(iii) A professional look in the interface:

With the help of my limited image processing skill and coding skill, I tried to create agood looking graphics interface for my game.

(iv) Different levels of game play:

The game comes with two different levels; each is different in the level of difficulty. This feature adds to the variation of the game.



Additional features:

I. Main menu:

The main menu is like a dashboard that gives the user access to any part of the program.

II. High Score:

The game also has a High Score option that can be accessed through the main menu. The game stores the top three scores from each level and shows them in the High Score.

III. Instructions:

The game has an option which show you the instructions on how to play the game.

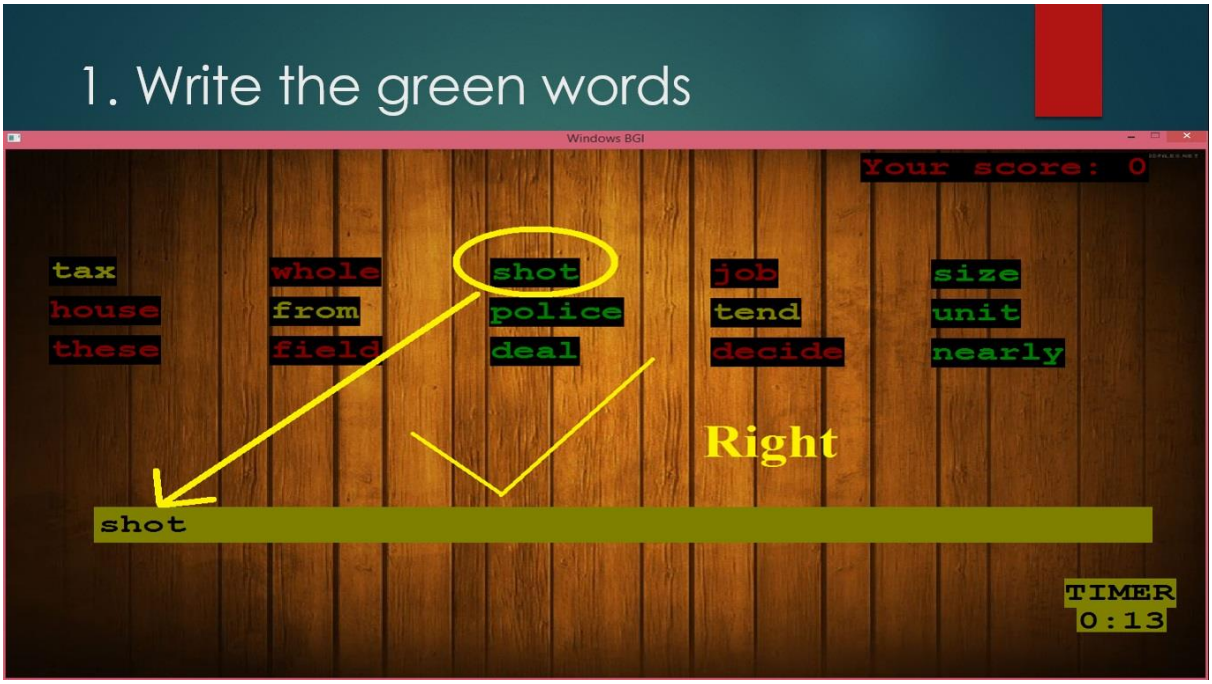
(iv) Exit option:

The game can be terminated at anytime using the EXIT option, which is located at the main menu.

Bonus features:

(i) Use of pictures in instructions:

In the Instruction option several images have been included to give the gamer a better understanding of the game.



(ii) Change of appearance:

The game changes its appearance several times throughout the game. The menu bar, gameplay, leader boards, instructions all of these options have different appearances.

Source Code properties:

(i) The source was written by following an efficient modular programming system, dividing the whole code in different separate parts. This method was encouraged by our teachers and it made our tasks a lot easier while combining all the features and joining the whole project together.

