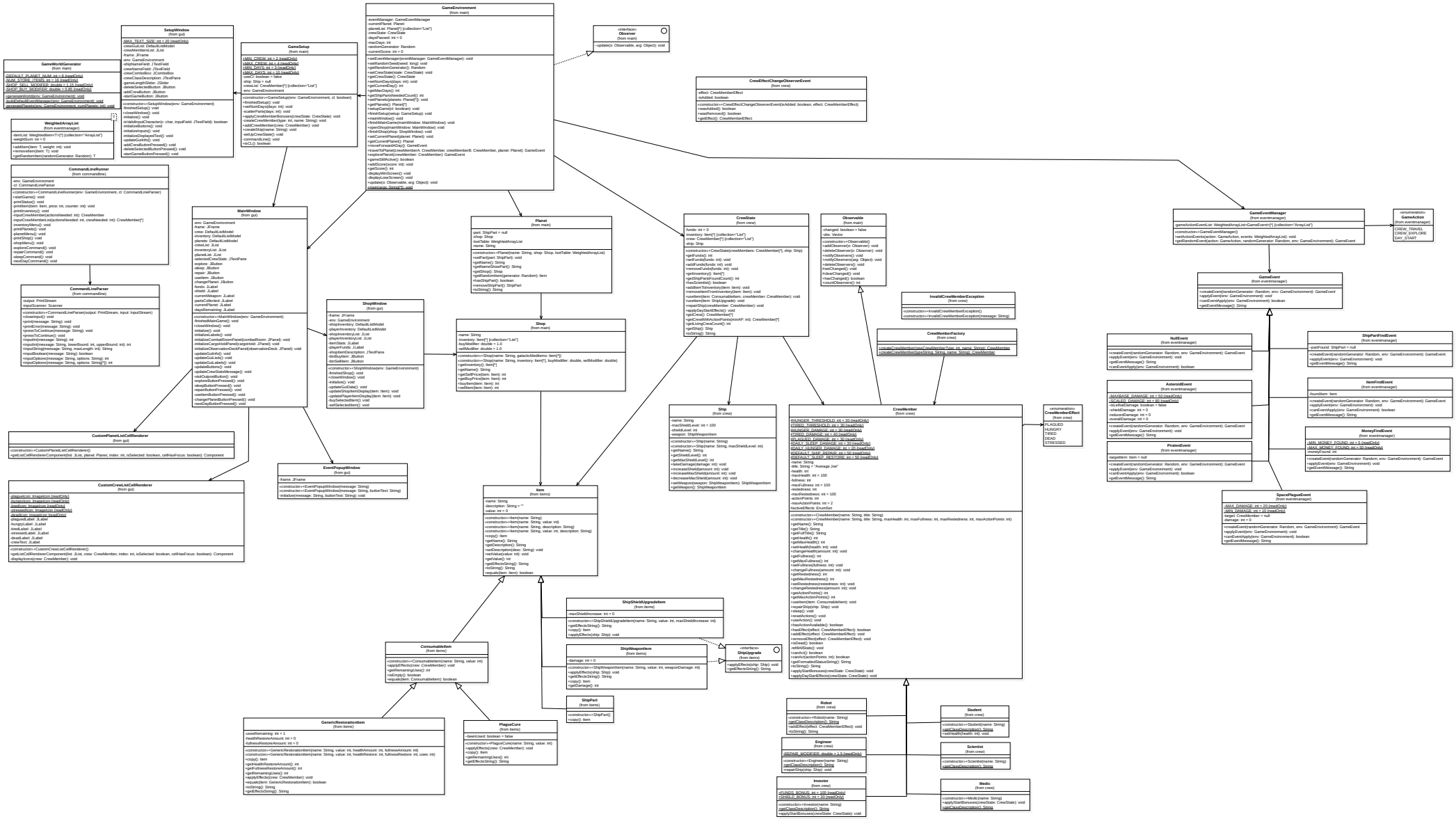
GameEnvironment
(from main)
eventManager: GameEventManager
currentPlanet: Planet
planetList: Planet[*] {collection="List"}
crewState: CrewState
daysPassed: int = 0
maxDays: int
randomGenerator: Random
currentScore: int = 0
+setEventManager(eventManager: GameEventManager): void
+setRandomSeed(seed: long): void
+getRandomGenerator(): Random
+setCrewState(state: CrewState): void
+getCrewState(): CrewState
+setNumDays(days: int): void
+getCurrentDay(): int
+getMaxDays(): int
+getShipPartsNeededCount(): int
+setPlanets(planets: Planet[*]): void
+getPlanets(): Planet[*]
+setupGame(): boolean: void
+finishSetup(setup: GameSetup): void
+mainWindow(): void
+setMainWindow(mainWindow: MainWindow): void
+openShop(mainWindow: MainWindow): void
+freshShop(shop: ShopWindow): void
+setCurrentPlanet(planet: Planet): void
+getCurrentPlanet(): Planet
+moveForwardADay(): GameEvent
+travelToPlanet(crewMemberA: CrewMember, crewMemberB: CrewMember, planet: Planet): GameEvent
+explorePlanet(crewMember: CrewMember): GameEvent
+gameStillActive(): boolean
+addScore(score: int): void
+getScore(): int
+displayWinScreen(): void
+displayLoseScreen(): void
+update(o: Observable, arg: Object): void
+toString(): String
SetupWindow
(from gui)
-MAX_TEXT_SIZE: int = 20 {readOnly}
-crewGuiList: DefaultListModel
-crewMembersList: JList
-frame: JFrame
-game: GameEnvironment
-shipNameField: JTextField
-crewNameField: JTextField
-crewComboBox: JComboBox
-crewClassDescription: JTextPane
-gameLenghSlider: JSlider
-deleteSelectedButton: JButton
-addCrewButton: JButton
-startGameButton: JButton
+constructor(+SetupWindow(env: GameEnvironment)
+finishedSetup(): void
+closeWindow(): void
+initialize(): void
-isValidInputCharacter(c: char, inputField: JTextField): boolean
-initializeButtons(): void
-initializeInputs(): void
-initializeDisplayedText(): void
-updateGuiInfo(): void
-addCrewButtonPressed(): void
-deleteSelectedButtonPressed(): void
-startGameButtonPressed(): void
GameSetup
(from main)
+MIN_CREW: int = 2 {readOnly}
+MAX_CREW: int = 4 {readOnly}
+MIN_DAYS: int = 3 {readOnly}
+MAX_DAYS: int = 10 {readOnly}
-useC: boolean = false
-ship: Ship = null
-crewList: CrewMember[*] {collection="List"}
-env: GameEnvironment
+constructor(+GameSetup(env: GameEnvironment, cl: boolean)
+finishedSetup(): void
+setNumDays(days: int): void
+scatterParts(days: int): void
+applyCrewMemberBonuses(crewState: CrewState): void
+createCrewMember(type: int, name: String): void
+addCrewMember(crew: CrewMember): void
+createShip(name: String): void
+setUpCrewState(): void
+commandLine(): void
+isC(): boolean
GameWorldGenerator
(from main)
-DEFAULT_PLANET_NUM: int = 8 {readOnly}
-NUM_STORE_ITEMS: int = 16 {readOnly}
-SHOP_SELL_MODIFIER: double = 1.15 {readOnly}
-SHOP_BUY_MODIFIER: double = 0.85 {readOnly}
