Santosh, Tangail-1902

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY



**A Report on**

**C Programming Project**

**ICT-1210**

*MAKING A SIMPLE BUBBLE CATCHING GAME*

***A PROJECT SUBMITTED TO THE DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY, MAWLANA BHASANI SCINCE AND TECHNOLOGY UNIVERSITY, SANTOSH, TANGAIL, BANGLADESH.***

**SUPERVISED BY**

**KAWSAR AHMED**

**LECTURER**

**DEPT OF ICT**

**MBSTU**

**SUBMITED BY**

**MD RUHUL AMIN AMRITA KAMKAR**

**ID: IT-15022 ID: IT-140**

**SESSION: 2014 - 2015 SESSION: 2013–2014**

**APPROVAL**

A report on “**A SIMPLE BUBBLE CATCHING GAME**”, submitted by *Md. Ruhul Amin*, ID : IT-15022,Session : 2014-2015, *Amrita Kamkar,* ID : IT-140, Session : 2013-2014 in the ***Department of Information and Communication Technology (ICT)*** ,**MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY**,Santosh,Tangail-1902,has been accepted as satisfactory for the partial fulfillment of the requirement for the degree of B.Sc.(Engg.) in Information and Communication Technology under the course ICT-1210: Course Title : “Project”.

Signature of Supervisor

KAWSAR AHMED

Lecturer

Department of Information and Communication Technology

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh, Tangail-1902.

**PREFACE**

**Special Thanks**

Special thanks to dear sir ***Kawsar Ahmed****.* He helps to develop the project and basic structure. His insights, expertise, and energy contributed greatly to the success to this project. He has helped on several on my project and, as always, his efforts are appreciated. We also wish to express my gratitude to the sir, especially for invaluable suggestions and critical review of this project.

Finally, we wish to thank many friends for their helpful cooperation.

**ACKNOWLEDGEMENT**

The authors feel proud to have the opportunity to express their heart-felt and most sincere gratitude to a number of peoples who have them in courses of this project.

This project was performed under the supervision of ***Kawsar Ahmed*** Lecturer, **Department of Information and communication Technology (ICT),** *Mawlana Bhashani Science and Technology University, Santosh, Tangail.* First and foremost, the authors wish to concede their profound indebtedness to him for guidance, valuable suggestions, encouragement and cordial cooperation.

We are also grateful to the rest of the teachers of our respective Department, **Department of Information & Communication Technology** for helping us in many ways.

Heartiest thanks to my Department of the University, Information & Communication Technology for providing me such an opportunity for the accomplishment of our project smoothly.

Lot of thanks to those who were the respondents of our project work & soulfully helped us a lot for the collection of data for our project.

Finally, we would like to express our gratitude to the development staff and friends, who have inspired us in this project.

Ruhul

&

Amrita

April, 2017

|  |  |  |  |
| --- | --- | --- | --- |
| **Contents** | | | |
| **Chapter 1: INTRODUCTION** | | | Page No |
|  | 1.1 | Introduction |  |
| 1.2 | C Programing |  |
| 1.3 | Graphics design |  |
| 1.4 | Mouse Programing |  |
| 1.5 | Conclusion |  |
| **Chapter 2: BACKGROUND** | | | |
|  | 2.1 | Introduction |  |
| 2.2 | Background |  |
| 2.3 | Conclusion |  |
| **Chapter 3: METHODOLOGY** | | | |
|  | 3.1 | Introduction |  |
| 3.2 | Welcome Page ( Mouse Programing ) |  |
| 3.3 | Bubble catching ( C Programing ) |  |
| 3.4 | Graphics Design |  |
| 3.5 | Conclusion |  |
| **Chapter 4: RESULT OF PROJECT** | | | |
|  | 4.1 | Introduction |  |
| 4.2 | 1 Welcome Page |  |
| 4.3 | Bubble catching |  |
| 4.4 | Random Bubble falling |  |
| 4.5 | Conclusion |  |
| **Chapter 5: CONCLUSION** | | |  |
| * **Reference** | | |  |
| * **Appendix** | **1.** | List of header file |  |
| **2.** | Source Code |  |

**Chapter 1: Introduction**

**1.1: Introduction**

**1.2: About C Programing Language**

**1.3: About Graphics Design**

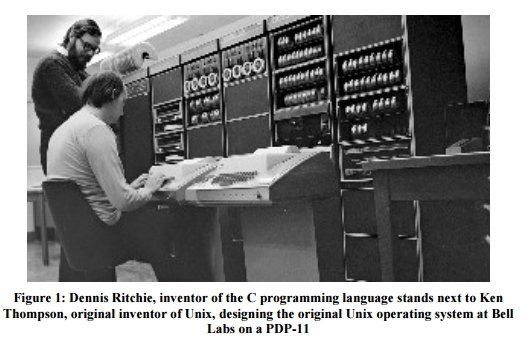
**1.4: About Mouse Programing**

**1.5: Conclusion**

**1.1: Introduction**

In this Chapter we discuss about the history of C Programing, Graphics Design and Mouse Programing. We also Express, How upgrading this programs day by day.

**1.2: About C Programing Language**

The computing world has undergone a revolution since the publication of The C  
Programming Languagein 1978. Big computers are much bigger, and personal computers have capabilities that rival mainframes of a decade ago. During this time, C has changed too, although only modestly, and it has spread far beyond its origins as the language of the UNIX operating system.

The growing popularity of C, the changes in the language over the years, and the creation of compilers by groups not involved in its design, combined to demonstrate a need for a more precise and more contemporary definition of the language than the first edition of this book provided. In 1983, the American National Standards Institute (ANSI) established a committee whose goal was to produce an unambiguous and machine-independent definition of the language C, while still retaining its spirit. The result is the ANSI standard for C.

The standard formalizes constructions that were hinted but not described in the first edition, particularly structure assignment and enumerations. It provides a new form of function declaration that permits cross-checking of definition with use. It specifies a standard library, with an extensive set of functions for performing input and output, memory management, string manipulation, and similar tasks. It makes precise the behavior of features that were not spelled out in the original definition, and at the same time states explicitly which aspects of the language remain machine-dependent.

I have chosen to follow that book’s organization in this book’s structure. The main difference is that their book is machine independent and gives lots of examples based on manipulating text, while this book is machine dependent.

**1.3: About Graphics Design**

**Computer graphics** are [pictures](https://en.wikipedia.org/wiki/Pictures) and [movies](https://en.wikipedia.org/wiki/Movie) created using [computers](https://en.wikipedia.org/wiki/Computer) – usually referring to [image](https://en.wikipedia.org/wiki/Image) data created by a computer specifically with help from specialized graphical hardware and software. It is a vast and recent area in [computer science](https://en.wikipedia.org/wiki/Computer_science). The phrase was coined by computer graphics researchers Verne Hudson and [William Fetter](https://en.wikipedia.org/wiki/William_Fetter) of [Boeing](https://en.wikipedia.org/wiki/Boeing) in 1960. It is often abbreviated as **CG**, though sometimes erroneously referred to as [CGI](https://en.wikipedia.org/wiki/Computer-generated_imagery).

**Figure 2: graphics programing lab**

Important topics in computer graphics include [user interface design](https://en.wikipedia.org/wiki/User_interface_design), [sprite graphics](https://en.wikipedia.org/wiki/Sprite_(graphics)), [vector graphics](https://en.wikipedia.org/wiki/Vector_graphics), [3D modeling](https://en.wikipedia.org/wiki/3D_modeling), [shades](https://en.wikipedia.org/wiki/Shader) , [GPU](https://en.wikipedia.org/wiki/GPU) design, and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), among others. The overall methodology depends heavily on the underlying sciences of [geometry](https://en.wikipedia.org/wiki/Geometry), [optics](https://en.wikipedia.org/wiki/Optics), and [physics](https://en.wikipedia.org/wiki/Physics). Computer graphics is responsible for displaying art and [image data](https://en.wikipedia.org/w/index.php?title=Image_data&action=edit&redlink=1) effectively and meaningfully to the user, and processing image data received from the physical world. The interaction and understanding of computers and interpretation of data has been made easier because of computer graphics. Computer graphic development has had a significant impact on many types of media and has revolutionized [animation](https://en.wikipedia.org/wiki/Animation), [movies](https://en.wikipedia.org/wiki/Movies), [advertising](https://en.wikipedia.org/wiki/Advertising), [video games](https://en.wikipedia.org/wiki/Video_game), and [graphic design](https://en.wikipedia.org/wiki/Graphic_design) generally.

