Acquiring Design: A Case Study

Beyond simply providing access to digital collections online, how can museums create online environments for user discovery and exploration? Given the infinite canvas of the web, what new interactions can people have with a museum's collection in digital space? Furthermore, with access to individual object's metadata, what relationships and narratives could users uncover themselves?

The project also questioned how we understand the history of individual objects within the context of a collection. By examining when objects were created, acquired and exhibited, can we uncover trends that are indicative of changes in society's values, signify importance to our collective history, or illustrate an institution's philosophy?

In the context of these questions, I designed and developed *Acquiring Design*, a web-based interface for exploring relationships within the Cooper Hewitt Design Museum's collection.

The project was initially developed while attending a continuing education web design class at the School of Museum of Fine Arts in Boston in early 2015. Since then, Acquiring Design is a personal project I continue to explore and develop.

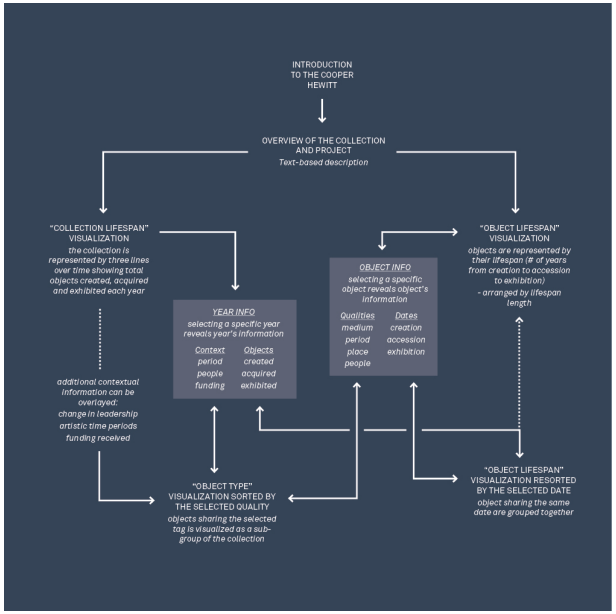
Beyond experimenting with how we might interact with digital collections, the project also provided opportunity to explore the newly released Cooper Hewitt API and learn the common data visualization library, D3.

