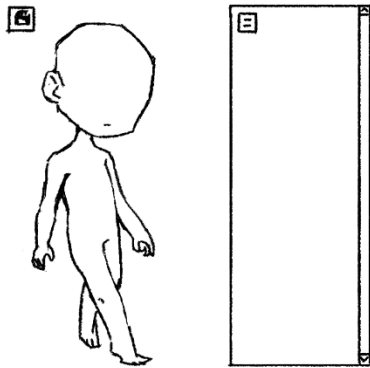


## Final Project Code + Design Critique

1. My project is a dress up game where the user can click on items to dress up a base. The items are stored in an array of objects with data about each item such as its file path, position, and width. This array is used to both place the image on the base, as well as serve as icons in the side box. The items in the side box are drawn using a data join with the items array. When a user hovers over any of these items, the background changes color. There is an array containing every item type that is currently selected. When a user clicks on an item in the side scroll box, it calls a function which updates in the current selected item for that type and calls a function that redraws the character. The user can also change the skin tone of the base by selecting one of the boxes, which also updates the current selected base and calls the character drawing function. There is a row of buttons below the item box and dress-up scene to clear items and change the background animation.



2. This is the initial sketch of my application. I originally wanted there to be a screenshot function to save the image. However, once I implemented it using html2canvas, I didn't like how it looked so I got rid of it. I also couldn't get the mousedown and mouseup events to detect anything, so I also did not implement the drag into place function to assign items and used a click event listener instead. I also planned on having filters, but for some reason the inline CSS for visibility was not working so that idea was also scrapped. I added buttons to clear different categories, and moved the item box to the left. I also added a background music player and a title.
3. At first glance, the product looks aesthetically cohesive. However, it could be better if there was an undo button. The lack of filters might be okay since there are not that many items, but the lack of items could also be bothersome. Additionally, the page does not resize well for a smaller screen.
4. I constructed the application around an array of item objects containing objects with information on each svg item and an array of the items selected. If there is no item selected for a category, the item in the selected array is set to an empty svg. I think this made sense because having an array of all the items made it easy to use a data join to display all of them in the sidebar, and then having another array of only selected items from the array of all item objects made it easy to update and draw the items on the canvas because I had access the attributes of each item.

```
function drawChar() {
  character.selectAll("image")
    .remove();

  var drawnItems = character.selectAll("image")
    .data(selectedItems)
    .enter()
    .append("svg:image")
    .attr("xlink:href", function(d) {return d.file;})
    .attr("x", function(d) {return xScale(d.x);})
    .attr("y", function(d) {return yScale(d.y);})
    .attr("width", function(d) {return d.width;});
}
```

5.

This is the function I wrote to draw the character. Using a data join for the selectedItems array, it draws the items that are currently in the array on the canvas using the attributes of each item to get the url, get the position, and resize the image. I chose this approach because it was the most straightforward way of drawing every element that I had stored in my selectedItems array. This also made it simpler to reset items and update items later on, because I could iterate through the array and set every item to be empty.

```
var body = character.append("svg:image")
  .attr("xlink:href", "bases/" + selectedBody)
  .attr("class", "base")
  .attr("x", xScale(baseX))
  .attr("y", yScale(baseY))
  .attr("width", 150);

//pants,skirts
var bottom = character.append("svg:image")
  .attr("xlink:href", "images/" + selectedBottom.file)
  .attr("x", xScale(selectedBottom.x))
  .attr("y", yScale(selectedBottom.y))
  .attr("width", selectedBottom.width);

//shirts,dresses
var top = character.append("svg:image")
  .attr("xlink:href", "images/" + selectedTop.file)
  .attr("x", xScale(selectedTop.x))
  .attr("y", yScale(selectedTop.y))
  .attr("width", selectedTop.width);

//shoes
var shoes = character.append("svg:image")
  .attr("xlink:href", "images/" + selectedShoes.file)
  .attr("x", xScale(selectedShoes.x))
  .attr("y", yScale(selectedShoes.y))
  .attr("width", selectedShoes.width);

//head -----
var head = character.append("svg:image")
  .attr("xlink:href", "bases/" + selectedHead)
  .attr("class", "base")
  .attr("x", xScale(headX))
  .attr("y", yScale(headY))
  .attr("width", 150);

//eyes + mouth
var face = character.append("svg:image")
  .attr("xlink:href", "images/" + selectedFace.file)
  .attr("x", xScale(selectedFace.x))
  .attr("y", yScale(selectedFace.y))
```

6. This is an alternative way to draw the character. Instead of storing the selected items as an array, they would be stored as separate variables. Then, in the drawChar function, each object would be drawn. While the variable names make it easier to read, this is a less efficient way of drawing the character because there is a lot of repetitive code.

7. An alternative way I would create this application would be to use an array of objects containing urls for each item where one is already pre-positioned on a 500 x 700 svg canvas so each item can just be placed at 0,0, and the other is icon size for the scroll bar. I think this would be easier to code because there would be no need to deal with positioning and a better design choice.

8. One thing I found difficult was positioning and resizing the images. I had initially drawn the items in illustrator without deciding on a size for all the items and didn't decide what size the screen would be. When I started coding I guess and checked the position of each of the images, which ended up being very inefficient because they were inaccurate and when the window resized, the character moved offscreen. I ended up

resizing the images in illustrator, and setting up scales using the width to position the images on the canvas.

9. I think the application could be improved if there was an undo button, if there were more items, and filters, as well as if the window could be resized to be smaller without messing up the layout by pushing the screen to the next line.
10. I wanted to have filters for the category of each item, and I successfully implemented a method that filtered out items that were in the selected category in the itemBox. However, the items could no longer be clicked and the on click event listener was no longer detecting anything when items in the filtered lists were clicked. I need to know more about how the on() method works in order to understand why the items were no longer clickable after filtering.