UJ DUNGEONS

Dokumentacja techniczna

Opis

Dokument skupia się na technicznych aspektach aplikacji UJ Dungeons.

Spis treści

Wprowadzenie

Dokument przedstawia techniczną stronę projektu UJ Dungeons, do której należą wzorce, typy kontrolerów, modeli czy też widoków.

Ma również na celu przybliżenie użytkownikowi sposobu ich działania oraz powodzie, dla których zostały wykorzystane.

Wzorzec architektoniczny Model-View-Controller

Wzorzec architektoniczny służący do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika.

Aplikacja UJ Dungeons jest grą posiadającą interfejs graficzny. Z tego też powodu nasz zespół postanowił zastosować wzorzec MVC. Nasz Model-View-Controller składa się z następujących elementów:

- **Model** - do obsługi algorytmów i zarządzania mechaniką gry, tutaj korzystamy z naszych komnat (chambers),

- **Controller** - kontroler dostarcza informacje, które mają być wyświetlone, a także obsługuje zdarzenia przychodzące z widoków. Znajdują się one w paczce Controllers. Wywołują metody w chambers i jej rezultaty przekazują do widoków,

- **View** - przekazuje zdarzenia do kontrolera, wyświetla wyniki.

Typy kontrolerów

Wcześniej wymienione w rozdziale (Wzorzec architektoniczny Model-View-Controller) kontrolery w aplikacji UJ Dungeons dzielą się na następujące typy:

- **Kontrolery komnat** - wykonują metody i dla poszczególnych komnat, dziedziczą po ChamberController,
- **Kontroler wyboru klasy** - obsługuje scenę z wyborem klasy postaci i jej nazwą,
- **Kontroler menu** - obsługa wyświetlania statystyk i informacji o naszym bohaterze,
- **Kontroler głównej sceny** - zarządza scenami pojawiającymi się na ekranie, doczepia do nich kontrolery i ładuje na ekran widok,

Kontrolery znajdują się w paczce Controllers.

Typy modeli

W aplikacji UJ Dungeons modele występują w postaci:

- Komnaty - dziedziczą po Chamber (Chamber.java). Wykonuje się w niej cała logika aplikacji dotycząca lokacji (cała paczka Chambers),

- Typy postaci - dziedziczą po chambers. Logika aplikacji odpowiedzialna za interakcję między postaciami w grze (paczka Models.Character),

- Itemy - ekwipunek, opis statystyk wraz z działaniem (paczka Others).

Typy widoków

Widokami w grze UJ Dungeons są sceny, grafiki, które użytkownik widzi podczas korzystania z aplikacji. Są to:

- **Komnaty** - wyjście w prawo lub w lewo + dodatkowa aktywność związana ze specyfiką lokacji,

- **Plansza kończąca grę** - zwycięstwo lub porażka (_winScreen.fxml lub _diedScreen.fxml)

- **Widok do wyboru postaci i jej imienia** – jedna z początkowych scen umożliwiająca użytkownikowi na wprowadzenie imienia dla bohatera w grze (_classSelectScreen.fxml),

- **Głowna scena** (root_layout.fxml) - wyświetla statystyki i udostępnia przestrzeń do wyświetlania lokacji,

- **Scena decydująca o chęci rozpoczęcia gry lub opuszczenia aplikacji** (_mainMenuScreen.fxml)

Wszystkie widoki znajdują się w folderze resources.

Wzorzec fabryka

Wzorzec ten służy do tworzenia obiektów bez konieczności ujawniania szczegółów związanych z procesem ich tworzenia. Pozwala na zastosowanie jednego interfejsu do tworzenia różnych typów obiektów lub rodzajów ich implementacji.

W przypadku naszej aplikacji, UJ Dungeons jest grą opartą na mechanice RPG (Role-playing Game), która oparta jest na zdobywaniu nowych przedmiotów oraz ciągłe ulepszanie swojej postaci. Sprawia to, że w podczas rozgrywki pojawia się duża ilość ekwipunku. W tej sytuacji idealnie sprawdza się omawiany wzorzec.

Schemat działania fabryki w aplikacji UJ Dungeons:

1. Do funkcji dostarczamy parametry: poziom, typ itemu, profesja gracza,
2. Następnie fabryka na ich podstawie zwraca odpowiedni item,
3. Itemy jakie zwraca określone są w klasie Item i każdy z nich dziedziczy po klasie Item.

Wzorzec strategia

Jest behawioralny wzorzec projektowy pozwalający zdefiniować rodzinę algorytmów, umieścić je w osobnych klasach i uczynić obiekty tych klas wymienialnymi.

Wzorzec ten wykorzystujemy do budowania mapy i jej użytkowania. Mapa tworzona jest poprzez wybranie pierwszej komnaty, a następnie są wskazywane komnaty do których użytkownik powinien się udać, gdy wybierze opcje 1 lub opcje 2.

Część komnat ma tylko jedno wyjście i wtedy należy tylko tę opcje obsłużyć.

Ponadto komnaty mają swoje indywidualne funkcje, które są rozpatrywane i obsługiwane przez kontroler danej komnaty.