10/16

10:40 AM - Started working on exercise 4

Completed problemA, very straightforward
Completed problemB except for FizzBuzz which fails the tests

- Reread the problem and figured out that I misinterpreted it, the words are supposed to only replace one element and not the whole array

Completed problemC except for the second fin due to incorrect instructions, will fix later
Can't seem to get ProblemD both() function to work

- Reread the instructions, forgot to pass the parameter into the callbacks

12:20 PM - Stopped working

10/17

3:30 PM - Fixed and completed problemC

10/18

10:30 AM - Started working on exercise 4

Started working on problemF

- Drawn lines have hard corners instead of rounded edges like the example
    - Googled, found lineCap property
- Test fails for some strange image comparison reason
    - It appears to be detecting a blank canvas, drawing the clock outside of the loop seems to fix it

Completed problemE, also pretty straightforward

11:40 AM - Stopped working

10/23

10:30 AM - Started working on exercise 5
Working on problemA

- Frequently referring to the textbook to learn how to do stuff
- Can't get the colors to be the right size on renderPalette, not sure how getBoundingClientRect works
    - Typed into console, found that it's an object with a width property, so I just needed to select that property

Working on problemB

- Can't seem to get new tasks to take on value of inputted text
    - Forgot that inputtedText is part of the state object and not a standalone variable

Working on problemC

- Instructions say to pass in `'2'1 to canvas.getContext, that seems off
    - Pulled up previous example and found that it's supposed to be '2d'
- Not sure how to get color input value
    - Googled, it's just the value property of the color input

Working on problemD
- Not sure how jQuery works yet, quickly skimmed book section and pulled up jQuery documentation
- Looking up jQuery functions for each step

12:20 PM - Stopped working

10/24
7:20 PM - Started working on exercise 5 problemD

Working through the huge event listener
- Can't get remove button to work
    - Testing in console seems to work
    - I just forgot the period in the class selector aaa

7:50 PM - Finished work

10/25
12:30 AM - Started working on game

Basic idea is to make an endless game where you avoid scrolling obstacles and move with a grappling hook
Start off simple and slowly add things in order of complexity
- Setting up a simple page with a canvas
- Set up game state and player objects
- Set up a game loop based on some simple examples found through google
- Draw a circle where the player is
    - Size and position are really wonky
    - Googled, found that I need to set canvas size within the javascript, not just css
- Handle basic player movement
    - Googled how to handle keyboard input
    - Added booleans to game state object representing arrow key input
    - Make a function that handles movement
    - Add velocity attributes to player, make keyboard buttons modify velocity
    - Got player to move, but it leaves a trail
    - Looked back to exercise 4 clock, figured out I need to clear canvas every frame

    - Added gravity and a jump as a placeholder movement mechanic because grappling is complicated
    - Added player position constraints to prevent going off screen
- Draw a simple 3-colored background
    - Got idea to have it constantly change hue, trying to figure out how to dynamically set colors
    - HSL seems to be what I want, but hsl isn't a Javascript function
    - Googled and figured out that color takes hsl(...) as a string, not a function
- Draw a targeting laser in the direction of the mouse
    - Googled to figure out the math behind calculating the angle to the mouse, found the useful Math.atan2
    - Want to change laser appearance if clicking; implemented mouse eventlisteners and added associated boolean in game state
    - Made the grapple gray and transparent if mouse up, lit up if mouse down

3:30 AM - Stopped working
10:30 AM - Started working

- Start working on obstacles
- Set up an array to hold all obstacles
- For simplicity each obstacle will just be a square with an x and y coord
- Set up functions for spawning, moving, and drawing obstacles
- As a placeholder, just add one obstacle in middle of screen while I test drawing
- Draw each obstacle as a black square with bright stroke of game hue
- To move obstacles, loop through every obstacle and move it to the left every frame to represent scrolling
- For obstacle spawning, start off just by borders on the top and bottom of screen
    - Got them to appear and move, but they appear every frame and overlap
    - Set up game variable to 'delay' the spawning so they only spawn after one full obstacle width of movement has occurred
- Slightly more complex, add randomly generated obstacles
    - Make a separate 'delay' variable from the border to control spawning
    - Random generation looks bad; should try to snap to grid
    - Managed to snap to discrete Y values, but can't figure out how to do it with X yet
- Handle collisions between player and boxes
    - Googled circle/rectangle collisions, found stackoverflow solution that I adapted
    - Collision detection works, want to make a simple game over state
        - Game over should desaturate the colors and stop the game loop
        - Stopping the loop isn't working; looked through examples and found that they stored it in a variable so it could be accessed later
        - Implemented restarting with R key that just starts the loop again
- Want to remove random obstacles and start spawning actual patterns

    - Basic idea: Set up object with function names associated with ‘weights’, then put those into an array of patterns and shuffle it, then spawn those over time
    - Set up pattern strings going into a queue, but can’t figure out how to call the strings as functions
        - Googled, found solutions that use window[functionName]() and object[functionName](), but those spewed errors

12:20 PM - Stopped working

10/28

12:30 AM - Started working on game

- Continuing work on patterns
    - Attempted to debug by console.logging but couldn’t find anything wrong
    - Decided to check if the function was defined before calling it, which fixed everything
- Set up a few basic patterns of obstacles, kind of tedious
    - Each pattern spawns blocks and then sets the delay until the next pattern
- Next goal was to set up a scoring system
    - Set up DOM element and just add a point whenever a pattern spawns
        - Feels off since it’s disconnected from what the player is actually doing
    - Idea: Spawning a pattern also spawns a ‘checkpoint’ x value that gives a point when passed
    - Created checkpoint array
    - Each pattern pushes a checkpoint array at a certain point
    - Moving obstacles also moves the checkpoint
    - Added function to loop to update score
        - Check the frontmost checkpoint and if the player passed it, remove it and increase score
- Finally time to deal with grappling and movement
    - First, figure out line collision with obstacles so that we know where to target
        - Googled line collision with square, found helpful code snippet to find the intersection of two line segments
        - Basic procedure is to calculate the angle between the player and the cursor and define a line between them, then loop through all the obstacles and check if they intersect with that line, then get the closest intersection to the player, which is where the target is
    - Then, allow the player to ‘hook’ onto a point
        - Calculate the intersection point every frame until the player clicks, then lock that point until they release the mouse
        - Need to account for movement of obstacles, so move the grapple point with the obstacles every frame
    - Finally, figure out the hard part - movement

