Mastermind

Team 2

Lyndsay Barnes (ljbarnes), Casey Beise (cmbeise) and Will Greene (wgreene)

Introduction- The Original Wordle

This game titled, “The Original Wordle” is designed to mimic the game Mastermind implemented by the programming language Java. The goal of the game is to guess a randomized, hidden sequence of four colors in the correct order while gaining the least amount of points. This is a two player game in which player’s take turns trying to guess a randomized sequence of colors in the least amount of guesses possible. The maximum number of guesses per sequence is 8 guesses. A player gains one point for each guess and receives two extra points if the sequence is not guessed within the maximum number of guesses allowed. Feedback regarding the guess success is provided back (in the form of white and black key pegs) to the player after each guess. White key pegs will represent a color that has been guessed correctly but placed in the wrong position of the sequence. Black key pegs will be displayed for each color guessed correctly and in the correct position of the sequence. When a player receives four black key pegs as feedback the player has guessed the hidden sequence of four colors correctly. The game is won by the player who has the least amount of points at the end of the last round. Please note that the number of rounds is determined by the players at the start of the game.

Software Requirements

Mastermind will be executed as follows (error handling is at the bottom of the Software Requirements) :

```
java TheOriginalWordle randomSeed
```

```
    Welcome to Mastermind!
A game of logic, with a bit of chance.
```

Players will enter their names (p1Name & p2Name), and number of rounds (numRounds).

```
This game is for 2 players.
Player 1 name? p1Name
Player 2 name? p2Name

Number of rounds? numRounds

Great! 4-color codes have been initialized for both players. Let’s begin.
```

The round, guess number, total points and player name will be shown for each player. Players will enter their guesses (p1Guess & p2Guess). Below is an example for Round 1.

```
[Round 1, Guess 1] Total Points for p1Name : 0
p1Name, please enter guess (R-ed, G-reen, B-lue, Y-ellow) : p1Guess
[Round 1, Guess 1] Total Points for p2Name : 0
p2Name, please enter guess (R-ed, G-reen, B-lue, Y-ellow) : p2Guess
```

Guesses will be displayed as below. There will be 2 grids - one for each player.

p1Name Guess Log

R	G	B	Y	w	b		

p2Name Guess Log

R	G	B	Y	w	b		

Players will receive 1 point each round (unless they guess the code). Below is an example for Round 2, assuming neither player guessed their code on the first try.

```
[Round 1, Guess 2] Total Points for p1Name : 1
p1Name, please enter guess (R-ed, G-reen, B-lue, Y-ellow) : p1Guess
[Round 1, Guess 2] Total Points for p2Name : 1
p2Name, please enter guess (R-ed, G-reen, B-lue, Y-ellow) : p2Guess
```

If a player guesses correctly, after visual has been displayed, output :

```
Congratulations, pXName! You have guessed the code!
pXName will continue guessing.
```

Game will continue while omitting the other player's guesses. If both players guess correctly before round 8, skip to round summary and, if applicable, begin a new round.

If one or more players finishes guess 8 without guessing correctly, output for each player :

```
pXName, you did not guess the code. 2 points will be added to your Total Points.
```

At the end of each round, output :

```
Totals
p1Name : totalPoints points
p2Name : totalPoints points
```

If applicable, start a new round. Output :

```
New codes have been initialized for both players.
```

Let’s begin [Round **numRounds++**]

[Round 2, Guess 1] p1Name … etc.

At the end of the game, after the round summary, output :

Congratulations, pXName! You are the winner!

In case of a tie - instead of the above, output :

Oh no! It’s a tie! Try playing again to settle the tiebreak.

Finally, output :

Play again (Y-es, Q-uit)?

User can select to play again or quit the program.

Error Handling

For player name input, any name longer than 50 characters, output :
“Name too long - max 50 characters”
“Player X name?”

For guess input, anything other than R, r, B, b, Y, y, G, g, output :
“Invalid input : use characters R, B, Y, G only.”
“pXName, please enter guess (R-ed, G-reen, B-lue, Y-ellow) :”

For number of round input, anything not between 1 and 10, inclusive, output
“Invalid number of rounds. Please enter an integer between 1 and 10, inclusive.”
“Number of rounds?”

For “play again” input, anything other than Y or Q, output :
“Invalid input.”
“Play again (Y-es, Q-uit)?”