Grading Rubric –
Tutorial 14, Review

Description	Pts	Your Score
1. Use your editor to open the ag_squares_txt.html, ag_squares_txt.js, and ag_cards2_txt.js files from the html14 > review folder. Enter your name and the date in the comment section of each file, and save them as ag_squares.html, ag_squares.js, and ag_cards2.js respectively.	**4**	
2. Go to the ag_squares.html file in your editor. Link the page to the ag_card2.js and ag_squares.js files in that order, loading the files asynchronously. Take some time to study the HTML code in the file and then close it, saving your changes.	**4**	
3. Go to the ag_cards2.js file in your editor. Bob has already created an object literal for the squareGame object containing the cardGrid array for storing five poker hands and two methods: the calcRowPoints() method for calculating the point total for a poker hand in a row indicated by an index argument, and the calcColumnPoints() method for calculating the point total for a poker hand in a column indicated by an index. Complete the squareGame object by adding the follow- ing properties and methods (make sure you separate the properties and methods with a comma.) a. The gametotal property of the squareGame variable, which stores the total points achieved in the Poker Squares game. Set its value to 0. b. The wintotal property of the squareGame variable, which stores the point total required for winning the game. Set its value to 50. c. The gameResult() method, which returns the text string "Winner" if gameTotal is greater than or equal to winTotal; otherwise returns the text string "No Winner".	**10**	
4. Below the squareGame object, insert code for the insertcard() method of the pokerHand object prototype. The purpose of this method is to insert a card into a poker hand at a specified index. The method has two arguments: the card argument referencing a pokerCard object and the index argument specifying the location where the card should be placed. Insert a command into the function to make this.cards[index] equal to the value of the card argument.	**4**	
5. Document your work with comments, and save your changes to the file	**4**	
6. Open the ag_squares.js file in your editor.	**3**	
7. Go to the playPokerSquares() function. Within the function add an onclick event handler to the startButton object that runs an anonymous function when the user clicks the Start Button on the web page. Within the anonymous function, do the tasks laid out in Steps 8 through 13.	**4**	

8. Set up the initial game board by doing the following: a. Set the gameTotal property of the squareGame object to 0. b. Remove the current game score by changing the value of the gameScore input box on the page to an empty text string. c. Remove the current game result by changing the text content of the gameResult element on the page to an empty text string. d. Remove the current row and column totals by looping through the contents of the rowSumCells and columnSumCells object collections, setting the text content of each cell to an empty text string. e. Remove the current card images by looping through the cardlmages object collection, setting the source of every inline image to the "ag_trans.gif" file.	**14**	
9. Create a new pockerDeck object named myDeck and use the shuffle() method to randomize the order of its cards.	**4**	
10. Create a new pokerCard object named mystartercard. Apply the shift() method to the cards array of myDeck to store the first card from the deck in myStarterCard. Change the src attribute of the newCard inline image by calling the cardlmage() method for the myStarterCard object.	**7**	
11. The starter card is added to the board by clicking a cell in the grid table where the user wants the card placed. For every image in the cardlmages collection, create an onclick event handler that does the following: a. Applies the cardlmage() method to the myStarterCard object to display the image of the current card in the event object target. b. Stores the row number and column number of the clicked image in the rowNum and colNum variable. (Hint: Use the charAt() method to retrieve the second and third characters of the id attribute of the event object target.) c. Applies the insertCard() method to the squareGame.cardGrid[rowNum] object to insert a card into the new grid. Use myStarterCard as the poker card and colNum as the location in the insertCard() method. d. After the card has been placed within the grid, it cannot be changed. Set the onclick event handler of the event target to null to prevent the user from re-clicking the cell later in the game.	**12**	
12. Finally, test whether the user has completed the grid table. Within the onclick event handler of the previous step, test whether there are more than 27 cards left in the deck. If there are more than 27 cards left, the game continues. Shift the next card from myDeck into the myStarterCard object and change the src attribute of newCard to display the image of the next starter card.	**9**	
13. Otherwise, if the grid table is completed and the game is over, calculate the game score and totals for the poker hands in each row and column as follows: a. Indicate that the game is over by changing the src attribute of the newCard image to the "ag_cardback3.png" file. b. Calculate the row poker hand totals by creating a for loop with a counter variable that goes from 0 to 4. Declare the rowTotal variable equal to the value returned by the calcRowPoints() method of squareGame object, using	**14**	

your counter variable as the parameter value. Add rowTotal to the value of the gameTotal property of the squareGame object. Display the rowTotal value in the element with the ID rowindexsum where index is the value of your counter variable. c. Calculate the column totals with another for loop as you did in the previous step for the row totals. Use the calcColumnPoints() method to calculate the totals for each column poker hand, adding the column total to the gameTotal property and displaying column total value in the element with the ID colindexsum. d. Change the value of the gameScore input box to the value of the gameTotal property of the squareGame object. e. Show whether the user won or lost by changing the text content of the gameResult element to the text returned by the squareGame object's gameResult() method.		
14. Document your work with comments and save the file.	**4**	
15. Open the ag_squares.html file in your browser. Click the Start button to begin a game. Verify that you can add cards to the grid by clicking cells within the table. Play a complete game by filling out the grid and verify that when all cells are filled, the page displays: a) row and column totals for every hand, b) the overall point total for the game, c) a message indicating whether the player won or lost, and d) a card image showing that the game is over.	**3**	
TOTAL	**100**	

YOUR SCORE: _____