

21.05.2023

Planet Trade Game Report

Advanced Programming for Effective
OOP

NURİ CAN ÖZTÜRK
18070006034

Planet Trade Game

Planet Trade Game is a turn-based multiplayer game. Each players can buy item, sell item, buy fuel, change spaceship, buy spaceship and travel to another planets.

The game universe consists of the following elements:

- A black hole, which explodes and creates a galaxy.
- A galaxy, which contains a set of planets located at specific distances from each other in terms of light-years.
- Each planet features a market where commodities are bought and sold.
- A commodity has a name, unit volume, and decay ratio.
- Each supply of a commodity at a market has a current supply amount, unit buy price, and unit sell price.
- A player has a name, current money, spaceship, and current planet.
- A planet has a name, a market, unit fuel price, and spaceship parking price per turn.
- A spaceship has a name, buy price, a list of cargo, volume capacity, speed in terms of light-years per turn, current fuel, fuel capacity, and fuel usage per light-year.
- A cargo contains a commodity and its quantity.

Each turn players can following actions:

- Buy new market items from the market of the current planet as much as the capacity of the spaceship and the supply of the market allows. The buy operation causes the current money drop with the amount calculated by unit buy price of the market item in the market and the amount
- Sell any cargo in the spaceship. The sell operation causes increase in the current money with amount calculated by the cargo amount and unit sell price of the commodity in the market.
- Buy fuel as much as the fuel capacity of the spaceship allows. It causes the current Money drop with the amount calculated by the unit fuel price at the current planet
- Plan journey to another planet. If this is done in one turn the player will be at the target planet in the next turn if the spaceship has sufficient amount of fuel which is calculated by the fuel usage of the spaceship and the distance between the planets. Otherwise the player stays at the same planet causing a drop in the current money by the parking price of the current planet.

Each turn game should update itself:

- At each turn player actions are validated by the game and if they are valid necessary updates are performed accordingly which includes:
 - the current money of each player
 - current planet of each player
- The current cargo of each player are decayed.
- At each market of each planet the following changes randomly
 - unit buy price and unit sell price of each market item
 - current supply of each market item (with relatively small amount

PRINCIPLES

Item -1-	Consider static factory methods instead of constructors	project/planettrade/factory - [all classes]
Item -2-	Consider a builder when faced with many constructor parameters	project/planettrade/entity/Commodity.java project/planettrade/entity/Planet.java project/planettrade/entity/SpaceShip.java
Item -15-	Minimize the accessibility of classes and members	project/planettrade/entity/Galaxy.java project/planettrade/entity/Market.java
Item -17-	Minimize mutability	Factory classes and if possible everywhere
Item -28-	Prefer Lists to Array	project/planettrade/entity/Market.java project/planettrade/entity/Galaxy.java
Item -29-	Prefer generic types	project/planettrade/util/exception/ExceptionUtil.java project/planettrade/util/exception/ISupplier.java
Item -30-	Favor generic methods	project/planettrade/util/exception/ExceptionUtil.java
Item -42-	Prefer Lambdas to anonymous classes	project/planettrade/entity/PlanetTradeGameContext.java methods: - changeSupplyAmountsOnMarket, - changePriceOnMarket
Item -43-	Prefer method references to lambdas	project/planettrade/action/actions/SellItem.java - method: sellItems project/planettrade/entity/PlanetTradeGameContext.java - method: changeSupplyAmountOnMarket project/planettrade/factory/MarketFactory.java - method: createMarket
Item -44-	Favor the use of standard functional interfaces	project/planettrade/util/exception/ExceptionUtil.java - (Consumer in parameters)
Item -45-	Use streams judiciously	project/planettrade/action/actions/SellItem.java - method: sellItem - used if necessary everywhere
Item -51-	Design method signatures carefully	project/planettrade/util/exception/ExceptionUtil.java (handleException methods are overloading)
Item -34-	Use enums instead of int constants	project/planettrade/action/ActionType.java (enum class) (used on ActionGenerator class)
Item -58-	Prefer for-each loops to traditional loops	project/planettrade/util/LinkedList.java - method: forEach, line = 137 - used forEach method lists and streams
Item -73-	Throw exceptions appropriate to the abstraction	project/planettrade/util/exception/ExceptionUtil.java project/planettrade/factory/MarketFactory.java - method: createMarket
Item -18-	Favor composition over inheritance	project/planettrade/entity/PlayerImpl.java
Item -46-	Prefer side-effect-free functions in streams	project/planettrade/entity/PlanetTradeGameContext.java - changeSupplyPricesOnMarket (map, forEach)

REQUIREMENTS

R1	project/planettrade/entity/BlackHole.java
R2	project/planettrade/entity/Galaxy.java
R3	project/planettrade/entity/Market.java
R4	project/planettrade/entity/Commodity.java
R5	project/planettrade/entity/Commodity.java
R6	project/planettrade/entity/PlayerImpl.java project/planettrade/entity/IntelligentPlayer.java
R7	project/planettrade/entity/Planet.java
R8	project/planettrade/entity/SpaceShip.java
R9	project/planettrade/entity/Cargo.java
R10	project/planettrade/factory/GalaxyFactory.java
R11	project/planettrade/factory/CommodityFactory.java
R12	project/planettrade/factory/SpaceshipFactory.java
R13	project/planettrade/entity/PlayerImpl.java project/planettrade/entity/IntelligentPlayer.java - method: prepareForGame defined on action: - project/planettrade/action/actions/SelectPlanet.java
R14	project/planettrade/entity/PlayerImpl.java project/planettrade/entity/IntelligentPlayer.java - method: prepareForGame defined on action: project/planettrade/action/actions/BuySpaceship.java
R15	project/planettrade/factory/MarketFactory.java
R16	project/planettrade/action/actions/BuyItem.java
R17	project/planettrade/action/actions/SellItem.java
R18	project/planettrade/action/actions/BuyFuel.java
R19	project/planettrade/action/actions/PlanTravelling.java
R20	-----
R21	project/planettrade/entity/PlanetTradeGameContext.java Trigger Method: - updateGame
R22	project/planettrade/entity/PlanetTradeGameContext.java Trigger Method: - updateGame
R23	-----
R24	project/planettrade/entity/PlayerImpl.java project/planettrade/entity/IntelligentPlayer.java - method: prepareForGame