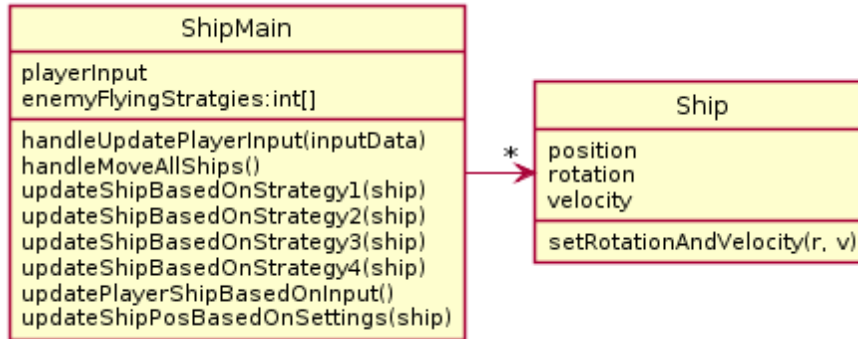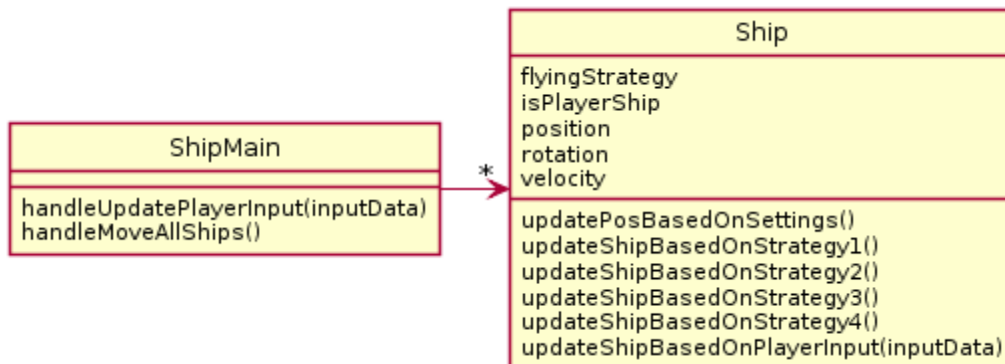**Design Problems, HW3**   Name: _____ CM: _____

Spaceship: In a single player spaceship racing game, the mechanics surrounding ships are particularly important. Each ship has its own current position, rotation, and velocity. Rotation and velocity can be increased or decreased by the controller - this is how the ship is steered. When a ship or multiple ships are computer controlled there are 4 different flying strategies that control how the ship moves in the game. When the player controls his/her ship, the ship gets input from a gamepad which sets rotation and velocity. There are two commands to consider: updatePlayerInput which is called when a player presses the gamepad and moveAllShips which updates the position of all ships in the game based on their current settings.

Solution A

ShipMain
playerInput enemyFlyingStratgies:int[]
handleUpdatePlayerInput(inputData) handleMoveAllShips() updateShipBasedOnStrategy1(ship) updateShipBasedOnStrategy2(ship) updateShipBasedOnStrategy3(ship) updateShipBasedOnStrategy4(ship) updatePlayerShipBasedOnInput() updateShipPosBasedOnSettings(ship)

*

Ship
position rotation velocity
setRotationAndVelocity(r, v)

Solution B

Ship
flyingStrategy isPlayerShip position rotation velocity
updatePosBasedOnSettings() updateShipBasedOnStrategy1() updateShipBasedOnStrategy2() updateShipBasedOnStrategy3() updateShipBasedOnStrategy4() updateShipBasedOnPlayerInput(inputData)

ShipMain
handleUpdatePlayerInput(inputData) handleMoveAllShips()

*

Explain the problems with each one (and give the number of the OO Principles violated)

Problems With A

Problems With B

Write your UML solution that fixes all the problems on the **back of this sheet of paper.**