(ii) I tried to use as many different types of programming tools as possible, so that it covers the maximum amount of what we have been taught in C language. The coding part of this project contains-

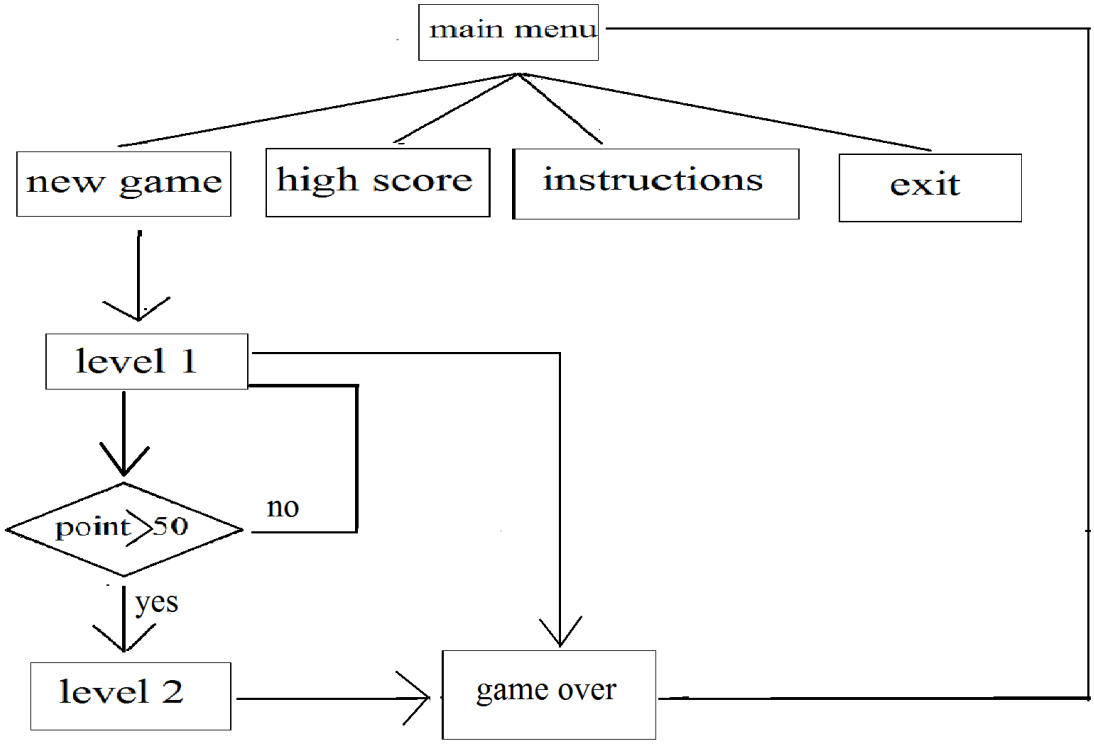
a. Basic input output
b. File input output
c. Conditional logic
d. For loop and While loop
e. Single arrays
f. String handling
g. Mouse operations
h. Keyboard operations
i. Structures
j. Sorting
k. Reading image
l. Graphics operations.

And so on.

Graphical interface of the game:

The main menu:

The game structure:



The game flow is shown in the figure drawn above. After opening the program, it takes the usertothe main menu. From the main menu, the user can take one of four possible options. These options are the "New Game" submenu, the "High score" screen, the "Instruction submenu" and the "Exit"option.

In the new game submenu will take the player directly to level 1. Scoring 50 points in level 1theplayer can have access to level 2.

