

CS 5630: Final Project

Vista Marston & Chris Marston
Fall 2022



Process Book

Table of Contents



Initial & Updated Project Proposals

- Initial Project Proposal
 - Summary
- Updated project Proposal
 - Change of Design

Process Book

- Data Processing
- Piece Count Scatter Plot
- Trends Over Time
- Themes & Sets Only vs All Variables
- The Squiggler
- Heatmap

Initial Project Proposal



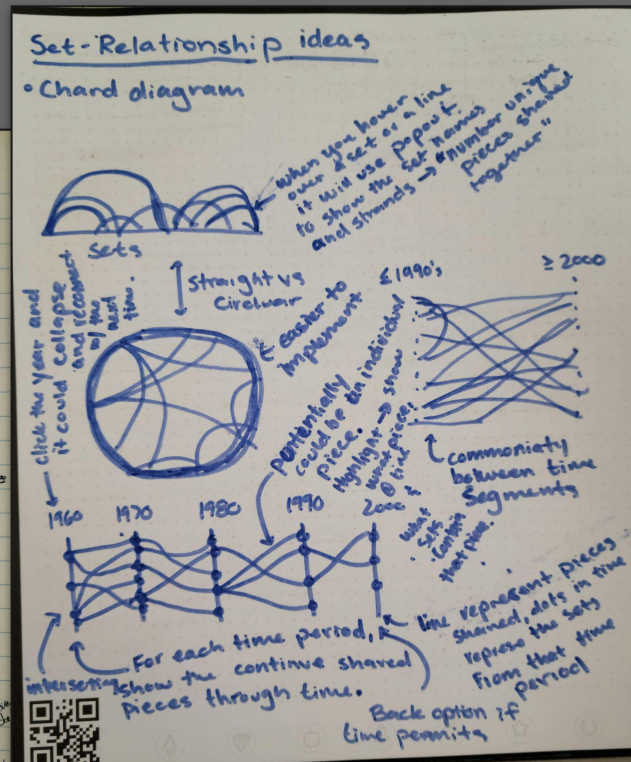
Summary



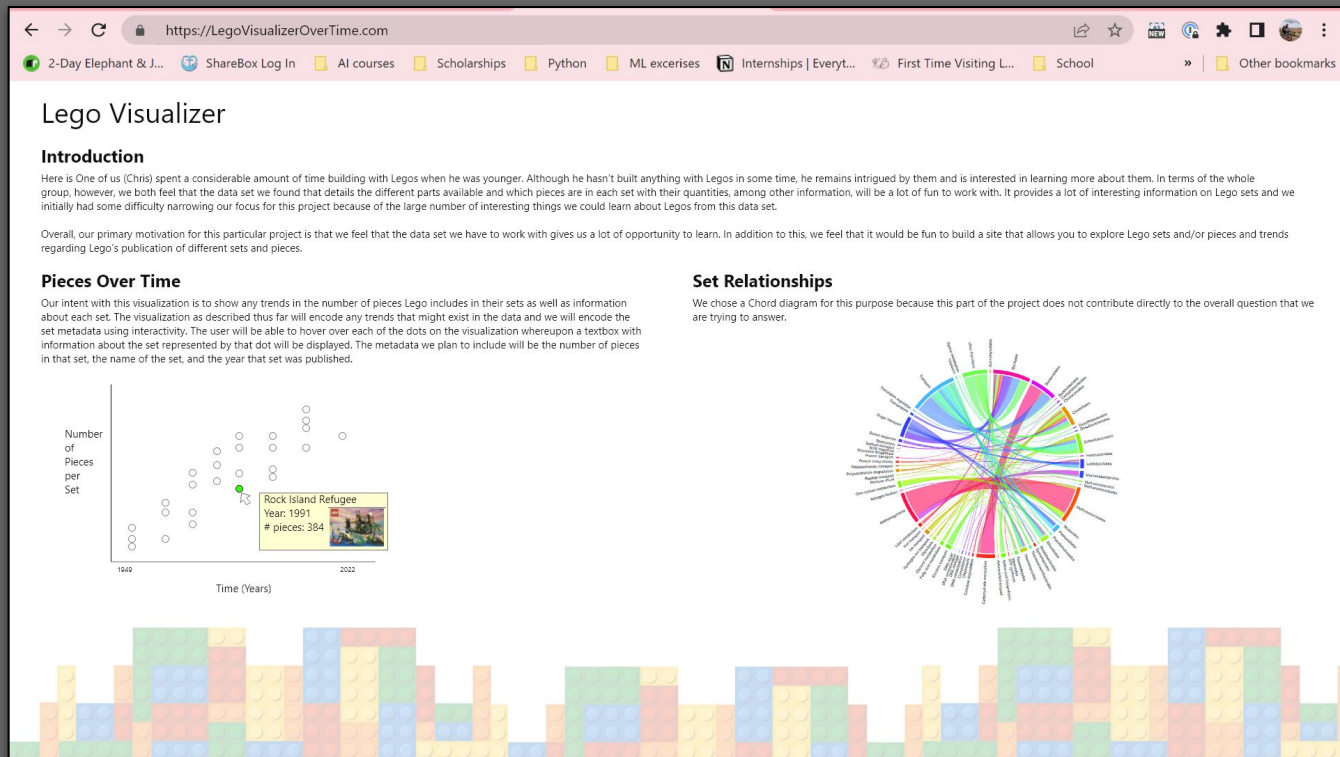
Our initial intent with this project was to generate visualizations that allow a user to explore how Legos have changed over time as well as how sets and pieces are related. Rebrickable, a site for posting custom Lego builds, maintains a database of all extant Lego sets and pieces. From this data we planned to generate two visualizations: a chord diagram that shows the relationships between sets, themes, or pieces and a scatter plot that shows how the number of pieces has changed over time.

