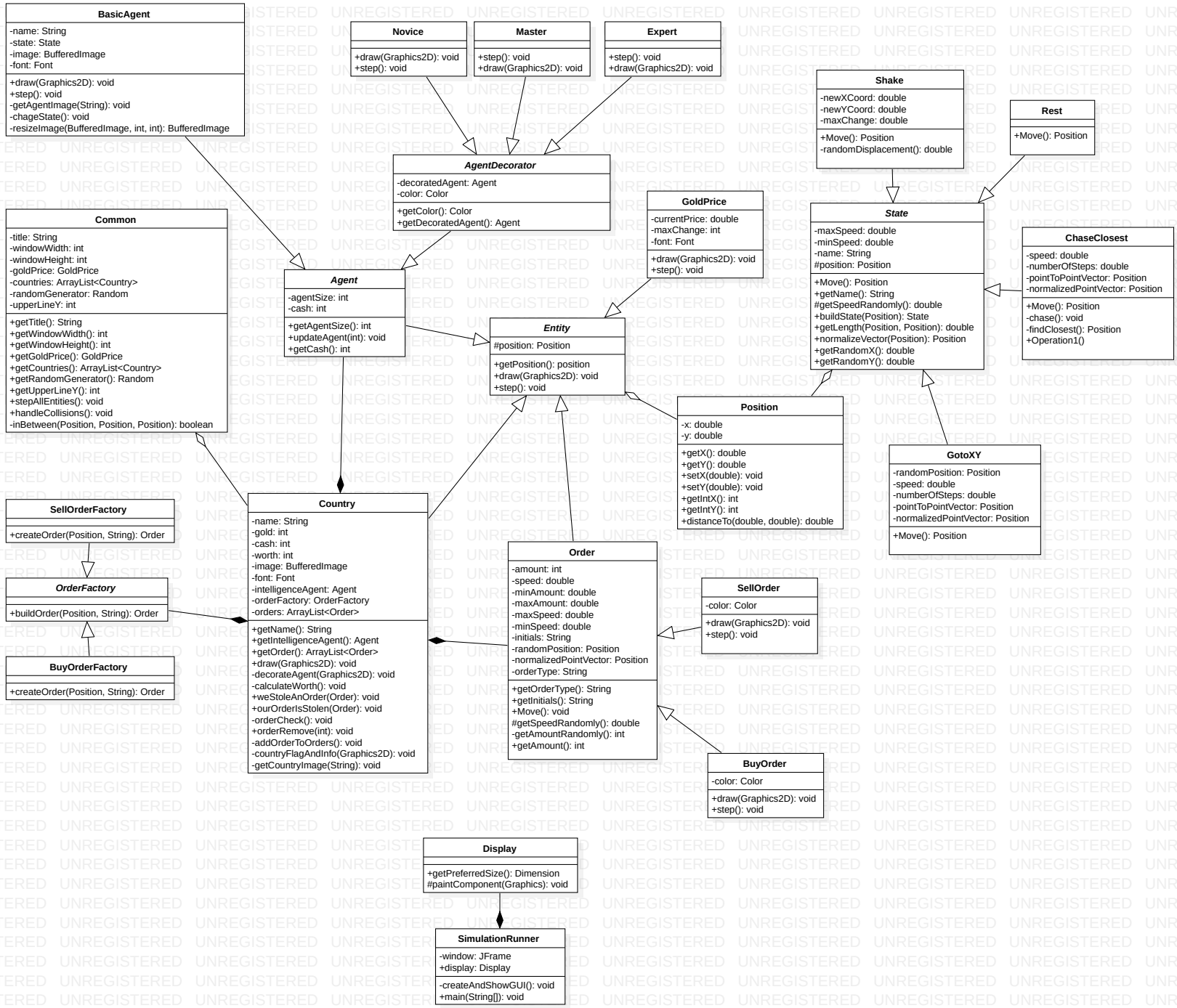
Model.Main


BasicAgent
-name: String
-state: State
-image: BufferedImage
-font: Font
+draw(Graphics2D): void
+step(): void
-getAgentImage(String): void
-chageState(): void
-resizeImage(BufferedImage, int, int): BufferedImage
Novice
+draw(Graphics2D): void
+step(): void
Master
+step(): void
+draw(Graphics2D): void
Expert
+step(): void
+draw(Graphics2D): void
Shake
-newXCoord: double
-newYCoord: double
-maxChange: double
+Move(): Position
-randomDisplacement(): double
Rest
+Move(): Position
AgentDecorator
-decoratedAgent: Agent
-color: Color
+getColor(): Color
+getDecoratedAgent(): Agent
GoldPrice
-currentPrice: double
-maxChange: int
-font: Font
+draw(Graphics2D): void
+step(): void
State
-maxSpeed: double
-minSpeed: double
-name: String
#position: Position
+Move(): Position
+getName(): String
#getSpeedRandomly(): double
+buildState(Position): State
+getLength(Position, Position): double
+normalizeVector(Position): Position
+getRandomX(): double
+getRandomY(): double
ChaseClosest
-speed: double
-numberOfSteps: double
-pointToPointVector: Position
-normalizedPointVector: Position
+Move(): Position
-chase(): void
-findClosest(): Position
+Operation1()
Common
-title: String
-windowWidth: int
-windowHeight: int
-goldPrice: GoldPrice
-countries: ArrayList<Country>
-randomGenerator: Random
-upperLineY: int
+getTitle(): String
+getWindowWidth(): int
+getWindowHeight(): int
+getGoldPrice(): GoldPrice
+getCountries(): ArrayList<Country>
+getRandomGenerator(): Random
+getUpperLineY(): int
+stepAllEntities(): void
+handleCollisions(): void
-inBetween(Position, Position, Position): boolean
Agent
-agentSize: int
-cash: int
+getAgentSize(): int
+updateAgent(int): void
+getCash(): int
