 ***ChatU***p

BY

*AARYA ARUN*

*HARDIK G*

*ANIRUDDH G*

COMPUTER PROJECT

2016-2017

A CHAT APPLICATION USING

DEV-C++

ACKNOWLEDGEMENT

We are really grateful to those who helped us to complete our Computer Science project of creating an offline chat app using Dev C++ and graphics within the given time-frame.

This assignment could not have been completed without the guidance and encouragement of our computer science teacher Mrs Smitha Ravindran and the entire computer science department. We want to thank her for all her help.

We also extend our sincere thanks to Kavitha ma’am for all her valuable inputs during the project building phases.

Last but not the least, we would like to thank our school Sri Kumaran Children’s Home and CBSE board for providing us with the opportunity to learn & perform this experiment.

This project provided us ample opportunities to display teamwork - we achieved our goals together as a team.

INDEX

Content Pg.no

1. Introduction 6

1. Project documentation 7
2. Project flow diagram 8
3. Class diagram 9
4. Functions 10
5. Source code 11
6. Output screenshots 22
7. Scope for improvement 24

Bibliography 25

INTRODUCTION

**Dev-C++** is a [free](https://en.wikipedia.org/wiki/Free_software) full-featured [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) distributed under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) for programming in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%252B%252B). It is written in [Delphi](https://en.wikipedia.org/wiki/Delphi_(programming_language)).

It is bundled with, and uses, the [MinGW](https://en.wikipedia.org/wiki/MinGW) or [TDM-GCC](https://en.wikipedia.org/wiki/TDM-GCC) 64 bit port of the [GCC](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) as its compiler. Dev-C++ can also be used in combination with [Cygwin](https://en.wikipedia.org/wiki/Cygwin) or any other GCC-based compiler.

Bloodshed **Dev**-**C++** is a full-featured Integrated Development Environment (IDE) for the C/**C++**programming language. It uses Mingw port of GCC (GNU Compiler Collection) as it's compiler.

This project has been made entirely using Dev C++ with the header file graphics.h.

***GRAPHICS.H:*** *An interface which 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.*

SYSTEM REQUIREMENTS

HARDWARE:

1. RAM 2GB OR HIGHER

2. In order to enable the chat program to work, computers sharing the same memory space with access to some common files must be used.

SOFTWARE : OS:WINDOWS XP OR ABOVE

IDE: BLOODSHED DEV-C++ *(VERSION 4.9.9.2 OR ABOVE****)***

PROJECT DOCUMENTATION

**AIM:** To facilitate communication between parties, especially within connected school lab systems without the usage of the internet.

**Target audience**: This application can be used by people of all ages to stay connected. As this works without the internet, it can be used in offices with connected workstations to maintain contact via messages even in the absence of the internet.

**Need for the project:**

* Maintenance of communication without usage of internet via messages in connected workstations.
* Caters to the ever growing demands to be constantly connected in this fast-paced world where technology is able to achieve such constant communication channels for important message transfers
* Can also be upscaled to have more utilities for sharing emoji, etc.

PROJECT FLOW DIAGRAM

Instructions for usage page

Login page (login with username)

Menu (for chat options)

Chat room

Logs out to allow another user login

Deletes previous chat records

Exits the program

1 2 3 4

CLASS DIAGRAM

The program has only one class in it, which is the chat class on which the program is built.

Class name: Chat

Class members: ~no class members~

Class functions:

void type()

void read()

void del()

void copy()

void loop()

void func()