Some initial design sketches.

Some initial design sketches.



Initial Design Proposal



Updated Project Proposal



Change of Design



Once we began processing the data from the Rebrickable database, it immediately became clear that we were dealing with far more data than we had anticipated. Our initial design was conceived under the assumption that we would have only a few hundred sets and a few dozen themes to visualize. Since 1949, the first year in the Rebrickable data, Lego has published thousands of sets under a few hundred themes. Far more than our design could effectively visualize.

After reevaluating the data, we decided that a redesign was necessary.

Updated Project Proposal

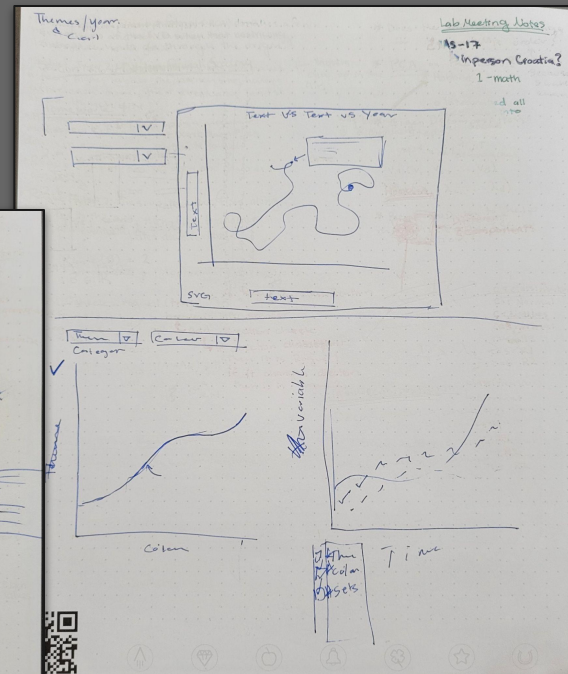
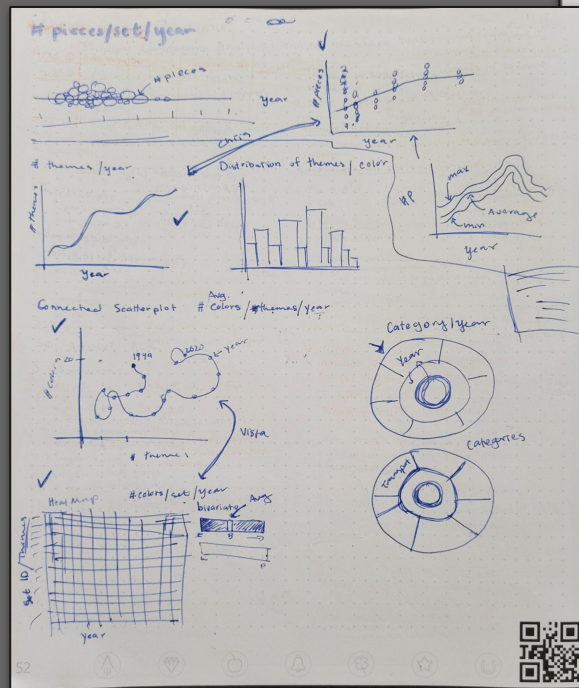


