Ruby 101

Intermediate Ruby: Building the LogEntry Class

Intermediate Ruby:

Each LogEntry object

IP Address
Time Stamp
Request
Response code
File Size
Http Referer
User Agent

Method to load data
Method to parse a row - Already created

Ruby 101

Intermediate Ruby:

```
def LogEntry
    attr_accessor :ip_address, :time_stamp, :request,
        :response_code, :file_size, :http_referer, :user_agent
    def initialize row = nil
        if row
            row.gsub! /\t/, "    "
            match_data = parse_row row
            set_properties match_data
        end
    end
    def set_properties match_data
        @ip_address = match_data[1]
        @request = match_data[10]
        @response_code = match_data[11]
        @file_size = match_data[12]
        @http_referer = match_data[13]
        @user_agent = match_data[14]
    end
```

Intermediate Ruby:

```
Changes to LogFile class:
        def initialize
                cd "/"
                @log_entries = Array.new
        end
def load_file
        if File.file?(@file_path + @directory.entries[@directory_index])
                @file_name = @directory.entries[@directory_index]
                log_array = IO.readlines(@file_path + @file_name)
                log_array.each_with_index do |log, index|
                        @log_entries[index] = LogEntry.new log
                end
                @log_entry_index = 0
                @list_start = 0
                true
         else
                false
         end
end
```

Intermediate Ruby:

Controller: keep the indexes within boundaries:

```
def file_dialog_move increment
        @log_file.directory_index += increment
        @log_file.directory_index = 0 if @log_file.directory_index < 0
        @log_file.directory_index = @log_file.directory.entries.length - 1 if
                @log_file.directory_index > @log_file.directory.entries.length - 1
…


def log_list_move increment
        @log_file.log_entry_index += increment
        @log_file.log_entry_index = 0 if @log_file.log_entry_index < 0
        @log_file.log_entry_index = @log_file.log_entries.length - 1 if
                @log_file.log_entry_index > @log_file.log_entries.length - 1
…
```

Intermediate Ruby:

View updates: in LogListView's update method

```
def update log_file
...
    total_columns = $stdout.winsize[1] - 28
    # (space after ip address, response code, and file size)
    text_column_size = total_columns / 3
    row = "\e[K" + entry.ip_address +
        "\e[17G" + entry.request.slice(0, text_column_size) +
        "\e[#{text_column_size + 17 + 1}G" + entry.response_code +
        "\e[#{text_column_size + 17 + 1 + 4}G" + entry.http_referer.slice(0,
                                                 text_column_size) +
        "\e[#{2 * text_column_size + 17 + 2 + 4}G" + entry.user_agent.slice(0,
                                                 text_column_size) +
        "\e[#{3 * text_column_size + 17 + 3 + 4}G" + entry.file_size.slice(0, 7) +

                                                 "\n"
    row = red(row) if index == log_file.log_entry_index
    print row
...
```