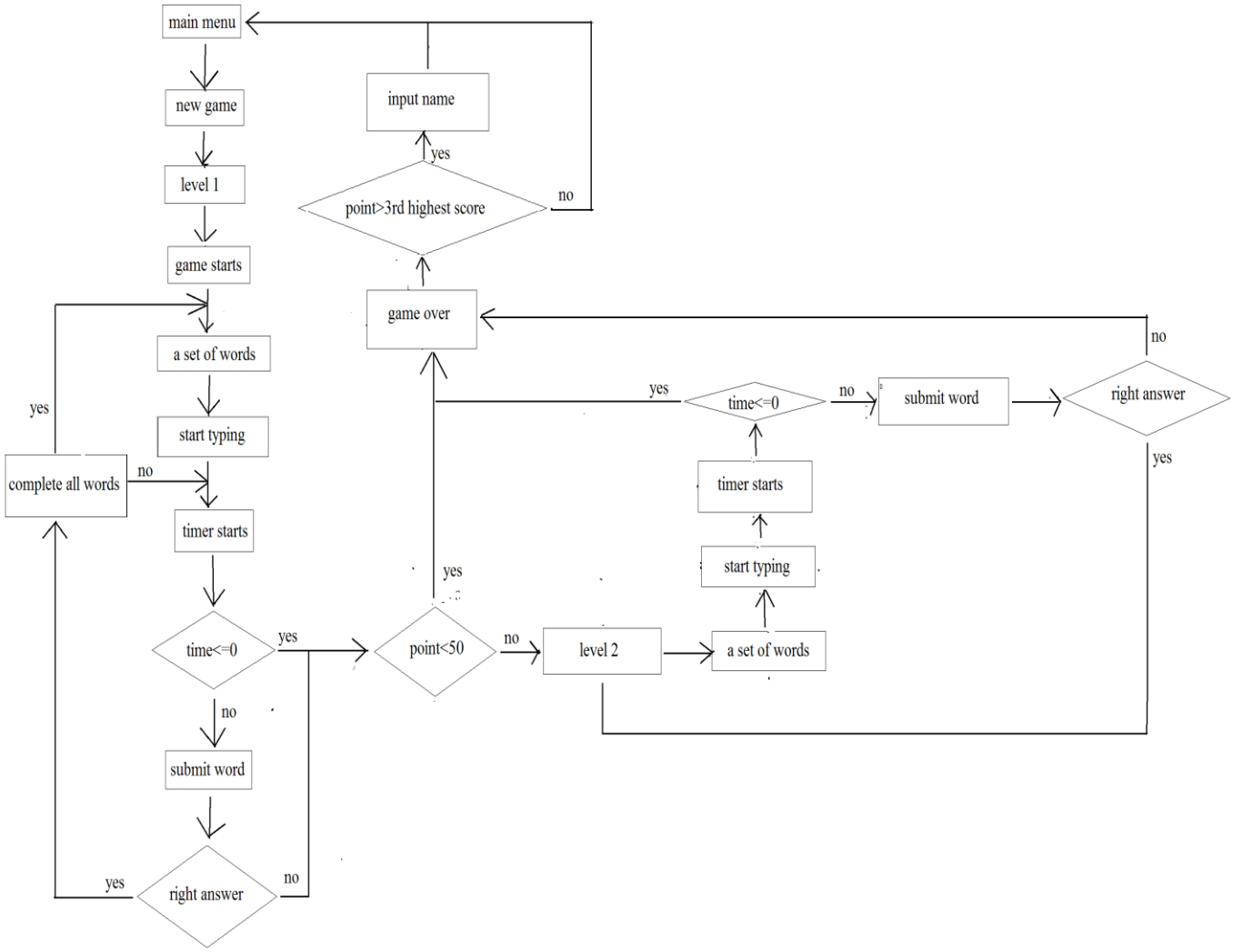
The High score feature shows the top 3 scores and the scorers name on the screen.

The instruction submenu gives a complete understanding about how to play the game. It alsoincludes images to give the player a better understanding of the game.

The Exit submenu will take you out of the game.

How the gameplay works:

The logic steps of the game play operation are not very complex. The basic idea is shown below with the help of a flow diagram.



Challenges in coding:

A huge number of words:

We needed a huge number of words for this game. So we searched for 1000 common words. Then we copied 1000 words from a sight to a file. Then we read those words from that file.

Generating random words everytime:

We needed to generate random word everytime. But we could not use rand() function because it gives the same value everytime the game starts. So we used srand() function.

Timer:

It was a tough work calculating the timer. We used clocks_per_second() function and clock() function to get the pc time at the beginning of the game and after that we continuously took the pc time every moment. From that we calculated the elapsed time.

High score:

For the high score option we needed to store the high scores and read it every time the game started. So we used structure and also printed the high scores in a file after each change in the high score structure.

A better understanding of the game:

To give the player a better understanding of the game we had to use several images in the instructions option.

Handling bugs:

The interactive menu is done by calling blocks. A certain amount of blocks are made to make this game easier to code, such as main_menu, instructions, high score,exit etc.