Due to the size of the dataset, we needed visualizations that would be able to effectively display a large amount of data in a concise and understandable format. To accomplish this, we chose to construct a scrolling story + exploration based website where the user could get a high-level overview of changes in Legos over time, but also explore deeper with interactivity. Additionally, we opted to focus solely on how variables have changed over time instead of how sets relate to each other through pieces.

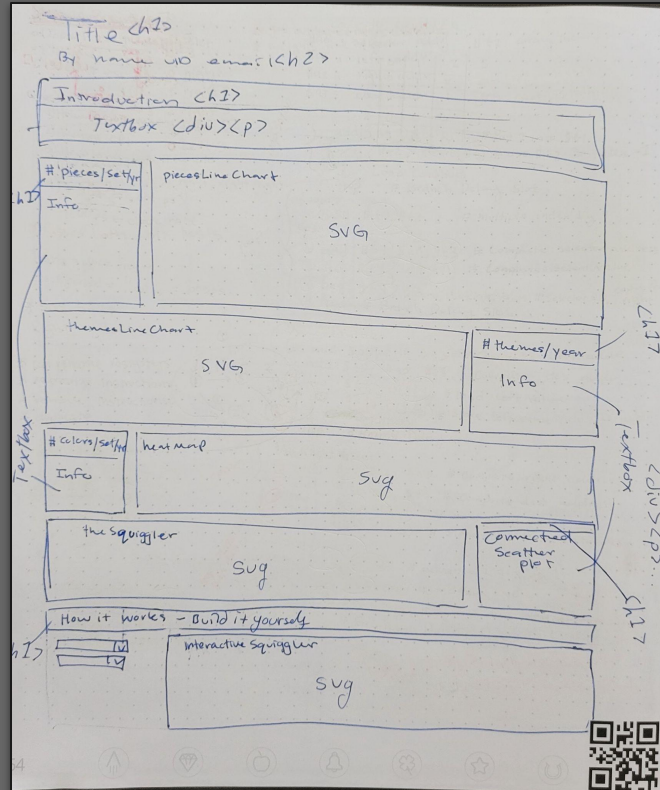
The new design consists of four separate visualizations, each using a basic type of visualization to ensure that the large amounts of data are able to be effectively visualized.

- Our initial scatter plot depicting each set, its piece count, and the year it was published.
- A line plot with time on the x-axis and the number of themes, unique colors, average pieces per set, and number of sets published – all per year – on the y-axis.
- An connected scatter plot that displays the user's selection of number of themes, average pieces per set, or number of unique colors on both the x and y-axes, where each dot is a year.
- A heatmap that displays how variables in the dataset had changed over time in.

Some redesign sketches.



Updated Design Proposal



Process Book



Data Processing



Initially, because the data was coming directly from a database in the form of CSV files – one for each database table – we expected to do little to no data processing. However, it quickly became clear that we would need to join the data from the tables together.

This issue did not pose to large a challenge as we were able to write a script to group the data into an array of objects that could be easily managed with JavaScript. This script runs as part of our website, which leads to slower load times but keeps the original CSV files in the source code for the website.


Additionally, when designing our data processing script, we chose to only process the data from the CSV files that would be used in the visualizations on the website. The variables we collated are as follows.

