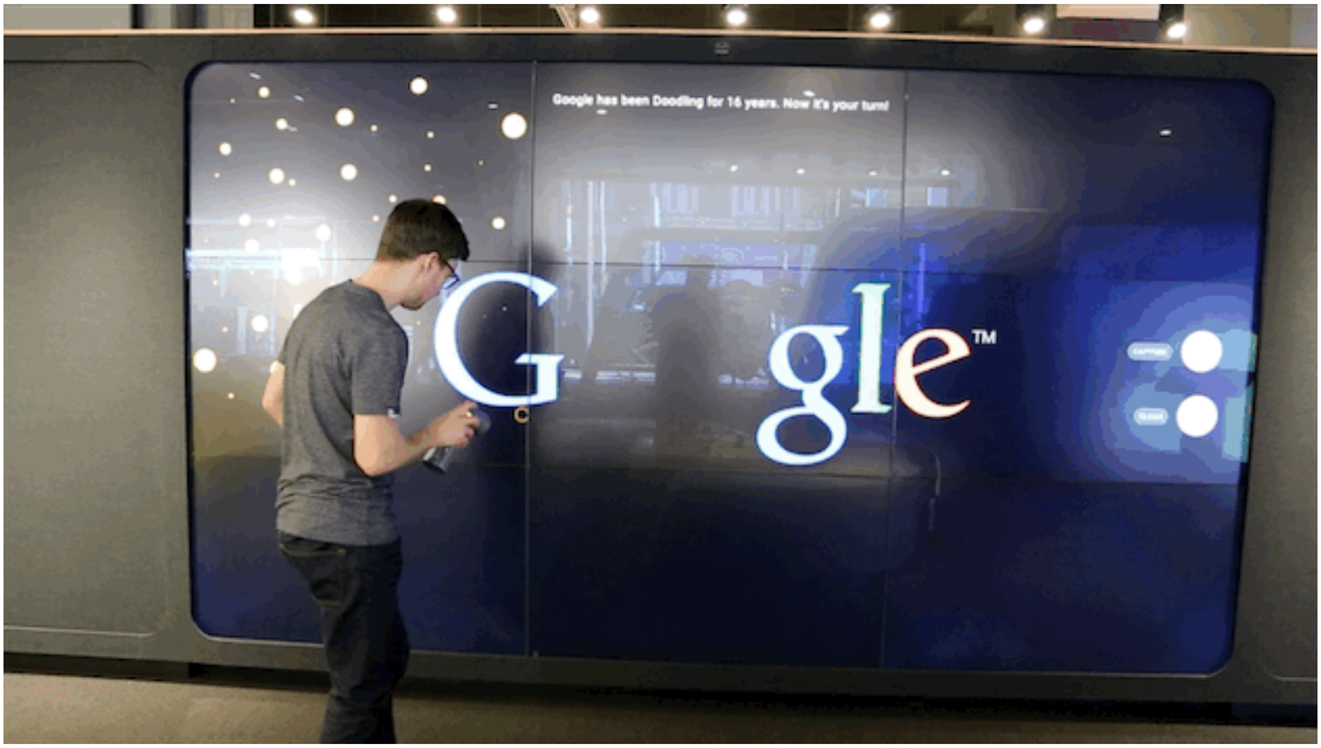


link to gif

Limit yourself to 2 hours. There's no single "right" answer, but many valid approaches that can be taken. This exercise is more about how you would approach a problem like this and not having a ready-to-build system.

Challenge: Google wants a physical manifestation of Google's famous daily Doodle.

The idea is to let visitors create art on the Google logo with a digital graffiti experience. The experience will have physical cans that guests can pick up and start spraying with. There's also a sharing component - users can record their doodle and get a link to a shareable gif of their digital art.