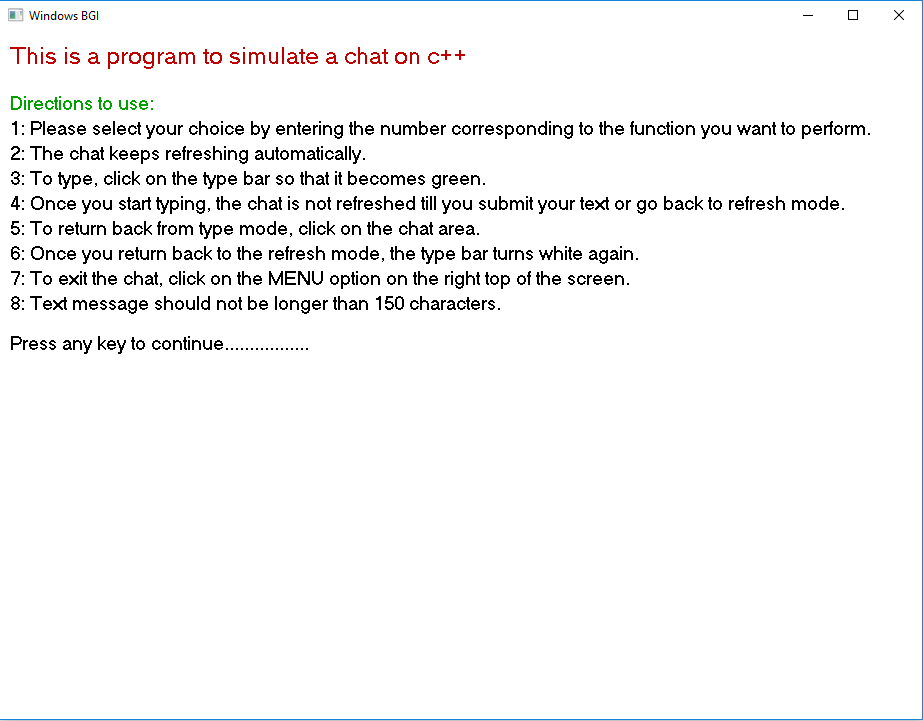
FUNCTIONS

These are the most important functions in the program with prototype and overall functionality.

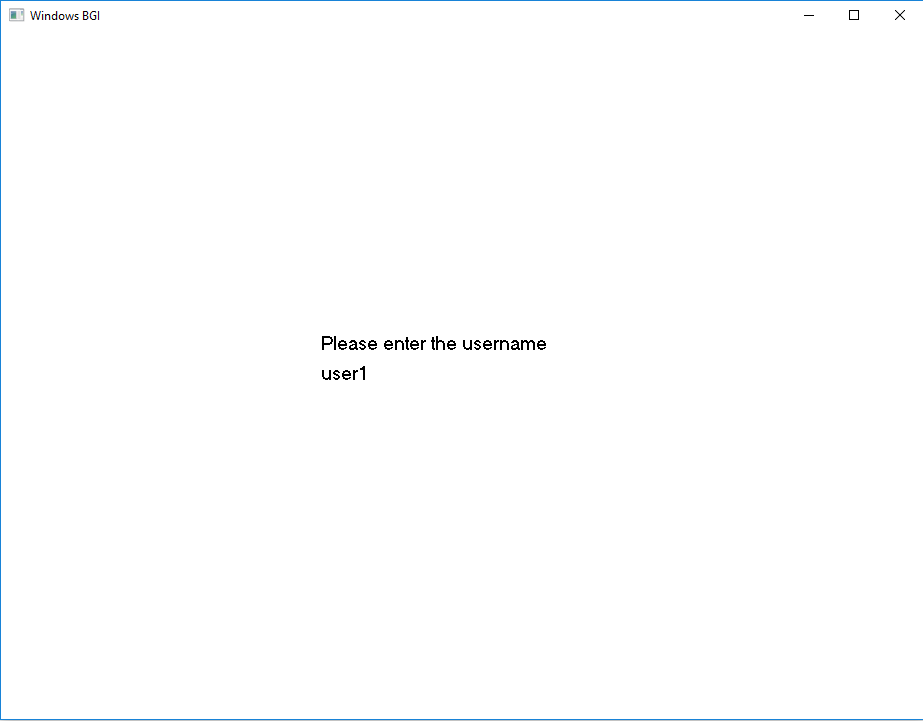
1. **void type()**: Function to type a message into the Chat file
2. **void read()**: Function to display the chat by reading from the Chat file
3. **void del()**: Function to delete the chat when the user chooses to do so
4. **void copy()**: Function to handle the layout of the text in the file to get desired display on the graphics screen
5. **void input(char \*buffer, int x, int y)**:Function to help take user name from the graphics window
6. **void loop()**: Function to decide which function has to be performed based on the position of the mouse pointer
7. **void select\_user()**: Function to take in the user name of from the user to provide identification
8. **void input\_text(char \*buffer, int x, int y)**: Function to help take the text message typed by the user from the graphics window.
9. **void func()**: Function to take the user choice and perform the desired task through a menu
10. **int main()**: Runs the entire program