The functions used in the source code:

User defined functions:

- print_highscore()
- instruction()
- high_score_function()
- upoin()
- level1()
- level2()
- gameover()

Created Block:

- Main_menu:
- New game:
- High score:
- Instructions:
- Exit:

Functions from "graphics.hpp":

1. setviewport()
2. clearviewport()
3. cleardevice()
4. settextstyle()
5. outtextxy()
6. setbkcolor()
7. getmaxwidth()
8. getmaxheight()
9. initwindow()
10. setcurrentwindow()
11. floodfill()
12. readimagefile()
13. setcolor()
14. kbhit()
15. getch()
16. delay()
17. initgraph()
18. ismouseclick(WM_LBUTTONDOWN)
19. ismouseclick(WM_MOUSEOVER)
20. getmouseclick()

Overview and conclusion:

The project was a very good way to encourage young students to develop their skills and make something useful with their knowledge. This project is a perfect example of all the exceptionalthings that can be done with some basic C programming.

This project will also encourage young minds who want to be a game developer in the future. The project promotes originality and creativity and the skill of working in a team.

The project also summarizes the basic tools of C language as they had to be implemented while making the game.

The source code:

The "main.cpp" file:

```
#include <bits/stdc++.h>
#include <graphics.h>
int rem;//time remaining for both level1 level2
using namespace std;

int maxx,mayy,point=0;//maximum  height, width, point
FILE *input= fopen("input.txt","r");//  file of words

char poin[100];
char abc[100];
char s[1005][15];

struct high_scores
{

   char name[100];
   int scor;

};

struct high_scores hs[15];


void print_data()//printing high score in file
{
   FILE *output= fopen("score.txt","w");
   int i;
   for(i=0; i<3; i++)
   {
      fprintf(output,"%s\n",hs[i].name);
      printf("%s\n",hs[i].name);
      fprintf(output,"%d\n",hs[i].scor);
      printf("%d\n",hs[i].scor);
   }
   fclose(output);

}
void high_score_function(int score)
{

   int i,j,pos=0;
   char in[100];
   char fahim;
   setbkcolor(3);
   readimagefile("game.jpg",0,0,maxx,mayy);
   settextstyle(10,0,5);
   setcolor(0);
   outtextxy(500,200,"Write  your name");//name of high scorer
   setviewport(500,400,900,450,1);
   clearviewport();
```

```
   while(1)
   {
      if(kbhit())//keyboard  input
      {
         in[pos]=getch();
         while(kbhit())
         {
            fahim=getch();
         }
         if(in[pos]==8)//backspace  handling
         {
            if(pos==0) continue;
            pos--;
            in[pos]='\0';
         }
         else
         {
            if(in[pos]==' '||in[pos]==13)//submitting  word by enter or space
            {
               in[pos]='\0';
               break;
            }
            else pos++;
            in[pos]='\0';
         }
         setviewport(200,400,600,450,1);
         clearviewport();
         outtextxy(5,5,in);//showing  word
      }
   }
   if (score>=hs[0].scor)//sorting
   {
      strcpy(hs[2].name,hs[1].name);
      hs[2].scor=hs[1].scor;
      strcpy(hs[1].name,hs[0].name);
      hs[1].scor=hs[0].scor;
      strcpy(hs[0].name,in);
      hs[0].scor=score;
   }
   else if(score>=hs[1].scor)
   {
      strcpy(hs[2].name,hs[1].name);
      hs[2].scor=hs[1].scor;
      strcpy(hs[1].name,in);
      hs[1].scor=score;
   }
   else if(score>=hs[2].scor)
   {
      strcpy(hs[2].name,in);
      hs[2].scor=score;
   }
   setviewport(0,0,maxx,mayy,1);
   cleardevice();
   return;
}

void upoin()//updating  point
{
```

```
    char pnt[100];
    int i=0;
    int x=point;
    while(x)//converting integer to string
    {
        pnt[i++]=x%10+'0';
        x/=10;
    }
    if(i==0)
        pnt[0]='0',i++;
