My design goals were informed by the game I was making the interactive poster for, Katamari Damacy. I wanted to somehow incorporate a representation of the katamari and its sticking mechanic for picking up objects. I wanted to give the poster a spacey, fun, abstract feeling with minimal graphics and colorful text colored by word or line. I chose colors that resemble the first katamari with its colorful rings and soft primary and tertiary colors. I started by coding the rotating image on the cursor location, which was originally an image of the earth; then I created a small 3 frame animated gif image for the prince and positioned it also in relation to the rotating earth image.

I created my own TextBlock class to group text info with positional info and a boolean that reflects whether the text has gotten within a radius of the katamari. I basically just check if the text is stuck and if its not, it draws in its original position, and if the text is stuck it will travel and rotate with the katamari.

I kept the earth image rotating in the lower right area of the screen and made my own earth image to replace the placeholder; I also added a counterrotating atmosphere image overtop of the earth for effect. I wanted to make this little graphic interactable so I made the rotation reference a variable for angle that is incremented with scroll wheel.

I designed out most of my poster by individually creating TextBlock objects with positions relative to the canvas position, then I created an array and used loops to create and update TextBlock objects for the entire sentence while selecting from a colors array in order.

I wanted to make the experience more complete with sounds so I found a video with katamari tech demo sounds and clipped a few for the katamari plucking sounds, which I select from at random and play when a TextBlock is stuck. I wanted to add a song from the Katamari Damacy OST, Lonely Rolling Star, but it was too large so I clipped the intro, a loop of the chorus, and the outro and stitched them together with logic in the draw() function to play the intro, loop the chorus three times, and then play the outro.

It was difficult at first figuring out how p5.js handles translation, rotation, and matrix states so it took me a lot of time to figure out how to position and rotate all of the elements how I wanted to. I learned a lot about javascript and p5.js doing this project and I think it turned out nice!