Wojskowa Akademia Techniczna

im. Jarosława Dąbrowskiego w Warszawie



PROGRAMOWANIE ZDARZENIOWE

Dokumentacja projektu

Gra *Pacman*

**Autor: Prowadzący**: Sebastian Bajor por. mgr inż. Dawid Bugajewski

1. **Opis gry**

Gra polega na kierowaniu żółtej kulki czyli Pacmana za pomocą strzałek poprzez labirynt pełen kulek do zebrania. Gracz „zjadając” kulki zwiększa liczbę zdobytych punktów. Zjedzenie wszystkich kulek powoduje przejście do kolejnej mapy rozgrywki i dopisanie graczowi dodatkowych 25 punktów. W grze należy unikać czterech duszków – zielonego, czerwonego, różowego i żółtego, które poruszają się w „pozornie” losowy sposób po całej mapce. Kontakt z duszkiem powoduje utratę życia i koniec rozgrywki.

1. **Wymagania**

Wymagania co do realizacji projektu zostały sprecyzowane przez Prowadzącego:

* Aplikacja musi zawierać elementy programowania zdarzeniowego
* Aplikacja musi być zgodna i spójna z wybraną konwencją popularną dla danego języka programowania
* Aplikacje muszą przechowywać stan w bazach danych
* Dostęp do baz danych powinien odbywać się z wykorzystaniem ORM lub korzystać z interfejsów baz obiektowych

1. **Wykorzystane technologie:**

* Java 11
* Baza danych Postgres
* Maven
* Hibernate

1. **Testy aplikacji**

|  |  |
| --- | --- |
| Przypadek testowy 1 | |
| Nazwa | Rozpoczęcie rozgrywki |
| Warunki wstępne | Aplikacja musi być uruchomiona |
| Kroki wykonania | W panelu głównym aplikacji gracz wciska przycisk „New Game” |
| Oczekiwany rezultat | 1. Następuje uruchomienie gry 2. Pacman zostaje ustawiony na środku mapki 3. Duszki zostają ustawione w rogach mapki i zaczynają się poruszać |

|  |  |
| --- | --- |
| Przypadek testowy 2 | |
| Nazwa | Zderzenie z duszkiem |
| Warunki wstępne | 1. Aplikacja musi być uruchomiona 2. Gracz musi być w trakcie rozgrywki |
| Kroki wykonania | W trakcie rozgrywki musi zajść kolizja pomiędzy Pacmanem, a którymś z duszków |
| Oczekiwany rezultat | 1. Rozgrywka zostaje zakończona 2. Pojawia się komunikat z ilością zdobytych punktów 3. 3 Pojawia się pole tekstowe do wpisania nicku w celu zapisu wyniku do bazy danych |

|  |  |
| --- | --- |
| Przypadek testowy 3 | |
| Nazwa | Przejście do kolejnej mapy rozgrywki |
| Warunki wstępne | 1. Aplikacja musi być uruchomiona |
| Kroki wykonania | 1. Pacman musi zjeść wszystkie kulki znajdujące się na mapie 2. Pacman nie może dać się złapać żadnemu z duszków |
| Oczekiwany rezultat | 1. Następuje zmiana mapy na kolejną 2. Pacman zostaje ustawiony na środku nowej mapy, a duszki w jej rogach 3. Do puli zebranych punktów zostaje dopisane dodatkowe 25pkt. |

**5.** **Diagram przypadków użycia**

* Przypadek użycia: Gracz



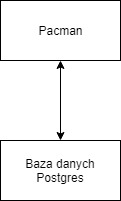
**6. Komponenty graficzne**

Klasy rozszerzające JPanel: kilka klas (między innymi GamePanel, AboutPanel, MenuPanel itd.) rozszerza klasę JPanel. Nadpisuję tu metodę paintComponent(Graphics g), co pozwala między innymi dodać tło czy narysować mapę w panelu gry. Wszystkie klasy rozszerzające JPanel używają GridBagLayout do ułożenia komponentów.

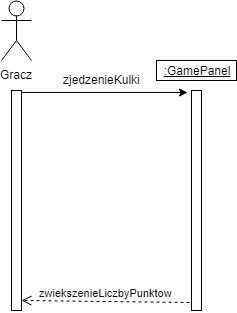
**7. Baza danych**

Baza danych, z której korzystam jest bardzo prosta – zawiera 1 encję (Highscores), w której każdemu graczowi odpowiada jego nick i liczba zdobytych punktów podczas rozgrywki. Przy starcie aplikacji dane są pobierane z bazy i zapisywane w tabeli w panelu HighScoresPanel. Po zakończeniu rozgrywki gracz zapisuje liczbę zdobytych punktów do bazy. Odbywa się to poprzez wpisanie w okienku nicku, które również informuje o ilości zdobytych punktów przez gracza.

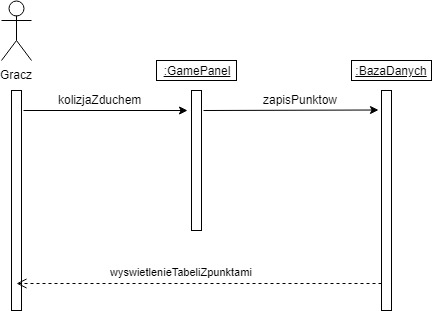
**8. Architektura systemu**

****

**9. Diagramy sekwencji**

****

**a)**

****

**b)**