- Each year as an array of objects, each representing a set
- Each set object contains the following variables,
 - Year published
 - Set name
 - Theme name
 - Piece count
 - Number of unique colors
 - An array of color objects,
 - Color ID
 - Color name
 - Miscellaneous unused variables

Data Object Structure

```
▼ 0: Array(2)
  0: "1949"
  1: Array(5)
    ▼ 0:
      ▼ colors: Array(2)
        ► 0: {id: '4', name: 'Red', rgb: 'C91A09', is_trans: 'f'}
        ► 1: {id: '15', name: 'White', rgb: 'FFFFFF', is_trans: 'f'}
        length: 2
        ► [[Prototype]]: Array(0)
        num_color: 2
        num_parts: 12
        set_name: "Small Doors and Windows Set (ABB)"
        theme_name: "Supplemental"
        year: "1949"
        ► [[Prototype]]: Object
```

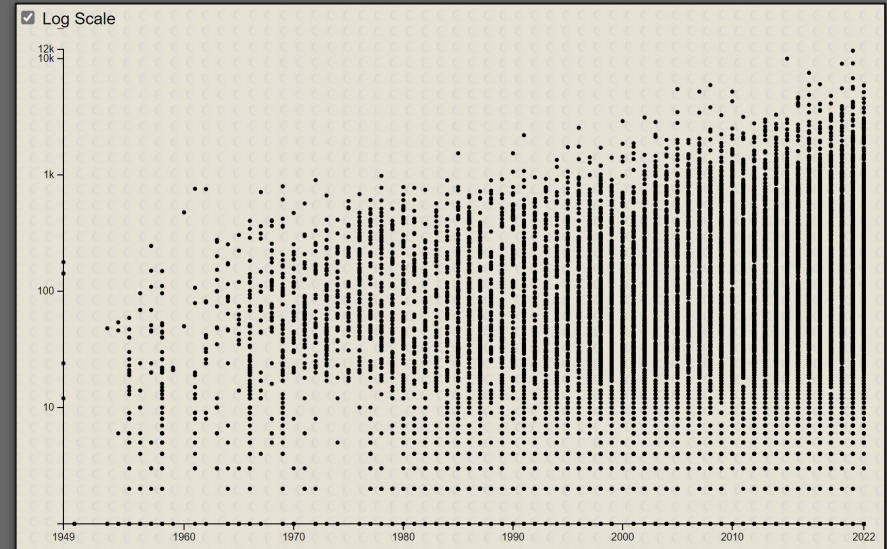
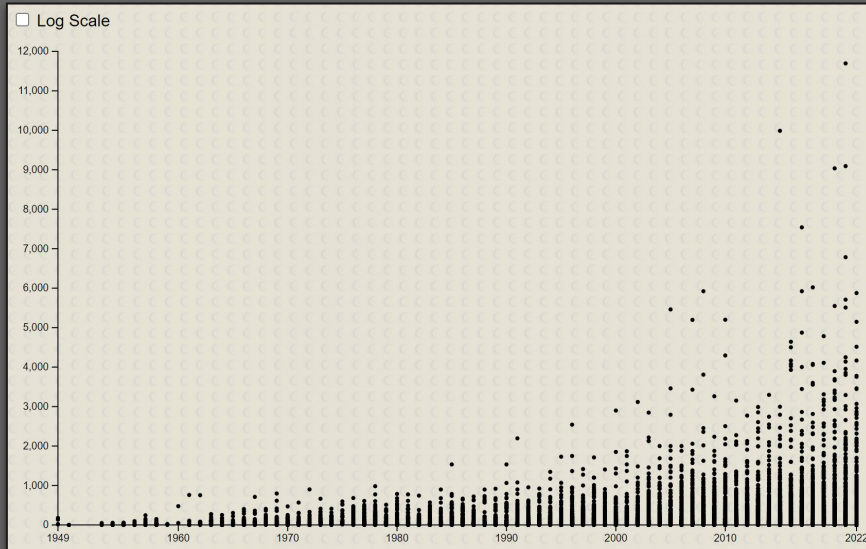
Piece Count Scatter Plot



Upon implementation of the scatter plot, we discovered that the linear scale we used for the y-axis resulted the compression of a large amount of the data due to a few outliers – sets with a very large piece count. To remedy this, we added a toggle to allow the user to switch the plot's scale between linear and logarithmic.

Additionally, we found that simply viewing the chart was not enough. There were several outliers that one might want to explore, so we added tooltips to the dots on the visualization to allow the user to view the set name and piece count in a popup tooltip box.

