table of contents

## part one

Discussion of language choice 2

Platform and hardware considerations 2

Key modules: list and description 3

Files: definition and description 4

## part two

Main module: IPO chart 5

All modules: structure chart 6

startUp: pseudocode 7

startUp: flowchart 8

Data dictionary 9

Test data 11

part one

# discussion of language choice

The language of choice for this project is Ruby version 1.9.3. Ruby is a programming language which boasts several features that make it a suitable selection for r.IRC.

Ruby follows the object-oriented paradigm, which allows for the easy creation of custom classes and objects suitable for use with r.IRC. Furthermore, individual objects can be given methods by which the data inside the object can be manipulated whilst being completely isolated from its environment. The use of developer-defined objects and methods allows for increased understanding, as well as ease of maintenance and increased modularity.

Ruby’s popularity has allowed for over 70 thousand user-made extensions, known as gems. These can be freely added to a base Ruby installation, and provide extra functionality in the form of extra classes and libraries. Such extensibility has encouraged the choice of Ruby as the programming language, primarily due to the availability of GUI libraries such as Shoes.

# platform & hardware considerations

r.IRC is designed to be cross-platform, officially supporting Windows 7+ and GNU/Linux. This is due to the interpretive nature of Ruby and the multi-platform capabilities of Shoes, wherein the latter makes use of native window controls to maintain an immersive application experience across operating systems. Mac OS X is not officially supported due to lack of test machines running OS X, but an untested OS X executable may be provided on request.

r.IRC does not feature any computationally nor graphically intense functionality, and thus is not particularly hardware taxing. Any modern or slightly old machine should be able to run r.IRC as intended. The performance of the host operating system can be used as a pre-emptive performance indicator; a generally smooth experience with the operating system should directly correlate with the capability to operate r.IRC without major hardware implications.

# key modules: list & description

|  |  |
| --- | --- |
| main | Calls all the other modules, in their specified order |
| startUp | First module called by main. Reads saved user profiles from file into class instances. Reads settings from file into class instances. |
| rirc | Primary container module. Starts the GUI, and splits execution into two threads: rircChat and rircInterface. Threads close when an exit flag is triggered. |
| rircChat | Runs in tandem with rircInterface. Controls the primary chat area, including history panel, outgoing message box, and participant list. Also controls tab bar – allows for easier implementation of switching of tabs. Splits execution into three threads: rircChatBar, rircChatReceive, and rircChatSend. The latter two have separate instances for each chat tab. |
| rircChatReceive | Runs in tandem with rircChatSend. Infinite loop to scan for incoming messages, displaying them in the history panel. |
| rircChatSend | Runs in tandem with rircChatReceive. Infinite loop to send entered messages to the server. Calls other functions when certain commands are entered. |
| rircInterface | Runs in tandem with rircChat. Controls the rest of the interface, including menu bar, status bar, and top controls. Exit is triggered from this module. |
| newTab | Called when a new chat tab is opened, including at start of GUI. Launches a dialog prompting input of details for new chat tab. Can be called by both rircChatBar and rircInterface. |
| closeTab | Called when a chat tab is closed. Sends commands to leave the IRC channel and close the connection to the server. Unsets the variables related to that particular session. |
| shutDown | Called by main when the rirc module exits. Closes the GUI. Unsets variables and classes. |

# files: definition and description

|  |  |
| --- | --- |
| profilesFile | This file stores all the user defined profiles for r.IRC. The layout for each profile is as follows:  [profileID] {must be positive unique integer}  [profile name]  [nickname]  [real name]  [username] {optional, value is ‘-1’ if empty} |
| settingsFile | This file stores all the preferences and settings used by r.IRC. Currently only holds a single value, that of the default profile referenced by profileID. |
| sessionsFile | This file stores information about prior chat sessions. For example:  [20140101:145932] Connection established to irc.rizon.net  [20140101:145951] Joined channel #channel  [20140101:151654] Left channel #channel  [20140101:151654] Closed connection to irc.rizon.net  Information relating to the content of messages is NOT stored. |

Notes:

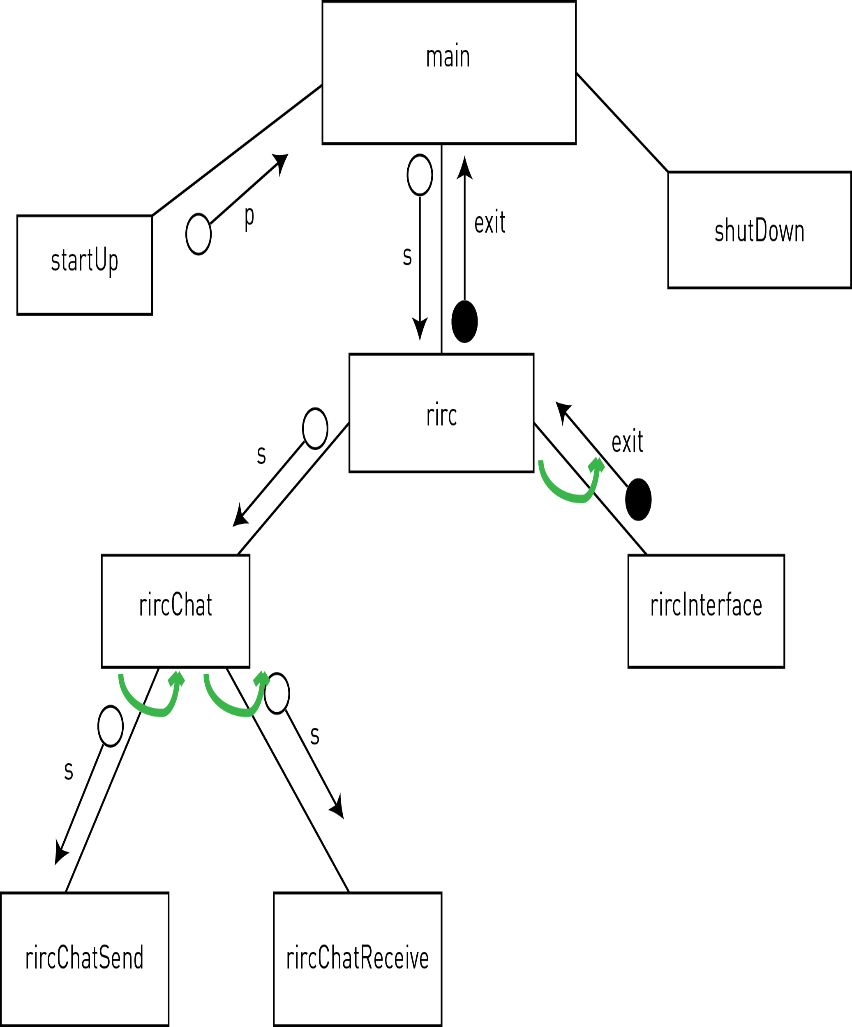
* r.IRC configuration files are accessed in such a way to allow for commented lines and blank lines to be ignored.
* Entire lines may be commented out using Bash-style comments, with a ‘#’ as the first character of the line.
* Lines with no characters or lines only containing a newline character are outright ignored.

part two

# main module: ipo chart

|  |  |  |
| --- | --- | --- |
| input | processes | output |
| Server location  Connection port | Attempt connection to server | “Successful connection”  OR  “Connection failed” |
| Username  Real name  Nickname | Send identifying details  to server |  |
| Channel name | Send join channel request  to server |  |
| Outgoing messages | Receive incoming messages  Send outgoing messages | Incoming messages |

