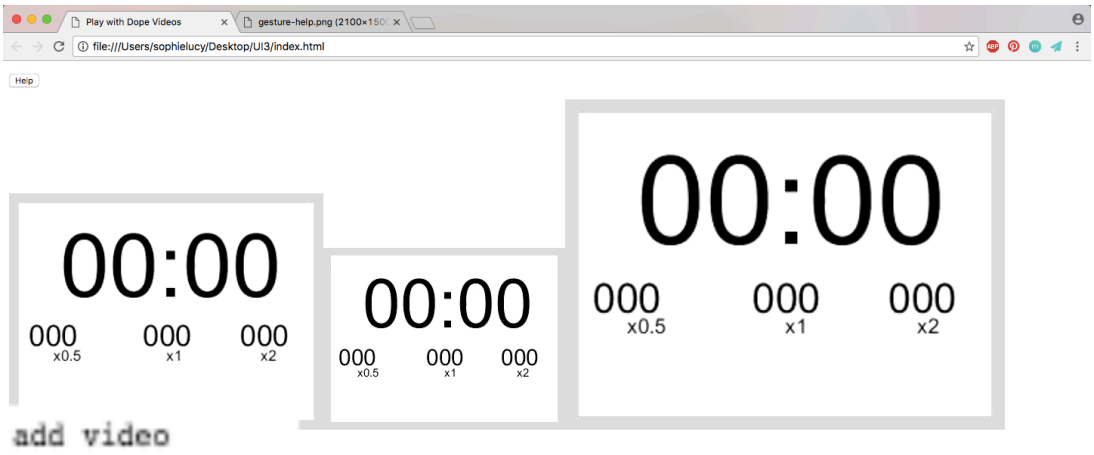
COMS W4170
Sophie Lucy
sjl2185

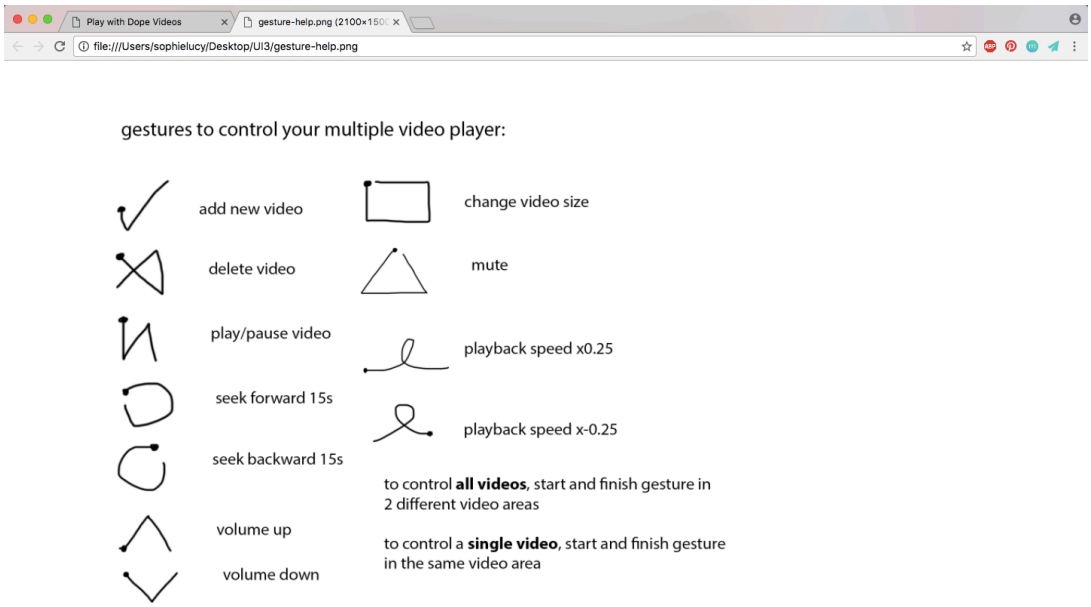
What the application does

The application allows the user to build a custom video player experience where they are able to add multiple videos and control their experience through single-stroke gestures.



