Jewel

-point : int
-type : String
-matchFound : boolean

+Jewel(int, String)
+match(int, int) : void
+leftDiagonalMatch(int, int) : boolean
+rightDiagonalMatch(int, int) : boolean
+horizontalMatch(int, int) : boolean
+verticalMatch(int, int) : boolean
+getPoint() : int
+getType() : String

Diamond

+Diamond()
+match() : void

Square

+Square()
+match() : void

Triangle

+Triangle()
+match() : void

Wildcard

+Wildcard()
+match() : void

MathematicalSymbol

-symbol : String

+MathematicalSymbol(String)
+match() : void
+getSymbol() : String

FileIO

+readCommand(String) : ArrayList<String[]>
+readGameGrid(String) : ArrayList<ArrayList<String>>
+readLeaderboard(String) : ArrayList<String[]>
+writeOutput(ArrayList<String[]>, ArrayList<ArrayList<String>>) : void

<<interface>>
Comparable<Player>

+compareTo(Player) : int

Leaderboard

-players : ArrayList<Player>

+createPlayers(ArrayList<String[]>) : ArrayList<Player>
+sortPlayers(ArrayList<Player>) : ArrayList<Player>

Player

-name : String
-score : int

+Player(String, int)
+getName() : String
+getScore() : int
+compareTo(Player) : int

Game

-matrix : ArrayList<ArrayList<Jewel>>
-name : String
-totalScore : int
-collectedScore : int

+createMatrix(ArrayList<ArrayList<String>>) : ArrayList<ArrayList<Jewel>>
+selectCoordinate(int, int, ArrayList<ArrayList<Jewel>>) : void
+slideDown() : void
+playGame(FileWriter, ArrayList<String[]>, ArrayList<ArrayList<String>>) : void
+printMatrix(FileWriter) : void
+getMatrix() : ArrayList<ArrayList<Jewel>>
+getTotalScore() : int
+addScore(int) : void
+setCollectedScore(int) : void
+getName() : String
+setName(String) : void

This project consists of 11 classes including the Main class. First of all, the inputs are given in program arguments. The inputs are taken as parameters in the FileIO class methods to read them and make them arraylists. Then the playGame method is run and commands are iterated over to do the necessary actions. When a coordinate is selected and is valid, the program sees which jewel is chosen and searches if it matches with other jewels according to some rules. For this reason, the Jewel class is abstract and the match method is abstract. Matching methods are defined for each direction in the Jewel class. For every Jewel subclass, including mathematical symbols, these directional matching methods are used to minimize duplicate codes. When the match is found, the jewels pop and remaining jewels fall. The falling method (which is slideDown) pseudocode is given in the code to better understand how it works. Then the updated matrix is printed in file. If the selected coordinate is not valid, the next coordinate is taken. When E command is given, game ends and user's total score is printed in file. After that the name is asked and a new player object is created for the user. Leaderboard input is read in FileIO and the player objects are also created in there. User is added to leaderboard file too. They are printed in the output of leaderboard. Then, all players are compared according to their scores and the user's rank is given at the end of the file. compareTo method is overriden from Comparable interface. Player class implements the Comparable interface so comparison is possible. Finally, the game ends here.