# all modules: structure chart



# startup: pseudocode

BEGIN startUp

OPEN profilesFile

READ profileCount FROM profilesFile

FOR i = 1 TO profileCount STEP 1

FOR j = 1 TO 5 STEP 1

READ profiles[i][j] FROM profilesFile

NEXT j

NEXT i

CLOSE profilesFile

OPEN settingsFile

FOR i = 1 TO 1 STEP 1

// only one setting currently

// will add more settings later

READ settings[i] FROM settingsFile

NEXT i

CLOSE settingsFile

FOR i = 1 TO profileCount STEP 1

IF profiles[i][1] == settings[1] DO

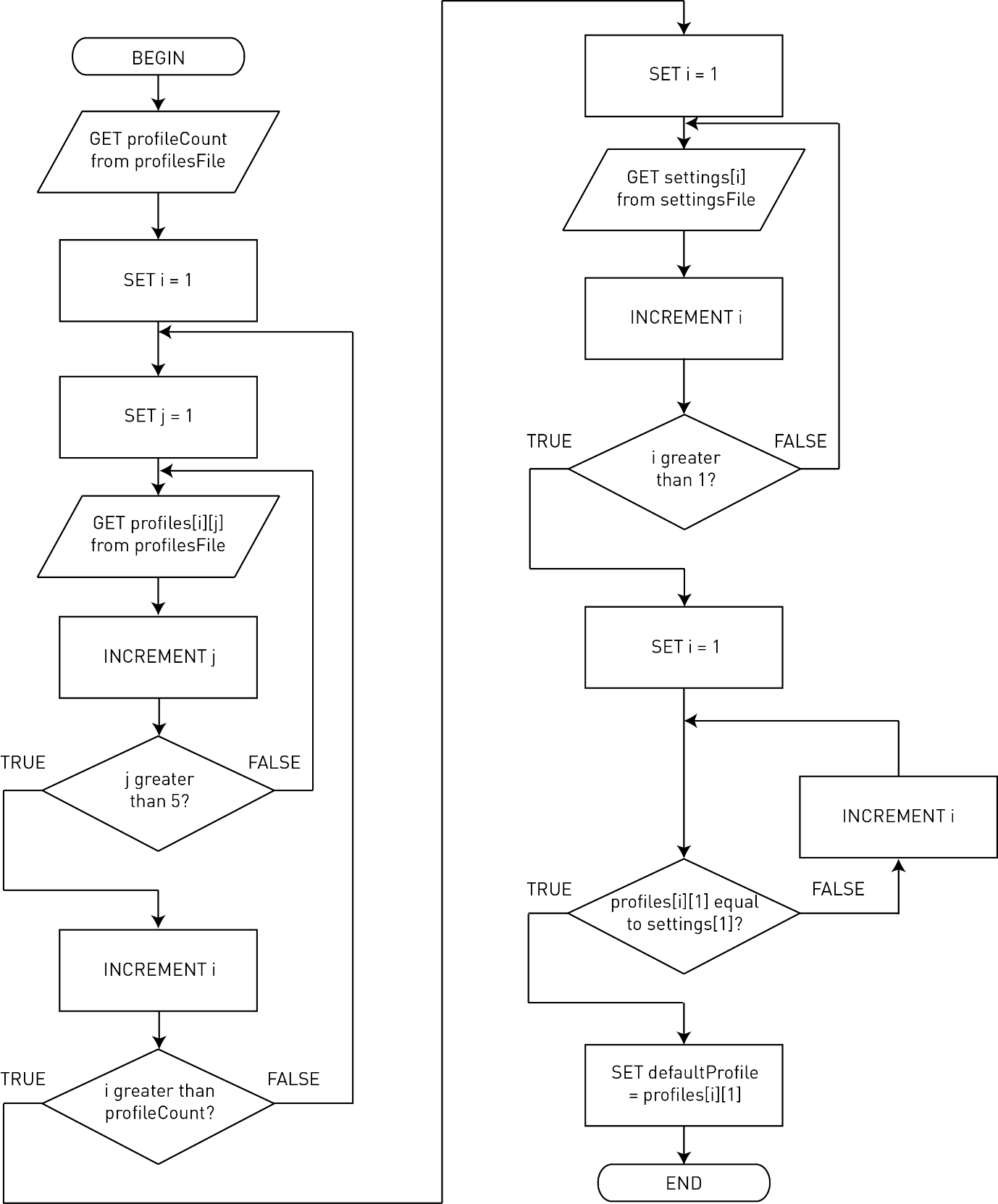
SET defaultPorilfe = profiles[i][1]

END IF

NEXT i

END startUp

# startup: flowchart



# data dictionary

Variables

|  |  |  |  |
| --- | --- | --- | --- |
| name | length | type | description |
| s | {variable} | Array of TCPSockets | Stores the TCPSocket to each server. One for each chat tab. |
| p | {variable} | Array of Profiles | Stores the Profile used for each chat tab. |
| c | {variable} | Array of strings | Stores the channel joined on each chat tab. |
| incomingMessage | 512 | String | Stores the next incoming message |
| outgoingMessage | 512 | String | Stores the next outgoing message |
| threads\_rirc | {variable} | Array of Threads | Stores the two threads rircChat and rircInterface |
| threads\_rircChat | {variable} | Array of Threads | Stores the threads for each tab in rircChat |

Class: Profile

|  |  |  |  |
| --- | --- | --- | --- |
| name | length | type | description |
| profileID | 3 | Integer | Identifies the profile. Unique. |
| profileName | 32 | String | User-friendly profile name. |
| nickname | 9 | String | Display name |
| realname | 64 | String | Name shown when using /whois |
| username | 64 | String | Optional identifying name |
| load | - | Method | Loads profiles from file into an array of class instances |
| create | - | Method | Writes a new profile to file |
| delete | - | Method | Removes a profile from file |
| modify | - | Method | Saves changes to a profile to file |
| count | - | Method | Returns the number of loaded profiles |

Class: Profile

|  |  |  |  |
| --- | --- | --- | --- |
| name | length | type | description |
| defaultProfile | 3 | Integer | Identifier of the default profile |
| load | - | Method | Loads settings from file into a class instance |

# test data

File: profilesFIle

|  |  |  |
| --- | --- | --- |
| name | value | expected output |
| profileID | 0 | {pass} |
| profileID | 999 | {pass} |
| profileID | -1 | {fail} |
| profileID | 1000 | {fail} |
| profileID | asdf | {fail} |
| profileID | {null} | {fail} |
| nickname | as3f | {pass} |
| nickname | {null} | {fail} |
| nickname | qwertyuio | {pass} |
| nickname | qwertyuiop | {fail} |

rircChatSend

|  |  |
| --- | --- |
| value | expected output |
| 0 | PRIVMSG :0 |
| {null} | {null} |
| /bye | QUIT |
| this is a message | PRIVMSG :this is a message |
| //this is a fwdslash | PRIVMSG :this is a fwdslash |
| /this is a fwdslash | /this: no such command |