The precursor sciences to the development of modern computer graphics were the advances in [electrical engineering](https://en.wikipedia.org/wiki/Electrical_engineering), [electronics](https://en.wikipedia.org/wiki/Electronics), and [television](https://en.wikipedia.org/wiki/Television) that took place during the first half of the twentieth century. Screens could display art since the [Lumpier brothers](https://en.wikipedia.org/wiki/Lumiere_brothers)' use of mattes to create special effects for the earliest films dating from 1895, but such displays were limited and not interactive. The first [cathode ray tube](https://en.wikipedia.org/wiki/Cathode_ray_tube), the [Braun tube](https://en.wikipedia.org/wiki/Braun_tube), was invented in 1897 - it in turn would permit the [oscilloscope](https://en.wikipedia.org/wiki/Oscilloscope) and the military [control panel](https://en.wikipedia.org/wiki/Control_panel_(engineering)) - the more direct precursors of the field, as they provided the first two-dimensional electronic displays that responded to programmatic or user input. Nevertheless, computer graphics remained relatively unknown as a discipline until the 1950s and the post-[World War II](https://en.wikipedia.org/wiki/World_War_II) period - during which time, the discipline emerged from a combination of both pure [university](https://en.wikipedia.org/wiki/University) and [laboratory](https://en.wikipedia.org/wiki/Laboratory) academic research into more advanced computers and the [United States military](https://en.wikipedia.org/wiki/United_States_military)'s further development of technologies like [radar](https://en.wikipedia.org/wiki/Radar), advanced [aviation](https://en.wikipedia.org/wiki/Aviation), and rocketry developed during the war. New kinds of displays were needed to process the wealth of information resulting from such projects, leading to the development of computer graphics as a discipline.

**1.4: About Mouse Programing**

[](https://en.wikipedia.org/wiki/File:3-Tastenmaus_Microsoft.jpg)A **computer mouse** is a [pointing device](https://en.wikipedia.org/wiki/Pointing_device) (hand control) that detects [two-dimensional](https://en.wikipedia.org/wiki/Two-dimensional_space) motion relative to a surface. This motion is typically translated into the motion of a [pointer](https://en.wikipedia.org/wiki/Pointer_(user_interface)) on a [display](https://en.wikipedia.org/wiki/Computer_monitor), which allows a smooth control of the graphical.

A computer mouse with the most common standard features:

* Two buttons (left and right) and
* A scroll wheel, which can also act as a third button.

Physically, a mouse consists of an object held in one's hand, with one or more buttons. Mice often also feature other elements, such as touch surfaces and "wheels", which enable additional control and dimensional input.

**In this lesson, we shall learn:**

* How to do something when input *events* occur.  Events covered are **key Pressed** (when a key on the keyboard is pressed), **mouse Clicked**(when the left button on the mouse is pressed), and **mouse Moved** (when the mouse is moved within the graphics canvas), and
* How to get the X and Y coordinates where the mouse was when its left button was clicked.

**THE USER INTERFACE**

These days, most of the interaction with a computer or any device with a computer in it, is via what's called a **G**raphical **U**ser **I**nterface.  In computer-speak it's commonly called the **GUI**, (pronounced goo-ee).

On a desk side computer system, the GUI provides ways for you to launch programs, display stuff, etc... All with a mouse pointing device.  When you are interacting with programs, you use a pointing device (a mouse) to open up menus, choose items from the menus, check boxes that determine how your program should do things.  It's a point-and-click environment.  When it is time to enter text, you have the keyboard.

On a calculator or mobile phone, much of the interaction is done with buttons - buttons for numbers, menu buttons, and arrow buttons for moving around within the menu system.  For text entry, sometimes the buttons are used, but some devices have tiny key pads (made popular by the Blackberry messaging device).

The first two computers I got to use only dealt with cardboard cards with holes punched in them for input.

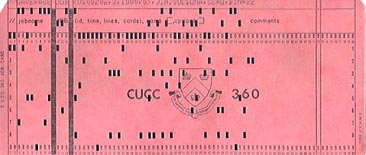


Fig: Shows what a punch card looked like

But, by the time I got my first job, my interface was through a teletype terminal, and this was a major step forward!  I could actually interact with the computer, not just submit something for it to do and then wait for something to appear on the printer.  Figure: 1 shows a Model-33 Teletype, like the one I used for years.  Note the yellow paper-tape reader it had on its left side.  Paper tape was used similarly to punched card

**Figure: 1**

By the 1970's, this magical kingdom in the world of computer science appeared, commonly called PARC.  Actually it is Xerox PARC (PARC stands for Palo Alto Research Center) and it still exists.  A group of scientists there built what was to be the future of computing: networks of *“*personal*"* computers were prototyped.  Much of what you see on a computer these days, the user interface, was invented at PARC.  The researchers all had computers (Altos) with high-resolution graphics displays and a pointing device (a mouse).  The mouse, or "X-Y position indicator for a display system" which is what it was originally called, was invented by Doug Engelbart at SRI (Stanford Research Institute) in the early 1960's.  The computers the researchers used were connected by a high-speed network (Ethernet).  They used this environment that they built themselves to do all of their work.  Figure: 2 is a picture of an Alto workstation.

**Figure: 2**

The first fancy integrated user interface that was built for the Alto (that's most applicable to what I want to talk about) was Alan Kay's vision of Daybook.  Alan and a few other researchers including Adele Goldberg and Dan Ingalls, built a graphically-oriented system for working withPersonal Dynamic Media, a programming environment they called Smalltalk.  The Smalltalk system introduced a new concept of breaking the graphical display into pieces called "windows" which held separate information.  As I can fit concepts from this system into our learning, I will.

We've been doing lots of graphical output, now it's time to interact with a user - to get some input via your computer systems' GUI.  In the following few sections I cover *events* that your program can respond to.  The *events* that logo supports are **key Pressed**, **key Released**, **key Typed**, **mouse Clicked**, and **mouse Moved**.  I'm going to start with the mouse since it is the most fun!

**1.5 Conclusion**

From this chapter we knew about the history of C Programing, Graphics Design and Mouse Programing. We also know, how upgrading this programs day by day.

**Chapter 2: Background**

**2.1: Introduction**

**2.2: Background**

**2.3: Conclusion**

**Chapter 3: Methodology**

**3.1: Introduction**

**3.2: Header File**

**3.2: Welcome Page**

**3.3: Analog Clock**

**3.4: Egg Game**

**3.5: Conclusion**

**3.1: Header File**

|  |  |  |
| --- | --- | --- |
| S. NO | Header Files | Description |
| 1 | #include<bits/stdc++.h> | It contains all standard c & c++ header file |
| 2 | #include<iostream> | Doing the input output operations. |
| 3 | #include<graphics.h> | Creates the graphics window on the screen. |
| 4 | #include<conio.h> | **Standard Input / Output Streams Library** |
| 5 | #include<dos.h> | It contains functions for handling interrupts, producing sound, date and time functions etc. |
| 6 | #include<math.h> | This is used to do all mathematical calculation. |
| 7 | #include <ctime> | All time related functions are defined in this header file. |

**Some of its inbuilt functions are:**

1. clrscr() - It is used to clear the screen
2. cgets() - Reads a string from the keyboard
3. getch() - Reads a character from keyboard
4. getche() - Reads a character from keyboard and echoes it to the screen
5. kbhit() - Determines if a keyboard key was pressed
6. putch() - Writes a character to the screen
7. cputs() - Writes a string to the screen

**1. #include<graphics.h>**

This interface provides access to a simple graphics library that makes it possible to draw lines, rectangles, ovals, arcs, polygons, images, and strings on a graphical window.

