Gra w zgadywanie

Marcin Baranowski

Grudzień 2022

1 Wstęp

Przedstawiona dokumentacja obejmuje zagadnienia dotyczące projektu zaliczeniowego pod tytułem "Gra w zgadywanie" napisanym w języku Ruby, na przedmiot "Programowanie w języku Ruby" na Uniwerystecie Gdańskim w ramach piątego semestru studiów licencjackich na kierunku "Informatyka".

2 Poziomy wykonania programu

2.1 Poziom 1

- wczytywanie i sprawdzanie czy podana liczba jest taka sama.
- jeżeli liczba liczba została odgątnięta, program przesyła gratulacje oraz kończy swoje działanie.
- jeżeli nie, program podaje czy dana liczba jest za mała, czy za duża oraz prosi o ponowne podanie liczby.

2.2 Poziom 2

- po wpisaniu słowa "koniec", program kończy swoje działanie, oraz przesyła wiadomość "żegnaj".

2.3 Poziom 3

- po odgadnięciu liczby program pyta użytkownika, czy chciałby zagrać jeszcze raz.
- jeżeli użytkownik odpowie "tak", gra rozpocznie się ponownie z nową wylosowaną liczbą.

2.4 Poziom 4

- po odgadnięciu liczby pytamy użytkownika o imie oraz zapisujemy go do struktry danych.
- jako struktury danych używamy mapy/struktury.
- po zakończonej grze, zapisujemy w ilu próbach użytkownik odgadł liczbę.
- na zakończenie programu wypisujemy listę użytkowników, wraz z ich wynikami

2.5 Poziom 5

- program zapisuje wyniki użytkowników do pliku hallOfFame.txt, tworząc w ten sposób tabele wyników
- program za każdym uruchomieniem wczytuje dane z pliku do listy wyników
- program informuje o pobiciu rekordu globalnego

3 Instrukcja obsługi programu

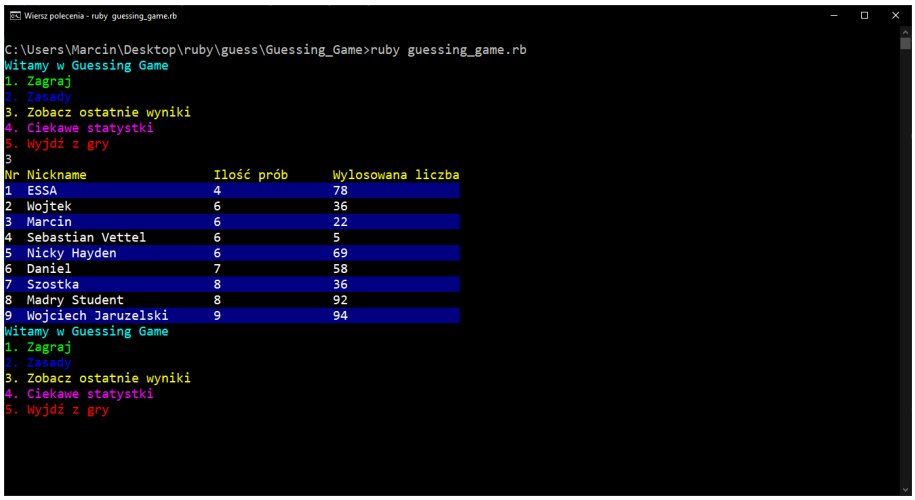
3.1 Ale najpierw jak go uruchomić?

Oczywiście, żeby uruchomić program należy posiadać zainstalowany Ruby'yego na swoim komputerze osobistym. Później w terminalu, należy zainstalować bibliotekę colorize, za pomocą polecenia **gem install colorize**. Po zainstalowaniu biblioteki uruchamiamy program komendą **ruby skrypt.rb**

3.2 Menu główne - co, gdzie i jak

Po skompilowaniu naszego programu, ukaże się na menu główne. Zobaczymy w nim cztery dostępne opcje tj. "zagraj", "zasady, "sprawdź wyniki", "ciekawe statystki" oraz "wyjście". Do użycia jednej z tych opcji należy wpisać odpowiadajacy jej numer, po czym kliknąć enter.

Rysunek 1: Menu główne programu



Rysunek 2: Tabela z wynikami

3.3 Rozgrywka

Po wybraniu opcji "zagraj" program poprosi nas o podanie liczby. Jeżeli zgadniemy "ukrytą" liczbę, program poprosi nas o podanie swojego imienia oraz zapisze nasz wynik do pliku. Jeżeli nie zgadniemy, program powiadomi nas czy nasza wpisana liczba jest za mała, czy za duża oraz poprosi o ponowne podanie liczby, do momentu w którym nie odgadniemy poprawnej liczby.