+generateWorld(env: GameEnvironment): void
-buildDefaultEventManager(env: GameEnvironment): void
-generatePlanets(env: GameEnvironment, numPlanets: int): void
WeightedArrayList
(from eventmanager)
-itemList: WeightedItem<T>[*] {collection="ArrayList"}
-weightSum: int = 0
+add(item:(item: T, weight: int): void
+removeItem(item: T): void
+getRandomItem(randomGenerator: Random): T
CommandLineRunner
(from commandline)
-env: GameEnvironment
-cl: CommandLineParser
+constructor(+CommandLineRunner(env: GameEnvironment, cl: CommandLineParser)
+startGame(): void
-printStatus(): void
-printItem(item: Item, prefix: int, counter: int): void
-printCrew(): void
-inputCrewMember(actionsNeeded: int): CrewMember
-inputCrewMemberList(actionsNeeded: int, crewNeeded: int): CrewMember[*]
-inventoryMenu(): void
-printPlanets(): void
-planetMenu(): void
-printShop(): void
-shopMenu(): void
-exploreCommand(): void
-repairCommand(): void
-sleepCommand(): void
-restCommand(): void
CommandLineParser
(from commandline)
-output: PrintStream
-inputScanner: Scanner
+constructor(+CommandLineParser(output: PrintStream, input: InputStream)
+closeInput(): void
+print(message: String): void
+printError(message: String): void
+pressToContinue(message: String): void
+pressToContinue(): void
+input(message: String): int
+input(message: String, lowerBound: int, upperBound: int): int
+inputString(message: String, maxLength: int): String
+inputBoolean(message: String): boolean
+inputOptions(message: String, options: String): int
+inputOptions(message: String, options: String[*]): int
MainWindow
(from gui)
-env: GameEnvironment
-frame: JFrame
-crew: DefaultListModel
-inventory: DefaultListModel
-planets: DefaultListModel
-crewList: JList
-inventoryList: JList
-planetList: JList
-currentStatus: JTextPane
-explore: JButton
-sleep: JButton
-repair: JButton
-useItem: JButton
-changePlanet: JButton
-fuelIs: JLabel
-shieldI: JLabel
-currentWeapon: JLabel
-partsCollected: JLabel
-currentPlanet: JLabel
-daysRemaining: JLabel
+constructor(+MainWindow(env: GameEnvironment)
-finishedMainGame(): void
+closeWindow(): void
-initialize(): void
-initializeLabels(): void
-initializeCombatRoomPanel(combatRoom: JPanel): void
-initializeCargoHoldPanel(cargoHold: JPanel): void
-initializeObservationDeckPanel(observationDeck: JPanel): void
-updateGuiInfo(): void
-updateGuiLists(): void
-updateGuiLabels(): void
-updateButtons(): void
-updateCrewStatusMessage(): void
-updateCrewDisplay(): void
-explorePlanetButtonPressed(): void
-sleepButtonPressed(): void
-repairButtonPressed(): void
-useItemButtonPressed(): void
-changePlanetButtonPressed(): void
-nextDayButtonPressed(): void
ShopWindow
(from gui)
-frame: JFrame
-env: GameEnvironment
-shopInventory: DefaultListModel
-playerInventory: DefaultListModel