Entity
#position: Position
+getPosition(): position
+draw(Graphics2D): void
+step(): void
Position
-x: double
-y: double
+getX(): double
+getY(): double
+setX(double): void
+setY(double): void
+getIntX(): int
+getIntY(): int
+distanceTo(double, double): double
GotoXY
-randomPosition: Position
-speed: double
-numberOfSteps: double
-pointToPointVector: Position
-normalizedPointVector: Position
+Move(): Position
SellOrderFactory
+createOrder(Position, String): Order
OrderFactory
+buildOrder(Position, String): Order
BuyOrderFactory
+createOrder(Position, String): Order
Country
-name: String
-gold: int
-cash: int
-worth: int
-image: BufferedImage
-font: Font
-intelligenceAgent: Agent
-orderFactory: OrderFactory
-orders: ArrayList<Order>
+getName(): String
+getIntelligenceAgent(): Agent
+getOrder(): ArrayList<Order>
+draw(Graphics2D): void
-decorateAgent(Graphics2D): void
-calculateWorth(): void
+weStoleAnOrder(Order): void
+ourOrderIsStolen(Order): void
-orderCheck(): void
+orderRemove(int): void
-addOrderToOrders(): void
-countryFlagAndInfo(Graphics2D): void
-getCountryImage(String): void
Order
-amount: int
-speed: double
-minAmount: double
-maxAmount: double
-maxSpeed: double
-minSpeed: double
-initials: String
-randomPosition: Position
-normalizedPointVector: Position
-orderType: String
+getOrderType(): String
+getInitials(): String
+Move(): void
#getSpeedRandomly(): double
-getAmountRandomly(): int
+getAmount(): int
SellOrder
-color: Color
+draw(Graphics2D): void
+step(): void
BuyOrder
-color: Color
+draw(Graphics2D): void
+step(): void
Display
+getPreferredSize(): Dimension
#paintComponent(Graphics): void
SimulationRunner
-window: JFrame
+display: Display
-createAndShowGUI(): void
+main(String[]): void