

Gra w zgadywanie

Marcin Baranowski

December 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 Uniwersytecie 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 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 imię oraz zapisujemy go do struktury 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", "sprawdź wyniki", "ciekawe statystyki" oraz "wyjście". Do użycia jednej z tych opcji należy wpisać odpowiadający jej numer, po czym kliknąć enter.

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. Jeżeli gra nas sfrustruje to zawsze możemy z niej wyjść, nawet w trakcie zgadywania liczby. Wystarczy zamiast liczby wpisać "koniec".

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 imię 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 imię, ilość prób, zgadywaną liczbę 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 gracza.

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 biblioteka `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 Statystyki

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

5 Kod programu

```
1  # Author => Marcin Baranowski
2
3  # Guessing Game
4  # Punkt 1 - done
5  # Punkt 2 - done
6  # Punkt 3 - done
7  # Punkt 4 - done
8  # Punkt 5 - done
9  # Punkt 6 - not done
10 # Punkt 7 - not done
11 # Punkt 8 - not done
12
```

```

13 require 'date'
14 require 'colorize'
15
16 String.disable_colorization = false
17
18 $gracz = Struct.new(:name, :counter, :number_to_guess, :date)
19
20 def saveResults(gracz)
21   File.open("hallOfFame.txt", 'a') do |f|
22     f.write
23     ↪ "#{gracz[:name]},#{gracz[:counter]},#{gracz[:number_to_guess]},#{gracz[:date]}\n"
24   end
25 end
26
27 def extractPlayers()
28   hall_of_fame = []
29
30   if(File.file?('hallOfFame.txt'))
31     File.foreach("hallOfFame.txt") { |each_line|
32       arr = each_line.split(",")
33       hall_of_fame.push($gracz.new(arr[0], arr[1].to_i,
34       ↪ arr[2].to_i, arr[3]))
35     }
36   end
37
38   sorted_hall_of_fame = hall_of_fame.sort_by(&:counter)
39   return sorted_hall_of_fame
40 end
41
42 def printResults()
43   results = extractPlayers()
44   position = 1
45
46   results.each { |p|
47     if results.index(p) % 2 == 0
48       puts "#{position}. #{p.name}, ilość prób: #{p.counter},
49       ↪ zgadywana liczba #{p.number_to_guess}"
50       .colorize(:color => :light_white, :background => :blue)
51     else
52       puts "#{position}. #{p.name}, ilość prób: #{p.counter},
53       ↪ zgadywana liczba #{p.number_to_guess}"
54       .colorize(:color => :light_white)
55     end
56     position += 1
57   }
58 end
59
60 def stats()
61   hall_of_fame = extractPlayers()
62   sum = 0.0

```

```

59     divider = 0.0
60     numbers = []
61     results = []
62
63     hall_of_fame.each { |p|
64         results.append(p.counter)
65         numbers.append(p.number_to_guess)
66         sum += p.counter
67         divider += 1.0
68     }
69
70     average = sum / divider
71     puts "\nŚrednia ilość prób do odgadnięcia ukrytej liczby =>
72     ↪ #{average.to_f}".light_white
73     puts "Najmniej prób przed zgadnięciem liczby =>
74     ↪ #{results.min}".light_cyan
75     puts "Najwięcej prób przed zgadnięciem liczby =>
76     ↪ #{results.max}".light_green
77     puts "Najmniejsza zgadnięta liczba => #{numbers.min}".light_yellow
78     puts "Największa zgadnięta liczba => #{numbers.max}".light_red
79     puts ""
80 end
81
82 def checkIfNewRecord(player)
83     hall_of_fame = extractPlayers()
84     results = []
85     hall_of_fame.each { |p|
86         results.append(p.counter)
87     }
88
89     puts results.min
90
91     if player.counter < results.min
92         puts "Właśnie ustanowiłeś nowy rekord w Guessing Game!
93         ↪ Gratulacje".light_green
94     end
95 end
96
97 def game()
98     target = rand(1..100)
99     game_over = false
100     counter = 0
101
102     puts "Teraz będziesz zgadywał liczbę".light_cyan
103
104     while !game_over
105         puts "Podaj liczbę".light_white
106         input = gets
107         counter += 1

```

```

105         if input == "koniec\n"
106             puts "żegnaj".light_red
107             exit
108         elsif input.to_i > target
109             puts "za duża".light_red
110         elsif input.to_i < target
111             puts "za mała".light_yellow
112         else
113             puts "Brawo zgadłeś".light_green
114             game_over = true
115         end
116     end
117
118     puts "Podaj swoje imie i nazwisko, abyśmy mogli zapisać twój
119     ↪ wynik".light_blue
120     names = gets
121
122     player = $gracz.new(names.strip, counter, target, Time.now)
123
124     checkIfNewRecord(player)
125     saveResults(player)
126
127     playAgain()
128 end
129
130
131 def playAgain()
132     puts "Czy gramy jeszcze raz? (Y/N)".light_magenta
133
134     choice = gets
135
136     if choice.upcase == "Y\n"
137         menu()
138     elsif choice.upcase == "N\n"
139         puts "Koniec działania programu".light_red
140         exit
141     else
142         puts "Nie ma takiej opcji".light_yellow
143     end
144 end
145
146 def menu()
147     begin
148         puts "Witamy w Guessing Game".light_cyan
149         puts "1. Zagraj".light_green
150         puts "2. Zobacz ostatnie wyniki".light_yellow
151         puts "3. Ciekawe statystyki".light_magenta
152         puts "4. Wyjdź z gry".light_red
153     end

```

```

154         option = gets
155
156         case option.to_i
157         when 1
158             game()
159         when 2
160             printResults
161             menu()
162         when 3
163             stats()
164             menu()
165         when 4
166             puts "papa".light_magenta
167             exit
168         else
169             puts "Nie ma takiej opcji"
170         end
171     rescue Interrupt => e
172         puts "papa".light_magenta
173     end
174 end
175
176
177 menu()

```