-shopInventoryList: JList
-playerInventoryList: JList
-itemState: JLabel
-playerFunds: JLabel
-shopItemDescription: JTextPane
-shopFunds: JLabel
-itemSellItem: JButton
+constructor(+ShopWindow(env: GameEnvironment)
-finishedShop(): void
+closeWindow(): void
-initialize(): void
-updateGuiDate(): void
-updateShopItemDisplay(item: Item): void
-updatePlayerItemDisplay(item: Item): void
-buyItemPressed(): void
-sellItemPressed(): void
-nextDayButtonPressed(): void
Planet
(from main)
-part: ShipPart = null
-shop: Shop
-lootTable: WeightedArrayList
-name: String
+constructor(+Planet(name: String, shop: Shop, lootTable: WeightedArrayList)
+setPart(part: ShipPart): void
+getName(): String
+getNameShowPart(): String
+getShop(): Shop
+getRandomItem(generator: Random): Item
+hasShipPart(): boolean
+removeShipPart(): ShipPart
+toString(): String
Shop
(from main)
-name: String
-inventory: Item[*] {collection="List"}
-buyModifier: double = 1.0
-sellModifier: double = 1.0
+constructor(+Shop(name: String, galacticMelBerm: Item[*])
+constructor(+Shop(name: String, inventory: Item[*], buyModifier: double, sellModifier: double)
+getInventory(): Item[*]
+getName(): String
+getSellPrice(item: Item): int
+getBuyPrice(item: Item): int
+buyItem(item: Item): int
+sellItem(item: Item): int
Item
(from items)
-name: String
-description: String = ""
-value: int = 0
+constructor(+Item(name: String)
+constructor(+Item(name: String, value: int)
+constructor(+Item(name: String, description: String)
+constructor(+Item(name: String, value: int, description: String)
+copy(): Item
+getName(): String
+getDescription(): String
+setDescription(desc: String): void
+setValue(value: int): void
+getValue(): int
+getEffectsString(): String
+toString(): String
+equals(item: Item): boolean
ShipShieldUpgradeItem
(from items)
-maxShieldIncrease: int = 0
+constructor(+ShipShieldUpgradeItem(name: String, value: int, maxShieldIncrease: int)
+getEffectsString(): String
+copy(): Item
+applyEffects(ship: Ship): void
ShipWeaponItem
(from items)
-damage: int = 0
+constructor(+ShipWeaponItem(name: String, value: int, weaponDamage: int)
+applyEffects(ship: Ship): void
+getEffectsString(): String
+copy(): Item
+getDamage(): int
ShipPart
(from items)
+constructor(+ShipPart()
+copy(): Item
ConsumableItem
(from items)
+constructor(+ConsumableItem(name: String, value: int)
+applyEffects(crew: CrewMember): void
+getRemainingUses(): int
+isEmpty(): boolean
+equals(item: ConsumableItem): boolean
GenericRestoreItem
(from items)
-usesRemaining: int = 1
-healthRestoreAmount: int = 0
-fullnessRestoreAmount: int = 0
+constructor(+GenericRestoreItem(name: String, value: int, healthAmount: int, fullnessAmount: int)
+constructor(+GenericRestoreItem(name: String, value: int, healthRestore: int, fullnessRestore: int, uses: int)
+copy(): Item