    pnt[i]='\0';
    for(int j=0; j<i/2; j++)
    {
        swap(pnt[j],pnt[i-j-1]);
    }
    strcpy(poin,"Your score: ");
    strcat(poin,pnt);
    setviewport(950,0,getmaxx(),50,1);
    setcolor(RED);
    outtextxy(20,5,poin);
    //setviewport(0,0,maxx,mayy,1);
}

void gameover()
{
    setviewport(0,0,maxx,mayy,1);
    cleardevice();
    settextstyle(10,0,7);
    char go[20];
    strcpy(go,"Game Over");
    setcolor(0);
    readimagefile("game.jpg",0,0,maxx,mayy);
    outtextxy(350,200,go);
    outtextxy(250,350,poin);
    delay(2000);
    return;
    //getch();
    //closegraph();
}

bool cmp(char s1[],char s2[],int x,int y)//comparing subsequence
{
    int len1=strlen(s1);
    int len2=strlen(s2);
    if(y==len2&&x==len1)
        return 1;
    if(x==len1)
        return 0;
    if(s1[x]==s2[y])
        return cmp(s1,s2,x+1,y+1);
    else
        return cmp(s1,s2,x+1,y);
}

void print()//instructions
{
    setviewport(0,0,maxx,mayy,1);
    cleardevice();
```

```
    readimagefile("New folder/Screenshot (33).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (34).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (35).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (36).jpg",0,0,maxx,mayy);
    getch();
    readimagefile("New folder/Screenshot (38).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (37).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (39).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (40).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (41).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (42).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (43).jpg",0,0,maxx,mayy);
    getch();
    getch();
    readimagefile("New folder/Screenshot (44).jpg",0,0,maxx,mayy);
    getch();
    getch();
}

int level1()
{
    settextstyle(10,0,5);
    int rem=25;
    maxx=getmaxwidth(),mayy=getmaxheight();
    int window_1;
    window_1=initwindow(maxx,mayy);//maximizing  window
    setcurrentwindow(window_1);
    char ck[17][20];//for keeping random words
    char in[15];//input  from player
    for(int i=0; i<712; i++)//input  from file
    {
        fscanf(input,"%s",s[i]);
    }
    int a[17],x;//status of a word and x for random number
    setbkcolor(0);
    cleardevice();
    srand(time(NULL));//for random, random function
    for(int i=0; ; i++)
    {
        setviewport(0,0,maxx,mayy,1);
```

```
setbkcolor(0);
readimagefile("k.jpg",0,0,maxx,mayy);//background  image
//setbkcolor(WHITE);
for(int j=0; j<15; j++)//taking  15 words
{
   if(j==14)//making  combination  of color
   {
      a[j]=1;
      setcolor(GREEN);
   }
   else
   {
      a[j]=rand()%3;
      if(a[j]==1)
         setcolor(GREEN);
      else if(a[j]==2)
         setcolor(BROWN);
      else
         setcolor(RED);
   }
   x=rand()%712;
   strcpy(ck[j],s[x]);//keeping random word for compare
   int b=(50*(j/5));
   settextstyle(10,0,5);
   outtextxy(50+(j%5)*250,j+150+b,s[x]);//showing  words
}
if(point>50)
{
   outtextxy(100,600,"Level  2 unlocked.");
   outtextxy(100,670,"You  may progress to level 2 after level 1");
}
char TIME[100];
int time2,time,time1;
int pos=0,k=0,j=0,b;
upoin();
setbkcolor(6);
setviewport(100,500,1300,550,1);
clearviewport();
setcolor(0);
if(i==0)
{
   k=1;
   outtextxy(5,5,"write  here");//waiting  for start typing
   in[0]=getch();
   clearviewport();
   in[1]=NULL;
   outtextxy(5,5,in);
}
time1=clock();
while(pos<15)
{
   while(pos<15)
   {
      if(kbhit())//detecting  keyboard click
      {
         in[k]=getch();
         if(in[k]==' '||in[k]==13)
         {
```

```
setviewport(100,500,1300,550,1);
clearviewport();
setviewport(0,0,maxx,mayy,1);
if(point>50)
{
