table of contents

Storyboard 2

User manual 3

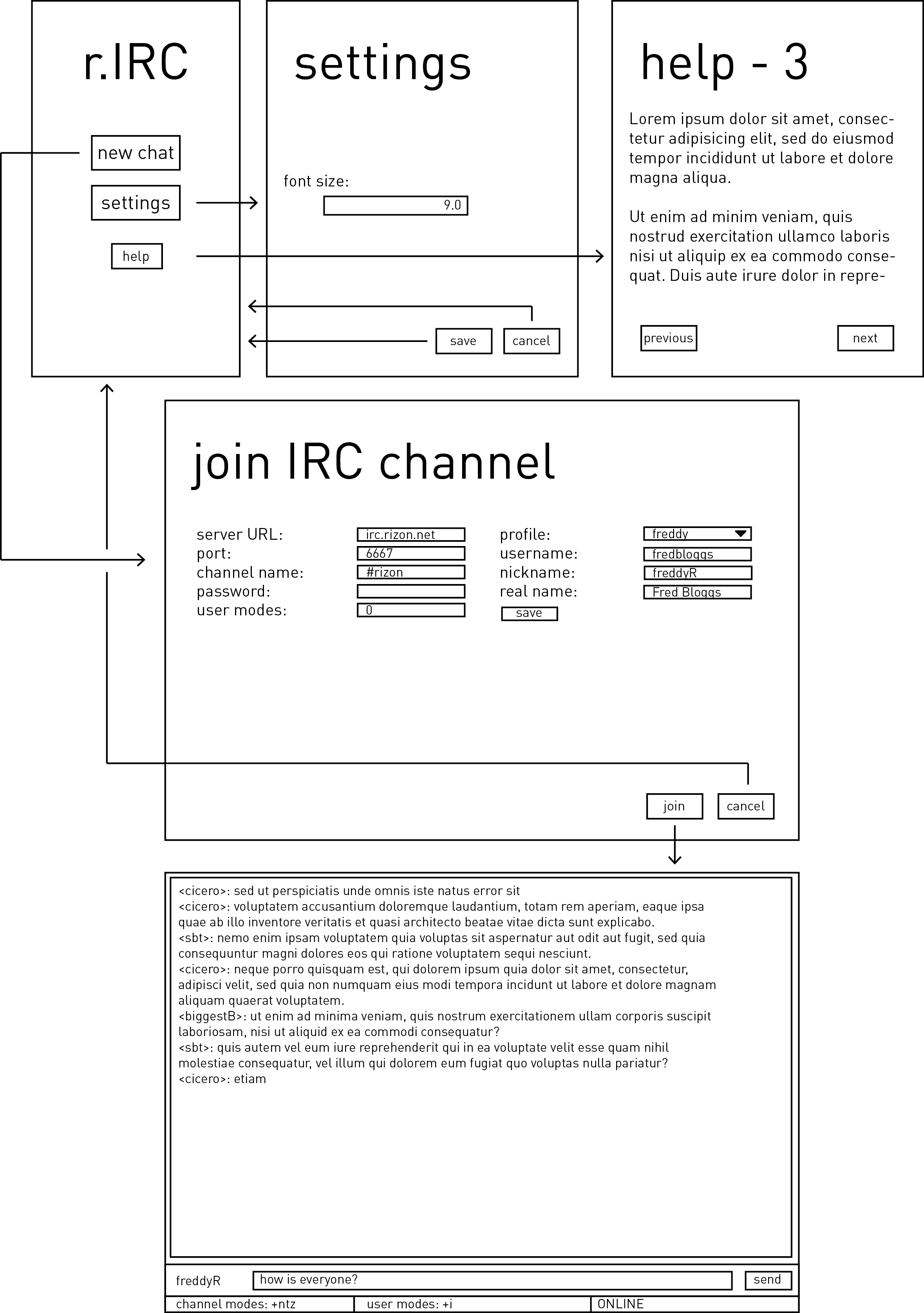
Installation manual 6

Project report 7

Peer check reports 9

Bibliography 10

# storyboard



# user manual

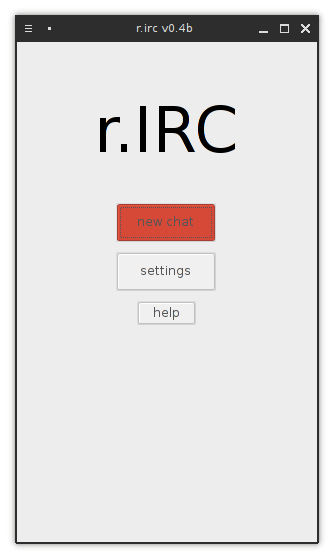
This user manual is designed for users that have little to no prior experience in using Internet Relay Chat. More experienced users are advised to conduct individual online research into the RFC Internet Relay Chat protocol.

What is IRC?

IRC stands for Internet Relay Chat, and is a method of online text-based communication between computing devices. r.IRC is an IRC client: it serves as an interface for humans to send and receive messages via IRC in a user-friendly manner.

