JavaScript - Level 1

Final Project – JavaScript Game

Course Value: 80%

Due: End of day, November 16, 2018

Instructions:

1.) Create a single page game using JavaScript
2.) The game can be one of the following games (listed in order of difficulty). You can also create another game not listed below (see your instructor for approval):
    a. Number guessing game
    b. Dice game
    c. Tic Tac Toe
    d. Hangman
    e. Whack-a-Mole
    f. Matching Game
    g. Blackjack
    h. Asteroid Hunter
3.) Each of the above listed games parameters and requirements are listed on the following pages
4.) Both the simpler guessing game and the most advanced Asteroid Hunter game will be marked the same. So, if you are struggling with JavaScript select one of the simpler games. If on the other hand you want to challenge yourself then select one of the more difficult games.
5.) To aid in your coding I have provided basic shell code with almost no styling for all the game types listed above.
6.) Try to code it on your own at first, but if you struggle use the code samples, but try to incorporate your own variable names and modify the code to fit your coding style.
7.) The games should be styled using CSS and images. Be creative, a good portion of the mark is on the design and usability of the game.
8.) See the following pages for the game parameters
9.) Upload your JavaScript game to your portfolio server and place a link on your landing page

Notes:

- The game does not have to be responsive. The game will be marked on a large desktop monitor

Marking Criteria:

- All instructions followed	5 marks
- JavaScript, HTML and CSS properly implemented	25 marks
- Quality, Creativity and usability of the design of the game	50 marks
- **Total**	**80 marks**

Number Guessing Game

Game Parameters:

- Create a number guessing game where the user has to guess a number between 1 and 100 that is randomly generated by the program.
- The game should consist of an input field where the user can enters his/her guess.
- The program will keep track of the number of guesses it takes for the user to guess the random number.
- The program will validate the data entered by the user, making sure that the data is a number and that it is between 1 and 100. If the user entered invalid data display an error message to him/her.
- The program will give feedback to the user about whether or not their guess was too high or too low.
- If the user guessed correctly display one of three messages depending on the number of guesses it took the user to guess the correct number.
    - If the user guessed correctly on the first guess display this message:
        - Wow mind reader, you guessed correctly on your first guess
    - If the user took between 2 and 5 guesses display this message:
        - Not bad it only took you (x) guesses to guess the correct number
    - If the user took more than 5 guesses than display this message:
        - You guessed correctly. It took you (x) guesses to guess the correct number
    - Note:
        - Replace (x) with the actual number of guesses it took to guess the correct number
- When the user guesses the correct number prevent the user from guessing again. If the user tries to guess again, ask them to refresh the page to play again.
    - Alternatively you can ask the user if they want to play again and then reset the game

Dice Game

Game Parameters

- Create a dice game where a user plays against the computer. The user and the computer each roll a pair of dice 3 times. After the third roll of the dice the player with highest score wins
- The scoring for the game works as follows:
    - If any of the players two dice comes up as a 1 then the score for that round for the player is 0
        - Example:
            - Player rolls a 6 and a 1
                - Player score = 0
    - If the player rolls a pair of the same numbers then the players score is the total of the two dice times 2
        - Example:
            - Player rolls a pair of 5's
                - Player score is (5 + 5) * 2 = 20
    - If the player rolls any other combination of dice other than the ones mentioned above then the players score is the total value of the two dice
        - Example:
            - Player rolls a 3 and a 2
                - Player score is 3 + 2 = 5
- The game should provide a text or graphical output showing the following:
    - The current rolled dice by the player and the computer
    - The score for this round for the player and the computer
    - The total score for the game
- The game should provide a button that will roll the dice for the player and the computer
- After three rolls of the dice the game should total up the scores and display a message displaying who the winner was
- The game should provide a button that will reset the game and start a new game

Tic Tac Toe

Game Parameters

- Create a 2 player game of Tic Tac Toe
- Each player gets to enter an “X” or an “O” in one of nine squares. The first player to get three “X” or three “O” in a row wins. If neither player gets three in a row and the game board is full then the game is a draw
- The game should provide a 9 square graphical grid playing board.
- The game should start with the “X” player
- When a player clicks on an empty square then that square should fill in with either an “X” or an “O” depending on the player.
- If the player clicks on an already filled in square then either nothing happens or you can display a message to the player saying this square is unavailable or something similar
- The game should check each turn to determine if there is a winner. If there is a winner then the game should announce who the winner is.
- If the game board is full without a winner then the game should announce that it is a draw.
- At the end of the game the game should ask if the users if they want to play again. If the users want to play again then the game should reset the game board and start a new game with the “X” player starting first.

