## drawTalk

Designing and developing speech driven design software for users of all abilities

### question:
Can we develop inclusive and accessible design software to benefit a wider range of users without sacrificing the purpose, use and integrity of the application?

### idea:
Develop an application built for users who have a disability first, then expand features to make it useful for the ‘general user’

### method:

<table>
<thead>
<tr>
<th>DESIGN</th>
<th>DEVELOPMENT</th>
<th>EXPERIENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determined necessary elements for basic graphic design</td>
<td>Used Google’s Speech-to-Text API to capture user input</td>
<td>Users are able to control the application entirely by speech</td>
</tr>
<tr>
<td>Mapped words to drawing actions/commands</td>
<td>Rendered results of captured input on a SVG canvas</td>
<td>Users can draw various shapes, move them, change their colors, add text, add images, define a grid, and export their designs</td>
</tr>
</tbody>
</table>

### outcome, next steps:

| Successful answer to research question | Focus on accessibility efforts at work | Design a procedure for developing inclusive software at a large scale, propose this to other companies |

| Stay involved in a11y (accessibility) community | Further accessibility features like sound recognition |
drawTalk

Designing and developing speech driven design software for users of all abilities

ZOE INGRAM • UCLA IRI • MAY 2018
problem:

Many companies do not incorporate accessibility needs into the foundation of their applications during the design and initial development stages, therefore many accessibility solutions tend to be hacky - and don’t always work.

idea:

Develop an application built for users who have a disability first, then expand features to make it useful for the “general user.”

This application would be a web based design tool.
question:

Can we develop inclusive and accessible design software to benefit a wider range of users without sacrificing the purpose, use and integrity of the application?
**approach:**

Broke away from the usual practice of trying to develop for the optimal user: *young, tech-fluent, web + software intuitive*

Incorporated standard accessibility solutions according to W3 guidelines

Built my own set of accessibility guidelines based on interviews with accessibility engineers, designers, and people of many different abilities

**influence:**

Accessibility engineering, design software, principals of universal design, inclusive development, empathetic engineering
## methods:

<table>
<thead>
<tr>
<th>DESIGN</th>
<th>Determined essential shapes, actions, and behaviors necessary for basic graphic design</th>
<th>Mapped words and simple phrases to shapes, actions and behaviors, as well as basic UI commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVELOPMENT</td>
<td>Used Google’s Speech-to-Text API to capture user input</td>
<td>Rendered results of captured input on a SVG canvas</td>
</tr>
<tr>
<td>EXPERIENCE</td>
<td>Users are able to login, navigate, and control the application entirely by speech</td>
<td>Users can draw various shapes, move them, change their colors, add text, add images, define a grid, and export their designs</td>
</tr>
</tbody>
</table>
obstacles:

DEVELOPMENT TIME

More features
Cleaner code
Better implementation

SPEECH RECOGNITION SOFTWARE

Super finicky
Requires “clean” speech

USER TESTING

Many more user groups to test
demo
outcome:

We can develop inclusive and accessible design software to benefit a wider range of users without sacrificing the application.
next steps:

Focus on accessibility efforts at work
Stay involved in a11y (accessibility) community

Design a procedure for developing inclusive software at a large scale, propose this to other companies

Further accessibility features like sound recognition