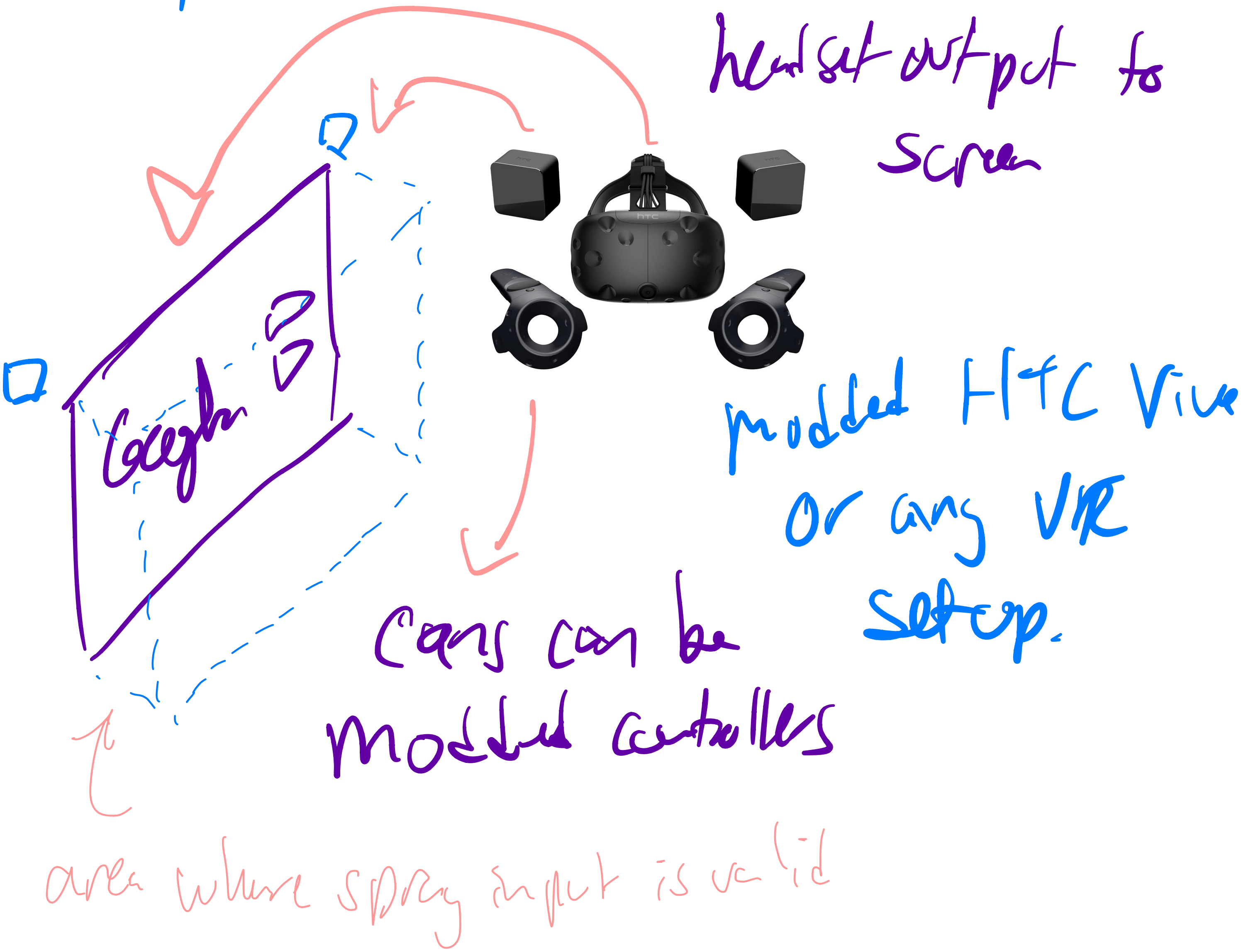
The high level user flow / system flow is:

1. System has an idle state showing pre-generated content
2. User picks up a can (there are 2 total) and begins to draw by pointing it at the screen and pressing a physical button on the can.
3. The experience requires the cans to be tracked in three dimensional space (in front of the screens)
4. The closer they get to the screen the finer their spray stroke width, the further they are, the wider the stroke becomes.
5. The application running on the screens should seamlessly draw a users strokes as they paint with the cans

Given that rough outline, can you talk through what this system might look like? What is your approach? What would you prototype—What are your initial thoughts on tracking the cans in 3D space? What would be your approach to building the front-end application? What are the components of the entire system, how are they connected, how do they interact?
What other team members / what skill-sets would you like to have on your team? Make any assumptions that you need to, such as "There is an API that does X."

Main problems

- How to track cans in 3D space?


headset output to screen
Google
modded HTC Vive or any VR setup.
Cans can be modded controllers
area where spray input is valid

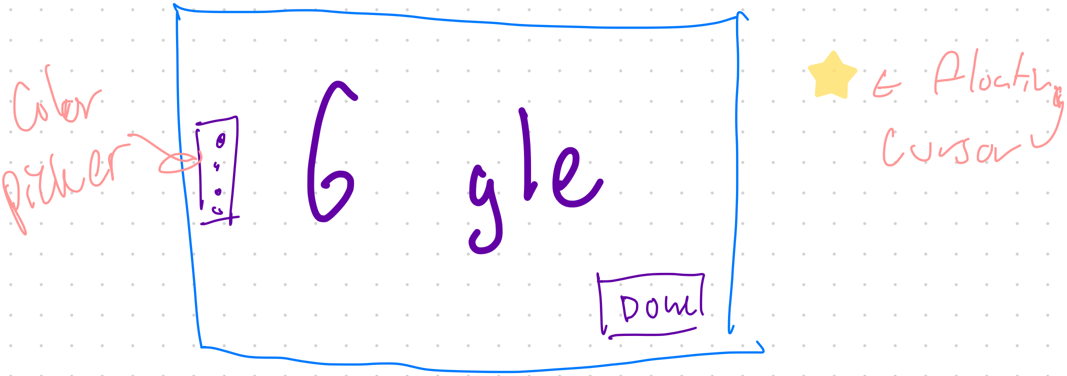
– I actually am not intimate w/ HTC Vive, however I have worked in VR. Most likely using Unity because it has easy VR integration

