|  |
| --- |
| abstract Brick |
| -Rectangle rect |
| +Brick()  +draw(Graphics g) : void  +getBrick() : Rectangle  +abstract getImage() : Image  +abstract updateBrick() : void  +abstract getHits() : double  +abstract updateLaser() : void  +abstract updateBullet() : void  +setLoc(x,y) : void  +move() : void |

|  |
| --- |
| MetalBrick extends Brick |
| no instance variables |
| +MetalBrick()  +updateBrick() : void  +getImage() : Image  +getHits() : double  +updateLaser() : void  +updateBullet() : void |

|  |
| --- |
| NormalBrick extends Brick |
| -double hits  -String file  -boolean hasPowerUp  -PowerUp p |
| +NormalBrick(double hits, double hP)  +getHits() : double  +getPowerUp() : PowerUp  +setPowerUp(boolean has)  +getImage() : Image  +updateBrick() : void  +updateLaser() : void  +updateBullet() : void |

|  |
| --- |
| PowerUp |
| -String type  -Rectangle powerUp  -Rectangle text |
| +PowerUp(String type)  +setLoc(int x,int y) : void  +getPowerUp() : Rectangle  +getImage() : Image  +draw(Graphics g) : void  +fall() : void  +getType() : String |

|  |
| --- |
| Paddle |
| -int speed  -boolean right  -boolean left  -Rectangle paddle  -String file |
| +Paddle(int speed)  +draw(Graphics g) : void  +getImage() : Image  +getPaddle() : Rectangle  +resetPaddle() : void  +getSpeed() : int  +setSpeed(int s) : void  +normalPaddle() : void  +laserPaddle() : void  +gunPaddle() : void  +longPaddle() : void  +flip() : void |

|  |
| --- |
| Ball |
| +int xSpeed  +int ySpeed  -Rectangle ball  -Ellipse2D.Double circle |
| +Ball()  +Ball(int x, int y, int xS, int yS)  +getImage() : Image  +getBall() : Rectangle  +getCircle() : Ellipse2D.Double  +resetBall() : void  +setXSpeed(int x) : void  +setYSpeed(int y) : void |

|  |
| --- |
| interface Level |
| no instance variables |
| +setLoc(int loc, int x, int y) : void  +placeBricks() : void  +getBricks() : ArrayList<Brick> |

|  |
| --- |
| Level 1-34 |
| -ArrayList<Brick> b |
| +Level[1-34]()  +setLoc(int loc, int x, int y) : void  +placeBricks() : void  +getBricks() : ArrayList<Brick> |

|  |
| --- |
| Laser |
| -Rectangle l  -int speed |
| +Laser(int x)  +getImage() : Image  +draw(Graphics g) : void  +move() : void  +getLaser() : Rectangle |

|  |
| --- |
| Laser |
| -Rectangle b  -int speed |
| +Bullet(int x)  +getImage() : Image  +draw(Graphics g) : void  +move() : void  +getBullet() : Rectangle |

|  |
| --- |
| Arrow |
| -Rectangle arrow  -int loc  -int[] angle  -int[] x  -int[] y  -int xSpeed  -int ySpeed  -int ang  -String fileName |
| +Arrow()  +left() : void  +right() : void  +getArrow() : Rectangle  +draw(Graphics g) : void  +getImage() : Image |

|  |
| --- |
| ImageLoader |
| -TreeMap<String, BufferedImage> cache |
| +loadCompatibleImage(String path) : BufferedImage  +loadImage(String path) : BufferedImage  +getCompatibleImage(Image img) : BufferedImage |

|  |
| --- |
| Game extends JPanel implements ActionListener, KeyListener |
| -AraryList<Ball> balls  -ArrayList<Level> levels  -ArrayList<Brick> b  -Paddle p  -Ball ball  -Rectangle r  -Arrow a  -Timer timer  -int gameWidth  -int gameHeight  -JLabel background  -ArrayList<PowerUp> pU  -ArrayList<Laser> leftLasers  -ArrayList<Laser> rightLasers  -ArrayList<Bullet> bullet  -int[] locs  -int loc  -int highscore  -int currentLevel  -int lives  -int score  -int ammo  -int contacts  -int moves  -boolean laser  -boolean gun  -boolean cat  -boolean n  -boolean bomb  -boolean pause  -boolean shake |
| +main() : static void  +addLevels() : void  +Game()  +actionPerformed() : void  +keyPressed(KeyEvent e) : void  +keyReleased(KeyEvent e) : void  +keyTyped(KeyEvent e) : void  +setUpGame() : void  +update() : void  +paint(Graphics g) : void  +usePowerUp(PowerUp e) : void  +movePaddle() : void  +handleBrickContacts() : void  +handleLasers() : void  +handleBullets() : void  +handlePowerUps() : void  +handleBallDirection() : void  +moveBricks() : void  +onlyMetal() : Boolean  +nextLevel() : void  +reset() : void  +removeBalls() : void  +handleInfiniteLoop() : void  +endGame() : void |

The purpose of this project is to use java to closely emulate the game Brick Breaker from the Blackberry phone. There will be four main sections of class interactions: the bricks, the ball, the paddle, and the levels / game. Both the normal brick and metal brick extend the abstract class brick. Additionally, the power up brick extends the normal brick. The power up class gives provides the power up to the brick. The ball class is an independent class. The paddle class is the super class for three other types of paddles: gun, laser, and long. Finally, all of these part will come together to form different levels.

One new topic to me is the use of key listeners. I have used action listeners in the past, but never key listeners. Another new topic to me will be the use of the Graphics2D class. In the past, I have always used JLabels but key listeners work better with Graphics2D. Finally, I will be using JPanel for the first time. For blackjack, I simply used a JFrame. Through doing this project, I wish to learn how a game actually operates rather than simply playing a game and not thinking about the code that is involved.