

Model

SHAPE
- xCoord = int - yCoord = int - initialY = int - lastXCoord = int - lastYCoord = int - vSpeed = int - hSpeed = int - width = int - height = int
+ Shape(int newX, int newY, int newWidth, int newHeight) + getX/YCoord () : int + getLastX/YCoord() : int + getInitialY(): int + setX/YCoord(int newX/Y) + getH/Vspeed(): int + getHeight/Width(): int + resetY()

Drawable
+ draw(Graphics g)

PlayerShip
+ playerShip(int screenLength, int newY, int size, int newSpeed) + inBounds(int boardWidth) + move(String direction) + draw(Graphics g) + toString(): String

InvadersGameLogic
- screenHeight = int - screenWidth = int - alienShot= Shot - playerShot= Shot - alienInvaders = AlienArray - ship = playerShip - barrier = Barrier - gameStatus = String - gameVersion = String
+ InvadersGameLogic(String version) + getGameStatus() : String + getArray() : AlienArray + getShip(): playerShip + getShot(): playerShot + getAlienShot(): alienShot + getBarrier(): barrier + getScreenWidth() / Height(): int + moveAliens() + checkStatus() + shipMovement() + moveAlienShot() + checkAlienHit() + checkShipHit() + handleShotInteraction() + shotAttempt() + randInt(int min, int max) + shotGeneration()

Alien
- isAlive = boolean
+ Alien(int hSpeed, int vSpeed, int newSize) + isAlive(): boolean + destroyAlien() + reachedEnd(int boardHeight): Boolean + moveRight() + moveDown() + moveLeft() + draw(Graphics g)

AlienArray
- numAliens = int - rowsAliens = int - moveRight = boolean - aliens = Alien[][] - leftBoundary = int - rightBoundary = int
+ AlienArray(String version) + getAliens(): Alien[][] + getNumAliens():int + getRowsAliens():int + createAlienArray() + setAliens() + checkBoundary() + aliensMovement(int width) + drawAlienArray(Graphics g) + setGUlaliens()

Shot
- shotFired= boolean
+ Shot(int startingRow, int newSpeed) + Shot(int startingRow, int newSpeed, int newWidth, int newLength) + shotFired (Boolean shotStatus) + getShotFired(): boolean + tryShot(int ShipLocation) + newAlienShot(int newX, int newY) + moveShot() + checkHit(int targetX, int targetY, int targetDiameters): boolean + checkGUIBarrierHit(Barrier barrier, int boardWidth,int boardHeight) : boolean + checkTextHit(int targetRow, int targetCOL, int lastCol): boolean + alienShotShip(int shipXCoord, int shipYcoord): boolean + inBounds(int height) + checkBarrierHit(Barrier barrier, int boardWidth, int boardHeight): boolean + draw(Graphics g)

Barrier
- barrier1HP = int - barrier2HP = int - barrier3HP = int - location = int - barrier1 = Color - lastBarrier1 = Color
+ Barrier(screenWidth: int, screenHeight: int) + getBarrier1HP()/2HP()/3HP(): int + updateBarrier1/2/3() + getLocation(): int + setLocation(newLocation: int) + draw(Graphics g)

Control

InvadersGameController
- keyboard = Scanner - logic = InvadersGameLogic - gui = InvadersGameGUI - text = InvadersGameText
+ InvadersGameController() + playGui() + playText()

TimerListener

KeyboardListener
+ keyPressed(KeyEvent e) + keyReleased(KeyEvent e) + keyTyped(KeyEvent e)

RunMe
+ main(String[] args)

View

InvadersGameText
- boardHeight = int - boardWidth = int - board = char[][]
+ createBoard() + validLocation(int x,int y):boolean + printBoard() + createBarriers(int rightBarrier, int leftBarrier, int traversePoint) + emptyBarriers(int rightBarrier, int leftBarrier, int traversePoint) + drawBarriers(Barrier barrier) + drawShip(PlayerShip ship) + drawShot(Shot shot) + drawAliens(AlienArray array) + drawAlienShot(Shot shot, Barrier barrier, PlayerShip ship) + drawCurrentState(PlayerShip ship, Shot shot, Shot alienShot, AlienArray array)

InvadersGameGUI
- windowWidth = int - windowHeight = int - gameStatus = String - toDraw = Object[]
+ InvadersGameGUI(InvadersGameLogic newLogic) + updateScreen()
Canvas extends JComponent + paintComponent(Graphics g)