**Some of its inbuilt functions are:**

|  |  |  |
| --- | --- | --- |
| S.NO | Inbuilt Functions | Description |
| 1. | [**initGraphics()**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:initGraphics)[**initGraphics(width, height)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:initGraphics) | -Creates the graphics window on the screen. |
| 2. | [**drawArc(bounds, start, sweep)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawArc)[**drawArc(x, y, width, height, start, sweep)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawArc) | -Draws an elliptical arc inscribed in a rectangle. |
| 3. | [**fillArc(bounds, start, sweep)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:fillArc)[**fillArc(x, y, width, height, start, sweep)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:fillArc) | -Fills a wedge-shaped area of an elliptical arc. |
| 4. | [**drawImage(filename, pt)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawImage)[**drawImage(filename, x, y)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawImage)[**drawImage(filename, bounds)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawImage)[**drawImage(filename, x, y, width, height)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawImage) | -Draws the image from the specified file with its upper left corner at the specified point. |
| 5. | [**getImageBounds(filename)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:getImageBounds) | -Returns the bounds of the image contained in the specified file. |
| 6. | [**drawLine(p0,p1)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawLine)[**drawLine(x0, y0, x1, y1)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawLine) | -Draws a line connecting the specified points. |
| 7. | [**drawRect(bounds)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawRect)[**drawRect(x, y, width, height)**](http://web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/graphics.html#Function:drawRect) | -Draws the frame of a rectangle with the specified bounds. |

**Table - 3.4**: Built In Functions of graphics.h

**2. #include<ctime.h>**

The ctime.h header defines four variables types, two macro and various functions for manipulating date and time.

**Some of its inbuilt functions are:**

|  |  |  |
| --- | --- | --- |
| S.NO | Inbuilt Functions | Description |
| 1. | **time\_t** | - Alias of a fundamental arithmetic type capable of representing times |
| 2. | **ctime()** | -Used to return string that contains date and time information. |

**Table - 3.5:** Built In Functions Of ctime.h

**3. #include<math.h>**

The math.h header file defines some function to calculate mathematical operation.

**Some of its inbuilt functions are:**

|  |  |  |
| --- | --- | --- |
| S.NO | Inbuilt Functions | Description |
| 1. | **pow()** | Raise to power. |
| 2. | **sqrt()** | Calculate square root of a number. |

**4. #include<iostream>**

The iostream header file defines some function for the operation of input and output that means I/O operation.

**Some of its inbuilt functions are:**

|  |  |  |
| --- | --- | --- |
| S.NO | Inbuilt Functions | Description |
| 1. | **Cout<<** | Used for console output |
| 2. | **Cin>>** | Used for console input |
| 3. | **fopen()** | Opens a file in specified path |
| 4. | **fclose()** | Closes a opened file |
| 5. | **fscanf()** | Scan data from file |
| 6. | **fprintf()** | Write data to file |

**3.3 Conclusion**

In this chapter we discussed about different methods and functions which we used in our project. We also discussed every header files and their including functions elaborately.

**Mouse programing**

In this lesson, we learned about the events that TG will pass on to our programs if we want to know about them.  They are:

* + When the mouse is clicked on the graphics canvas.  Your program can do something by defining a procedure named mouse Clicked.
  + When the mouse is moved around on the graphics canvas.  Your program can do something by defining a procedure named mouse Moved.  The mouse must have been previously clicked on the graphics canvas to activate it.
  + When a key is pressed on the keyboard.  Your program can do something by defining a procedure named key Pressed.  The mouse must have been previously clicked on the graphics canvas to activate it.

We learned about the **mousex** and **mousey** operators which output the X and Y values for the coordinate where the mouse was when it was clicked or moved.

**Graphics program**

In this lesson, we learned about how graphics programing work and which sector we use graphics programing,

Computer graphic design provide students with the skills to enter careers in web, graphic and print design. Skills acquired range from technical to creative. These courses may be part of a degree program or offered in stand-alone format. Schools offering [3D Animation degrees can also be found in these popular choices](http://learn.org/articles/What_Skills_Will_I_Learn_in_a_Computer_Graphic_Design_Courses.html#searchSummaryAndResults).

**Appendix:**

1. **List Of Header Files:**
2. #include<conio.h>
3. #include<stdio.h>
4. #include<iostream.h>
5. #include<graphics.h>
6. #include<stdlib.h>
7. #include<dos.h>
8. #include<string.h>
9. #include<math.h>

**2. Source Code:**

#include<bits/stdc++.h>

#include<graphics.h>

#include<windows.h>

#include<math.h>

#include<iostream>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<dos.h>

#include<string.h>

#include <ctime>

using namespace std;

int mx, my;

char arr[10000]="" ;

int play\_push=true;

double x = 10 , y = 200 ;

double cx1 = 5, cy1 = 0 , cr1 = 1 ;

int score = 0, s = 0, c= 1, g=0;

int temp=0, ck=0;

float ball\_speed=1;

float timer=2000;

int i=1;

void over();

void play();

void start();

void uprocess();

void ins();

void uprocess()

{

cleardevice();

setcolor(BLACK);

outtextxy(10,40,"??? Sorry, option is under constructed");

setcolor(BLACK);

setbkcolor(WHITE);

delay(10);

getch();

start();

}

void ins()

{

cleardevice();

setcolor(WHITE);

outtextxy(10,20,"Just catch the ball in the limited time.");

outtextxy(10,45,"Catch ball and earn score.");

outtextxy(10,70,"For moving in various rotation you can follow");

outtextxy(10,95,"these instructions: ");

outtextxy(10,125,"1. press 4 for left, 6 for right");

outtextxy(10,150,"2. press 8 for up, 2 for down");

outtextxy(10,175,"3. press 1 for left-down, 3 for right-down");

outtextxy(10,200,"4. press 7 for left-up, 9 for right-up");

setcolor(BLACK);

outtextxy(10,235,"Idea is given by Amrita and Developed by Ruhul");

outtextxy(10,270,"Special thanks to Dear Kawsar Ahmed Sir.");

getch();

start();

}

void over()

{

FILE \*f;

f=fopen("High Score","r");

fscanf(f,"%d",&temp);

if(temp<score)

{

f=fopen("High Score","w");

fprintf(f,"%d",score);

fclose(f);

}

else

{

fclose(f);

}

cleardevice();

settextstyle(3,0,5);

setcolor(BLACK);

outtextxy(170,40,"GAME OVER");

sprintf(arr, "Your level %d", i);

outtextxy(180,100,arr);

sprintf(arr,"Your Score %d", score);

outtextxy(180,160,arr);

setcolor(BLACK);

getch();

start();

}

void play()

{

z:

for(; i<=100; i++)

while(true)

{

timer=timer-0.89;

if(timer < 0)

{

over();

}

if(kbhit())

{

char ch=getch();

if(ch=='8')

{

y-=10;

}

if(ch=='2')

{

y+=10;

}

if(ch=='6')

{

x+=10;

}

if(ch=='4')

{

x-=10;

}

if(ch=='9')

{

x+=10;

y-=10;

}

if(ch=='7')

{

x-=10;

y-=10;

}

if(ch=='1')

{

x-=10;

y+=10;

}

if(ch=='3')

{

x+=10;

y+=10;

}

if(ch==' ')

{

outtextxy(250,360,"Press Any key to Continue");

getch();

