

Merchant simulation - projekt symulacji na laboratorium Programowanie Obiektowe.

Gusta Oskar, Zajęcia: K00-65g (wt 19:05)

23.06.2022r.

Spis treści

1	Koncept	3
2	Opis działania	3
3	Szczegóły techniczne	4

1 Koncept

Program wykonuje symulację której celem jest pokazanie przebiegu handlu na zadanych algorytmach i z określoną losowością.

2 Opis działania

Po uruchomieniu projektu zadanymi argumentami decydującymi o losowości zostaje użytkownikowi przedstawiona możliwość kontynuowania symulacji która wcześniej została zapisana. Jeśli użytkownik posiada plik z zapisanym stanem symulacji, może on go wczytać w tym kroku. W przeciwnym wypadku wyświetlane jest pytanie o typ strategii i algorytm odpowiedzialny za przemieszczanie się. Następnie po wczytaniu lub wybraniu stworzeniu nowej symulacji użytkownikowi jest prezentowana właściwa część programu. Na górze ekranu wyświetlane są wioski z podstawowymi danymi, są one losowo generowane co trzecią turę symulacji i na początku. Pod wioskami znajdują się dane kupca: pozycja, ilość i konsumpcja jedzenia na jeden 'dzień' symulacji, ilość pieniędzy, informacja czy kupiec został zaatakowany na ostatniej ścieżce do wioski oraz ilość posiadanych dóbr. Użytkownik ma dostęp do konsoli sterowania z poziomu której może poruszać symulację do przodu o jeden lub kilka iteracji a także zapisać, wyjść oraz otworzyć pomoc z poziomu symulacji.

Co każdy epoch kupiec przemieszcza się do wioski za pomocą wybranego algorytmu i płaci cenę proporcjonalną do przebytej drogi. W trakcie podróży kupiec może zostać zaatakowany z prawdopodobieństwem które zwiększa się z odległością konieczną do przebycia. Jeśli dojdzie do ataku kupiec traci część pieniędzy oraz część z każdego typu posiadanego dobra, w tym jedzenia. Następnie sprawdzana jest ilość jedzenia, jeśli jest nie wystarczająca symulacja kończy się i użytkownik jest pytany o zapis, po czym następuje wyjście z programu. Jeśli zapas jedzenia jest wystarczający następuje przejście do sekcji sprzedaży i kupna. Algorytmy kupują i sprzedają w proporcji z góry określonej, uporządkowanej przez ceny dóbr w wiosce. Cykl programu powtarza się do momentu końca symulacji lub manualnego wyjścia z opcjonalnym zapisem.

Wybrane parametry mają wpływ na ogólną losowość symulacji oraz dotkli-

wość ataku, jeśli on nastąpi. Wybrana strategia decyduje o losowości, konsumpcji jedzenia oraz o obniżce cen kupna. Wybrany algorytm przemieszczania się odpowiada za to czy nasz kupiec będzie przemieszczał się do wioski która jest najbliższej czy do tej której *Price Index* obliczany jako średnia algorytmiczna cen jest najniższy.

3 Szczegóły techniczne

Projekt posiada poprawne zapisywanie i odczytywanie stanu symulacji do formatu JSON z uwzględnieniem polimorfii oraz rekursywnych odwołań (*io.json.Converter*). Projekt korzysta z polimorfii na potrzeby wyboru strategii (Wykorzystanie *simulation.environment.EPOCHS*, Implementacja *simulation.strategy.StrategyType*). Projekt korzysta z klas abstrakcyjnych oraz dziedziczenia (Na przykład *simulation.goods.StockBase*). Projekt korzysta z Javadoc do tworzenia dokumentacji poszczególnych klas, metod, stanów i pakietów. Projekt posiada schemat klas zamieszczony w tym samym folderze w systemie git. Projekt korzysta z kontroli źródła.

Wykonano przy użyciu L^AT_EX.