Linear & Log Scales of the Pieces Scatter Plot



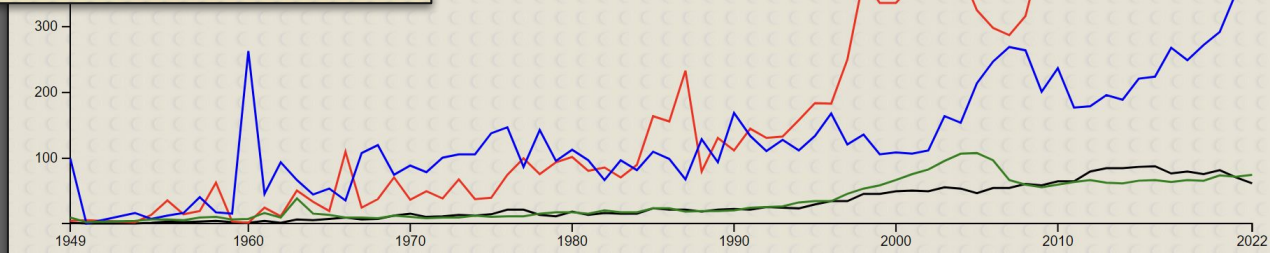
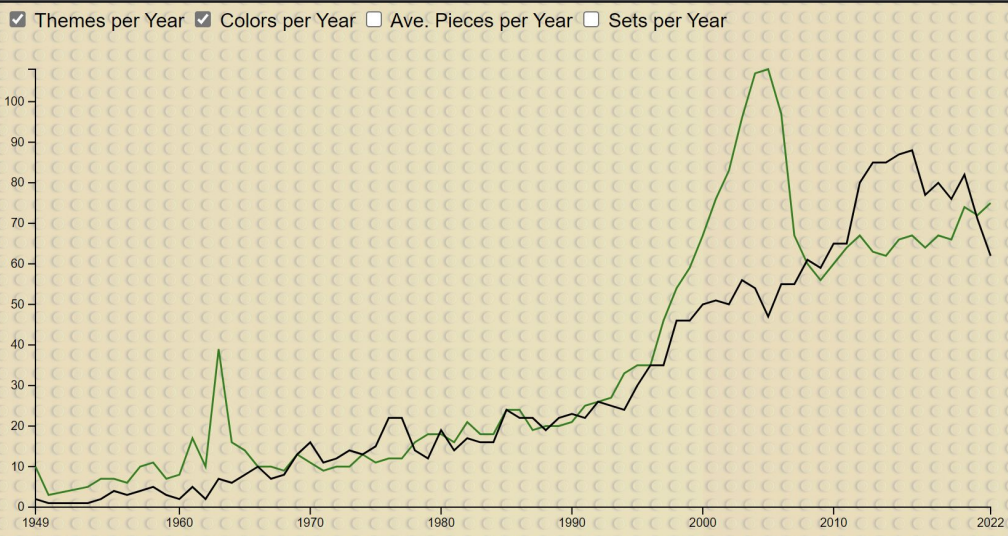
Trends Over Time



The line chart that shows the changes in themes, colors, average pieces, and sets-published per year was intended to show how different variables changed relative to each other over time. Unfortunately, we failed to consider the fact that the number of pieces in sets will more than likely be much larger than the number of sets or themes published each year. This trend also holds true for the number of unique colors used in sets. Because of this, the lines plotted for the number of sets and themes become squished, when plotted on the same y-scale as the number of pieces and colors.

To remedy this, we chose to add an interactive feature that allows the user to toggle which lines are displayed on the plot. When different selections are made, the y-axis scale adjusts so as to best display the current selection of lines together. This adjustment allows the user to explore the trends in the sets and themes in more detail by removing the lines for pieces and colors.

