-frame: JFrame +WIDTH = 464: int +HEIGHT = 500: int +main(String[] args): void main::Controller -thread: Thread -g: Graphics2D -image: BufferedImage -state = STATE.MENU: STATE -game: Board main::Board main::ImageLoader main::MainMenu -gameOver: GameOver -bounds = { new Rectangle(70, 310, 310, 55) }: Rectangle[] -serialVersionUID = 1L: long -ImageFG: BufferedImage -winScreen: WinScreen +titleF = "/BrickBreakerForeground2.png": String -titleScreenForeground: Image -PDL: Paddle -menu: MainMenu +titleB = "/BrickBreakerBackground3.png": String -BALL: Ball[] -titleScreenBackground: Image -pickLevel: PickLevel +PaddleImg = "/Paddle2.png": String -TILES: Brick[] +mousePoint = new Point(0, 0): Point +MainMenu(): ctor +BrickImg = "/Bricks7.png": String -lives: Image +bigFont = new Font("TimesRoman", Font.PLAIN, 25): Font +tick(): void +BallImg = "/Ball2.png": String -powerups: Powerup[] +render(Graphics g): void +smallFont = new Font("TimesRoman", Font.PLAIN, 18): Font +FireBallImg = "/FireBall.png": String -gridPos = new int[8][10]: int[][] -running = true: boolean +ArrowImg = "/Arrow.png": Štring -score = 0: int -serialVersionUID = 1L: long +PUMultiBallImg = "/PUMulitBall.png": String -GStarted = false: boolean -lastTime: long +PUGrowthImg = "/PUGrowth.png": String -debug = false: boolean -fps: double -livesLeft = 2: int +PUFireballImg = "/PUFireball.png": String +score = 0: int -NOofBALLS = 1: int handlers::MouseHandler +ImageLoader(String path): ctor +level = 0: int-ALLBRICKS = 0: int +getImage(): BufferedImage +MOUSEDOWN = false: boolean +Controller(): ctor -level: int +getSubImage(int section): BufferedImage +hasPressed = false: boolean +addNotify(): void +Board(int level): ctor handlers::KeyHandler -init(): void +MouseHandler(): ctor +init(): void +run(): void +LEFT = false: boolean +mouseClicked(MouseEvent e): void +tick(): void +mouseEntered(MouseEvent e): void +RIGHT = false: boolean -display(): void +render(Graphics g): void +UP = false: boolean +switchStates(STATE state): void +mouseExited(MouseEvent e): void +switchStates(STATE state, int level): void +mousePressed(MouseEvent e): void +KeyHandler(): ctor +mouseReleased(MouseEvent e): void +keyTyped(KeyEvent e): void files::Files +keyPressed(KeyEvent e): void +mouseDragged(MouseEvent e): void +mouseMoved(MouseEvent e): void +keyReleased(KeyEvent e): void +SaveProgress(boolean[] scores): void main::GameOver main::WinScreen main::PickLevel -mainMenu: Rectangle -mainMenu: Rectangle -arrow: Image -background: Image -background: Image -page = 1: int ~levels = { new Rectangle(50, 200, 75, 75), new Rectangle(150, 200, 75, 75), new Rectangle(250, 200, 75, 75), new Rectangle(350, 200, 75, 75) }: Rectangle[] +GameOver(): ctor +WinScreen(): ctor ~arrows = { new Rectangle(5, Frame.HEIGHT - 90, 50, 50) }: Rectangle[] +tick(): void +tick(): void +render(Graphics g): void +render(Graphics g): void +PickLevel(): ctor +tick(): void +render(Graphics g): void parts::Powerup parts::Ball parts::Brick parts::Paddle files::Level +MAX_LEVEL = 3: int ~loader: ImageLoader ~loader: ImageLoader +powerup: int ~loader: ImageLoader +bounds: Rectangle +width = 23: int -width = 50: int +bounds: Rectangle +levels = new int[4][4][9]: int[][][] +height = 23: int -height = 25: int +width = 100: int +lockedLevels = new boolean[MAX_LEVEL + 1]: boolean[] -x, y: int $-\text{level0} = \{ \{ 3, 0, 3, 0, 2, 0, 3, 0, 3, 0, 2, 0, 3, 0,$ -width = 25: int +height = 30: int -x: int -x: int -height = 25: int -x: int -y: int -y: int +remove = false: boolean -dx = 0.25: double +level: int -y: int -speed = 2: int+originalLevel: int +moveSpeed = 10: int -dy = -6: double -image: Image -dir = 0: int -addScore = false: boolean +Paddle(int x, int y): ctor +getLevel(int level): int[][] +dropPowerup = false: boolean -speed = 5: int+Powerup(int x, int y, int type): ctor +getImage(): Image +fillLevels(): void +hasDied = false: boolean +bounds: Rectangle +setX(int pos): void +tick(): void -rand: Random +bounds: Rectangle +getX(): int +getX(): int -rand: Random -timer: Timer +getY(): int +getY(): int +onFire = false: boolean -powerup = 0: int +getWidth(): int +getWidth(): int -fireSec = 0: int +getImage(): Image +getHeight(): int +Brick(int x, int y, int level): ctor +isColliding(Powerup powerup): boolean +getImage(): Image +Ball(int x, int y, boolean onFire): ctor +getImage(): Image +setX(int pos): void +moveLeft(): void +moveRight(): void +setX(int pos): void +setY(int pos): void +render(Graphics g): void +setY(int pos): void +getX(): int +getY(): int +getX(): int +getWidth(): int +getY(): int +setOnFire(int seconds): void +getHeight(): int +hasCollided(): void +tick(): void +checkBrickCollision(Brick[] bricks): void +destroyed(): void +checkPaddleCollision(Paddle paddle): void +hasPowerup(): int +switchDirections(Brick brick): void +render(Graphics g): void +hitPaddle(Paddle paddle): void +render(Graphics g): void

main::Frame