Unit 4 Project: OOP Game Show App
Grading Rubric

	< B	B	A
PHRASE CLASS	• No constructor or constructor doesn't properly initialize a phrase property • Is missing any of the following methods: addPhraseToDisplay(), checkLetter(), showMatchedLetter()	• Includes constructor that receives a phrase parameter and initializes a phrase property set to the phrase • Includes addPhraseToDisplay() method which adds the phrase to the gameboard • Includes checkLetter() method which checks if a letter is in the phrase • Includes showMatchedLetter() method which reveals the letter(s) on the board that matches the player's selection	• N/A
GAME CLASS CONSTRUCTOR	• No constructor or constructor doesn't properly initialize the missed, phrases, or activePhrase properties • Phrases added to the game include numbers, punctuation or special characters	• Includes a constructor that initializes a missed property set to 0, a phrases property set to an array of five Phrase objects, and an activePhrase property set to null initially • Phrases added to the game only include letters and spaces	• N/A

GAME CLASS METHODS	• Is missing any of the following methods: startGame(), getRandomPhrase(), handleInteraction(), checkForWin(), removeLife(),gameOver()`	• Includes startGame() method that hides the start screen overlay, sets the activePhrase property to a random phrase, and calls the addPhraseToDisplay() method on the active phrase
		• Includes getRandomPhrase() method that randomly retrieves one phrase from the phrases array
		• Includes handleInteraction() method that:
		  ○ Disables the selected letter's onscreen keyboard button
		  ○ If the phrase does not include the guessed letter, the wrong CSS class is added to the selected letter's keyboard button and the removeLife() method is called
		  ○ If the phrase includes the guessed letter, the chosen CSS class is added to the selected letter's keyboard button, the showMatchedLetter() method is called on the phrase, and the checkForWin() method is called. If the player has won the game, the gameOver() method is called
			• N/A

		• Includes checkForWin() method that checks if the player has revealed all of the letters in the active phrase • Includes a removeLife() method that removes a life from the scoreboard (one of the liveHeart.png images is replaced with a lostHeart.png image), increments the missed property, and if the player has lost the game calls the gameOver() method • Includes gameOver() method that displays a final "win" or "loss" message by showing the original start screen overlay styled with either the win or lose CSS class	
APP.JS	• Clicking the "Start Game" button doesn't create a new Game object or doesn't start a new game • Clicking an onscreen keyboard button doesn't result in a call to the handleInteraction() method or the method is called for the incorrect keyboard button • Clicking the spaces between and around the onscreen keyboard buttons results in the handleInteraction() method being called	• Clicking the "Start Game" button creates a new Game object and starts the game • Clicking an onscreen keyboard button results in a call to the handleInteraction() method for the clicked keyboard button • Clicking the spaces between and around the onscreen keyboard buttons does not result in the handleInteraction() method being called	• Event listener has been added for the keydown or keyup event so that pressing a physical keyboard button results in the handleInteraction() method being called for the associated onscreen keyboard button
RESETTING THE GAMEBOARD	• After a game is completed, the gameboard isn't reset so that clicking the "Start Game" button fails to load a new game	• After a game is completed, the gameboard is reset so that clicking the "Start Game" button loads a new game	• N/A
HTML AND CSS	• Provided HTML or CSS is not used	• Provided HTML and CSS is used	• App styles have been personalized and changes have been noted in the

Nice job with the hangman animation!

			README.md file	
DOCUMEN TATION	• No code comments. • Repository does not include a README.md file.	• Code comments have been added explaining how the functions work. • Repository includes a README.md file with name of the project, a description of the project, and your name.	• Code comments use JSDocs throughout that include a description and any relevant tags (i.e. parameter and return). • The README.md file for your repo explains anything your user or fellow developers might need to know to use the project.	JSDocs was used in some places. It should be used throughout to be useful for auto generation of documentation.