+getHealthRestoreAmount(): int
+getFullnessRestoreAmount(): int
+getRemainingUses(): int
+applyEffects(crew: CrewMember): void
+equals(item: GenericRestoreItem): boolean
+toString(): String
+getEffectsString(): String
PlagueCure
(from items)
-beenUsed: boolean = false
+constructor(+PlagueCure(name: String, value: int)
+applyEffects(crew: CrewMember): void
+copy(): Item
+getRemainingUses(): int
+getEffectsString(): String
CustomPlanetListCellRenderer
(from gui)
+constructor(+CustomPlanetListCellRenderer()
+getListCellRendererComponent(list: JList, planet: Planet, index: int, isSelected: boolean, cellHasFocus: boolean): Component
CustomCrewListCellRenderer
(from gui)
-plagueIcon: ImageIcon {readOnly}
-hungryIcon: ImageIcon {readOnly}
-deadIcon: ImageIcon {readOnly}
-stressedIcon: ImageIcon {readOnly}
-deadIcon: ImageIcon {readOnly}
-plagueLabel: JLabel
-hungryLabel: JLabel
-tiredLabel: JLabel
-stressedLabel: JLabel
-deadLabel: JLabel
-crewText: JLabel
+constructor(+CustomCrewListCellRenderer()
+getListCellRendererComponent(list: JList, crew: CrewMember, index: int, isSelected: boolean, cellHasFocus: boolean): Component
-displayIcons(crew: CrewMember): void
EventPopupWindow
(from gui)
-frame: JFrame
+constructor(+EventPopupWindow(message: String)
+constructor(+EventPopupWindow(message: String, buttonText: String)
-initialize(message: String, buttonText: String): void
<interface>
Observer
(from main)
-update(o: Observable, arg: Object): void
CrewEffectChangeObserverEvent
(from crew)
-effect: CrewMemberEffect
-isAdded: boolean
+constructor(+CrewEffectChangeObserverEvent(isAdded: boolean, effect: CrewMemberEffect)
+wasAdded(): boolean
+wasRemoved(): boolean
+getEffect(): CrewMemberEffect
CrewState
(from crew)
-funds: int = 0
-inventory: Item[*] {collection="List"}
-crew: CrewMember[*] {collection="List"}
-ship: Ship
+constructor(+CrewState(crewMembers: CrewMember[*], ship: Ship)
+getFunds(): int
+setFunds(funds: int): void
+addFunds(funds: int): void
+subtractFunds(funds: int): void
+getInventory(): Item[*]
+getShipPartsFoundCount(): int
+hasScientist(): boolean
+addItemToInventory(item: Item): void
+removeItemFromInventory(item: Item): void
+useItem(item: ConsumableItem, crewMember: CrewMember): void
+useItem(item: ShipUpgrade): void
+repairShip(crewMember: CrewMember): void
+applyDayStartEffects(): void
+getCrew(): CrewMember[*]
+getCrewWithActionPoints(minAP: int): CrewMember[*]
+getLivingCrewCount(): int
+getShip(): Ship
+toString(): String
Observable
(from main)
-changed: boolean = false
-obs: Vector
+constructor(+Observable()
+addObserver(o: Observer): void
+deleteObserver(o: Observer): void
+notifyObservers(): void
+notifyObservers(arg: Object): void
+deleteObservers(): void
+setChanged(): void
+clearChanged(): void
+hasChanged(): boolean
+countObservers(): int
