

Specyfikacja wymagań

Mateusz Biegański, Anna Kramarska, Michał Sarzyński, Magda Suchodolska

3 kwietnia 2018

1 Wymagania funkcjonalne

2 Grupy użytkowników

1. Gracz:

- Bierze udział w każdej grze
- Może sprawdzić 10 dotychczas najlepszych wyników
- Po każdej grze może wprowadzić uzyskany wynik i swój pseudonim do rankingu

3 Funkcjonalność

1. Rozpoczęcie nowej gry i prowadzenie jej do momentu przegranej lub osiągnięcia maksymalnego wyniku
2. Wyświetlanie planszy z polem, na które mają spaść klocki
3. Sekwencyjne wysuwanie się klocków z prawej i lewej strony ekranu
4. Umieszczanie nowego klocka na szczycie wieży poprzez wciśnięcie spacji
5. Odcinanie części klocka, która nie zmieściła się w swoim polu docelowym
6. Przesuwanie widoku wieży w górę wraz z dokładaniem klocków
7. Zrestartowanie gry po zakończeniu poprzedniej
8. Przejście do rankingu najlepszych wyników
9. Wprowadzenie wyniku do rankingu po zakończeniu rozgrywki
10. Włączenie/wyłączenie melodii w tle

4 Harmonogram

Planujemy następujący harmonogram:

- **Sprint 0** - 12.03 - Wizja, harmonogram, wybór technologii
- **Sprint I** - 04.04 - Specyfikacja wymagań (wersja pierwsza), kod wykonywalny: rozpoczęcie nowej gry, plansza z jednym klockiem
- **Sprint II** - 25.04 - Specyfikacja wymagań (dopracowanie), architektura systemu (wersja pierwsza), kod wykonywalny: budowanie wieży (ładowanie klocków, odcinanie), kończenie gry
- **Sprint III** - 09.05 - Architektura systemu (dopracowanie), kod wykonywalny: przesuwanie widoku wieży
- **Sprint IV** - 23.05 - Plakat (wersja pierwsza), kod wykonywalny: grafika, melodia
- **Sprint V** - 30.05 - Plakat (dopracowanie), prezentacja, kod wykonywalny: ranking 10 najlepszych wyników, możliwość zapisu własnego wyniku

5 Technologie

- **System:** Linux
- **Język:** Java
- **Framework:** libGDX

6 Pozostałe narzędzia

- **Repozytorium:** Github
- **Testy jednostkowe:** JUnit