4 Opis działania programu - czyli co w kodzie piszczy

4.1 Zapisywanie wyników użytkownika

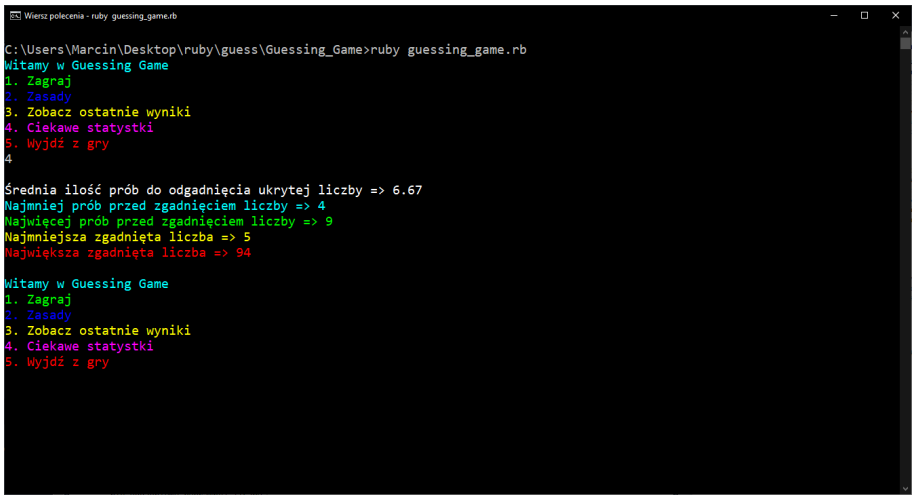
Zapisanie wyniku użytkownika odbywa się za pomocą struktury. Początkowo przez pierwsze etapy używałem hashmapy, która zawierała w sobie imie gracza jako klucz oraz jego wynik jako wartość, ale doszedłem do wniosku, że nie jest mi ona potrzebna, a nawet mnie ogranicza. Dzięki temu powstała osobna struktura gracza, zawierająca jego imie, ilość prób, zgadywaną liczbe oraz date. Taka struktura jest następnie wpisywana do pliku tekstowego, który zawiera wszystkie wyniki poprzednich użytkowników. Wczytanie tych wyników z pliku następuje za pomocą listy, która przechowuje wszystkie struktury graczy.

4.2 Kolorowe napisy

Zawsze warto dodać jakiś element graficzny, nawet jeżeli jest on niewielki, prawda? Tak jest i w tym przypadku. Do kolorowania wyświetlanych poleceń została wykorzystana bibiloteka colorize, o której już wcześniej wspomniałem w punkcie 3.1. Jest to zewnętrzna biblioteka, którą należy zainstalować. Umożliwia ona edycje wyświetlanych poleceń w szesnastu różnych kolorach! Umożliwia również zakolorowanie tła polecenia, tak jak zostało to użyte, przy okazji tablicy wyników.

4.3 Statystki

Funkcja stats() analizuje listę wyników, z której wylicza średnią ilość rund potrzebnych na zgadnięcie liczby, najmniejszą oraz największą wylosowaną liczbe. Jak widać, nie jest to za bardzo rozbudowana funkcjonalność programu. Na pewno jest to jedna z rzeczy, którą można byłoby rozbudwować w dalszych etapach tworzenia programu.



