

# **Dokumentacja wymagań projektowych**

## **Mastermind**

**Wersja 1.01**

**Przygotowane przez: Maciej Siciarz**

**Tomasz Słoczyński**

**IS Grupa 3**

**17.4.2015**

# **Spis treści**

## **Spis treści**

### **Historia edycji**

#### **1. Wprowadzenie**

- 1.1 Cel
- 1.2 Przeznaczenie dokumentu
- 1.3 Zakres produktu
- 1.4 Odniesienia

#### **2. Opis ogólny**

- 2.1 Perspektywa produktu
- 2.2 Funkcje produktu
- 2.3 Aktorzy i charakterystyki użytkowników
- 2.4 Środowisko uruchomieniowe

#### **3. Wymagania zewnętrzne**

- 3.1 Opis interfejsu użytkownika
- 3.2 Wymagania sprzętowe
- 3.3 Wymagane oprogramowanie

#### **4. Funkcje systemu**

- 4.1 Konfiguracja gry
- 4.2 Rozgrywka
- 4.3 Koniec rozgrywki
- 4.4 Pomoc

#### **5. Wymagania pozafunkcjonalne**

#### **6. Wymagania niefunkcjonalne**

#### **7. Plan ewaluacji**

- 7.1 Testy jednostkowe
- 7.2 Testy integracyjne
- 7.3 Testy akceptacyjne

#### **Dodatek A: Słownik**

## Historia edycji

Imię	Data	Powód	Wersja
Maciej Siciarz	16.4.2015	Wstępna wersja dokumentu	0.9
Tomasz Słoczyński	16.4.2015	Ostateczna wersja dokumentu	1.0
Maciej Siciarz	17.4.2015	Drobne poprawki	1.01

## 1. **Wprowadzenie**

### 1.1 **Cel**

Dokument opisuje cele i wymagania dla aplikacji Mastermind. Głównym celem aplikacji jest rozrywka.

### 1.2 **Przeznaczenie dokumentu**

Dokument ma na celu przybliżyć charakterystykę opracowanego programu dla klienta oraz zespołu wykonującego program.

### 1.3 **Zakres produktu**

Celem gry jest odgadnięcie wylosowanej kombinacji symboli w jak najmniejszej ilości prób. Gracz próbuje zgadnąć jaki symbol ustawiony jest na każdej z X (określona liczba) pozycji, pamiętając o tym, że kolejność symboli nie może być inna niż we wzorze. Gra kończy się, gdy gracz zgadnie symbol bądź skończy mu się limit strzałów.

Są trzy rodzaje oceny danego kodu:

- a. nie ma takiego znaku w kodzie
- b. znak istnieje w kodzie, ale nie na tej pozycji
- c. znak jest na dobrej pozycji

Kod losowany jest na początku gry przez komputer.

### 1.4 **Odniesienia**

Dokument został stworzony zgodnie ze standardem IEEE830.

## **2. Opis ogólny**

### **2.1 Perspektywa produktu**

Program jest tworzony przez studentów Informatyki Stosowanej na Wydziale Fizyki i Informatyki Stosowanej AGH na potrzeby przedmiotu Inżynieria Oprogramowania. Jest to samodzielna aplikacja.

### **2.2 Funkcje produktu**

1. Gracz wybiera poziom trudności - liczbę pól do zgadnięcia oraz liczbę prób.
  - a. Domyślnie: 4 pola - 9 prób; 5 pól - 12 prób; 6 pól - 15 prób
  - b. Manualne wprowadzenie wartości liczby prób.
2. Gracz wybiera jakiego typu symbole chce zgadywać
  - a. Litery
  - b. Cyfry
  - c. Kolory
3. Rozgrywka: Gracz zgaduje kombinację - w okienku obok otrzymuje informację dot. trafności danej próby. Sterowanie odbywa się za pomocą klawiatury.
4. Gra kończy się, gdy gracz zgadnie kombinację (wygrywa), bądź gdy skończy się limit ilości prób (przegrywa)
5. Po zakończeniu gry gracz może:
  - a. Zakończyć grę
  - b. Rozpocząć nową grę na takich samych ustawieniach (ilość pól i rodzaj symbolu)
  - c. Rozpocząć nową grę przy nowych ustawieniach
6. Wygrane zapisywane są do pliku (ranking Top10)