}

}

sprintf(arr,"Level %d Score = %d ",i,score) ;

settextstyle(SANS\_SERIF\_FONT,HORIZ\_DIR,1);

setcolor(GREEN);

outtextxy(450,20,arr);

setcolor(WHITE);

circle(cx1,cy1,cr1) ;

setcolor(BLACK\_PEN);

setbkcolor(WHITE\_PEN);

sprintf(arr,"TimeLeft = %.2f",timer) ;

outtextxy(450,50,arr);

cy1+=ball\_speed;

if(cy1>400)

{

cy1 = 0 ;

cx1 = rand() % 400 + 100 ;

}

bar(x, y, x+50, y+50);

rectangle(x,y,x+50,y+50) ;

setcolor(RED);

setfillstyle(1,GREEN);

rectangle(x,y, x+50,y-20);

double temp = (cx1-x) \*(cx1-x) + (cy1 - y ) \*(cy1 - y ) ;

temp = sqrt(temp);

if(temp < (50))

{

cx1 = rand() % 400 + 100 ;

cy1 = 0 ;

score++;

}

delay(10) ;

cleardevice();

if(score>s)

{

s=score;

cx1+=1;

cy1+=.5;

cr1+=1;

}

if(score==10\*i)

{

ball\_speed++;

i++;

timer=2000;

goto z;

}

if(score==40\*c)

{

cx1=5;

cy1=0;

cr1=1;

c++;

}

}

if(timer==0)

over();

}

void start()

{

int ck=0,temp=0;

while(true)

{

cleardevice();

mx=mousex();

my=mousey();

setfillstyle(15,2);

bar(0,0,650,450);

setfillstyle(1,BLACK);

bar(200,50,435,90);

settextstyle(8,0,3);

setbkcolor(BLACK);

setcolor(WHITE);

outtextxy(260,56,"New Game");

setfillstyle(1,WHITE\_BRUSH);

bar(200,120,435,160);

settextstyle(8,0,3);

setbkcolor(WHITE\_BRUSH);

setcolor(RED);

outtextxy(240,66+60,"Select Level");

setfillstyle(1,BLUE);

bar(200,190,435,230);

settextstyle(8,0,3);

setbkcolor(BLUE);

setcolor(WHITE);

outtextxy(230,66+130,"Highest score");

setfillstyle(1,WHITE\_PEN);

bar(200,250,435,295);

settextstyle(8,0,3);

setbkcolor(WHITE\_PEN);

setcolor(WHITE);

outtextxy(240,66+195,"Instructions");

setfillstyle(1,RED);

bar(200,320,435,360);

settextstyle(8,0,3);

setbkcolor(RED);

setcolor(WHITE);

outtextxy(290,66+260,"Exit");

if(ismouseclick(WM\_LBUTTONDOWN))

{

if(mx>200 && mx<435 && my>130 && my<170)

{

uprocess();

}

if(mx>200 && mx<435 && my>60 && my<100)

{

ck=1;

cleardevice();

goto yeah;

}

if(mx>200 && mx<435 && my>200 && my<240)

{

char arr[20];

FILE \*f;

f=fopen("High Score","r");

fscanf(f,"%d",&temp);

fclose(f);

sprintf(arr,"High Score : %d",temp);

cleardevice();

settextstyle(1,0,4);

outtextxy(150,200,arr);

getch();

}

if(mx>200 && mx<435 && my>250 && my<290)

{

ins();

}

if(mx>200 && mx<435 && my>320 && my<360)

{

exit(0);

}

clearmouseclick(WM\_LBUTTONDOWN);

}

delay(400);

cleardevice();

}

yeah:

if(ck==1)

{

play();

}

}

int main()

{

system("color D");

printf("Simple bubble touching game project.");

initwindow(650,450);

setbkcolor(BLACKNESS);

setcolor(BLUE);

settextstyle(BLUE,0,GREEN);

outtextxy(40,230,"WELCOME TO OUR C PROJECT.");

delay(1200);

cleardevice();

outtextxy(40,230,"WELCOME TO OUR C PROJECT..");

delay(1200);

cleardevice();

outtextxy(40,230,"WELCOME TO OUR C PROJECT...");

delay(1200);

cleardevice();

outtextxy(40,230,"WELCOME TO OUR C PROJECT....");

delay(1200);

cleardevice();

outtextxy(40,230,"WELCOME TO OUR C PROJECT.....");

delay(1000);

cleardevice();

outtextxy(200,200,"LET'S START");

delay(800);

cleardevice();

start();

getch();

closegraph();

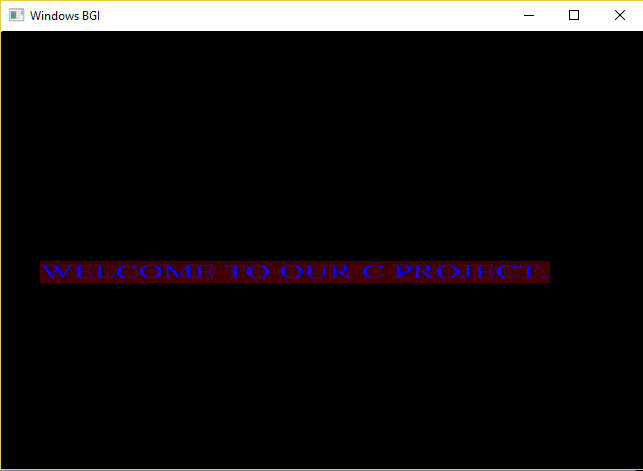
return 0;

