A2Prj Class Diagram

Starter
-current : Form
-theme : Resources
+init(context : Object) : void
+start() : void
+stop() : void
+destroy() : void

EnergyStation
<<Property>> -capacity : int
+EnergyStation(size : int, x : double, y : double, color : int, w : GameWorld)
+toString() : String

Base
<<Property>> -sequenceNumber : int
+Base(size : int, x : double, y : double, color : int, w : GameWorld, sequenceNumber : int)
+setColor(color : int) : void
+toString() : String

AttackStrategy
-npr : NonPlayerRobot
~AttackStrategy(enemyRobot : NonPlayerRobot)
+apply() : void

NonPlayerRobot
-NPR_MAX_POSSIBLE_DAMAGE : int = 1000
<<Property>> -strategy : IStrategy
+NonPlayerRobot(size : int, x : double, y : double, color : int, speed : int, maximumSpeed : int, energyLevel : int, energyConsumptionRate : int, world : GameWorld)
+invokeStrategy() : void
+changeStrategy() : void
+steerTowardNextBase() : void
+steerTowardPlayerRobot() : void
+setDamageLevel(damageLevel : int) : void
+move() : void
+toString() : String
-computeIdealSteeringAngle(xTargetPos : double, yTargetPos : double) : int

<<Interface>> IStrategy
+apply() : void

BlueButton
-setBlueButtonUnselectedStyle() : void
-setBlueButtonSelectedStyle() : void
+BlueButton(text : String)

<<Interface>> ISteerable

Fixed
+Fixed(size : int, x : double, y : double, color : int, w : GameWorld)

NextBaseStrategy
-npr : NonPlayerRobot
~NextBaseStrategy(enemyRobot : NonPlayerRobot)
+apply() : void

GameObjectIterator
+index : int = 0
+hasNext() : Boolean
+getNext() : GameObject
+Reset() : void

<<Interface>> IIterator
<<Property>> +next : GameObject
+hasNext() : Boolean
+Reset() : void

Robot
-SPEED_INCREASE : int = 5
-SPEED_DECREASE : int = 5
-MAX_STEERING_DIRECTION : int = 40
-MAX_POSSIBLE_DAMAGE : int = 100
<<Property>> -steeringDirection : int
<<Property>> -maximumSpeed : int
<<Property>> -energyLevel : int
<<Property>> -energyConsumptionRate : int
<<Property>> #damageLevel : int
<<Property>> -lastBaseReached : int
<<Property>> -lives : int
-initialXLocation : double
-initialYLocation : double
-initialEnergyValue : int
-robotInstance : Robot = null
#Robot(size : int, x : double, y : double, color : int, speed : int, maximumSpeed : int, energyLevel : int, energyConsumptionRate : int, world : GameWorld)
+getInstance(size : int, x : double, y : double, color : int, speed : int, maximumSpeed : int, energyLevel : int, energyConsumptionRate : int, world : GameWorld) : Robot
+reinitializeRobot() : void
+accelerate() : void
+brake() : void
+steerLeft() : void
+steerRight() : void
+toString() : String
-adjustSpeed(speed : int) : void
+move() : void
-clampSteering() : void

<<Interface>> ICollection
<<Property>> +iterator : IIterator
+add(go : GameObject) : void

GameObject
<<Property>> -size : int
-xPos : double
-yPos : double
<<Property>> -color : int
<<Property>> -world : GameWorld
+GameObject(size : int, x : double, y : double, color : int, w : GameWorld)
+getX() : double
+setX(x : double) : void
+getY() : double
+setY(y : double) : void
+toString() : String

Movable
<<Property>> -speed : int
<<Property>> -heading : int
+Movable(size : int, x : double, y : double, color : int, speed : int, heading : int, world : GameWorld)
+move() : void
+toString() : String
#locationBoundAdjust() : void

GameObjectCollection
-objectList : GameObject
+GameObjectCollection()
+add(go : GameObject) : void
+add(index : int, go : GameObject) : void
+size() : int
+getIterator() : IIterator

Drone
+Drone(size : int, x : double, y : double, color : int, speed : int, heading : int, world : GameWorld)
+move() : void
+toString() : String

SoundCommand
-gw : GameWorld
+SoundCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

RightCommand
-gw : GameWorld
+RightCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

CollideWithDroneCommand
-gw : GameWorld
+CollideWithDroneCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

LeftCommand
-gw : GameWorld
+LeftCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

GameWorld
-NPR_COLLISION_DAMAGE : int = 10
<<Property>> -height : int = 0
<<Property>> -width : int = 0
<<Property>> -count : int = 0
-sound : boolean = false
<<Property>> -numBases : int = 0
-goc : GameObjectCollection = null
+init() : void
+mCommand() : void
+dCommand() : void
+getLivesLeft() : int
+getSoundSetting() : boolean
+setSoundSetting(on : boolean) : void
+setDimensions(height : int, width : int) : void
+accelerate() : void
+brake() : void
+tick() : void
+changeStrategy() : void
+getNextBaseInSequence(sequenceNumber : int) : Base
+exit() : void
+reInitialize() : void
+getPlayerRobot() : Robot
+turnLeft() : void
+turnRight() : void
+collideWithDrone() : void
+collideWithNPR() : void
+collideWithBase(baseNum : int) : void
+collideWithEnergyStation() : void
+getGameObjectsIterator() : IIterator
+addRandomEnergyStation() : void
-fadeColor(obj : GameObject) : void
-updateMoveableObjects() : void

CollideWithEnergyStationCommand
-gw : GameWorld
+CollideWithEnergyStationCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

AccelerateCommand
-gw : GameWorld
+AccelerateCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

CollideWithNPRCommand
-gw : GameWorld
+CollideWithNPRCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

ExitDialogCommand
-gw : GameWorld
+ExitDialogCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void
-displayExitDialogue() : void

CollideWithBaseCommand
-gw : GameWorld
+CollideWithBaseCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void
-createBaseCollisionDialog() : int

HelpCommand
-gw : GameWorld
+HelpCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void
-displayHelpDialogue() : void

AboutCommand
-gw : GameWorld
+AboutCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void
-displayAboutDialogue() : void

TickCommand
-gw : GameWorld
+TickCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

BrakeCommand
-gw : GameWorld
+BrakeCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

MapViewLabel
-gwo : GameWorld
+MapViewLabel(gw : GameWorld)
+paint(g : Graphics) : void

StrategiesCommand
-gw : GameWorld
+StrategiesCommand(command : String, world : GameWorld)
+actionPerformed(ev : ActionEvent) : void

DebugMapViewCommand
-mv : MapView
-gw : GameWorld
+DebugMapViewCommand(command : String, mv : MapView, gw : GameWorld)
+actionPerformed(ev : ActionEvent) : void

ScoreView
-timeValue : Label
-lifeValue : Label
-lastBaseReachedValue : Label
-playerEnergyValue : Label
-playerDamageValue : Label
-soundSettingValue : Label
+ScoreView(gw : GameWorld)
+update(o : Observable, arg : Object) : void
+createUiElements(timeStep : int, livesLeft : int, lastBase : int, energyValue : int, damageValue : int, soundSetting : boolean) : void

Game
-gw : GameWorld
-mv : MapView
-sv : ScoreView
+Game()
-createGUI() : void
-createToolbar() : void
-createLeftContainer() : void
-createRightContainer() : void
-createBottomContainer() : void

MapView
-imgLabel : MapViewLabel = null
+update(o : Observable, arg : Object) : void
+createDebugMapviewImage(gw : GameWorld) : void