NM2207

Session 09

timerGame Challenge Brief

Overview of what we will do today:

- Add a way to count the number of clicks to a moving dot
- Change difficulty level depending on radio button selection

This assignment just brings it all together and lets you have fun with the design part! It builds on the circleMover codealong we were doing so far.

We return to a simpler version of the problem – one with a single circle on the screen. Today, we'll turn it in to a "race against time" click game where the user
a) Starts (and restarts) the game by clicking the button (already in place)
b) Can restart the game by clicking the button again (already in place)

c) Has to click a moving object as many times as possible within a fixed amount of time. When the time is up, the animation stops, and a dialog box pops up reporting the score
d) Can choose a level of difficulty that affects some aspect of the game. In this case, we want to change the size of the dot and its speed.

Try to write down the steps you will need to solve this challenge, starting from your existing code, before reading the steps provided below!!

1. What variables are you going to need? What functions will you write?
2. What are the key things you need to do, and where (in which functions) will you do them?

AFTER you have sketched out your strategy, variables, and functions, then see how they correspond to the steps below and code it up!

Steps for Part a and b (already provided to you in the template)

Reminder: Check to make sure things are working properly *at every step* in your development with console messages, etc.

Part a

You'll want variables to store the game startTime, the number object clicks, and whether the game is running.

Of course, put a click listener on your object and count clicks.

Change your toggle button text, and its behavior so that it "restarts your game" when it is pushed. The callback function should:

a) use the Date() object to store the new start time
b) start calling your draw() function so that the object animates
c) reset your click counter to 0
d) (It should do these things only if the game isn't already playing!)

Part b

Create an endGame function that

a) stops the animation by stopping the periodic calls to your draw() function
b) opens a confirm box to report how many clicks were made during the game
c) resets the graphical object back to the center

Stop the game after the time limit is reached by calling endGame(). Where will you do that?

Allow the user to choose different levels of difficulty

- For example, you could add radio buttons under your 'play' button to select difficulty – what might you change to make you game more/less difficult?

Comment your code thoroughly so that you and others will know what each part of the code does and how you were thinking.

Hints for Part c and d

1. **Observe the code and the alert windows carefully. Check where numClicks is declared and where you want to update it. Which element should get an event listener?**
2. **What are the properties tacking the size of the dot and its speed?**