Getting started

r.IRC comes pre-configured with optimal settings to allow for inexperienced users to avoid any steep learning curves. Press the New chat button to begin the process of connecting to an IRC server.

A new window will appear with input boxes to configure the connection. Many IRC servers exist on the Internet, and the user is encouraged to undertake further research in order to locate servers and channels that likeminded individuals may frequent. Popular IRC servers include:

* Rizon: irc.rizon.net
* freenode: irc.freenode.net
* QuakeNet: irc.quakenet.org

The port field designates the network port that your computer will use to connect to the server; the default is 6667 and thus this field is pre-emptively filled.

IRC channels are the individual “chatrooms” on an IRC server, wherein users can communicate with others in the same channel. Similar to servers themselves, many popular channels exist on large IRC servers, and users are encouraged to further research to find channels that suit their interests.

The password is an optional field used when connecting to password-restricted channels. Most channels are open and do not require a password, and this field can be left blank if the user is not trying to join a password-guarded channel.

User modes are flags that describe how the IRC server and all other users will view and interact with the user. There are a multitude of user modes, such as the invisibility flag which, when set, does not alert others in the IRC channel to the invisible user’s presence unless he or she sends a message.

IRC requires three identifying data fields from the user. These are known as nickname, username, and real name. The nickname is the most important of these, as it is the name displayed to other users when messages are sent. Nicknames have a maximum length of nine (9) characters and cannot contain several non-alphanumeric characters, as well as requiring a non-numeric character as the first character.

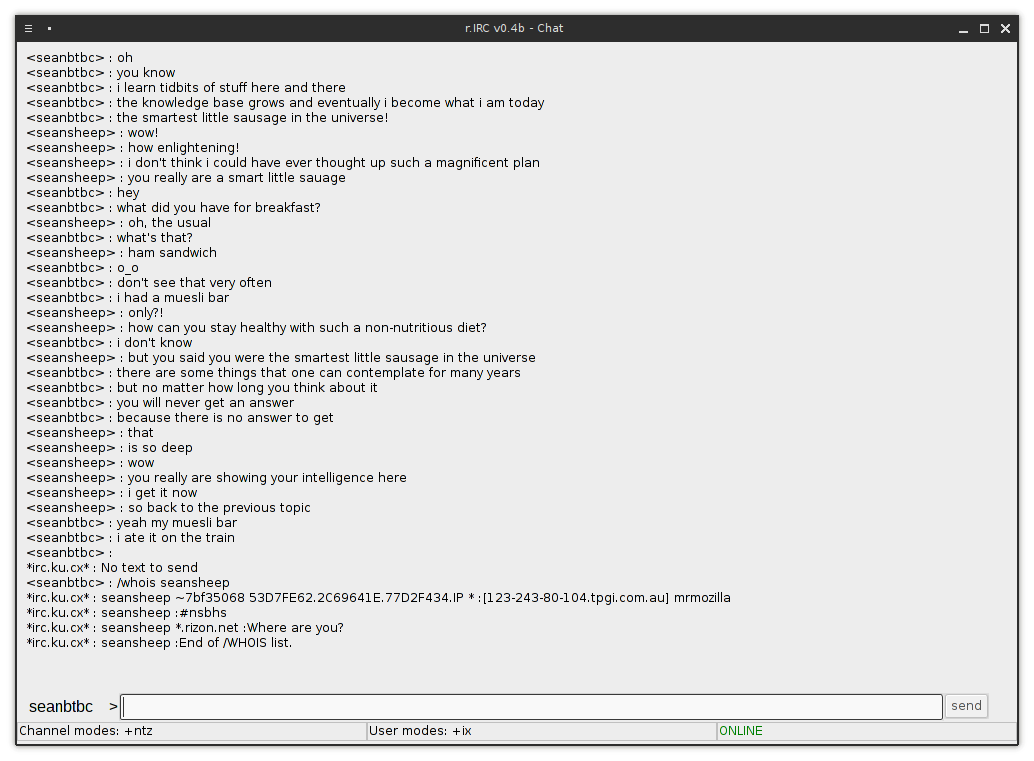
The username is the identifying name of the user on their current host. It is required in order for the IRC server to properly communicate with the client.

The real name is displayed to other users when they run the whois command. It does not necessarily have to be your real name.

One of r.IRC’s distinguishing features is its incorporation of a profile system, which saves user-defined sets of usernames, nicknames, and real names. This allows users to save significant amounts of time when using IRC with a frequently reused online identifier. Profiles can be saved and then selected using a drop-down menu, which automatically fills the relevant form fields with the saved data.

After all the aforementioned fields are filled, press the Join button and wait several moments for you to be connected to the IRC channel of your choice.

Interface



The main interface of r.IRC is extremely simple, consisting of a history pane, a message input box, and a status bar.

The history pane occupies the bulk of the window, and details the past messages and the output of any IRC commands.