3D Tracking

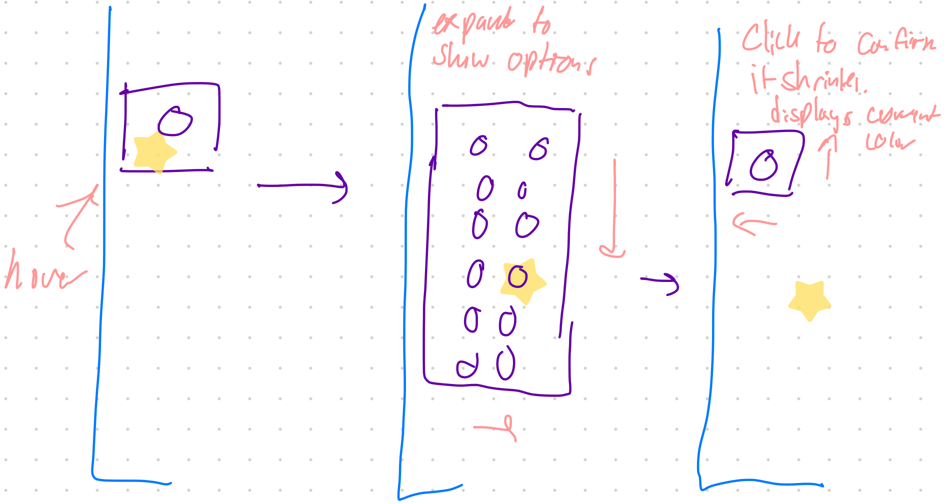
- Determine active area in the bounds of the trackers.
- determine the position of the real tv screen (real canvas) in the virtual space (virtual canvas). They will be calibrated to be the same. This is our canvas that displays the doodles.
- application should be able to determine distance between the real canvas and controller spray can since it can track the distance between the virtual canvas and controllers position in virtual space. This allows us to determine the stroke width of the spray.
- A simple circle can be used as a reticle to show where the spray paint will show up. The size of the circle can be dynamic to show the stroke width

Front end app.

- My gut tells me to use unity as I am very familiar with it but 1 am open to other engines.
- what will be displayed on the screen is a fixed camera view of the virtual canvas. The background is black with Google's logo sans o.



Color picker

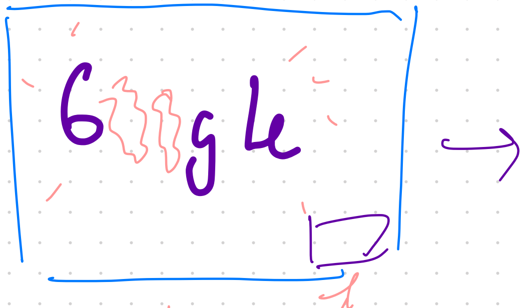


Drawing

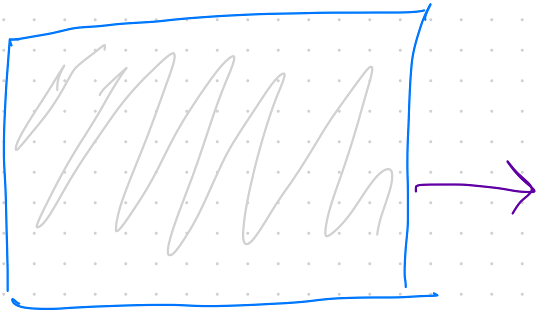
- the users drawing input is stored as an object of Point. Point has 4 properties: x coordinate, y coordinate, stroke width, and color. They're saved in a dynamic array such that the drawing can be played back for the gif output. There probably is an API to save the play back to gif.
- xy coordinates and color are self explanatory, stroke width is determined by the distance between the virtual canvas and controller
- there could be a bit of a dynamic stroke. Like the stroke has abit of vibration to make it more lively

Idk, like a wiggle.

Complete flow

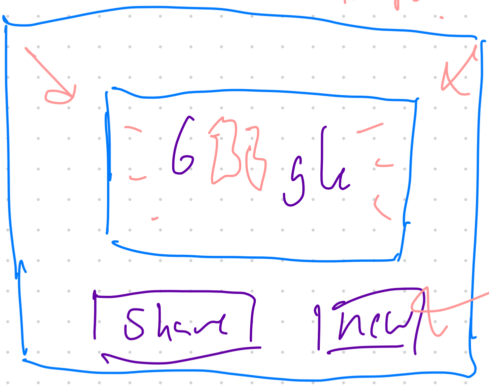


use the cam to "click" the done button



flash of white resembles a picture being taken

image shrinks to a photo snapshot



new doodle.

Users can click on share to get a gif

Team Compisition

- developer of user interface
- developer of the Point/Drawing system
- developer of getting HTC vive to be set up with the application and calibrating within the app
- fabricator for the spray paint cans and the holder for cans
- fabricator for installation