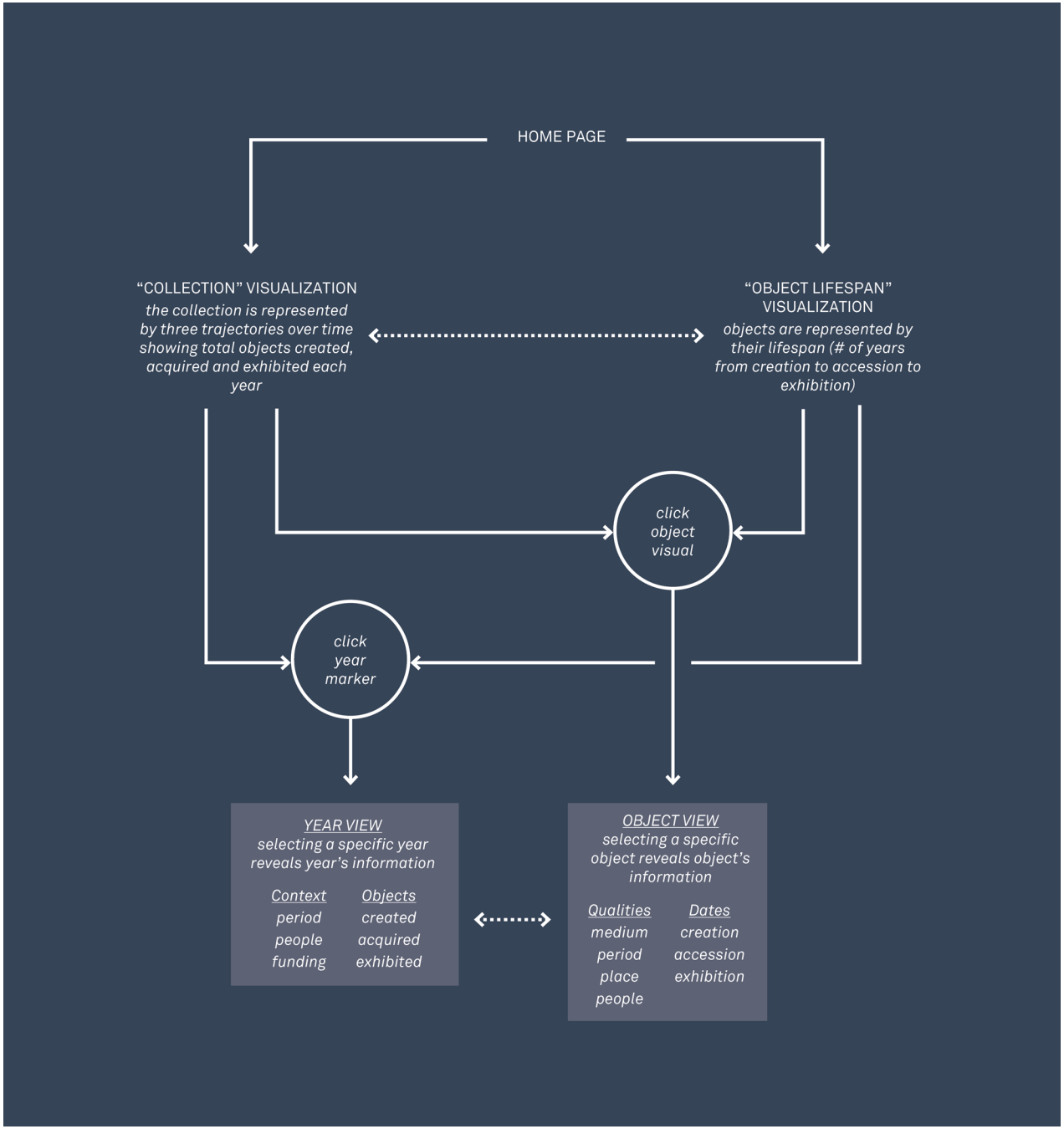
Exploratory Navigation: Multiple Ways To Explore The Collection

Navigation through the interface was inspired by choose-your-own-adventure stories. Within the framework of relationships, users jump from to related objects and years. I prioritized exploring and browsing the collection, allowing for multiple ways to end up at the same place. However, the initial user flow diagram, shown below, proved confusing.

By referencing Shneiderman's Visual Information Seeking Mantra—overview first, zoom and filter, then details-on-demand—I developed a clear structure and hierarchy for navigating various layers of the interface.

INITIAL USER FLOW DIAGRAM





Establishing Context Through Relationships Over Time

Acquiring Design presents the collections as a series of relationships between objects to establish a framework for users to explore and understand the collection.

Underlying the relationships are points in time: when an object was created, when it was acquired by the Cooper Hewitt and when it has been exhibited. These points in time establish common reference points between objects and provide context to understand individual objects.

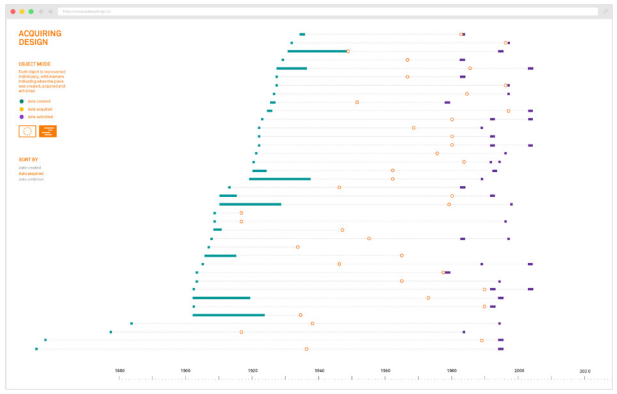
The interface presents two lenses through which to explore the collection:

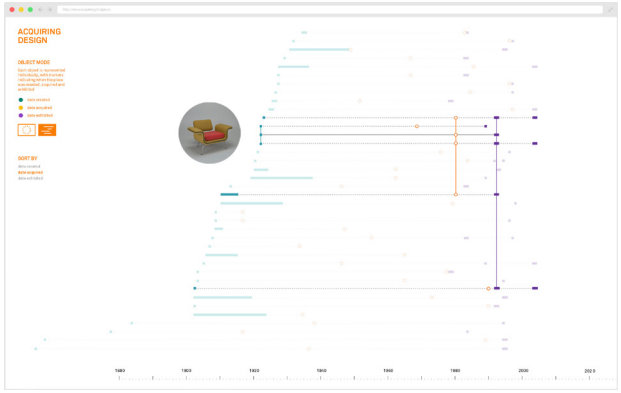
Object View: how individual objects relate to others with the same properties (i.e., designer, year, location)

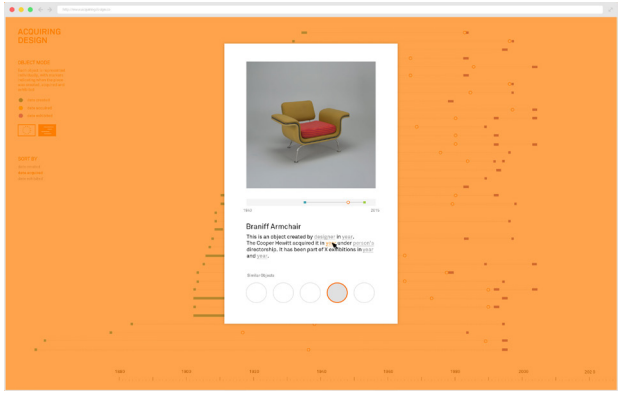
Aggregate View: how the collection as a whole has evolved over time (i.e., by department, volume of acquisitions, size of exhibitions)

Within each mode, interactions reveal additional relationships and information. Critical to layering additional information was maintaining the context of the overall collection and other relationships. Filtering or focusing on individual objects does not eliminate or hide the context of the entire collection.