InvalidCrewMemberException
(from crew)
+constructor(+InvalidCrewMemberException()
+constructor(+InvalidCrewMemberException(message: String)
CrewMemberFactory
(from crew)
+createCrewMember(newCrewMemberType: int, name: String): CrewMember
+createCrewMember(typeStr: String, name: String): CrewMember
Ship
(from crew)
-name: String
-maxShieldLevel: int = 100
-shieldLevel: int
-weapon: ShipWeaponItem
+constructor(+Ship(name: String)
+constructor(+Ship(name: String, maxShieldLevel: int)
+getName(): String
+getShieldLevel(): int
+getMaxShieldLevel(): int
+takeDamage(damage: int): void
+increaseShield(amount: int): void
+increaseMaxShield(amount: int): void
+decreaseMaxShield(amount: int): void
+setWeapon(weapon: ShipWeaponItem): ShipWeaponItem
+getWeapon(): ShipWeaponItem
CrewMember
(from crew)
-HUNGER_THRESHOLD: int = 30 {readOnly}
-TIRED_THRESHOLD: int = 30 {readOnly}
-STRESS_THRESHOLD: int = 30 {readOnly}
-PLAGUED_DAMAGE: int = 20 {readOnly}
-NEGLECTED_DAMAGE: int = 30 {readOnly}
-ROUTY_SLEEP_DAMAGE: int = 30 {readOnly}
-ROUTY_HUNGER_DAMAGE: int = 20 {readOnly}
-IDEEALLY_SHIP_REPAIR: int = 50 {readOnly}
-IDEEALLY_SLEEP_RESTORE: int = 50 {readOnly}
-name: String
-title: String = "Average Joe"
-health: int
-maxHealth: int = 100
-fullness: int
-maxFullness: int = 100
-restedness: int
-maxRestedness: int = 100
-actionPoints: int
-maxActionPoints: int = 2
-activeEffects: EnumSet
+constructor(+CrewMember(name: String, title: String)
+constructor(+CrewMember(name: String, title: String, maxHealth: int, maxFullness: int, maxRestedness: int, maxActionPoints: int)
+getName(): String
+getTitle(): String
+getFullTitle(): String
+getHealth(): int
+getMaxHealth(): int
+setHealth(health: int): void
+getFullness(): int
+setFullness(): int
+getMaxFullness(): int
+setFullness(fullness: int): void
+changeFullness(amount: int): void
+getRestedness(): int
+getMaxRestedness(): int
+setRestedness(restedness: int): void
+changeRestedness(amount: int): void
+getActionPoints(): int
+getMaxActionPoints(): int
+useItem(item: ConsumableItem): void
+repairShip(ship: Ship): void
+sleep(): void
+resetActions(): void
+useAction(): void
+hasActionAvailable(): boolean
+addEffect(effect: CrewMemberEffect): boolean
+addEffect(effect: CrewMemberEffect): void
+removeEffect(effect: CrewMemberEffect): void
+isDead(): boolean
+setBAStats(): void
+canAct(): boolean
+canActionPoints(int): boolean
+getFormatedStatusString(): String
+toString(): String
+applyStartBonuses(crewState: CrewState): void
+applyDayStartEffects(crewState: CrewState): void
<enumeration>
CrewMemberEffect
(from crew)
PLAGUED
HUNGRY
TIRED
DEAD
STRESSED
Robot
(from crew)
+constructor(+Robot(name: String)
+getClassDescription(): String
+addEffect(effect: CrewMemberEffect): void
+toString(): String
Engineer
(from crew)
