Web Scripting 1

Final Project: JavaScript Game

Course Value: 60%

Due: Game Progress Check-in: Feb 22, 2024 | Final Project: April 1, 2024 – 11:59 pm

Project Description:
Create an interactive game using HTML/CSS and JavaScript

Instructions

1. Select one of the following games (listed in order of difficulty). You can also create another game not listed below, but you must receive instructor approval

2. Suggested Games List (listed in order of difficulty, from easiest to the most difficult)
    - Dice Game
    - Tic Tac Toe (2 human players or 1 human player vs. the computer)
    - Wheel of Fortune
    - Whack-a-Mole
    - Matching Game
    - Blackjack

3. All the above-listed game parameters and requirements are listed on the following pages
4. The easiest number guessing game and the most difficult game will be marked the same. If you struggle with JavaScript, select one of the simpler games. If, on the other hand, you want more of a challenge, then select one of the more difficult games
5. The game should be nicely styled with CSS and have a well-laid-out and usable user interface (a a good portion of the mark is on usability). Submitting a project without any styling will result in a greatly reduced final mark
6. See the following pages for the game parameters
7. The game does NOT have to be responsive. It only needs work at a single screen size. I will be checking the games at a 1920px by 1080px screen resolution. If you would like me to view your game at a different resolution, just let me know in your submission document
8. You are permitted to use jQuery
9. You are permitted to use HTML / CSS libraries such as Bootstrap for the user interface

Submission

There are 2 submission processes for this assignment

1. **Game Progress Check-In**
    - **a.** Due: Feb 22, 2024
    - **b.** Arrange to meet with your instructor or the Lab Assistant on day 9, 10 or 11 of this course to discuss the progress of your game up to this point
        - i. It is your responsibility to arrange a meeting with your instructor or lab assistant
            1. Failure to do so will lead to a zero grade for the check-in portion of your mark for this project
        - ii. For this submission, you should have the following ready at a minimum
            1. A game selected
            2. Rough design for the game
                a. This can be a wireframe (digital or paper sketched)
                b. Screenshots of games you found on the internet that you wish to use as inspiration for your game
                c. Non-styled or styled HTML output
            3. A plan for completing the project by the due date
            4. Questions or concerns about the project
                a. (this is optional)
2. **Final Submission**
    a. Due: April 1, 2024
    b. Zip up your project folder
        i. Make sure to include the HTML, CSS, and JavaScript files
        ii. Include all images and other media required to make your project run
    c. Submit your project to the Project 01 drop box for the Web Scripting 1 course on The Learning Hub

Marking Criteria

- **Game progress Check-In: 10 marks**
    - Met the minimum requirements as described in the Game progress Check-in section of the Project Submission section
- **All Technical Requirements Met: 10 marks**
    - Followed all the instructions as set out in the instructions section of this document
    - Does your game fulfill all the game requirements as listed in the game descriptions at the end of this document?
        - If you decide to build your own game that is not one of the predefined games, then you and your instructor will agree to a set of requirements
- **HTML / CSS / JS Code Quality 20 marks**
    - Is the code error-free and valid?
    - JS code properly structured
        - Well named variables
        - Functions utilized for re-used code
        - THe code is nicely organized and well commented
- **Design and Usability if the Website 20 marks**
    - Does the game follow usability best-practices
        - Are the controls intuitive (or are instructions provided?)
        - Are the game elements nicely spaced out
        - Is the game text readable?

Marking Summary

- Technical requirements met: 10 marks
- HTML / CSS / JS code quality: 30 marks
- Design and usability of the website: 20 marks

Other information

If you have any questions or require assistance, send me a DM on Slack, or send me an email at michael_laroy@bcit.ca

Game Parameters

*** Have another game in mind, not listed below, just see your instructor for approval

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 the highest score wins
- The scoring for the fame works as follows:
    - If any of the player's two dice comes up a 1, then the score for that round for the player is 0
        - Example:
            - The player rolls a 6 and a 1
                - Player score = 0
    - If the player rolls a pair of the same numbers, then the player's score is the total of the two dice times 2
        - Example:
            - The player rolls a pair of 5's
                - Player score is (5+5) * 2 = 20
    - If the player rolls any other combination of dice than the ones mentioned above, then the player's score is the total value of the two dice.
        - Example:
            - The 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 fame should provide a 9-square graphical grid playing board
- The fame should start with the “X” player
- When a player clicks on an empty square, that square should fill in with either an “X” or an “O” depending on the players
- 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 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 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

Wheel of Fortune

Game Parameters

- Create a version of the Wheel of Fortune game
- The Wheel of Fortune game should randomly select a word or phrase from an array of words or phrases
- The player 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, then the game should display the correctly guessed letters in the position of the word where they are located
- If the player makes an incorrect guess, then the program should lose one life
- If the player loses all their lives, then the player loses the game
    - The number of lives or incorrect guesses a player can make before they lose is up to you
- The player can win the game by either of the following methods
    - Correctly guessing the word or phrase before all the letters have been shown
    - All the letters in the word or phrase have been displayed
- 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 and 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 user's 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 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 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, 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 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 game's object is to get a 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
- The player will initially be dealt two cards for this version of Blackjack
- After the player is dealt his/her initial two cards, his/her hand will be totaled. The player will then be given the option to either hold or to 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, the 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 hand 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 through 10 are worth the value of their number
- Once the player decides to hold and does not go over 21, the dealer plays its hand
- The dealer plays under special rules
- The dealer will continue to get more care as long as its hand's 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, his hand is totaled
- The player with the most points wins
- The game should provide a button to play a new game