Bonus

- Develop a human vs computer (AI) version of this game. A simple computer AI might just simply select a random square. A more advanced computer AI might try and select the best square
- See your instructor for code samples if you get stuck with this bonus version

Hangman

Game Parameters:

- Create a version of the hangman game
- The hangman game should randomly select a word from an array of words
- The user must guess the correct word by entering letters into an input box
- If the user guesses a letter that is contained in the selected word than the game should display the correctly guessed letters in the position of the word where they are located
- If the user makes an incorrect guess, then the program should display part of the hangman graphic
- If the user makes too many incorrect guesses and the entire hangman graphic is displayed, then the user loses the game
- If the user correctly guesses all the letters in the selected word than the user wins the game
- When the game is over either from making too many incorrect guesses or correctly guessing the word, then the game should give the user the option to play again

Whack-a-Mole

Game Parameters

- Create a single player game where the user must click on moles (it does not have to be moles) that randomly appear, then disappear after a random amount of time in boxes on the screen.
- Every time the user clicks on a mole they score a point.
- The game should run for a set time, then stop.
- The game should keep track of the users score and output the score to the screen.
- The game should have start and stop buttons (these buttons can be combined into a single button) that will start and stop the game when clicked.

Matching Game

Game Parameters

- Create a single player game where the user must match pairs of hidden pictures on a grid of at least 12 squares.
- The player can reveal the pictures hidden behind each square by clicking on the square.
- The player can reveal up to two pictures in a turn.
- If the player reveals two pictures that are the same in a turn, then the pictures remain revealed.
- If the player reveals two pictures that are not the same, then the pictures return to being hidden after a certain amount of time.
- If a user clicks on an already revealed picture than nothing should happen
- The game is completed when the user has matched all the pairs of pictures.
- The game should keep track and display to the user the number of turns he/she has taken.
- The game should keep track and display to the user the number of matches he/she has.
- The game should have a button that starts the game
- The user should have the ability to end the game at any time

Blackjack

Game Parameters

- Create a simplified version of the game of Blackjack where a single user plays a single round of Blackjack against the computer.
- For this version of Blackjack, the object of the game is to get as close to the value of 21 without going over 21.
- The computer acts as the dealer and the player plays against the dealer
- Whoever is closer to 21 wins the round
- For this version of Blackjack, the player will initially be dealt two cards.
- After the player is dealt his/her initial two cards his hand will be totaled. The player will then be given the option to either hold or ask for another card
- If the player holds then the dealer plays its cards. The dealer plays under special rules described later.
- If the player asks for another card, then the player is dealt another card and his hand is totaled.
- The player can continue to ask for cards until the player decides to hold.
- The player automatically loses if at any time the player goes over 21.
- The hands points are totalled in the following manner:
    - Aces are worth either 1 or 11 (if the hand goes over 21 then an ace can be worth 1 to allow the hand to go below 21)
    - Kings, Queens and Jacks are worth 10
    - The number cards 2 thru 10 are worth the value of their number
- Once the player decides to hold and the player did not go over 21 then the dealer plays its hand.
- The dealer plays under special rules
- The dealer will continue get more cards as long as its hands total points remain less than 17
- If the dealer's points are 17 or more then the dealer will hold
- The dealer will automatically lose if it goes over 21
- Once the dealer finishes playing then his hand is totaled
- The player with the most points wins
- The game should provide a button to play a new game

Asteroid Hunter

Game Parameters

- Create a side scrolling arcade shooter game
- For this version a user navigates a ship (does not have to be a ship) through an asteroid field (does not have to be asteroids)
- The user gets a certain number of lives. If the user's ship hits an asteroid their ship is destroyed, and the users number of lives decreases by one.
- The user's ship can shoot lasers that can damage asteroids. If an asteroid is hit a certain number of times, then the asteroid is destroyed.
- The user scores points whenever the user destroys an asteroid
- The game should display the number of lives the user has left
- The game should display the number of points the user has achieved
- When the users ship is destroyed, and their number of lives decreases to zero then the game is over
- When the game ends, display the users score and give them the option to play again

Bonus

- Have the game get progressively harder by increasing the number of asteroids on screen over time
- Create a high score board where users can enter their initials into the high score board