    - Idea is to have a physics-based grappling system that's affected by gravity and momentum
    - Googled and found a helpful stackoverflow thread with high-level instructions
    - Attempted to implement for like 2 hours but was much more complex than I anticipated couldn't get it to work consistently or well
    - While experimenting, had the idea of just pulling the player to the grapple point instead of messing with physics
    - Kind of worked, but felt unfun especially with gravity
    - Removed gravity and horizontal movement, limiting movement to just hooking onto a point and then pushing or pulling, which was interesting and fun
- Set up high score system
    - Added checks for high score each time score is updated, got high score to update and display but not persist
    - Googled how to save high scores locally, found useful examples that worked
    - Added button to reset high score

6:00 AM - Stopped working

10/29

1:00 AM - Started working on game

- Started working on improving aesthetics and implementing audio
    - Created sound effects with Bfxr
    - Found Sound.JS library, was easy to implement simple sound effects that played when game events happened
    - Goal for music is to use some sort of audio frequency analysis to affect the display of different game objects
    - Googled javascript audio visualization, but the examples didn't seem to work for me and couldn't get audio to play properly
    - Found Dancer.js library which was exactly what I needed
    - Added audio visualizations with a lot of tweaking:
        - Background stripes shake with different frequencies
        - Whole audio spectrum in the center of background
        - Player pulses with low frequencies
        - Added ring around player that makes larger pulses
        - Obstacle fill becomes brighter with high frequencies
        - Obstacle stroke becomes brighter with low frequencies
        - Obstacle sizes subtly pulse with high frequencies
        - Whole screen very slightly shakes in response to different frequencies
    - Added mute toggle
- Added pause feature that just toggles the game loop, will add graphics for it later
- Improved HTML and CSS to have a more reliable page layout

    - Canvas is now always centered, with stats sticking to its left and instructions sticking to its right
- Added a bunch of fun obstacle patterns
    - Making patterns was very tedious and I found myself re-typing a lot of the same long statements; made helper functions to add a block and add a checkpoint
- Added trail to player
    - Idea is to store player's previous positions and draw them as circles of decreasing size and opacity; previous positions should also move to the left to create illusion of motion
    - Got it to work, but it looks strange and too granular
    - Tried another approach of drawing lines to each point in order, which looked even weirder and caused ugly behavior when moving to the left
    - Tried drawing lines from the player to each individual point, which worked pretty well and looked the best out of all the options
- Crazy idea on a whim: moving obstacles
    - Idea is to add a moveY attribute to certain obstacles that causes them to move in one direction continuously
    - Implemented moving and made the obstacles wrap around the screen if they go off-screen
    - To allow grappling to moving obstacles, give the player object a reference to the currently grappled object, so while the player is grappled, move the grapple at the same rate as the object
        - Added bonus of making the grappled object light up
    - Created a few patterns with moving obstacles which is cool, but now I want to try obstacles that move sinusoidally
    - Add new attributes that control period and offset of obstacle, then make the obstacle move on a sine wave based on those plus game time
        - Had to figure out what to add to position each frame, math is hard
    - Got sinusoidal obstacles to work, but it destroyed grappling
        - New idea - give the player object a reference to the difference between the grapple's Y position and the object's Y position, and continually have the grappleY be the objectY + grappleYOffset
    - Made a few cool patterns with sinusoidal obstacles

7:30 AM - Stopped working

10/30

2:30 AM - Started working on game

- Added pause overlay when the game is paused
- Added text to the game over screen

- Implemented a simple menu that briefly explains the game and prevents it from autoplaying
- Add modal popup whenever player dies with a new high score
    - Kind of intrusive, so I'll make it automatically go away after one second
    - It stays up even if the player hits R to restart which feels bad, made it instantly dismiss if the player presses R
- Sometimes when hard refreshing, the menu fonts default to a weird serif font that looks bad, not sure how to fix yet

5:00 AM - Stopped working

10:40 AM - Started working on exercise 6

Got pretty much everything to work, but a lot of the tests fail

- Reread instructions; I implemented renderSearchResults slightly wrong and had it take in an array and not an object with an array
- A lot of tests still fail and seem to not be waiting for the results; will investigate later

11:20 AM - Started working on game

- Still not liking the modal, looking for slightly less intrusive alternatives
    - Materialize also has a modal that pops up from the bottom which I like more, switched to that
- Googled the font loading problem, tried a few solutions that didn't really work
    - Digging around and found Font Face Observer, which was exactly what I needed and fixed the issue

12:20 AM - Stopped working
6:00 PM - Started working on game

- Added little toast popups whenever a high score is achieved or the speed increases
- Improved aesthetics
    - Prevented hue from going into deep blues/purples which ruined the contrast and made the game unreadable
    - Hue no longer constantly cycles through the whole rainbow; instead, it stays in a particular color range and changes whenever the speed increases or the game is restarted
    - Made the hue cycle more rapidly when at max speed
- Ran code through a linter and fixed a few minor errors
- Improved page accessibility to screen readers by adding live regions and hiding unnecessary elements

7:30 PM - Stopped working