

Graphics Animation of Random Walk

Random Walk Assignment

So far, you've been able to use HTML, JavaScript and CSS to create a UI element representing a ball on the web page. You were also able to add animation to this element and make it move on the page.

In step 1 of this activity, you will add JavaScript code that will create three balls within a defined area of the screen. Then, in step two, you will make the three balls move randomly within that area.

The starter code for this activity includes the following:

- **Index.html**: This is the Html file where all elements are created. You will not need to update this file to solve this activity.
- **styles.css**: This is the CSS file containing all the styles needed for this activity. You will not need to update this file to solve this activity.
- **walk.js**: This is the JavaScript file containing all the functions needed for this activity. **You will need to update this file to solve this activity.**

The **walk.js** file contains code and comments that will help you through this activity. Here are some more details:

- The **area** variable is an object that defines the area of the screen in which the ball elements will be displayed. It has a set width and height, and a reference to the Html element called **area**. This reference is stored in the property **element**.
- The **initialize** function creates the area div on the Html page.
- The **moveTo** function takes a "ball" and "x,y" coordinates as parameters, then it sets the coordinates of that given ball to the given x,y coordinates.
- The **changeDirectionIfNecessary** function takes a "ball" and "x,y" coordinates as parameters, then it determines whether the ball trajectory should be reversed if the ball hits the area bounds.
- The **create** function has the following parameters as input:
 - **color**: This is a string representing the color of the ball (ex. red)
 - **dx**: The starting position of the ball on the x-axis
 - **dy**: The starting position of the ball on the y-axis
- The **update** function has the following parameters as input:
 - **ball**: This is a reference to the ball
 - **x**: The new desired position of the ball on the x-axis
 - **y**: The new desired position of the ball on the y-axis**Your task in this activity is to complete the implementation of the function **create** in Step 1, and the function **update** in step 2**

After you're done, uncomment the following block of code. You should see three balls: blue, red and green on the screen.

```
initialize();
const ball1 = create('blue', 4, 3);
const ball2 = create('red', 1, 5);
const ball3 = create('green', 2, 2);
```

```
moveTo(ball1, 1, 1);  
moveTo(ball2, 10, 10);  
moveTo(ball3, 20, 20);
```

Task

- Complete the implementation of the function `create`

Track movement of the Ball

Now that you've created three balls on the screen, it is time to make them move.

Your task is to edit the `update()` function to add movement to the balls.

The `update` function should use the input `ball`, `x` and `y` to move the ball on the screen using the `moveTo` function. Then, it needs to check the direction the ball should move in using the `changeDirectionIfNecessary` function every sixteen milliseconds. It should also keep calling itself every sixteen milliseconds to keep the balls moving.

After you're done, uncomment the following block of code. You should see three balls: blue, red and green moving on the screen:

```
update(ball1, 70, 0);  
update(ball2, 20, 200);  
update(ball3, 300, 330);
```

Task

- Implement the `update()` function to add movement to the balls