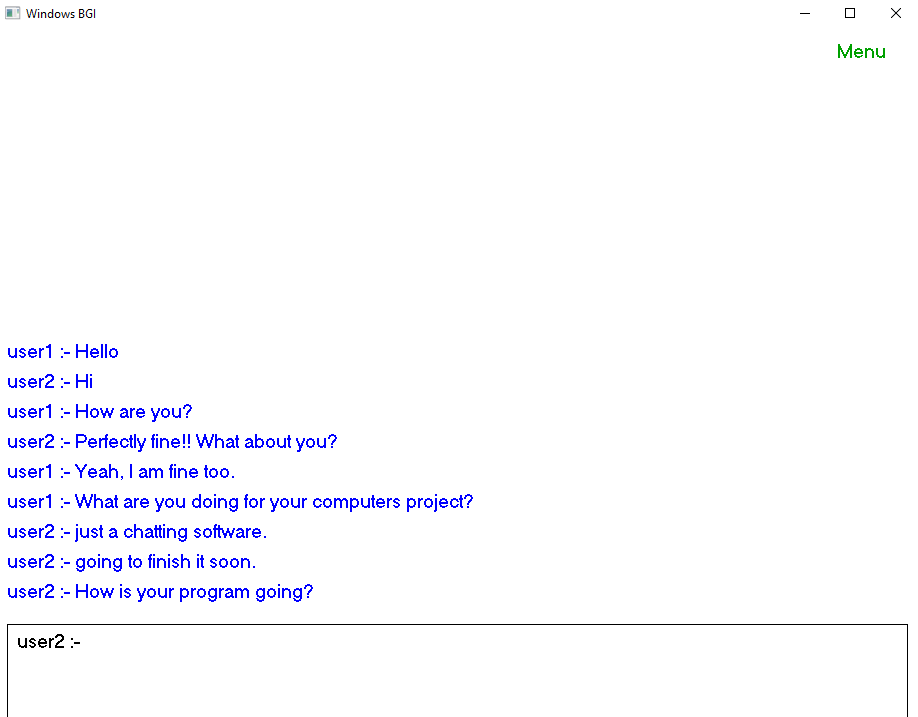
OUTPUT SCREENSHOTS



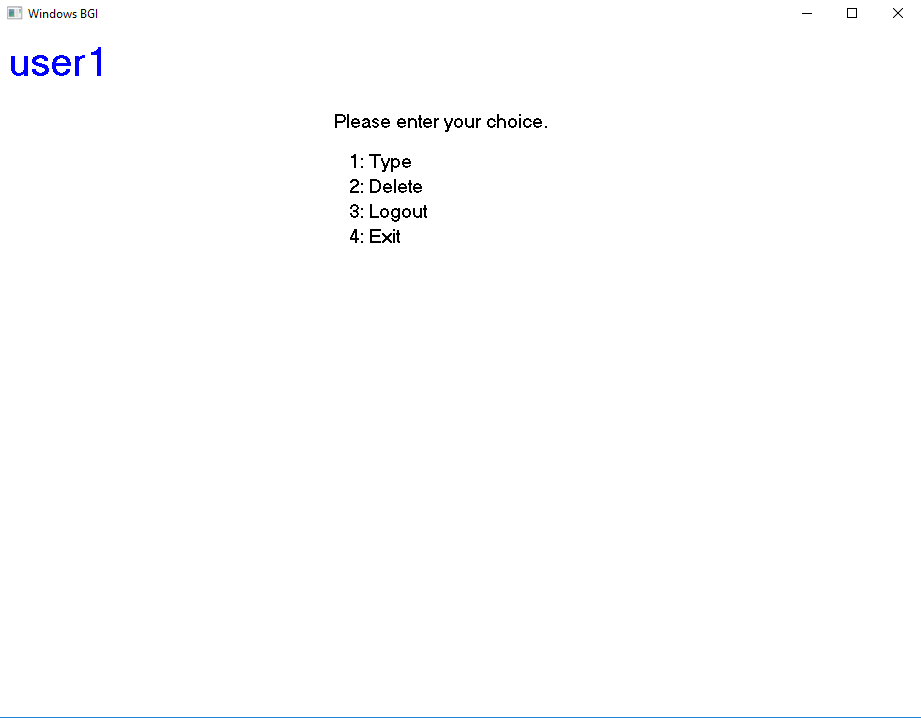
 Opening screen

Menu for the chat program

User login



User interface



Chat options for the user

SCOPE FOR IMPROVEMENT

Even though the current project works successfully with the use of simple principles of C++, text files and graphics.h, this application can definitely be expanded to have many more utilities.

1. **Emoticon sharing**: This was a feature that was put into the project but proved to be very buggy given the algorithm the program followed. Nevertheless it is highly feasible to have emoticons which the user can share instead of just messages.
2. **Accounts**: Another feature that was added but proved to be buggy was the account feature, which gave the user the option of logging in or creating an account with a personal password. This would also give the user a colour of their own for the chat application interface, to differentiate between users while going through the conversation. This also can be used as means to offer the options of personal chat, giving access to the conversation to only some members.
3. **Better interface**: Better graphics can be used to make the application more refined and usage of advanced animation in order to make the app more interactive and user friendly, doing away with the need for keyboard input for anything other than typing the actual message.
4. Increased amenities such as voice recording/gif/picture sharing/video calling, etc.
5. Can be connected to the internet to allow wider range of transmission and communication

**Bibliography**

1. Wikipedia-the free encyclopedia
2. [www.bloodshed.net/devcpp.html](http://www.bloodshed.net/devcpp.html)
3. C++ for class 12 by Sumita Arora