UML method descriptions

Aero Griffin, Trevor Averill

03/13/20

Screen

**getPlayerChoice** – provides player menu and asks user for a choice choices are to spin, guess(or solve) the puzzle, or buy a vowel returns an int that can be checked against constants in the Screen class

**gameScreen**- uses player and phrase information provided when called, displays the game screen with the phrase, players the current player and a menu for the player to make a choice of what to do

**inputConsanents**- uses phrase and available consanent information provided to display the phrase and ask the user to input a consanent to guess keeps asking for input until input is valid

**inputVowels**- uses phrase and available vowel information provided to display the phrase and ask the user to input a vowel to guess keeps asking for input until input is valid

**solveScreen**-uses phrase information provided to output the phrase info and ask user to guess what the phrase is by providing a string

**winScreen**- uses player information provided to display who won the game and how much money they won

**roundEndScreen**- uses player and winner information to display the results of the round that just ended including the totals each player has at the moment

Wheel

**Wheel**- Creates a new wheel with a provided round to set up the correct wheel values

**updateWheelRound**- updates the base values on the wheel to the values for the correct provided round

**spin**- animates the process of randomly selecting a wheel value and returns the result as an integer

**wheelFormat** – used to help draw the wheel with cleaner looking code format data centered and converts –1, and –2 into lose turn and bankrupt.

Phrase

**Phrase**- Creates a new Phrase object using provided phrase, category, and difficulty

**guess**- takes in a string representing a player’s guess and returns true if the guess was correct otherwise returns false

**letterGuess**- takes in a character representing a new character guess adds that character to the String for guessedLetters updates the display string and returns the number of that character that appears in the phrase

**toString**- formats and returns a String with the display phrase, difficulty, and category

**getGuessedLetters**- returns the String that stores all the previously guessed letters

Main

**main** this method sets up three rounds of wheel of fortune and a user provide number of players with a given difficulty

Round

**Round** creates a round to be played with a given array of players and number of players in the array and a phrase

**playRound** plays the round until a player wins or quits the game

**playTurn** plays out a single turn

**spin** this method is run when a user chooses to spin the wheel can bankrupt a player make them loose a turn or give them an amount of points based on the spin and whether a player has guessed correctly if player did not guess correctly allows next player to play

**guess** this method allows a player to attempt to solve a puzzle when the player guesses wrong passes turn on to the next player otherwise ends the round with a win for that player

**buyVowel** this method allows a player to buy a vowel when they have enough money when the player guesses a correct vowel it remains their turn otherwise the turn goes to the next player

**removeChar** this method removes the first instance of a given character from a given string

**nextPlayer** this method set current player to the next player in line

Player

**Player** creates a new player with a given name

**successfulGuess** this method adds won points to the players round score based on the number of the letter that was present and the amount that was spun

**spendPoints** this method subtracts a given amount from the players round score

**roundWin** this method adds the round score to the total score and resets the round score back to zero

**roundLoss** this method sets the round score to zero

**getTotalScore** this method returns the totalScore of a player

**getRoundSCore** this method returns the roundScore of a player

**getName** this method returns the name of a player

**toString** this method formats the players data into a neatly printable string and returns it

Game

**Game** this method creates a new game object this object has 3 rounds fully set up and ready to play

**play** this method plays the previously set up rounds of wheel of fortune

**readFile** this file sets up arrays of easy medium and hard puzzles to be randomly picked from when making rounds, reads each line from a file formated puzzles,category,difficulty and stores each one appropriately