### **2.3 Aktorzy i charakterystyki użytkowników**

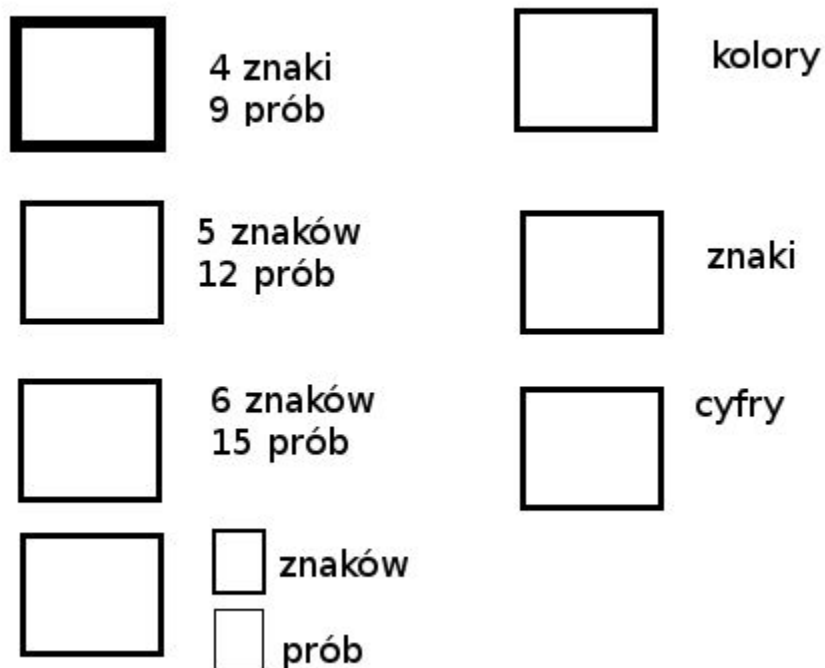
Gra przeznaczona jest dla każdego użytkownika.

### **2.4 Środowisko uruchomieniowe**

Program będzie uruchamiany na każdej platformie posiadającej Java Virtual Machine 1.8 (lub nowszej). Dodatkowo wymaga się, aby dostępne były urządzenia peryferyjne takie jak: klawiatura i monitor.

