NORTH
SOUTH
EST
WEST □ dashboard: final Map<ComponentPosition, Component>
□ discards: final List<Component> reserves: final Component[2]
crew: int
batteries: int
goods: final List < Good> protectedSides: final List<DirectionType> players: final List<Pair<PlayerData, Integer>> • getComponentAtPosition(ComponentPosition): Component startingDeck: final List<PlayerData> ■ Card
□ level: final int
□ isLearner: final bool
■ getLevel(): int
■ getIsLearner(): bool
■ resolve(): void • getPositionOfComponent(Component): ComponentPosition □ pile: final List<Card> getAllOccupiedPositions(): Set<ComponentPosition>
 getAllFreePositions(): Set<ComponentPosition>
 insertComponent(ComponentPosition, Component): void
 destroyComponent(ComponentPosition): void
 getDiscards(): List<Component>
 getReserves(): Component[2]
 countExposedConnectors(): int timeManagment: final TimeavaiableGoods: final List<Good> CREW CANNON ENGINE getPlayers(): List<Pair<PlayerData, Integer>> rewards: final List<ColorType> getPile(): List<Card> getIsBig(): boolgetDirectionFrom(): DirectionType getTimeManagment(): Timeshuffle(): void • getRewards(): List<ColorType> drawCard(): Card getCrew(): int movePlayer(PlayerData, int): voidmoveToStartingDeck(PlayerData): void setCrew(int): void getBatteries(): int moveToBoard(PlayerData): void setBatteries(int): void getAvaiableGoods(): List<Good> getGoods(): List<Good> getShields(): List<ShieldComponent> getCannons(): List<CannonComponent> getEngines(): List<EngineComponent>getCabins(): List<EngineComponent> calculateFirepower(int): int calculateEnginepower(int): int color: final ColorType
username: final String
ship: final Ship
credits: int
getColor(): ColorType
getUsername(): String
getShip(): Ship
getCredtis(): int
setCredits(int): void □ firepower: final int
□ credits: final int ComponentPosition
x: final int
y: final int
getX(): int
getY(): int **C** CountablePenaltyCombactZone timer: Timer
hourglassPos: int
getTimeLeft(): int
setTimeLeft(int): void
startTimer(): void
getHourglassPos(): int credits: final int **C** Component **C** CombactZoneCard E AlienType CANNON ENGINE penaltyNumber: final int □ goods: final List<ColorType>
□ days: final int □ days: final int
□ crew: final int days: final int goods: final List<ColorType> © EpidemicCard C SabotageCard C OpenSpaceCard GREEN YELLOW BLUE □ meteors: final List<Meteor> penaltyType: final MalusType □ lines: final List<Pair<CriteriaType, PenaltyCombartZone>> cannonFires: final List<CannonFire> connectors: final ConnectorType[4] cannonFires: final List < CannonFire > • getConnectors(): ConnectorType[4] getFirePower(): intgetGoods(): List<ColorType>getDays(): int getCrew(): intgetCredits(): intgetDays(): int getFirepower(): intgetCredits(): intgetDays(): intgetCrew(): int getCrew(): intgetGoods(): List<ColorType>getDays(): int getPenaltyNumber(): intgetPenaltyType(): Malustype getFirepower(): intgetCredits(): intgetDays(): intgetCannonFires(): List<CannonFire> • getCannonFires(): List<CannonFire> • getLines(): List<Pair«CriteriaType, PenaltyCombartZone» **C** CargoHoldsComponent **C** CannonComponent **C** CannonFire **C** EngineComponent humans: intalien: Alienstarting: bool □ isTriple: final bool
□ goods: final List<Good> **C** ShieldComponent □ direction: final DirectionType
□ isDouble: final bool direction: final DirectionType isBig: final booldirectionFrom: final DirectionType □ type: final AlienType getIsTriple(): bool
getGoods(): List<Good>
loadGoods(goods: List<Good>): void
unloadGoods(goods: List<Good>): void getHumans(): int
 setHumans(num): void
 getAlien(): Alien
 setAlien(alien): void

OBJE

 getIsTriple(): bool
 getBatteries(): int
 setBatteries(num): void • getType(): AlienType
• getIsDouble(): bool
• getDirectionType
• getDirectionType getAlien(): AlienType
 getType(): ColorType getIsBig(): boolgetDirectionFrom(): DirectionType • getIsDouble(): bool **C** Special Cargo Holds Component □ isDouble: final bool □ goods: final List<Good> • getIsDouble(): bool getGoods(): List<Good>