Themes & Sets Only vs All Variables



The Squiggler

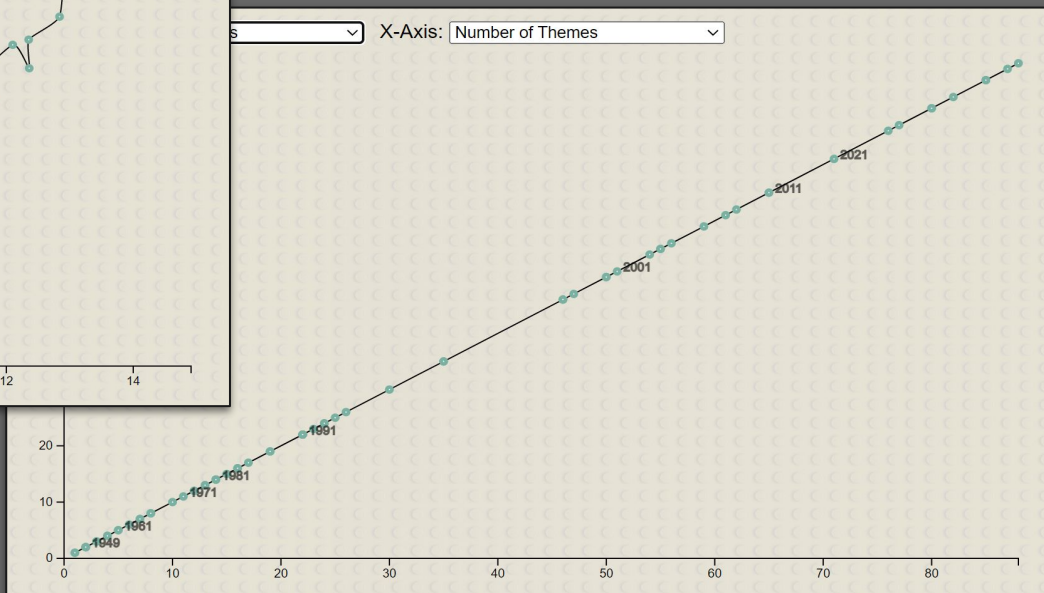
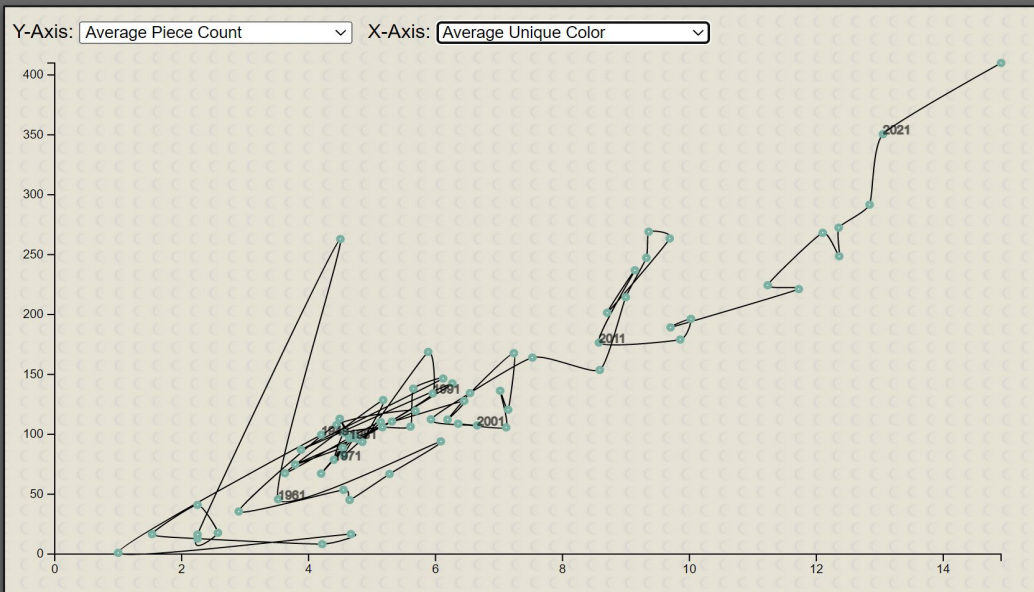


The connected line plot, affectionately nicknamed “The Squiggler,” worked out largely as we had anticipated. Our goal with this visualization was to allow the user to compare different variables with each other and see how they changed over time. To do this, we initially chose to add two dropdowns that allow the user to select the variables they want to display on the x and y-axes.