   outtextxy(100,600,"Level  2 unlocked.");
   outtextxy(100,670,"You  may progress to level 2 completing  level 1");
}
in[k]=NULL;
while(pos<15)
{
   if(a[pos]==0)//red word comparing
   {
      b=50*(j/5);
      outtextxy(50+(j%5)*250,j+150+b,ck[pos]);//changing  color of typed words
      j++;
      if(strcmp(in,ck[pos])==0)
      {
         //gameover();
         return time;
      }
      pos++;
   }
   if(a[pos]==2)//yellow  word comparing
   {
      b=50*(j/5);
      outtextxy(50+(j%5)*250,j+150+b,ck[pos]);
      j++;
      if(strcmp(in,ck[pos])==0)
      {
         point+=strlen(in);
         upoin();
         pos++;
         k=0;
         break;
      }
      pos++;
      if(pos==15)
      {
         k=0;
         break;
      }
   }
   if(a[pos]==1)//green word comparing
   {
      if(strcmp(in,ck[pos])!=0)
      {
         //gameover();
         return time;
      }
      else
      {
         b=50*(j/5);
         outtextxy(50+(j%5)*250,j+150+b,ck[pos]);//changing  color of typed words
         j++;
         point+=strlen(in);
         upoin();
         pos++;
```

```
                    k=0;
                    break;
                }
            }
        }
        upoin();
        k=0;
        break;
    }
    if(pos>=15)
    {
        k=0;
        break;
    }
    if(in[k]==8)//handling  backspace
    {
        in[k]='\0';
        k-=2;
    }
    in[k+1]='\0';
    setviewport(100,500,1300,550,1);
    clearviewport();
    settextstyle(10,0,5);
    printf("%s\n",in);
    setcolor(0);
    outtextxy(5,5,in);//showing  input
    //upoin();
    k++;
}
setbkcolor(6);
time2=clock();
if(time!=((time2-time1)/CLOCKS_PER_SEC))//handling  time
{
    setviewport(1200, 600, 1500, 675, 1);
    time =rem- ((time2-time1)/CLOCKS_PER_SEC);
    if(time<=0)
    {
        clearviewport();
        return 0;
    }
    else
    {
        setviewport(1200, 600, 1500, 675, 1);
        sprintf(TIME,"%d:%d",  time/60,time%60);
        if(time==9)
            clearviewport();
        setcolor(0);
        outtextxy(0,0,"TIMER");
        outtextxy(15,40,TIME);

    }
}
}
}
setviewport(0,0,maxx,mayy,1);
cleardevice();
rem--;
upoin();
```

```
    }
    //delay(200);
    //return time;
}

void level2()
{
    rem=20;
    settextstyle(10,0,7);
    srand(time(NULL));
    char in[50];
    setviewport(0,0,maxx,mayy,1);
    int a[7],x;
    clearviewport();
    int status[105];
    for(int i=0; ; i++)
    {
        char alu[100],balu[100],khalu[100];
        alu[0]='\0';
        balu[0]='\0';
        khalu[0]='\0';
        for(int j=0; j<5; j++)
        {
            x=rand()%712;
            strcat(alu,s[x]);
            //settextstyle(1,0,2);
            //outtextxy(i+50+j*200,j+50,s[x]);
            strcat(alu," ");
            //fprintf(output,"%d\n",i+j);
        }
        strcat(alu,"\0");
        printf("%s\n",alu);
        int len=strlen(alu);
        for(int i=0; i<len; i++)//handling  status
        {
            if(alu[i]==' ') continue;
            status[i]=rand()%3;
            char extra[2];
            extra[0]=alu[i];
            extra[1]='\0';
            if(status[i]==0) strcat(balu,extra);//for comparing  green words
            if(status[i]==0||status[i]==2)  strcat(khalu,extra);//for comparing both yellow  and green words
        }
        printf("%s\n",balu);
        int kk=0;
        setviewport(0,0,getmaxx(),getmaxy(),1);
        readimagefile("k.jpg",0,0,getmaxx(),getmaxy());
        //upoin();
        setbkcolor(0);
        for(int i=0; i<len; i++)//showing  words
        {
            if(status[i]==0) setcolor(GREEN);
            if(status[i]==1) setcolor(RED);
            if(status[i]==2) setcolor(YELLOW);
            if(alu[i]==' ')