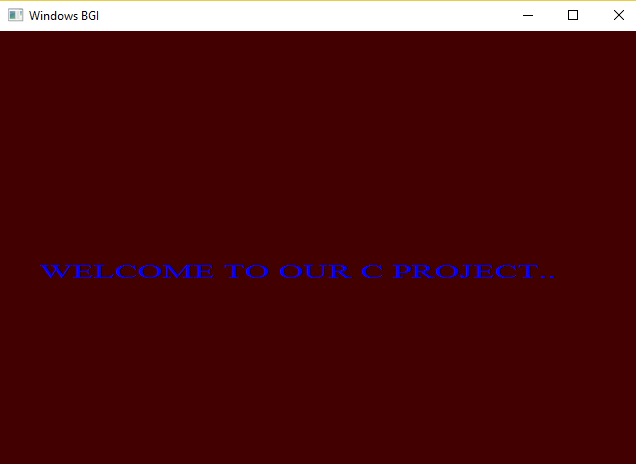
}

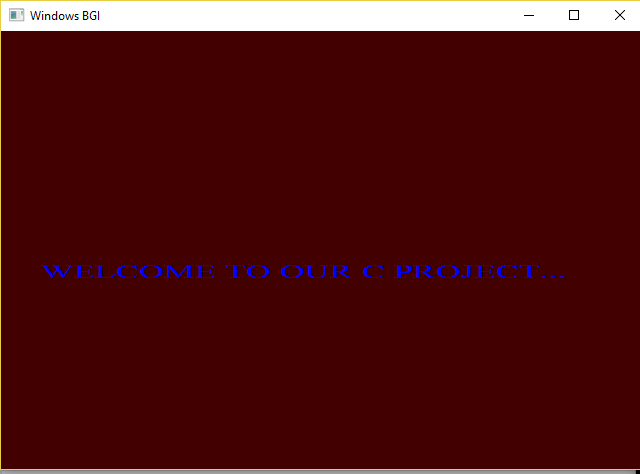
**Output:**

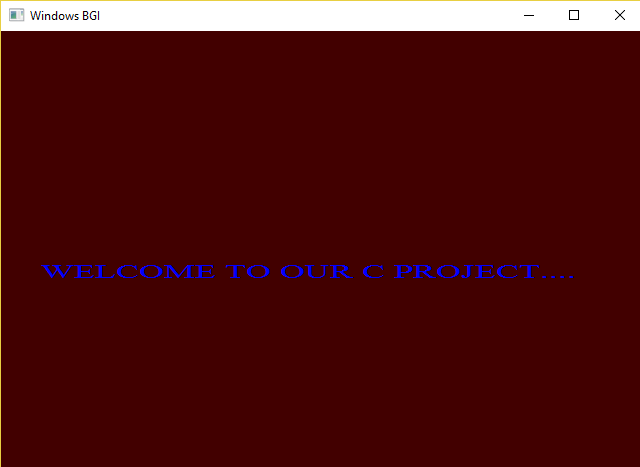
**Welcome page:**

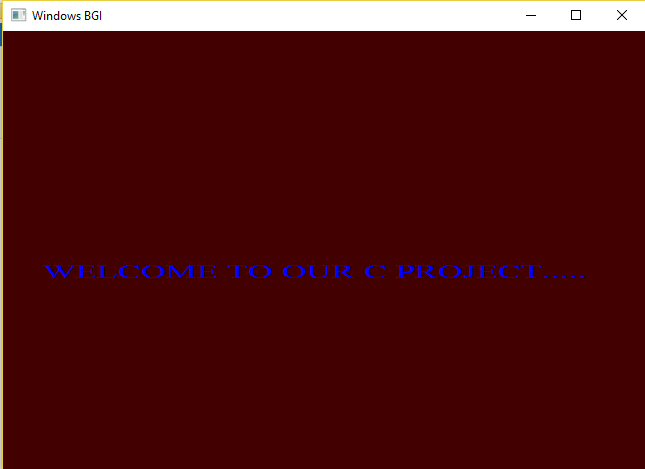
**Loading our game project….**

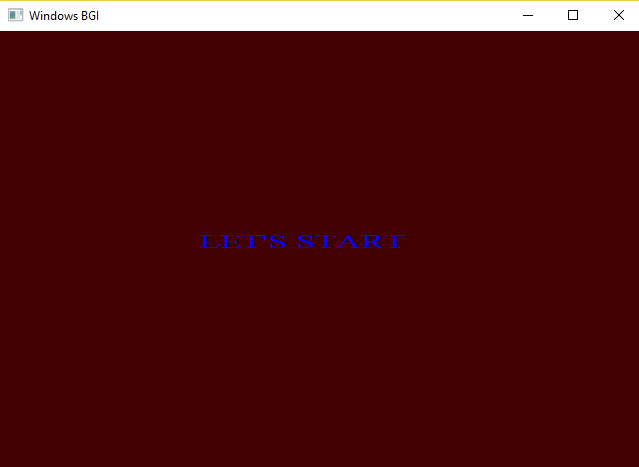




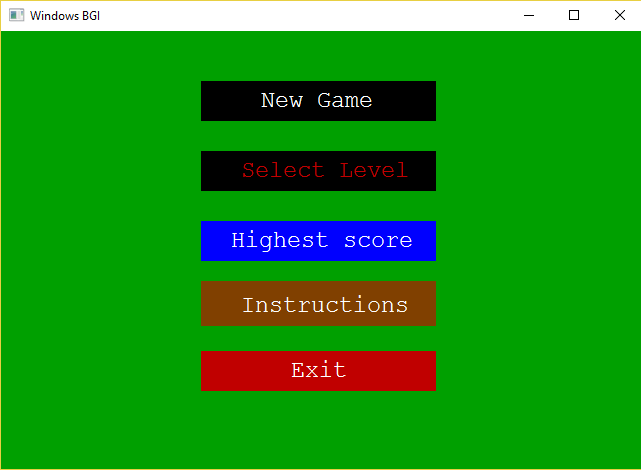




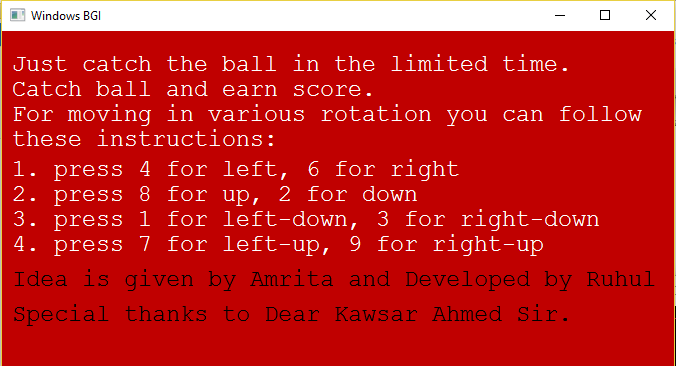




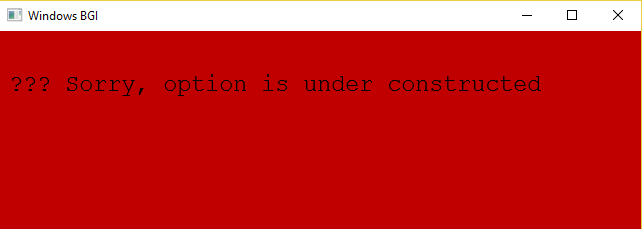
Game menu:

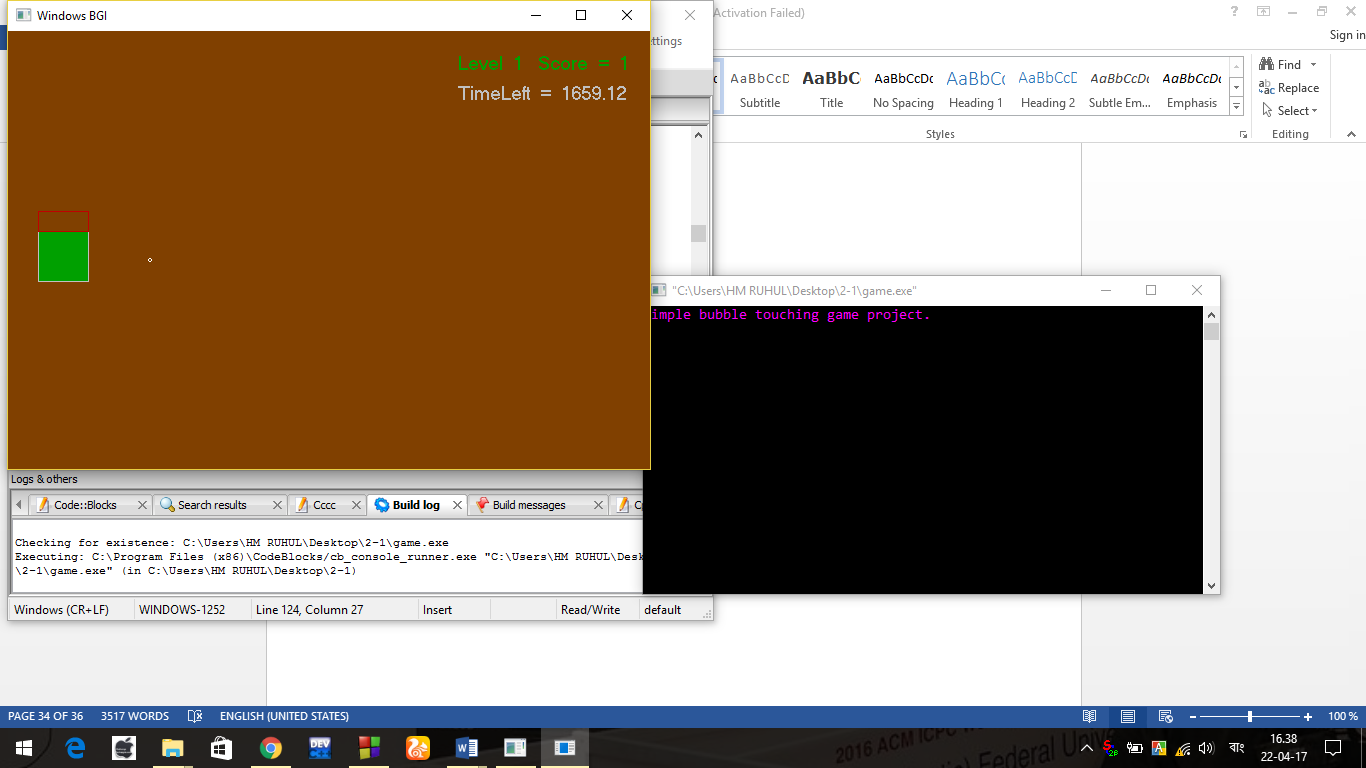


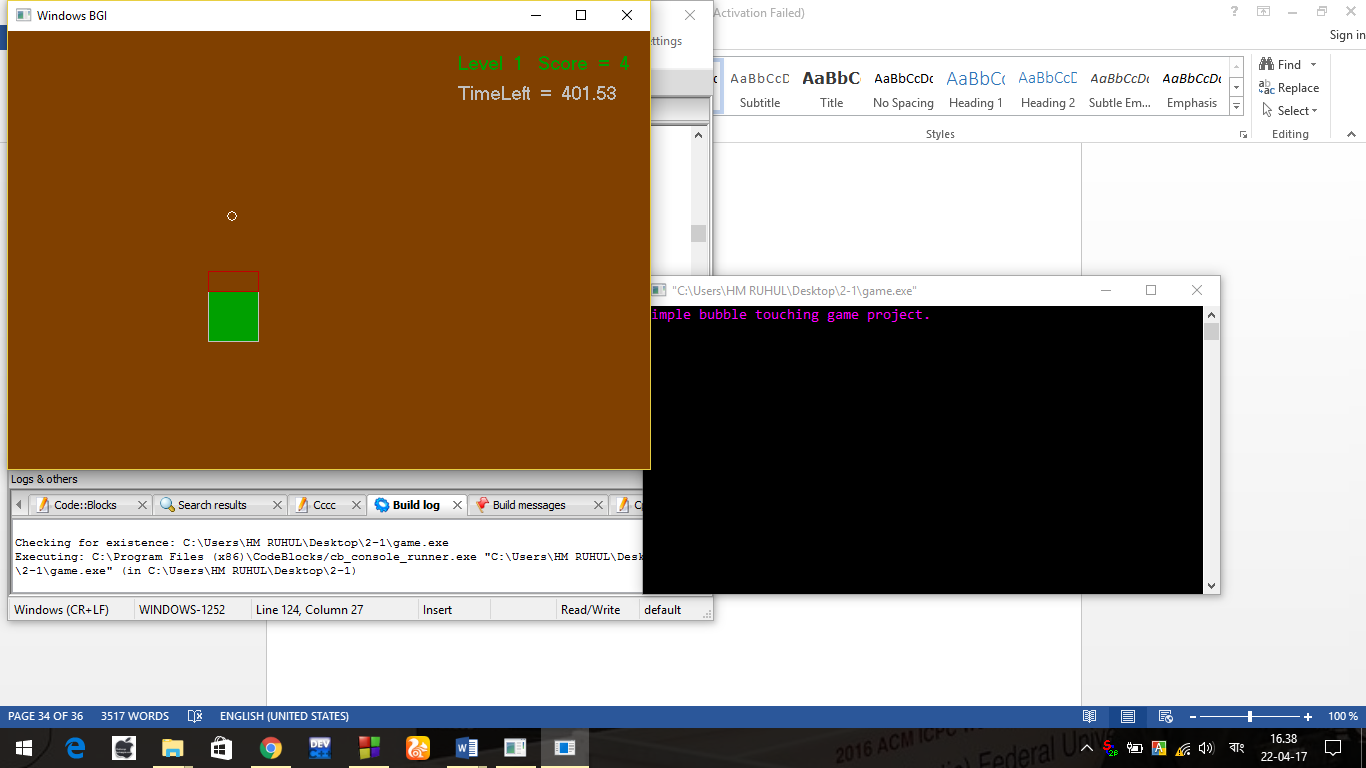
Game Instuructions:

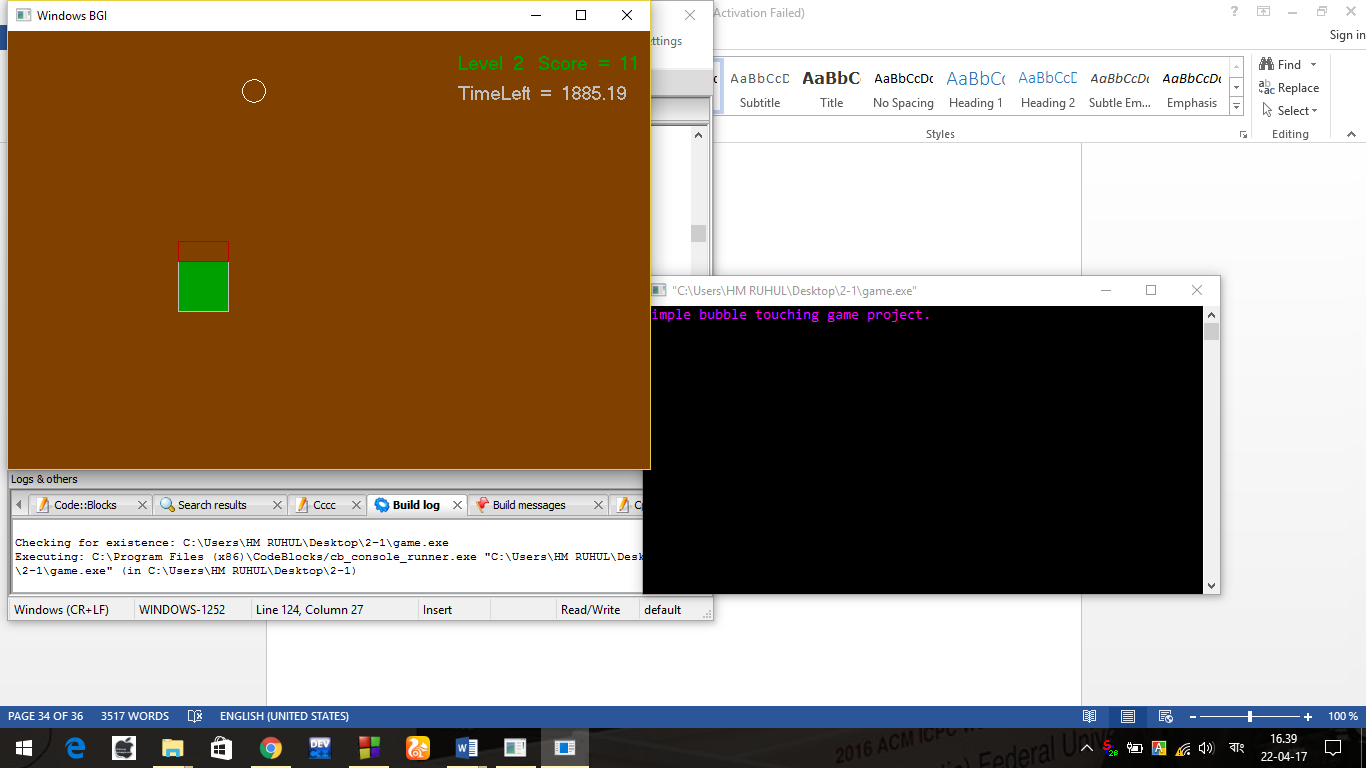


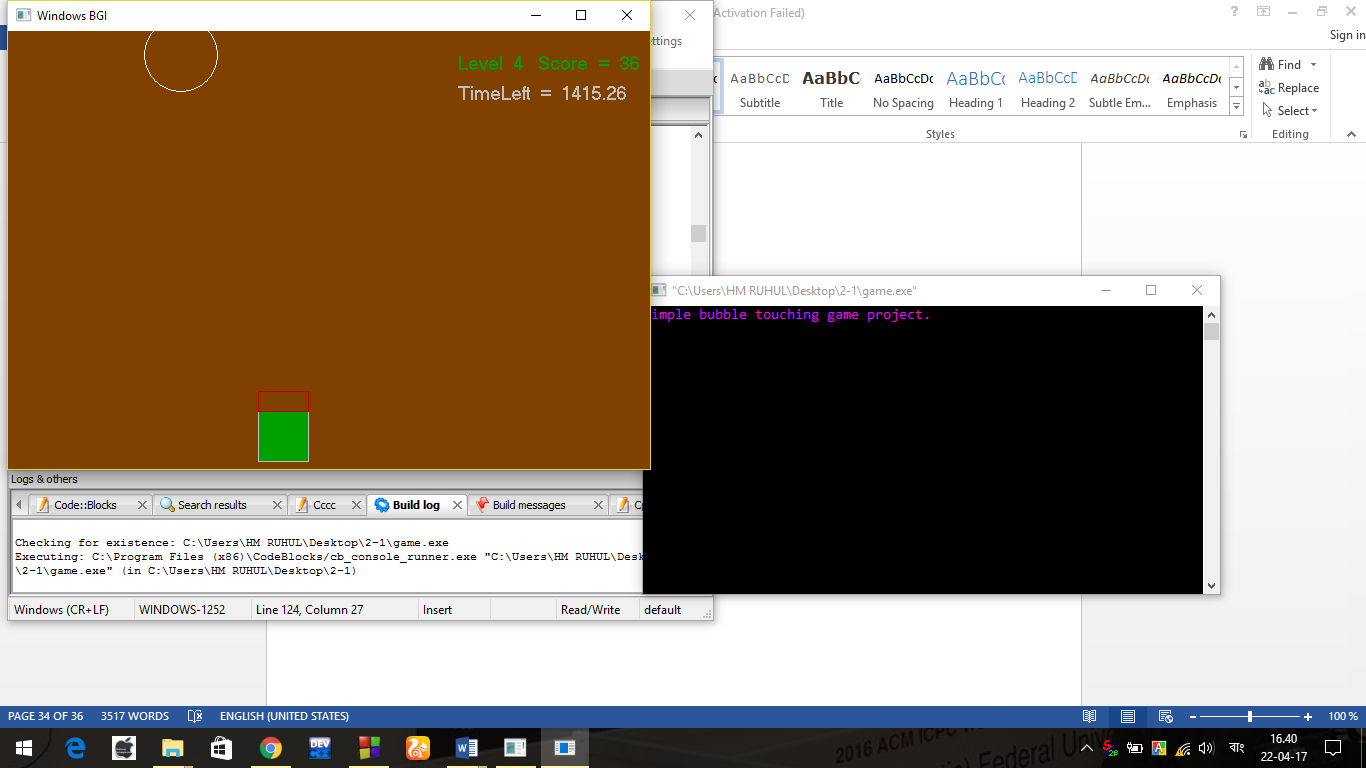
Level Choice:

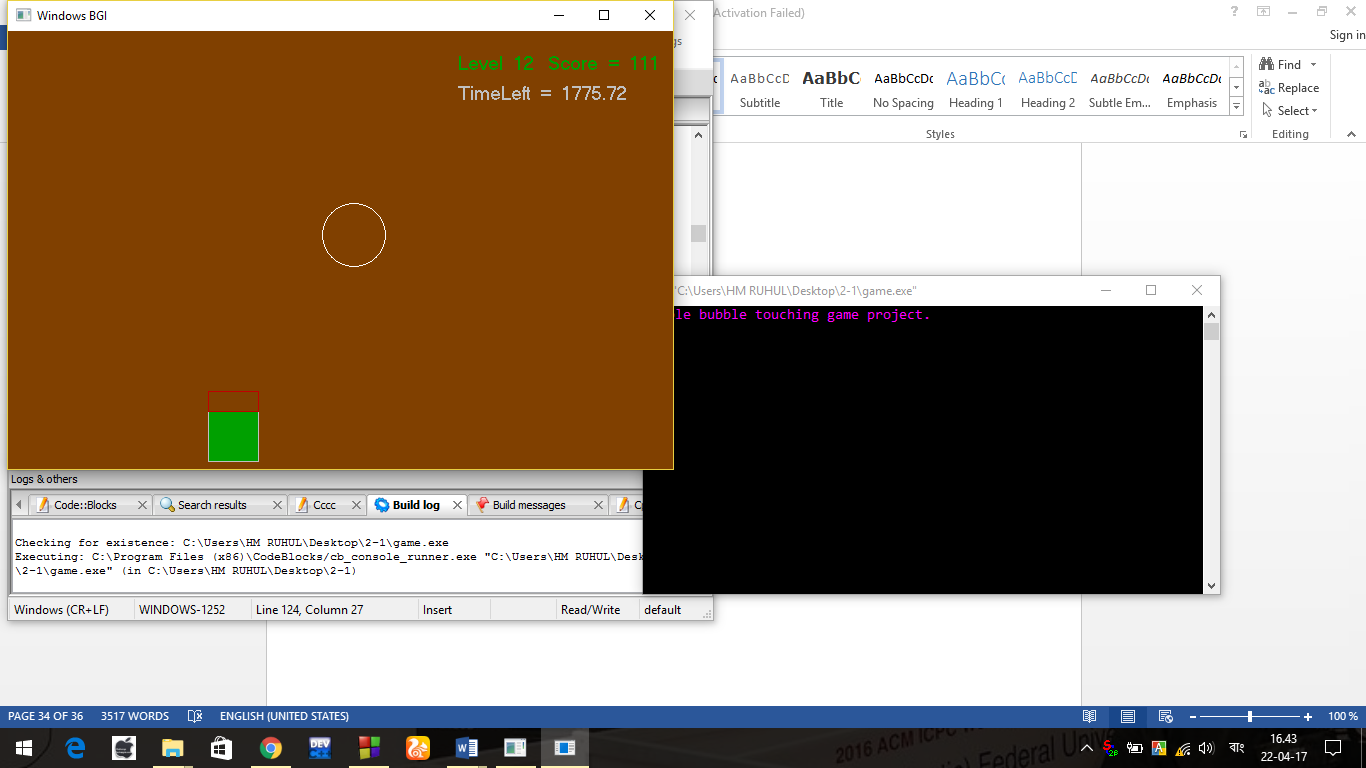


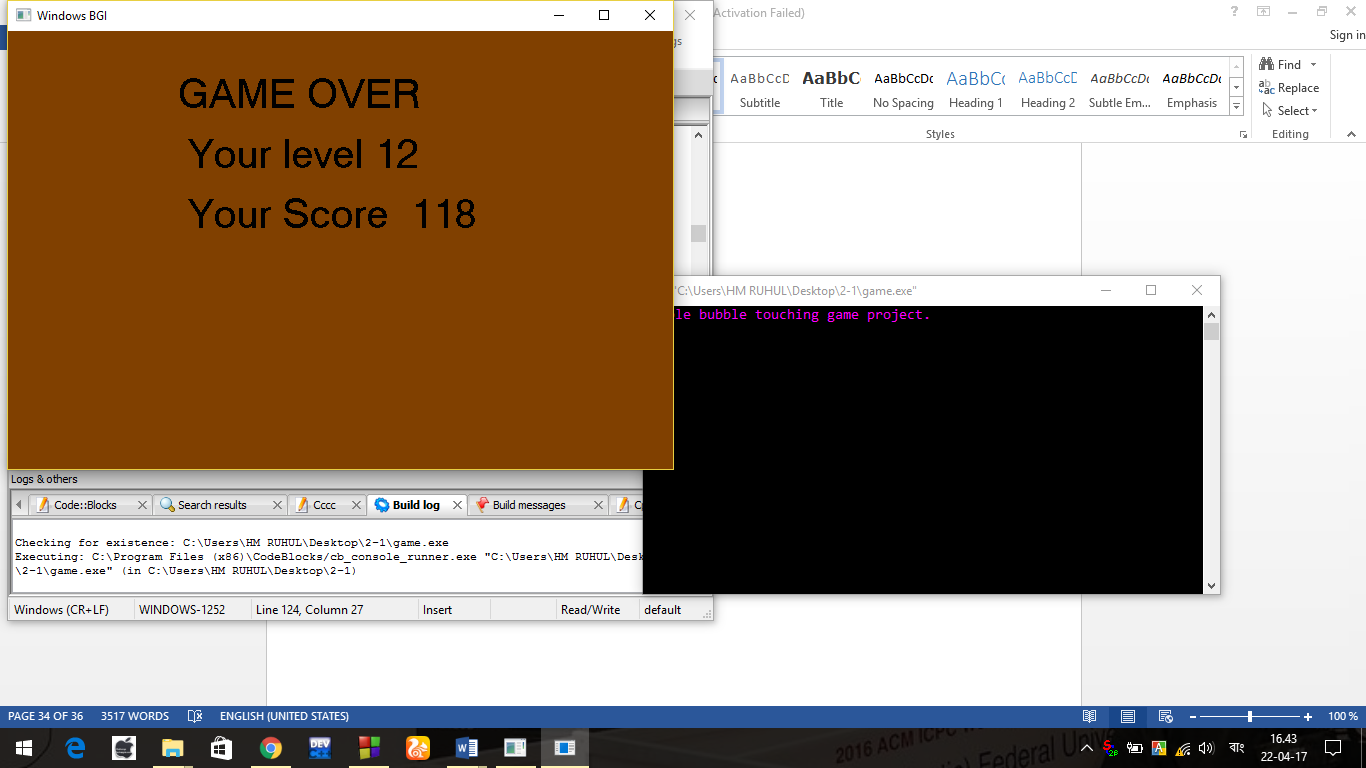
Starting of the game:

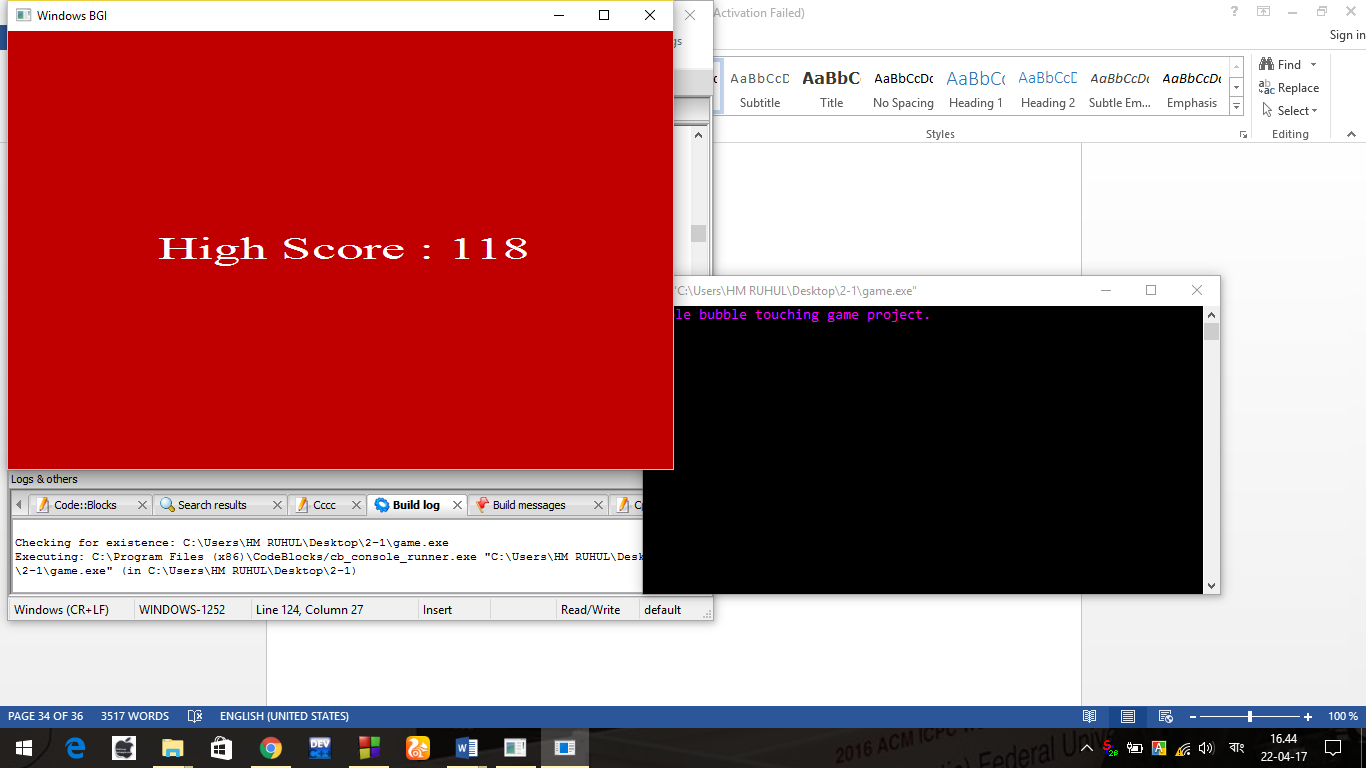


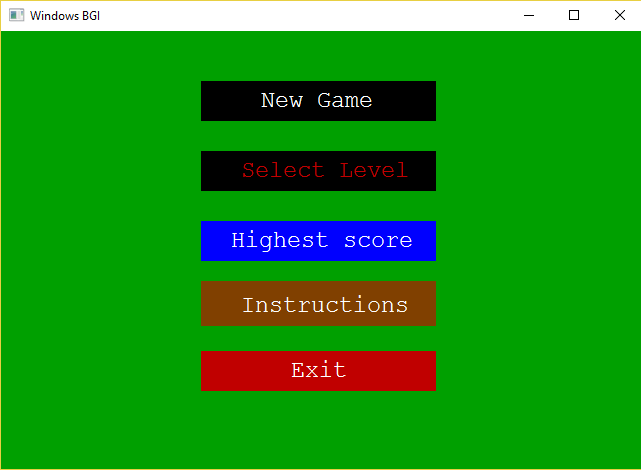












For any key press we can return to main menu.

Conclusion: In this chapter , we represented the result of our C programming project.

**REFERENCES**

1. "Home page". JosefPallweber.com. Retrieved 2015-11-07.

2. Churm, Thomas M. (September 26, 2010). "A Short History of Digital Clocks and Watches". Alarm Clock Blog. Retrieved 2015-11-07.

3. (German) Imperial Patent Office (October 27, 1890). "Patent No. 54093" (PDF). German Patent and Trademark Office. Retrieved 2015-11-07.

4. "Is digital more precise?". The German Clock Museum. April 2015. Retrieved 2015-11-07.

5. "The History of the Digital Watch". h2g2. April 30, 2003. Retrieved 2015-11-07.

6. "Radio Controlled LED Alarm Clock Instruction Manual — SM2442" (DOC). Zeon Ltd. Retrieved 2015-11-07.

7. "Stopped Clock". TV Tropes. Retrieved 2015-11-07.

8. Burton G. Andreas (1965). Experimental psychology. p.186

9. Neil J. Salkind (2006). Statistics for People who (think They) Hate Statistics: The Excel Edition. page 106.

10. ISO/IEC (1999). ISO/IEC 9899:1999(E): Programming Languages - C §7.19.1 para 1

11. "The GNU C Library – Introduction". Gnu.org. Retrieved 2013-12-05.

12. "Difference between C standard library and C POSIX library". stackoverflow.com. 2012. Retrieved 2015-03-04.

13. "C Standards". Kiel. Retrieved 24 November 2011.

14. https://en.wikibooks.org/wiki/C\_Programming/C\_Reference/math.h