# Animation Enrichment Bundle (Alpha release)

**Ye Huang (utorid: huang853)**

1. This library is intended to add a few interesting animations to the webpage without the developer worrying about any of the detailed implementations, it is mostly about manipulating points on canvas element (by passing in **any** canvas element) to display interesting animations. This library should mainly be used to decorate any of the header/logo/titles if needed, or any other text that the developer thinks can use it.

2. For the alpha release, I have implemented the “spin and float”, “bounce” and “text fall” features:
    - Spin and float: As the name suggests, by calling spin/float on an html element, a spinning/floating animation will be added to the element, and the animation will change based on the corresponding argument(s) passed in.
        - Developer’s log: This part of the project is used as a recap for my html, vanilla js, and css coding, however, due to its lack of correlation with what the project is supposed to do (it’s supposed to be about points manipulation on canvas element), I will consider removing them once the project is complete, but it will remain in the project for now as a show of effort.

    - Bounce: By passing in a canvas element, it will generate a specified number of moving points on the canvas that are of same/different colors at different locations. If any of the points come in contact with the border, it will bounce back.
        - Developer’s log: This part is not that impressive by its look, but it does bring a lot of potentials to my project; After implementing this feature, I now have a coordinated system of “point” objects (which I will discuss at the end of this report) work with, and I do not need to rely heavily on css for my animation (this feature has its own animation loop and does not use any of css apart from general style setting like a border or text alignment).

    - Text Fall: Very similar to Bounce, by passing in a canvas element and the string wished to be rendered, it will plot out the string using the “point” object on the canvas; After clicking on the canvas, the points will accelerate and starts the Bounce animation mentioned previously, and as the client clicks more, the points will accelerate faster.
        - Developer’s log: This part of the project is more practical and usable (and more interesting) than the previous 2 features in my opinion, I can personally see someone using this effect on their logo or title to add a bit more gist to the webpage. Since I didn’t use any external library to help me with identifying the string, I had to manual plot them out and then

store their coordinates in a json-like object inside its init function (fell free to use the suffix /pub/textFall.js to take a look at the huge matrix of coordinates if you are interested), which took me a lot of time and effort to do. However, after completing this, I now can practically turn any text given to points and plot them on the canvas.

3. Heroku url: https://salty-fortress-52521.herokuapp.com/ (last tested on: July 26, 2020)

   After entering the page, you will see a welcome message and a list of urls to navigate you to the corresponding features I have described above. Inside each demo, there would be a side note about the demo.

4. For the structure part, I meanly wish to talk about the “point” object I have been mentioning.

   Here is an example of point
   Point instance: {
   ```
   x = 100
   y = 100
   dx = 2
   dy = 2
   maxSpeed = 5
   canvas_width = 800
   canvas_height = 600
   ctx = canvas.getContext("2d")
   r = 5
   border_width = border_width
   ```
   }

   A point instance is created upon demands (either from client or the number of points needed for the text passed in), and will be drawn to the canvas (ctx in our example) according to its x and y position with a radius of x. Then, as the animation begins, it will update its position through their dx/dy properties. In order to bounce back from the border, it will have to know the width and height of the canvas as well as the thickness of the border around it.

   Please note that this is only the alpha release, and there will be more data added to the object, but these will be the basics to keep the animation running.

   Each “point” object will have a draw and update method, but the developers will probably mainly concern themselves with the update method, as it defines the behavior of the points.

The update method will update the location of the Point objects based on the conditions provided (like bouncing back upon hitting a wall, or accelerates if dx/dy is not at maxSpeed), and it will call the draw method at the end when all locations have been updated.
There are other methods that the animation relies on (for example, the animation loop method I briefly mentioned earlier, or the init method that generates all the point objects and have them ordered), but I don't think they should be made accessible to the developers, as they should not worry about how the animation is run.

5. When the library is finished, it will provide animations that interact with mouse-clicks and mouse-move, such as the points used to form a letter that will move away as the cursor draws close to them, then move back when the cursor moves away, or having randomly moving points form a piece of text that was preset by the developer upon mouse-clicks.

The more challenging part about the 2 features I described above will be how to let the points move back to their designated position after the event has occurred.
One way is to simply calculate the distance between its current position and the designated position, then change its dx and dy accordingly. But this will cause the graphical performance of the animation to suffer since it will look strange and jagged. I will do some research and try to find a way to make the animation look smooth and natural, so the points will come back with a curvy route, and as it approaches its destination, it should slow down till it hits the position and stops.

I will add in more features if time permits, as I started off the project with no clue about what I should do (hence the "spin and float" feature), but now I have found a goal to accomplish, and I believe I will strive to achieve it.