-REPAIR_MODIFIER: double = 1.5 {readOnly}
+constructor(+Engineer(name: String)
+getClassDescription(): String
+repairShip(ship: Ship): void
Investor
(from crew)
-FUNDS_BONUS: int = 100 {readOnly}
-SHIELD_BONUS: int = 30 {readOnly}
+constructor(+Investor(name: String)
+getClassDescription(): String
+applyStartBonuses(crewState: CrewState): void
Student
(from crew)
+constructor(+Student(name: String)
+getClassDescription(): String
+setHealth(health: int): void
Scientist
(from crew)
+constructor(+Scientist(name: String)
+getClassDescription(): String
Medic
(from crew)
+constructor(+Medic(name: String)
+applyStartBonuses(crewState: CrewState): void
+getClassDescription(): String
<interface>
ShipUpgrade
(from items)
+applyEffects(ship: Ship): void
+getEffectsString(): String
GameEventManager
(from eventmanager)
-gameActionEventList: WeightedArrayList<GameEvent>[*] {collection="ArrayList"}
+constructor(+GameEventManager()
+setActionEvents(action: GameAction, events: WeightedArrayList): void
+getRandomEvent(action: GameAction, randomGenerator: Random, env: GameEnvironment): GameEvent
<enumeration>
GameAction
(from eventmanager)
CREW_TRAVEL
CREW_EXPLORE
DAY_START
GameEvent
(from eventmanager)
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+canEventApply(env: GameEnvironment): boolean
+getEventMessage(): String
NullEvent
(from eventmanager)
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+getEventMessage(): String
+canEventApply(env: GameEnvironment): boolean
ShipPartFindEvent
(from eventmanager)
-partFound: ShipPart = null
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+getEventMessage(): String
AsteroidEvent
(from eventmanager)
-MAXBASE_DAMAGE: int = 50 {readOnly}
-SCALED_DAMAGE: int = 80 {readOnly}
-isLethalDamage: boolean = false
-totalDamage: int = 0
-reduceDamage: int = 0
-overallDamage: int = 0
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+getEventMessage(): String
ItemFindEvent
(from eventmanager)
-foundItem: Item
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+canEventApply(env: GameEnvironment): boolean
+getEventMessage(): String
MoneyFindEvent
(from eventmanager)
-MIN_MONEY_FOUND: int = 5 {readOnly}
-MAX_MONEY_FOUND: int = 50 {readOnly}
-moneyFound: int
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+getEventMessage(): String
PiratesEvent
(from eventmanager)
-targetItem: Item = null
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+canEventApply(env: GameEnvironment): boolean
+getEventMessage(): String
SpacePlagueEvent
(from eventmanager)
-MAX_DAMAGE: int = 20 {readOnly}
-MIN_DAMAGE: int = 10 {readOnly}
-target: CrewMember = null
-damage: int = 0
+createEvent(randomGenerator: Random, env: GameEnvironment): GameEvent
+applyEvent(env: GameEnvironment): void
+canEventApply(env: GameEnvironment): boolean
+getEventMessage(): String