Documentation for space battle project

**public abstract class Sector** :

Represents a gameboard in which ships are placed

*+ Sector(size: int)*

Constructor for sector, length and width are equal to size

*+ rowToLetter(row: int): String*

Returns a number  as corresponding letter to help the toString() method

*+ toString(): String*

Abstract method


**public class SpaceSector:**

extends from Sector and represents a player`s sector

*+ SpaceSector(size: int)*

Uses Sector constructor

*+ getSector(): Spaceship[][]*

Getter method

*+ setSector(sector: Spaceship[][])*

Setter method

*+ getPossiblePlacements(ship: Spaceship): boolean[][]*

Returns a boolean array containing all possible placements for as certain ship

*- placingShipPossible(ship: Spaceship, row: int, col: int): boolean*

Returns true if placing the ship`s anchor in a certain position is possible

*+ placeShip(ship: Spaceship, row: int, col: int)*

Places ship in a player`s sector

*+ getPosition(): int[]*

Returns the ship`s anchor point

*+ toString()*

Draws a players sector containing their ships and destroyed ships

*+ shipInSector(): boolean*

Returns true if a ship is in the sector, used to check if a ship is completely destroyed

**public class EnemySector**

represents a map of an opponent`s sector

*+ EnemySector(size: int)*

Uses Sector constructor

*+ setMap(map: int[][])*

Setter method

*+ getMap(): int[][]*

Getter method

*+ toString(): String*

Draws map of an opponent`s sector containing all the information known to a player


**public abstract class Spaceship**

represents a spaceship

*+ Spaceship(type: ShipType, shape: int[][], price: int)*

constructor that is given a ShipType, the ships shape as an int[][] array and the ship`s price

*+ getType(): ShipType + getShape(): int[][]*

Getter method

*+ getPrice(): int + getMarker(): int[]*

Setter method

*+ setMarker(marker: int[])*

Setter Method

*+ isPlaced(): boolean*

Returns true if Spaceship was placed in sector

*+ setPlaced(placed: boolean)*

Setter method

*+ isDestroyed(): Boolean*

Returns true if a field is a Wreck

*+ toString(): String*

Draws the ship`s shape to help place it

**<<enumeration>> ShipType**

Contains all possible types of ships that hava different names, shapes and prices

**public class Battleship, public class Carrier, public class Cruiser, public class Submarine, public class Corvette, public class Wreck**

classes that extend from Spaceship and represent ships of every possible type and have a constructor in which type, shape and price are set

**public class Commander**

represents a player

*+ Commander(sectorSize: int, coins: int)*

Constructor that is given the amount of coins and the size of a player`s sector

*+ getFleet(): Spaceship[]*

*+ setFleet(fleet: Spaceship[])*

*+ getSpaceSector(): Spacesector*

*+ setSpaceSector(spaceSector: Spacesector)*

*+ getEnemySector(): EnemySector*

*+ setEnemySectot(enemySector: EnemySector)*

*+ getCoins(): int*

*+ setCoins(coins: int)*

Getter and setter methods

*+ isDefeated(): boolean*

Returns true if a player`s sector only contains Wrecks and empty spaces

*+ getShipsToPlace(): Spaceship[]*

Returns array with all ships that haven`t been placed yet

*+ updateEnemyMap(row: int, col: int, hit: boolean)*

Updates the opponent`s map after every attack with new information

*+ addShipToFleet(type: ShipType)*

Adds a ship of a given type to a player`s fleet

*+ toString(): String*

Draws the player`s sector and a map of their opponent`s sector

*+ noPossiblePlacements(ship: Spaceship): Boolean*

Returns True if a ship can`t be placed in its current rotation

Documentation for space battle project

**Public class Alien**

Extends from Commander and represents the virtual player controlled by the computer

*+ Alien(sectorSize: int, coins: int)*

Constructor that is give the amount of coins and the size of a player`s sector

*- getRanShipInFleet(): Spaceship*

Returns a random ship inside the alien`s fleet

*- rotateRandom(ship: Spaceship)*

Rotates a ship a random amount of times

*- placeShipRandom(ship: Spaceship)*

Places ship in a random allowed tile

*- addRandomShips()*

Adds ships of random types to the Alien`s fleet util not enough coins are left to buy more ships

*+ getShotCoordinates(): int[]*

Returns a random shot coordinates for Alien, ensures tile hasn`t been attacked yet

*+ setRandom()*

Unused method to test game