<<interface>> IntersectionDetectable

intersects(player : Player) : boolean

Enemy

− speed : double
− movingUp : boolean
− radius : double

+ Enemy(x : double, y : double, radius : double, speed : double)
+ move() : void
+ intersects(player : Player) : boolean
= generateCirclePoints(x : double, y : double, radius : double, sides : int) : Point[]
+ paint(brush : Graphics) : void

SpinningRectangle

rotationSpeed : int

+ SpinningRectangle(initialPosition : Point, initialRotation : double)
+ move() : void
+ intersects(player : Player) : boolean
+ paint(brush : Graphics) : void

Point

+ y : double
+ x : double

+ Point(inX : double, inY : double)
+ getX() : double
+ getY() : double
+ setX(x : double) : void
+ setY(y : double) : void
+ clone() : Point

Polygon

+ rotation : double
+ position : Point
− shape : Point[]

+ Polygon(inShape : Point[], inPosition : Point, inRotation : double)
+ getPoints() : Point[]
+ contains(point : Point) : boolean
+ rotate(degrees : int) : void
− findArea() : double
− findCenter() : Point

Score

− score : int

+ Score()
+ decreaseScore(points : int) : void
+ getScore() : int

Timer

− startTime : long

+ Timer()
+ getElapsedTime() : long

WorldsHardestGame

gameCompleted : boolean
spinningRectangle2 : SpinningRectangle
spinningRectangle1 : SpinningRectangle
enemy9 : Enemy
enemy8 : Enemy
enemy7 : Enemy
enemy6 : Enemy
enemy5 : Enemy
enemy4 : Enemy
enemy3 : Enemy
enemy2 : Enemy
enemy1 : Enemy
enemies : ArrayList<Enemy>
player : Player
checkeredY : int
checkeredX : int
checkeredSize : int
height : int
width : int
− playerScore : Score
elapsedTime : double
− gameTimer : Timer

+ WorldsHardestGame()
+ paint(brush : Graphics) : void
− isInRightGreenArea(player : Player) : boolean
+ keyPressed(e : KeyEvent) : void
+ keyReleased(e : KeyEvent) : void
+ keyTyped(e : KeyEvent) : void
− runGameLoop() : void
− updateGame() : void
+ main(args : String[]) : void

Player

− speed : double
− spinning : boolean
− movingRight : boolean
− movingLeft : boolean
− movingDown : boolean
− movingUp : boolean

+ Player(initialPosition : Point, initialRotation : double)
+ setRotation(rotation : double) : void
+ setPosition(position : Point) : void
+ move(checkeredX : int, checkeredY : int, checkeredSize : int) : void
+ respawn() : void
~ collidesWithTopBorder(checkeredY : int) : boolean
~ collidesWithBottomBorder(checkeredY : int, checkeredSize : int) : boolean
~ collidesWithLeftBorder(checkeredX : int) : boolean
~ collidesWithRightBorder(checkeredX : int, checkeredSize : int) : boolean
+ setMovingUp(movingUp : boolean) : void
+ setMovingDown(movingDown : boolean) : void
+ setMovingLeft(movingLeft : boolean) : void
+ setMovingRight(movingRight : boolean) : void
+ setSpinning(spinning : boolean) : void
+ paint(brush : Graphics) : void

Game

buffer : Image
height : int
width : int
on : boolean

+ Game(name : String, inWidth : int, inHeight : int)
+ paint(brush : Graphics) : void
+ update(brush : Graphics) : void
− sleep(time : int) : void

What Went Well?
In the development of this game project, several aspects were very successful for me. Using the graphical user interface capabilities of java.awt proved to have very little issues for me. Implementing keyboard input for player control and incorporating collision detection mechanisms allowed me to demonstrate a solid understanding of game development. The design of the various game elements, such as the player, enemies, and spinning rectangles allowed for some creativity and change from the original game. I feel like I really got the freedom to make exactly what I wanted with this project for the most part.

What Didn't Go Well?
There were a few challenges that I encountered during the development process of this project. One of the main issues I ran into involved the collision between the player and the outer bounds of the checkered area. Since I used a rectangle and not a 'border', when checking for collision in this case, I was actually checking the specific coordinates of the square. This took a lot of brute force testing. The other main issue I had was actually making sure the project fulfilled all the requirements. By this I mean I had the game completed to the level I wanted before I had all of the necessary elements (i.e. 1 lambda expression).

If You Had More Time, What Would You Implement Next?
If I were given additional time for this project, there are several enhancements I could implement to elevate the game. One priority I would have would be refining the user interface. For example, I did not have the chance to implement a way for the user to restart after they win

the game. I would add a button to try again in the future. I would also get the chance to implement additional levels beyond the first one. One of the things I might do also would be make the game more closely resemble the original "worlds hardest game", but then again possibly not because the differences give it some personal flair. I could also incorporate sound effects or background music to make the game even more engaging.

Skills Learned/Demonstrated:
This project provided an opportunity for me to strengthen and demonstrate a lot of the programming and game development skills that I have learned in all of my computer science classes and experiences thus far. Using GUIs and awt specifically is something I have moderate experience with, but this project definitely allowed me to fine-tune some of my skills in that area. I also feel that it just generally helped me to grasp inner classes and anonymous classes from a more practical stance. This will hopefully help me in the future (such as on exams or in a career field). The other thing that was great for me was the fact that this project was a game. My top choice for career path is game development and this gave me a very nice intro.

Link to my video:
https://drive.google.com/file/d/1B6RqKEVHKVjya2o9nmRjSl8xmEnUaDJx/view?usp=sharing