            {
                kk+=3;
                continue;
```

```
    }
    char extra[2];
    extra[0]=alu[i];
    extra[1]='\0';
    settextstyle(10,0,5);
    setviewport(0,0,getmaxx(),getmaxy(),1);
    outtextxy(50+kk*24,100,extra);
    kk++;

}
int pos=0;
char fahim;
char TIME[100];
int time2,time,time1;
upoin();
setbkcolor(6);
setviewport(100,500,1300,550,1);
clearviewport();
setcolor(0);
if(i==0)
{
    pos=1;
    outtextxy(5,5,"write here");
    in[0]=getch();
    clearviewport();
    in[1]=NULL;
    outtextxy(5,5,in);
}
time1=clock();

while(1)
{
    if(kbhit())
    {
        in[pos]=getch();
        if(in[pos]==8)
        {
            if(pos==0) continue;
            pos--;
            in[pos]='\0';
        }
        else
        {
            if(in[pos]==' '||in[pos]==13)
            {
                in[pos]='\0';
                break;
            }
            else pos++;
            in[pos]='\0';
        }
        setviewport(100,500,1300,550,1);
        clearviewport();
        //setcolor(0);
        //rectangle(1,1,1199,50);
        //setcolor(BROWN);
        //floodfill(5,5,0);
        setbkcolor(6);
```

```
            setcolor(0);
            outtextxy(5,5,in);
          }
          time2=clock();
          if(time!=(int)((time2-time1)/CLOCKS_PER_SEC))
          {
            setviewport(1200, 600, 1500, 670, 1);
            time =rem- (int)((time2-time1)/CLOCKS_PER_SEC);
            if(time<=0)
            {
              clearviewport();
              break;
            }
            else
            {
              sprintf(TIME,"%d:%d",  time/60,time%60);
              if(time==9)
                 clearviewport();
              setcolor(0);
              outtextxy(0,0,"TIMER");
              outtextxy(15,40,TIME);

            }
          }
        }
        if(cmp(in,balu,0,0)==0||cmp(khalu,in,0,0)==0)//comparing  subsequence
        {
          rem--;
          setviewport(0,0,getmaxx(),getmaxy(),1);
          setbkcolor(RED);
          cleardevice();
          settextstyle(10,0,8);
          setcolor(4);
          if(time>0)
          {
            setcolor(0);
            outtextxy(400,300,"WRONG  ANSWER");
            delay(2000);
            //getch();
            return ;
          }
          else
          {
            setcolor(0);
            outtextxy(450,300,"TIME  OVER");
            delay(2000);
            return ;
          }
        }
        setviewport(0,0,maxx,mayy,1);
        cleardevice();
        point+=(strlen(in)+time*3);
        rem--;
        upoin();
      }
     setviewport(0,0,maxx,mayy,1);
     return;
}
```

```
int main(void)
{
   int gdriver = DETECT, gmode, errorcode;
   initgraph(&gdriver,&gmode,"d:\\tc\\bgi");
   maxx=getmaxwidth(),mayy=getmaxheight();
   int window_1;
   window_1=initwindow(maxx,mayy);
   setcurrentwindow(window_1);
   int ijk;
   FILE *inputt= fopen("score.txt","r");//taking high scores from file
   for(ijk=0; ijk<3; ijk++)
   {
      fscanf(inputt," %[^\n]",hs[ijk].name);
      fscanf(inputt,"%d",&hs[ijk].scor);
   }
   fclose(inputt);
   bool gg=0;
   int x,y;
   readimagefile("best.jpg",0,0,maxx,mayy);
   delay(3000);
   while(1)
   {
      readimagefile("normal11.jpg",0,0,maxx,mayy);
      int tt=0;
      while(1)
      {
         //cleardevice();
         //readimagefile("normal11.jpg",0,0,maxx,mayy);
         if(ismouseclick(WM_MOUSEMOVE))
         {
            getmouseclick(WM_MOUSEMOVE,x,y);
            if(x>=105 && x<=524 && y>=200 && y<=268)
            {
               readimagefile("new.jpg",0,0,maxx,mayy);
            }
            else if(x>=113 && x<=564 && y>=446 && y<=519)
            {