How to use

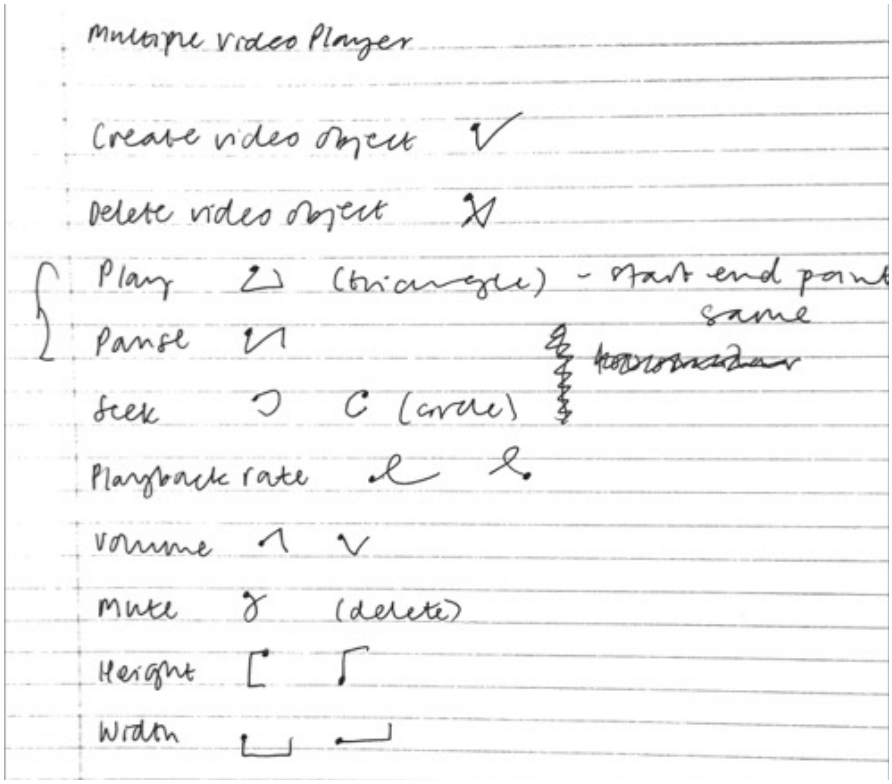
To use the application, the user opens the index.html file from the folder into their browser. The application is extremely minimal (adhering to the “aesthetics and minimalist design” heuristic). The “Help” button at the top of the page allows the user to immediately read about and understand how the video player works and following each gesture, the feedback given on the screen allows them to see the status of the video player in terms of the gestures that it recognizes and learn how the video player behaves.



Approach and Design Process

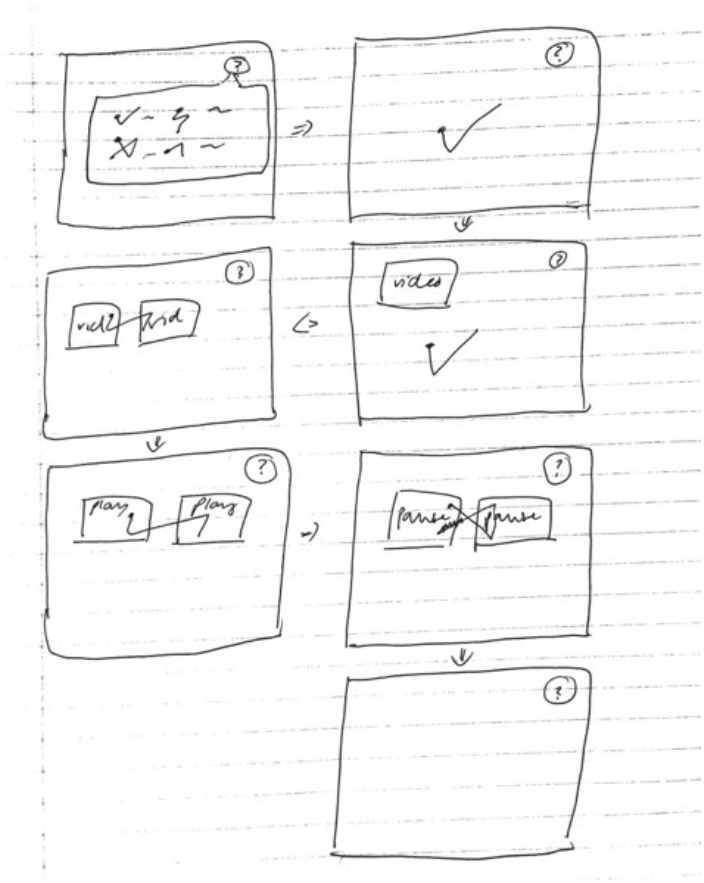
1- I looked at the existing gestures to see which ones would match each action required by the video player.

