

Web Programming II (420-H20-HR)

Assignment 3 – Animation & jQuery

| | |
|----------------|----------------------------------|
| Date assigned: | April 25 th , 2025 |
| Breakdown Due: | May 6th, 2025 |
| Due: | May 16th, 2025 |

Overall Objectives

These assignments are part of a large, final assignment which will demonstrate the learning objectives for the overall course. There are 3 assignments that will make up the assignments for the course.

The overall assignment is to build a one-person dice betting game. The user (player) gets to start with a sum of money and play the games against the computer. Each game has different limits and different pay outs.

The three assignments will be as follows:

- 1) Assignment 1: Set up the game rules and tests to fully test the game logic. You must use classes and functions with a singular purpose. There is no user input or output in this assignment, everything can be tested using automated test cases. This is an exercise in design and testing; let's make sure the base is correct before moving forward.
- 2) Assignment 2: Create and validate the form for the user information and use this information as part of the game play. Accept input from the user. Add the dice graphics to the screen – does not have to be interactive yet.
- 3) Assignment 3 (**this assignment**): Add graphics and animation to the game and use the stored player's winnings and basic information so that they can come back to the game if they stop. Create an animated splash page for the game startup page.

Basic Rules for Assignments

- 1) You **MUST** keep all the formatting for your HTML code in an external CSS file. Uses classes and ids whenever possible and employ good naming standards. This will come in very handy when completing the final assignment.
- 2) We will use a "mobile first" design philosophy. The games must be able to play on mobile phones, tablets, and desktops. They will likely behave slightly differently on each of these platforms. You are welcome to use bootstrap for the interface development.
- 3) Your JavaScript must be put into functions wherever possible, and all functions must be kept in external files and linked to the html file that is using it. Only use the JavaScript that has been taught in class. There is a lot of very bad code on the Internet...be careful. Please use the coding checklist as there will be marks for following the standard.

Learning Objectives

Upon successful completion of this assignment, the student will be able to:

- ❑ Create some page animation in CSS and JavaScript.
- ❑ Use jQuery validate on your form

Overview

Read and make sure you understand the rules for your game. If not ASK! You need to understand how your game works before you can code it. Read and make sure you understand the validation requirements for your game. If not ASK!

You will have some lab time to work on the assignment.

To do:

1. In this assignment you must work with the previous assignment that you completed. Copy over the files from assignment 2.
2. Make changes or fixes to the assignment based on feedback.
3. Fix any bugs you might find.
4. Make sure your images and animations are in good taste. Anything insulting, degrading or offensive will result in the student receiving a failing grade and being reported to Academic Services. If you are in doubt, ask.
5. Using the theme of your game environment, design a splash page which will be the first page that user sees when starting the application. It can be a door that opens; floodlights that scan over an image on the page, moving letters/images, etc... use your imagination. If you use images, make sure you attribute any copyright information as necessary. The following requirements must be met in the Splash page:
 - a) The file name must be index.html.
 - b) You must use a combination of JavaScript and CSS for your animation (NOTE: CSS animation means keyframes and/or transforms).
 - c) You must use both a canvas and a non-canvas animation in JavaScript (along with a CSS animation).
 - d) The animation must be related to your game design.
 - e) The game name must be displayed on the Splash page; and,
 - f) There must be a way to skip the animation and go to the next page.
 - g) When the animation is complete, or the user presses the Skip automatically jump to the intro.html page. When you get to the intro page, follow the rules as below.
6. When the intro.html first loads, check to see if any values exist in localStorage for the game.
 - a) If no localStorage exists, use the form to prompt the user for his information as usual and store the information **and the current date and time in localStorage**.
 - b) If localStorage does exist, check to make sure that the fields are relevant to the game (there might be other localStorage used). If the localStorage is for the game, do NOT

display the intro.html page, but skip immediately to the game.html file (use location.href).

- c) In game.html, get the information from localStorage and display the person's information as you did in the previous assignment, but now add the last time the page was visited. Such as (and yours will look different):

Welcome back, Adewole Adewumi.

Your phone number is: (343) 123-4321 and your postal code is A1A 1A1.

You have \$3,600 left in your bank roll.

Your last visit was April 20, 2023 at 8:57 a.m.

On the next line include "Not Adewole Adewumi? [Change your credentials](#)".

- d) Update the lastVisit localStorage key to the current date and time.
 - e) If the user selects Change your Credentials, remove all the localStorage variables (do NOT set them to null, remove them) and display intro.html (once again location.href). With no localStorage variables, this should prompt the user to enter information into a blank form.
- 7. Update game.html to include animation while the game is played (it does not matter what kind or combination of kinds). This includes things like, but not limited to:
 - a) Seeing the dice roll
 - b) Spinning images
 - c) An animation when a round starts (shooting stars, things exploding, mud splashing)
 - d) Lights flashing on winning with messages that grow and shrink or whatever
 - 8. Change the intro.html form to use the jQuery Validate JavaScript library. Make sure that you submit using a jQuery-submit to only submit when the form is valid.
 - 9. Add a generic error event handler to display a message for the user if an unexpected error occurs. When an uncaught/unexpected error occurs, display a message, and restart the game by calling intro.html.
 - 10. The flow for the completed game is as follows:
 - a) The index.html page is loaded, and the splash page animation starts to play
 - b) Once the animation ends or is skipped, go to the intro.html page
 - c) In intro.html, check to see if the correct fields exist in localStorage
 - i. If the user fields are there, redirect immediately to game.html.
 - ii. If the user fields are NOT there, display form.
 - d) Once the form is valid, store the values in localStorage and redirect to the game.html page
 - e) Playing the game is:
 - i. The user makes a bet.
 - ii. The user guesses the outcome of dice roll (**Note:** This depends on what game you are assigned to)
 - iii. The user rolls the dice.

- iv. The system determines if the user wins or loses.
- v. The system increases the user's bank if the user wins and decreases the bank if they lose.
- vi. The user goes back to making a new bet
- f) The game.html page must have a link (or button) which will:
 - i. Remove all values from localStorage
 - ii. Return to the intro.html page to allow new information to be entered
- g) When the user leaves, display a message thanking the user for playing and showing the final amount. This can be done with a message that is displayed or using a separate screen.

***** Both HTML pages must be responsive and look good at 768px, 992px, and 1200px*****

To Commit Progress/Submit

Ensure you commit regularly as you conduct the assignment. Each commit must have a meaningful message explaining in brief what was achieved. The commands are like the labs:

```
git add .  
git commit -m <<commit message>>  
git push origin main
```