               readimagefile("Istructions11.jpg",0,0,maxx,mayy);
            }
            else if(x>=125 && x<=529 && y>=320 && y<=394)
            {
               readimagefile("high score11.jpg",0,0,maxx,mayy);
            }
            else if(x>267&&x<411&&y>580&&y<648)
               readimagefile("exit.jpg",0,0,maxx,mayy);
            else
            {
               readimagefile("normal11.jpg",0,0,maxx,mayy);
            }
         }
         else if(ismouseclick(WM_LBUTTONDOWN))
         {
            getmouseclick(WM_LBUTTONDOWN,x,y);
            if(x>=105 && x<=524 && y>=200 && y<=268)
               break;
            else if(x>=113 && x<=564 && y>=446 && y<=519)
            {
```

```
            print();
            cleardevice();
            readimagefile("normal11.jpg",0,0,maxx,mayy);
        }
        else if(x>=125  && x<=529 && y>=320 && y<=394)
        {
            setbkcolor(3);
            setbkcolor(3);
            cleardevice();
            readimagefile("high.jpg",0,0,maxx,mayy);
            setcolor(0);
            settextstyle(10,0,6);
            outtextxy(340,100,"1.");
            outtextxy(400,100,hs[0].name);
            sprintf(abc,"%d",hs[0].scor);
            outtextxy(340,300,"2.");
            outtextxy(650,100,abc);
            outtextxy(400,300,hs[1].name);
            sprintf(abc,"%d",hs[1].scor);
            outtextxy(340,500,"3.");
            outtextxy(650,300,abc);
            outtextxy(400,500,hs[2].name);
            sprintf(abc,"%d",hs[2].scor);
            outtextxy(650,500,abc);
            outtextxy(300,600,"press  any key to back main menu");
            getch();
            cleardevice();
            readimagefile("normal11.jpg",0,0,maxx,mayy);
        }
        else if(x>267&&x<411&&y>580&&y<648)
        {
            gg=1;
            break;
        }
        else
        {
            readimagefile("normal11.jpg",0,0,maxx,mayy);
        }
    }
}
if(gg)
    break;
//setviewport(0,0,maxx,mayy,1);
setbkcolor(3);
cleardevice();
setcolor(WHITE);
settextstyle(10,0,10);
outtextxy(300,300,"Level  1");
delay(1500);
int t=level1();
settextstyle(10,0,10);
if(t<=0)
{
    setviewport(0,0,maxx,mayy,1);
    setbkcolor(RED);
    cleardevice();
    setcolor(0);
    outtextxy(350,300,"TIME  OVER");
```

```
            delay(2000);
        }
        else
        {
            setbkcolor(RED);
            cleardevice();
            setcolor(BLACK);
            outtextxy(350,300,"WRONG  ANSWER");
            delay(2000);
        }
        setviewport(0,0,maxx,mayy,1);
        cleardevice();
        if(point>50)
        {
            setbkcolor(3);
            cleardevice();
            setcolor(WHITE);
            settextstyle(10,0,10);
            outtextxy(300,300,"Level  2");
            delay(1500);
            level2();
        }
        gameover();
        if(point>hs[2].scor)
        {
            delay(2000);
            cleardevice();
            printf("kjddf");
            high_score_function(point);
            print_data();
        }
        point=0;
    }
    return 0;
}
```

Special thanks to:

1. Imran Bin Azad
   Lecturer, CSEDU
2. Ms. Nusrat Faria
   Lecturer, CSEDU
3. Hasnain Heickal Jami
   Lecturer, CSEDU
4. Syed Abrar Zaoad
   Student, CSEDU 22nd batch

References:

[1] https://en.wikipedia.org/wiki/Borland_Graphics_Interface
[2]https://www.google.com/search?q=letters+pattern+wallpaper+hd&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjd2K-1nq_XAhXLu48KHcINDeYQ_AUICigB&biw=1366&bih=662 for images
[2] Borland Graphics Interface for windows:
https://www.cs.colorado.edu/~main/bgi/doc/
[3] www.youtube.com
[4] C The Complete Reference (4th Edition) By Herbert Schildt

The End