The message input box is under the history pane, and allows the user to enter messages and IRC commands to be sent to the channel using the send button.

The status bar is at the bottom, and contains various statistics about your connection to this channel, including your current nickname, the current channel modes, your current user modes, and whether your connection to the channel is up or down.

# installation manual

Installation of r.IRC is relatively simple; in fact, it technically is not an installation process at all! r.IRC is a completely self-contained package, which means:

* All files required to run r.IRC are inside the provided package
* Removing r.IRC is just as simple as installing and will not leave any residue files
* Administrator privileges are generally not required to run r.IRC
* The application is completely portable – you can place it on a writable portable storage device such as a USB flash drive and it will work on any supported computer

To start using r.IRC, one must ensure they meet the following pre-requisites:

* A computer running either:
  + Windows 7 or above
  + Mac OS X Snow Leopard or above
  + GNU/Linux 3.x
* An installed Java Runtime Environment supporting Java SE 6+, such as:
  + Oracle Java Platform SE 6+
  + OpenJDK 6+
  + Sun JRE 6+

The installation process is as follows:

1. Ensure you satisfy all the pre-requisites listed above.
2. Browse and locate the folder named bin in your provided package.
3. Copy the folder and all its contents to your desired location (optional)
4. Open the bin folder and run the rirc.jar Java executable.

# project report

Reflection on work

The previous eight months of development have resulted in the application that is r.IRC today. Aspects of particular note during the process include time management, quality of craftsmanship, and the self-learning process.

Overall, time management has been of a high standard, with all mid-process checks and deadlines being met without the need for last-minute rushing. Changes or omissions of requirements initially outlined in the specification were primarily a result of a lack of technical feasibility rather than an inability to meet deadlines.

Contrastingly, quality of craftsmanship is more varied, especially when comparing work performed during different time periods. Although an effort has been made to ensure a consistently high quality across the entire development process, demotivation as a result of frustration often led to significantly lower quality output during certain time periods, particularly leading up to a necessary specification change. In reflection, following a more structured and better planned approach to developing software modules would have improved overall quality of craftsmanship.

An increasingly important concept in the software development process is the requirement of substantial amounts of self-learning. This is especially relevant due to the ever-evolving nature of the utilities used in the development of r.IRC, including the Shoes 4 GUI toolkit, which is still in early alpha stages. The trade-off of being able to access the newest and greatest features is the need to education oneself in these features on a regular basis, and the large time investment not only at once but also over the entire development process. In hindsight, the use of another GUI toolkit, even a less compatible and more unfamiliar utility such as the C-based GTK or the C++-based Qt Framework, may have led to increased time efficiency due to their stability.

Future direction

r.IRC may be complete and released, but there is still much that could be improved and augmented to the application. The most significant of these are limited by the development progress of the Shoes 4 GUI toolkit, which has previously hindered elements such as internal frame scrolling as well as causing a requirements shift away from a tabbed interface towards a multi-window approach. As Shoes 4 proceeds through its development process, it is expected that such features will be included and thus r.IRC will be able to support previously changed or removed requirements as well as adapt to new requirements.

Examples of future development concepts for r.IRC include:

* Extended message history box utilising internal scrolling
* Support for commonly used IRC colour codes
* Optional GUI elements to aid users in using IRC commands
* Support for third-party developed extensions

# peer check reports

I, Sean Batongbacal, made an informal peer check of r.IRC and found it to be of acceptable quality.

-

I, Austin Tankiang, have concluded an informal peer check of r.IRC, and have assessed it to be satisfactory.

-

# bibliography

Full source code

Full source code for r.IRC is available at: <www.github.com/mrmozilla/r.irc>

Utilities

Lovitt, Michael, 2014. *Rubular: a Ruby regular expression editor*. [online] Available at: <www.rubular.com>

Shoes Team, 2014. s*hoes/shoes4*. [online]

Available at: <www.github.com/shoes/shoes4>

Guides and documentation

Morin, Michael, 2014. *Parsing Using Named Capture Methods*. [online]

Available at: <ruby.about.com/od/irc/ss/Parsing.htm>

Ruby Community, 2014. *Documenting the Ruby Language*. [online]

Available at: <www.ruby-doc.org>

Shoes Team, 2013. *The Shoes Manual*. [online]

Available at: <www.shoesrb.com/manual/Shoes.html>

Network Working Group, 2000. *RFC 2812 – Internet Relay Chat: Client Protocol*. [online]

Available at: <tools.ietf.org/html/rfc2812>

Stack Overflow Community, 2014. *Stack Overflow*. [online]

Available at: <www.stackoverflow.com>

Special thanks

Austin Tankiang for invaluable input on the Ruby programming language and for permitting use of his IRC channel on Rizon for testing purposes.

Jesse Zhang for excellent suggestions regarding the intricacies of the IRC protocol and for putting up with the flood of debugging messages sent on aforementioned channel.

Victor H. Zhang for constructive advice relating to the implementation of threading in the Java programming language.