LAYERED INFORMATION: OBJECT VIEW

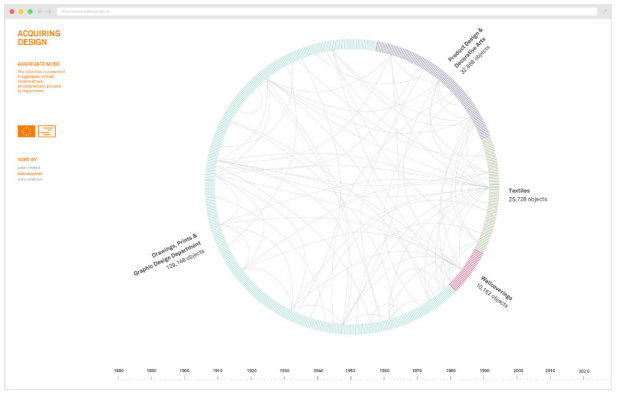


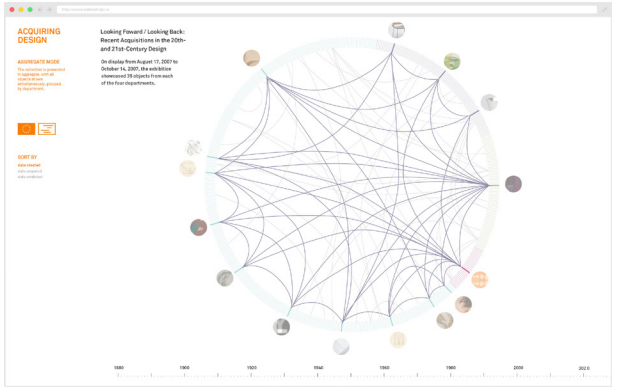


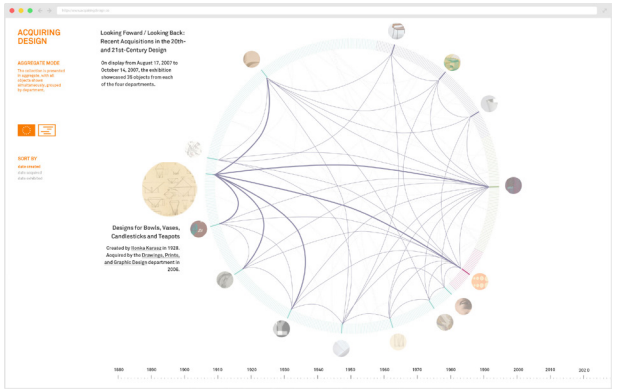


1. *Additional object information and related objects are revealed through hovering and clicking interactions.*

LAYERED INFORMATION: AGGREGATE VIEW



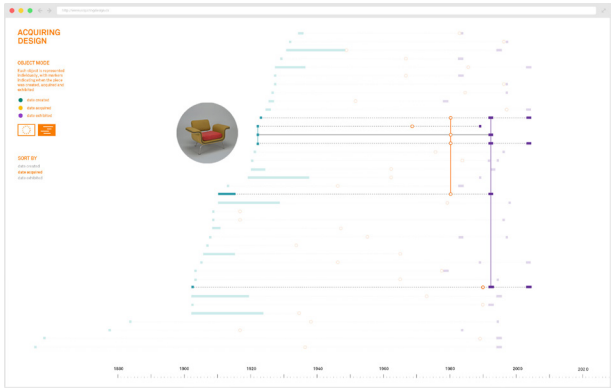




2. *Clicking on the network connections reveals which objects are related and how - whether through an exhibition, designer, location, etc.*

Coding the Interaction

On hovering over an object, any objects created, acquired or exhibited in the same year as the selected object are considered related. As such, they are highlighted and connected with vertical lines. This functionality required filtering the dataset to find matching objects and then adjusting their styling. The code snippet shows how this is achieve for objects acquired in the same year.

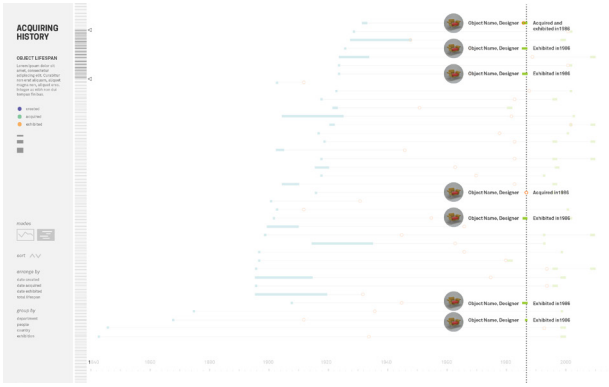


1. *The hover state interaction while exploring the collection in Object Mode reveals an image of the selected object and highlights objects related through common years.*

```
d3.selectAll(“g”).on(“mouseover”, function(d) {

// VARIABLES FOR "THIS" (the object currently hovered over)
var currentObject = d3.select(this);

//Acquired Marker Positions
var xPositionAcquired = parseFloat(currentObject.selectAll('g > circle').attr("cx"));
var yPositionAcquired = parseFloat(currentObject.selectAll('g > circle').attr("cy"));

// set up variables for the moused-over data
var selectedYearAcquired = d.yearAcquired; // year to compare with

// filter object selection to match mouseover object years
d3.selectAll('.object').filter(function(d) {
        return d.yearAcquired == selectedYearAcquired;
})
        .each(function(d,i) {
                d3.select(this).select('.acquired')
                .transition()
                .delay(100)
                .duration(250)
                .style('fill','orange');//change circle to filled

                var subSel = d3.select(this).select('g > circle').attr("cy");
                svg.append("line")
                .attr("class","acquiredLine")
                .attr("x1", x(format.parse(selectedYearAcquired)))
                .attr("y1", subSel) // y pos of matched marker
                .attr("x2", x(format.parse(selectedYearAcquired)))
                .attr("y2", yPositionAcquired) // y pos of highlighted marker
                .attr("transform", function(d,i) {
                return "translate(" + margin.left + "," + margin.top + ")";
                })
                .style('opacity','0')
                .transition()
                .delay(100)
                .duration(250)
                .style('opacity','1');
        })
        .transition()
        .style('opacity','1'); //matched objs stay solid
}); //end mouse over
```

