Packets

simulation

environment

goods

player

villages

strategy

io

arguments

json

console

app

Main
+static main(String[] Args)

Packet simulation

simulation.strategy

ConservativeStrategy
-foodConsumption: int = 10
-priceMultiplier: float = 1.0
-travelCost: float = 1.15

BalancedStrategy
-foodConsumption:int = 15
-priceMultiplier: float = 0.9
-travelCost: float = 1.05
+fluctuateFoodConsumption()

AggresiveStrategy
-foodConsumption:int = 20
-priceMultiplier: float = 0.7
-travelCost: float = 1
+fluctuateFoodConsumption()
+fluctuatePriceMultiplier()

<<Interface>> StrategyType
+getFoodConsumption(): int
+getPriceMultiplier(): float
+getTravelCost(): float

simulation.environment

<<Singleton>> Epochs
-count: int
+advance()
+advanceBy(int)
+getCount()

<<Singleton>> VillageMap
-locations: Set<Village>
+VillageMap(Set<Village>)
+getBestPrices(): Village
+getClosestToPlayer(): Village

simulation.player

Food
-consumption: int
-stock: int = 100
+Food(int)
+add(uint)
+consume()

<<Singleton>> Storage
-stock: Set<Product> = {}
-money: float = 100.0
-food: Food
+addProduct(Product)
+removeProduct(Product)
+getProduct(ProductType): float
+getMoney(): float
+subtractMoney(float)
+getFood(): Food

<<Singleton>> PlayerState
-isAttacked: bool = false
-currentPosition: Pair<int, int> = {0,0}
-lastTransaction: Transaction
+getIsAttacked(): bool
+setIsAttacked(bool)
+getCurrentPosition(): Pair<int, int>
+setCurrentPosition(Pair<int, int>)
+setLastTransaction(Transaction)
+getLastTransaction(): Transaction

simulation.goods

Product
-weight: float
-type: ProductType
+Product(float, ProductType)
+getWeight(): float
+getType(): ProductType
+addWeight(float)

Transaction
+Transaction(float, Product, TransactionType)

<<enumeration>> ProductType
+SPICE
+GEM
+SOAP

<<enumeration>> TransactionType
+BUY
+SELL

simulation.villages

Village
-position: Pair<int, int>
-goods: Map<Product, float>
+Village(Pair<int, int>)
+getProductByType(ProductType): Pair<Product, float>
+getPosition(): Pair<int, int>

Road
-startPos: Pair<int, int>
-endPos: Pair<int, int>
+Road(Pair<int, int>, Pair<int, int>)
+travel()
+calculateDistance(): float
+calculateRisk(float): float

Thugs
-stealMultiplier: float
+Thugs()
+steal()

io.console.input

io.arguments.Difficulty

io.console.Input

io.arguments.Difficulty

io.arguments.Difficulty

Packet io

io.console

<<enumeration>>
CommandType

+ADVANCE
+ADVANCE_BY
+ADVANCE_AUTO
+ADVANCE_STOP
+SAVE_QUIT
+SET_STRATEGY
+RESUME

Command

-type: CommandType
-param: int

+Command(CommandType, int)

Output

+displayVillages()
+displayStatus()

Input

-lastCommand: Command

+waitCommand()
+executeLastCommand()

simulation.player

simulation.environment

io.json

Converter

+saveToFile(string)
+readFromFile(string): Game

Game

-epochs: Epochs
-state: PlayerState
-map: VillageMap
-storage: Storage
-strategy: StrategyType
-isDead: bool

+Game(Epochs, PlayerState, VillageMap, Storage, StrategyType, bool)
+load()

io.arguments

Parser

+parse(String[] args): Difficulty

Difficulty

-rng: float
-stealMultiplier: float

+Difficulty(float, float)
+getRng(): float
+getStealMultiplier(): float

JSON Schema 'Game'

{
  "Epochs": {...},
  "PlayerState": {...},
  "VillageMap": {...},
  "Storage": {...},
  "StrategyType": {...},
  "isDead": bool
}