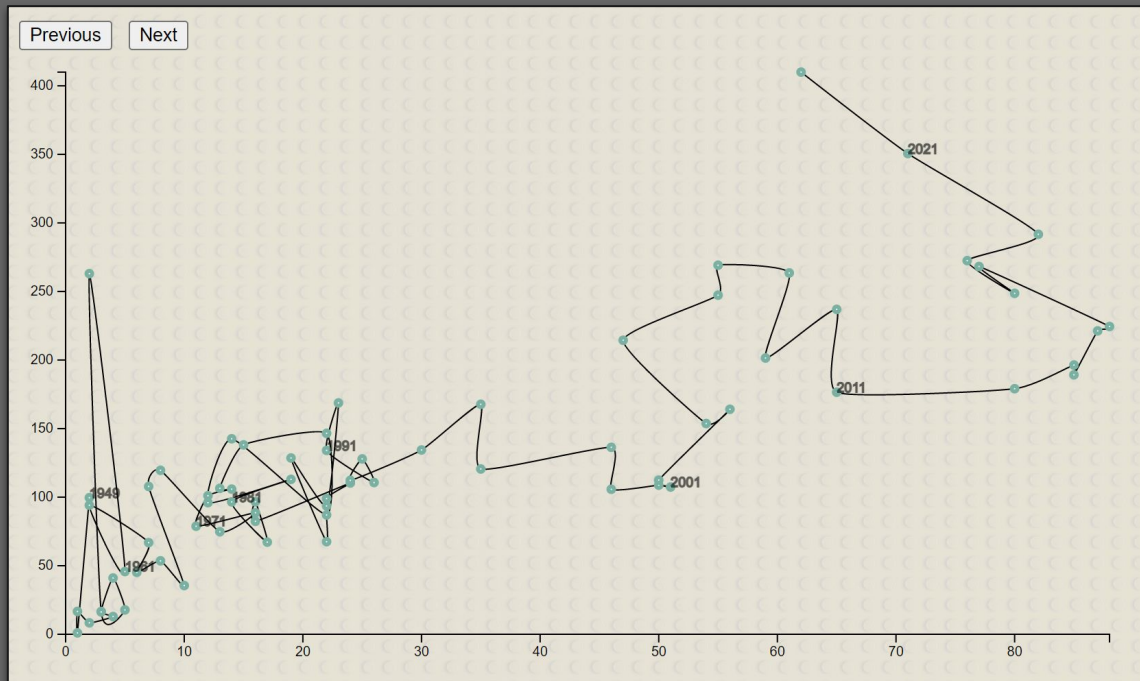
While the plot itself came out looking fine, there were a few unintended consequences associated with allowing the user such a high-level of freedom in choosing what is displayed on each axis. The first of these is that if the user selects the same variable for both axes, an uninteresting 45-degree line is plotted. The second is that a few of the combinations of variables produced large “hairballs” in the plot that were both impossible to interpret and unsightly.

Because of these issues, we chose to replace the dropdown selectors with “previous” and “next” buttons that allow the user to scroll through preselected axis choices. This change eliminated the 45-degree lines and the “hairballs” from the visualization and made the remaining visualizations more informative.

A “Hairball” and a Linear Relationship



Updated Squiggler



Heatmap



The heatmap visualization also gave us a bit of trouble. We had originally intended to use it to show how different variables relate to each other while still being able to show all of the individual sets, pieces, or themes. Since there are so many of them, the heatmap seemed to be a good way to show this. However, when first constructed the visualization, the relationships between the different variables – pieces vs themes for example – weren't strong enough to produce a very strong trend in the visualization. Additionally, because some years don't have elements that fall into certain categories, there were gaps in the heatmap that made it uninteresting and hard to discern any meaningful trends.

To adjust the heatmap, we selected two different relationships that produced stronger relationships and more compelling visualizations. These two relationships were 1) Frequency of each color used in sets published by lego over time and 2) the number of unique colors over time relative to the piece count of the sets. We then added a selection dropdown to allow the user to switch between the two visualizations and added tooltips to display metadata for each of the heatmap cells.

An additional adjustment that had to be made was the binning of sets in relationship 2 in order to be able to display the data in a concise manner. That is, sets were binned by the their piece count.

Sparse & Compelling Heatmaps