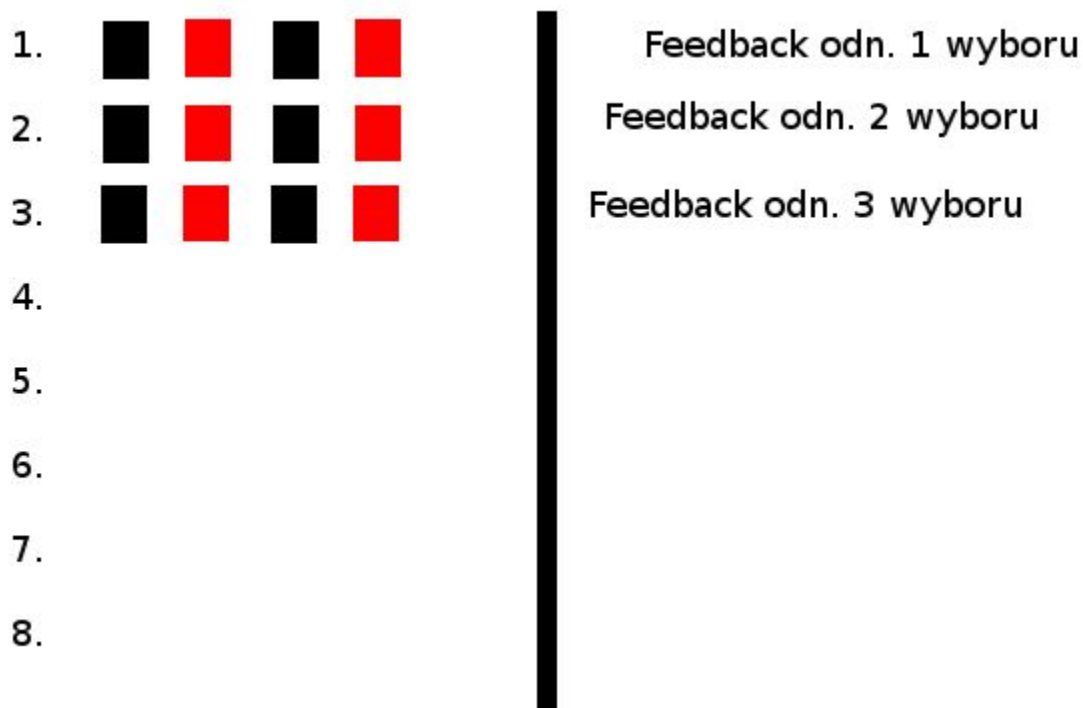
### 3. Wymagania zewnętrzne

#### 3.1 Opis interfejsu użytkownika



*Wstępny szkic ogólnego zarysu interfejsu graficznego okna konfiguracji*

W odpowiednich dużych prostokątach zostanie zaprezentowany w postaci X wybór opcji (tylko jeden X dla poziomu trudności oraz jeden dla rodzaju znaków), dodatkowo przy wyborze ostatniej opcji poziomu trudności aktywowane zostaną okienka do wpisania ilości znaków i ilości prób.



*Wstępny szkic ogólnego zarysu interfejsu graficznego okna gry*  
Po lewej gracz zgaduje, po prawej otrzymuje feedback.

Reszta okien będzie miała charakter czysto tekstowy.  
W każdym oknie na dole będzie wyświetlany pasek informujący o możliwości otworzenia okna pomocy.

### 3.2 Wymagania sprzętowe

Komputer posiadający klawiaturę i monitor.

### 3.3 Wymagane oprogramowania

Platforma posiadająca zainstalowaną Java Virtual Machine 1.8.

## 4. Funkcje systemu

### 4.1 Konfiguracja gry

#### 4.1.1 Opis i priorytet

Gracz konfiguruje poziom trudności gry oraz rodzaj symboli (znaki, liczby, kolorowe prostokąty) jaki ma być użyty podczas gry

Priorytet: wysoki

#### 4.1.2 Interakcja

Za pomocą klawiatury gracz ustala odpowiednie opcje, których zmiana zostaje odzwierciedlona na ekranie. Na koniec zmiany zostają potwierdzone odpowiednim klawiszem.

#### 4.1.3 Wymagania funkcjonalne

M-CONF: Wybór symboli będzie jednoznaczny (tzn. w postaci listy, której odpowiednie wpisy będą reprezentowały kolejne wybory). W przypadku wyboru poziomu trudności wybór będzie również jednoznaczny przy wyborze poziomu predefiniowanego. Jeśli gracz wybierze samodzielne wpisanie wartości, program poinformuje go o błędnym wprowadzeniu danych jeśli taka sytuacja nastąpi

### 4.2 **Rozgrywka**

#### 4.2.1 Opis i priorytet

Na początku losowany jest kod, niewidoczny dla gracza. Następnie gracz zgaduje symbol oraz sprawdza, zaś program sprawdza czy kod jest prawidłowy i daje graczowi odpowiednią informację zwrotną. Gra kończy się po osiągnięciu limitu prób lub po odgadnięciu odpowiedniego kodu.

