## Class Diagram: Tetros MADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY ( **SmartDraw SmartDraw** Smart Draw SmartDraw<sup>\*</sup> **SmartDray** MADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY ( SmartDraw<sup>\*</sup> Smart Draw Smart Draw SmartDraw<sup>\*</sup> **SmartDray** MADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY : **SmartDraw** SmartDraw<sup>\*</sup> **SmartDraw SmartDray** +tick() + render(g:Graphics) MADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY : - makeMove(game:MainGame) Smart Draw Smart Drav EndGame thread:Thread running:boolean +tick() Shape -bottomHit:boolean + render(q:Graphics) board:Board shape:Shape - tetromino :ShapeType - coords[][]:int - coordsTable:int[][][] - originalShape: Shape -img: BufferedI mage -introScreen: Intro + setShape(shape:ShapeType) + getShape():ShapeType -hud: HUD MADE WITH A TRIAL COPY OF - gameOver:EndGame -timeAfterLoss:int MADE WITH A TRIAL COP MADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY ( + setRandomShape() + setNewX(index:int,x:int) + setNewY(index:int, y:int) + getX(index:int):int **SmartDray** + stop( + run() + tick() Intro + getY(index:int):int + render() + clamp(val:int, min:int, max:int):int + setBottomHit(value:boolean) - board:Board - shape:Shape + render(g:Graphics) + isBottomHit():boolean + setNewShape() + getMainShape(): Shape + startGame( board: Board, shape: Shape) + getBoard():Board + enterPressed() + setOrigShape() MADE WITH A TRIAL COPY OF MADE WITH A BORIGAL COPY OF + getQrigShape(): Shape + main(String args[]) MADE WITH A TRIAL COPY OF MADE WINDOW, TRIAL COPY ( maxX:int maxY:int board:int[][] direction: String KeyInput + getBoard():int[][] + placeShape(shape:Shape) + checkFullRow():boolean - board:Board RunGame + dearBoard() + dearBoard() + moveLeft (shape:Shape) + moveRight (shape:Shape) + keyPressed(e:keyEvent) GameObject MADE WITH A TRIAL COPY O + moveDown(shape:Shape) + leftCollision(shape:Shape):boolean MADE WITH A TRIAL COPY O ADE WITH A TRIAL COPY OF MADE WITH A TRIAL COPY ( - tick() + main(args:String[]) -render(g:Graphics) + iet.comision(stape::Shape);boolean + rightCollision(shape:Shape);boolean + bottomCollision(shape:Shape);boolean + rotateLeft(shape:Shape, originalShape: Shape) + rotateRight(shape:Shape, originalShape: Shape) + setDirection(direction:String) + getDirection():String