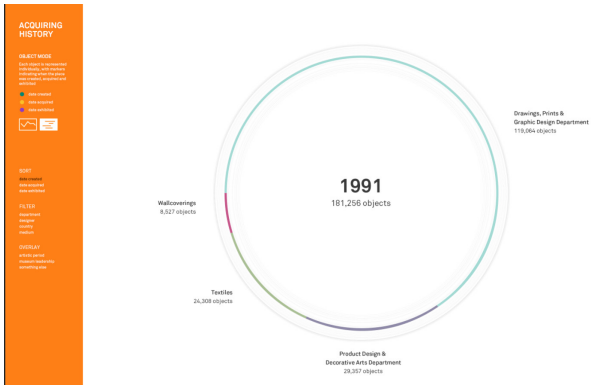
Design Explorations

I produced a number of iterations and design studies throughout the project. Furthermore, I maintained a process blog, *Working Through It*, documenting questions, code snippets, design iterations and commentary from others about the role of digital collections.

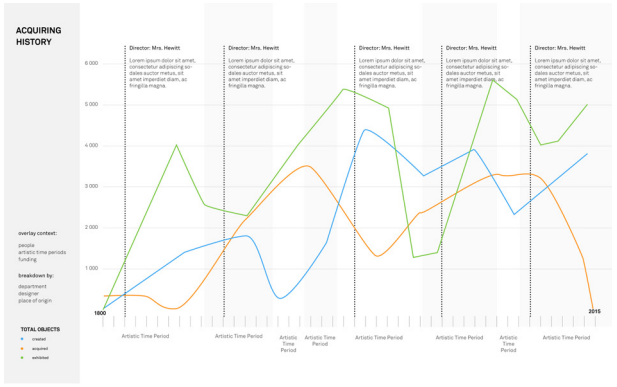


1.

2.



3.



4.

5.

1. *For the Object View, I explored different ways multiple objects could be shown relationally.*
2. *Initially, an overlay revealed information about a specific year.*
3. *Later, a revised version of the Aggregate View integrated this idea of displaying discrete information about each year.*
4. *The Aggregate View was initially designed as a line graph showing number of objects created, acquired and exhibited for each year. The museum's overall history was overlayed.*
5. *The representation of each object was abstracted from its actual form. Iterations explored how its visual representations could emphasize the overall "lifespan" of an object or the individual events over time.*

Continuing Development

As a side project, I continue to develop the live site. The current iteration uses only a small subset of objects from the entire collection. However, I continue to work with the Cooper Hewitt API methods so eventually the project will incorporate all objects available.

Furthermore, I continue to develop the Aggregate View on the live site. I've explored other design iterations that attempt to capture a year-based focus for the collection.

```
//Cooper Hewitt API token
var token = "2907bb23a319de02d7174829a85eef94";
var objectsTotal = 100;
var startYear = "1960";

var allObjectsDataset = [];



/* ------ QUERY COLLECTION FOR OBJECTS THAT HAVE IMAGES AND WERE CREATED

/* ------ AJAX REQUEST - returns JSON ------ */

var urlObjects ='https://api.collection.cooperhewitt.org/rest/?method=co

$.ajax({
        url: urlObjects,
success: function (response) {

console.log("AJAX requrest successful");

var objResponse = response; // variable to hold the response
var allObjects = objResponse.objects;

for (i = 0; i < allObjects.length; i++) {
allObjectsDataset.push(allObjects[i]);
console.log(allObjects[i]);
done();
                }
        }
}); // end main AJAX request
```

Code snippet referencing the Cooper Hewitt API in an attempt to gather the metadata of all objects meeting a set of criteria.