Rysunek 3: Zestawienie różnych statystk

5 Kod programu

```ruby
# Author => Marcin Baranowski

# Guessing Game
# Punkt 1 - done
# Punkt 2 - done
# Punkt 3 - done
# Punkt 4 - done
# Punkt 5 - done
# Punkt 6 - not done
# Punkt 7 - not done
# Punkt 8 - not done

require 'date'
require 'colorize'


$gracz = Struct.new(:name, :counter, :number_to_guess, :date)

#fukcja zapisująca wynik gracza do pliku tekstowego
def saveResults(gracz)
    File.open("hallOfFame.txt", 'a') do |f|
        f.write
        ↪  "#{gracz[:name]},#{gracz[:counter]},#{gracz[:number_to_guess]},#{gracz[:date]}\n"
    end
end

#funkcja, która wypisuje graczy z pliku .txt do listy.
```

```ruby
def extractPlayers()
    hall_of_fame = []

    if(File.file?('hallOfFame.txt')) #jezeli plik txt istnieje
        File.foreach("hallOfFame.txt") { |each_line|  #dla każdej jego
        ↪  linii
            arr = each_line.split(",")
            hall_of_fame.push($gracz.new(arr[0], arr[1].to_i,
            ↪  arr[2].to_i, arr[3])) #wprowadź gracza do listy
        }
    end

    sorted_hall_of_fame = hall_of_fame.sort_by(&:counter)
    return sorted_hall_of_fame
end

def rules()
    print "\nProgram losuje liczbe z zakresu od 1 do 100. Twoim zadaniem
    ↪  jest odgadnięcie tej liczby.\n".light_green
    print "Jeżeli trafisz w poprawną liczbe, program ci pogratulujei
    ↪  zapisze twój wynik.\n".light_green
    print "Jeżeli nie odgadniesz program poinformuje Cię, czy podana
    ↪  liczba jest za mała, czy za duża.\n".light_green
    print "Powodzenia\n".light_green
    puts ""
end

def printResults()
    results = extractPlayers()

    if results.empty?()
        puts "Tablica wyników jest pusta!".light_red
        return
    end

    position = 1
    puts "%-2s %-24s %-15s %-18s".light_yellow % ["Nr", "Nickname",
    ↪  "Ilość prób", "Wylosowana liczba"]
    results.each { |p|
        if results.index(p) % 2 == 0
            puts "%-2d %-24s %-15d %-17d".colorize(:color =>
            ↪  :light_white, :background => :blue) % [position, p.name,
            ↪  p.counter, p.number_to_guess]
        else
            puts "%-2d %-24s %-15d %-17d".colorize(:color =>
            ↪  :light_white) % [position, p.name, p.counter,
            ↪  p.number_to_guess]
        end
        position += 1
    }
```

```ruby
end

#funkcja odpowiadająca za przedstawienie statystk np. średniej ilości prób wymaganej do odgadnięcia liczby
def stats()
    hall_of_fame = extractPlayers()
    sum = 0.0
    divider = 0.0
    numbers = []
    results = []

    hall_of_fame.each { |p|
            results.append(p.counter) #lista zawierająca ilość prób
            numbers.append(p.number_to_guess) #lista zawierająca liczby które zostały wylosowane do odgadnięcia
            sum += p.counter
            divider += 1.0
    }

    average = sum / divider
    puts "\nŚrednia ilość prób do odgadnięcia ukrytej liczby => #{'%.2f' % average.to_f}".light_white
    puts "Najmniej prób przed zgadnięciem liczby => #{results.min}".light_cyan
    puts "Najwięcej prób przed zgadnięciem liczby => #{results.max}".light_green
    puts "Najmniejsza zgadnięta liczba => #{numbers.min}".light_yellow
    puts "Największa zgadnięta liczba => #{numbers.max}".light_red
    puts ""
end

#funkcja sprawdzająca czy gracz ustanowił nowy rekord w grze
def checkIfNewRecord(player)
    hall_of_fame = extractPlayers()
    results = []

    hall_of_fame.each { |p|
        results.append(p.counter)
    }


    if results.min == nil
        puts "Wow, jako pierwszy wpisałeś sie na liste wyników. Tym samym zajmujesz pierwsze miejsce!"
    elsif player.counter < results.min
        puts "Właśnie ustanowileś nowy rekord w Guessing Game! Gratulacje".light_green
    end
end

```

```ruby
#funkcja zawierająca logike gry
def game()
    target = rand(1..100)
    game_over = false
    counter = 0

    puts "Teraz będziesz zgadywał liczbe".light_cyan

    while !game_over
        puts "Podaj liczbe".light_white
        input = gets
        counter += 1

        if input == "koniec\n"
            puts "żegnaj".light_red
            exit
        elsif input.to_i > target
            puts "za duża".light_red
        elsif input.to_i < target
            puts "za mała".light_yellow
        else
            puts "Brawo zgadłeś".light_green
            game_over = true
        end
    end


    puts "Podaj swoje imie i nazwisko, abyśmy mogli zapisać twój wynik".light_blue
    names = gets

    player = $gracz.new(names.strip, counter, target, Time.now) #tworzymy nowego gracza

    checkIfNewRecord(player) #sprawdzamy czy pobił rekord globalny
    saveResults(player)#zapisujemy go i jego wynik do pliku txt

    playAgain()
end

#funkcja odpowiedzialna za zapytanie gracza czy chce grac dalej
def playAgain()
    puts "Czy gramy jeszcze raz? (Y/N)".light_magenta

    choice = gets

    if choice.upcase == "Y\n"
        menu()
    elsif choice.upcase =="N\n"
        puts "Koniec działania programu".light_red
```

```
158         exit
159     else
160         puts "Nie ma takiej opcji".light_yellow
161         playAgain()
162     end
163 end
164 
165 #funkcja obsługująca menu główne
166 def menu()
167     begin
168         puts "Witamy w Guessing Game".light_cyan
169         puts "1. Zagraj".light_green
170         puts "2. Zasady".light_blue
171         puts "3. Zobacz ostatnie wyniki".light_yellow
172         puts "4. Ciekawe statystki".light_magenta
173         puts "5. Wyjdź z gry".light_red
174 
175         option = gets
176 
177         case option.to_i
178         when 1
179             game()
180         when 2
181             rules()
182             menu()
183         when 3
184             printResults()
185             menu()
186         when 4
187             stats()
188             menu()
189         when 5
190             puts "papa".light_magenta
191             exit
192         else
193             puts "Nie ma takiej opcji"
194             menu()
195         end
196     rescue Interrupt => e
197         puts "papa".light_magenta
198     end
199 end
200 
201 
202 menu()
```