PlayerTest

+CLASSNAME: String = "Player"

+copy_constructorTest(): void
+test_anySpins_added(): void
+test_playerscore(): void
+test_number_of_strikes(): void
+Test_miniGame_count(): void

SquareTest

+CLASSNAME: String = "Square"

+ValueTest(): void
+SpinTest(): void
+setSpinsTest(): void
+setValueTest(): void

BoardText

+playBoard(int Spins, Player player): void

Square

#spins: int
#value: int
#title: String
#image: String

+setTitle(title:String): void
+setValue(value:int): void
+setSpins(spins:int): void
+getTitle(): String
+getSpins(): int
+getValue(): int

Player

-name: String
-numOfSpins: int
-score: int
-strikes: int
-awards: ArrayList<Square>

+getName(): String
+getSpins(): int
+getScore(): int
+getStrikes(): int
+setName(): void
+setSpins(newSpins:int): void
+addSpins(addedSpins:int): void
+setScore(newScore:int): void
+setStrikes(newStrikes:int): void
+addAward(newAward:Square): void
+getAwards(): ArrayList<Square>

BrickBreaker

+start(Stage primaryStage): void
+handle(ActionEvent t): void
+getScore(): int

ManageSpinsForPlinko

-spins: int

+getSpinsFromPlinko(): int
+setSpinsFromPlinko(int nSpins): void

MainText

-counter: int = 0

+main(args[]: String): void
+initiateGame(): void
+whichGame(): void

BoardTest

+CLASSNAME: String = "Board"

+PopulateTest(): void
+RandomTest(): void

MinGameText

-spinEarnedArray: ArrayList<Integer>
-randIndexGenerator: Random
-player: Player

+getSpinsFromMiniGame(): int
+setUserName(String userName): void
+setUpMiniGame(): void

Board

-gameboard: ArrayList<Square>

+addSquare(square:Square): void
+populateBoard(): void
+getList(): ArrayList<Square>
+RandomizeList(al: ArrayList<Square>): ArrayList<Square>

MainMethod

-player1: Player

+MainMethod extends Application
+start(Stage primaryStage): void
+main(args[]:String): void

window2

#GRID: int
-highlighter: Rectangle

+window2 extends Application
+start(Stage primaryStage): void
+prizeBox(int x, int y, int width, int height): Rectangle
+currentSelection(int choice): Rectangle
+random(): int

Plinko

-spinEarnedArray: ArrayList<Integer>
-randIndexGenerator : Random

+getSpinsFromPlinko(): int
+addNum(int num): void
+getValueFromSpinsEarnedArray(int i): int
+setUserName(): void
+setUpPlinko(): void
+main(args[]:String): void

FileIO

-fileName: String = "gameData.txt"

+write(name:String, score:int, spins:int, minigames:int): void
+readScore(): String
+readName(): String
+readSpins(): String
+readMiniGameCount(): String
+main(args[]: String): void

SnakeMain

+Direction: enum
+createContent(Stage primaryStage): Parent
+start(Stage primaryStage): void
-startGame(): void
+pause(Stage pauseStage, Stage primaryStage): void
+main(args[]: String): void

PlinkoGui

-spinEarnedArray: ArrayList<String>
-player: Player
-spins: ManageSpinsForPlinko

+start(Stage primaryStage): void
+checkForCollisions(Circle obj, Circle obj2): Boolean
+getXofCollision(Circle obj, Circle obj2): double
+giveSpinsToPlayer(Circle droppedBall, Bounds bounds, Player player): void
+dropBall(Circle droppedBall, Bounds bounds, ArrayList<Circle> pegs, Player player): void
+setUpPlinko(ArrayList<String> alstr): void
+getValueFromSpinsEarnedArray(int i): String
+getSpinsFromPlayer(): int
+getSpinsFromPlinko(): int