Priorytet: wysoki

#### 4.2.2 Interakcja

Wybór kodu odbywa się za pomocą klawiatury i jest jednoznaczny, zmiany natychmiast odzwierciedlone są na ekranie. Potwierdzenie odbywa się odpowiednim klawiszem, gra przechodzi wtedy do następnej próby i wyświetla informacje dot. trafności wyboru w poprzedniej.

#### 4.2.3 Wymagania funkcjonalne

M-GAME: Wybór symboli jest jednoznaczny, tzn. nie można wybrać spoza danych możliwości.



## 4.3 **Koniec rozgrywki**

### 4.3.1 Opis i priorytet

Gdy zakończona zostaje rozgrywka gracz staje przed wyborem: czy chce rozpocząć grę ponownie, czy zakończyć sesję. W przypadku zakończenia sesji oraz wygranej wynik (ilość prób po których odgadnięto kombinację) zostaje zapisany do listy top10. W przypadku rozpoczęcia gry ponownie gracz wybiera, czy chce zmienić coś w konfiguracji czy pozostawić aktualną.

Priorytet: średni

### 4.3.2 Interakcja

Wybór odbywa się za pomocą klawiatury i efekty (kolejne okna dialogowe) zostają prezentowane natychmiast po wybraniu odpowiednich opcji.

### 4.3.3 Wymagania funkcjonalne

M-END: Odpowiada za zakończenie rozgrywki: inicjację ponownie gry, konfiguracji lub zapis wyniku do top10.

M-CONF: W przypadku ponownej konfiguracji

M-GAME: Do ponownego rozpoczęcia gry

## 4.4 **Pomoc**

### 4.4.1 Opis i priorytet

Po wciśnięciu odpowiedniego klawisza wyświetlone zostanie okienko pomocy adekwatne do aktualnego momentu gry (konfiguracja, gra).

Priorytet: niski

### 4.4.2 Interakcja

Wciśnięcie klawisza wyświetli okno pomocy, ponowne wciśnięcie zamknie je.

### 4.4.3 Wymagania funkcjonalne

M-HELP: Wyświetli ekran pomocy

5. **Wymagania pozafunkcjonalne**

Nie dotyczy produktu.

6. **Wymagania нефunkcjonalne**

Nie dotyczy produktu.

7. **Plan ewaluacji**

7.1 **Testy jednostkowe**

Planowane jest przeprowadzenie testów jednostkowych.

7.2 **Testy integracyjne**

Planowane jest przeprowadzenie testów integracyjnych.

7.3 **Testy akceptacyjne**

Planowane jest przeprowadzenie testów akceptacyjnych.

## **Dodatek A: Słownik**

Platforma - sprzęt komputerowy na którym uruchamiamy aplikację.

JVM (ang. *Java Virtual Machine*,) – maszyna wirtualna oraz środowisko zdolne do wykonywania kodu bajtowego Javy. Jest to zestaw aplikacji napisanych na tradycyjne urządzenia i systemy operacyjne. Dostarcza środowiska, w którym może się wykonywać program skompilowany do kodu bajtowego Javy.

Mastermind - gra planszowa wynaleziona w 1970 roku, przeznaczona dla dwóch osób, polegająca na odgadnięciu przez jednego z nich ukrytego kodu. Kod składa się z cyfr (bądź kolorów), cyfry mogą się powtarzać, a długość kodu jest ustalana na początku.

Na odgadnięcie kodu pierwszy gracz ma kilka tur, po każdej turze każda cyfra przez niego wpisana zostaje poddana ocenie. Są trzy rodzaje oceny danego elementu kodu:

- a) nie ma takiej cyfry w kodzie
- b) cyfra taka istnieje, ale ma złą pozycję
- c) cyfra ma dobrą pozycję

Drugim graczem może być komputer lub osoba ludzka. Gra kończy się wtedy gdy upłynęła ustalona ilość tur (wygrywa wtedy gracz drugi) lub gdy sekwencja została odgadnięta (wygrywa wtedy gracz pierwszy).