2- I drew out new gestures that could be matched with the remaining actions



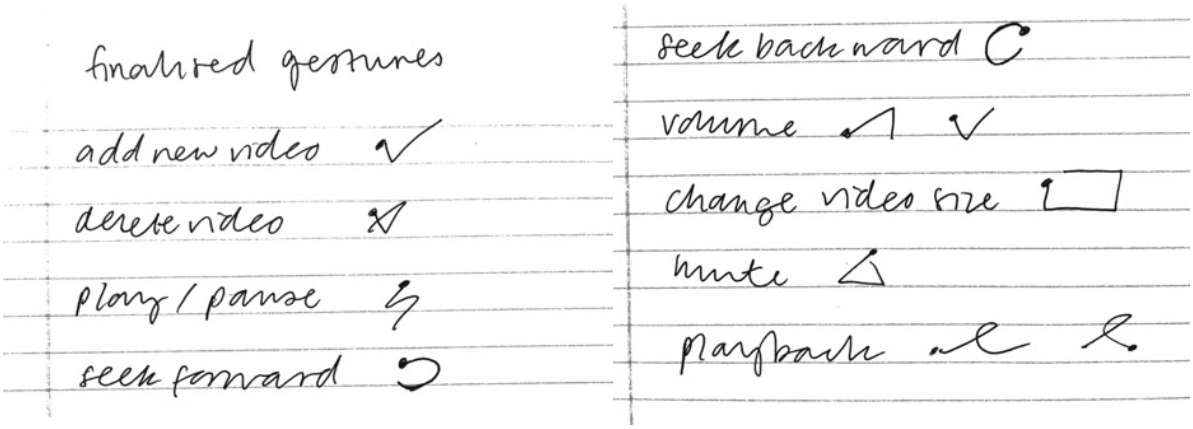
3- I played around with the recognizer to see how the gestures felt and found that there were some gestures that did not work well. For example a single horizontal stroke was often mistaken for a *check mark*, a *right square bracket* or *left curly brace* instead of the new gesture I had added. From testing out the different gestures I was able to narrow down the gestures I could use for my video player. Some gestures also made it difficult for the user to control all the videos simultaneously because the shape of the gesture would not allow for its start and end point positioned in different video areas. For example the *triangle* gesture for play would make it difficult to start and end in different video areas so I decided to combine both *play* and *pause* into the same gesture.

4- I made some sketches and a storyboard of how I would want the multiple video player to work and controlled by the user.



5- Using the existing recognizer, I added a line of code so that the gesture was outputted and could be added to the list of gestures.

6- While testing the gestures I had compiled, I found that there were some gestures that were being mistaken for others. For example, the *mute* gesture was being mistaken for the *playback slower* gesture.



7- As we are not able to use any buttons, sliders, keyboard etc, I wanted the position of the gesture to indicate whether the user would like to control all the videos or just a specific video. If the start and end point of the gesture are in the same area then only 1 video is being controlled. Alternatively, if the start and end point of the gesture are in different areas then all videos will be controlled.

Explanation of Gesture Design Choices

Gesture	Explanation
create video	The gesture to create a new video object is a check mark because of the ease in making the gesture which would make adding video extremely quick and easy.
delete video	The gesture makes an x mark that signifies crossing out the video/videos and therefore deleting it/them from the page which relates to the “match between system and real world” heuristic.
play/pause	I used the same gesture for both *play* and *pause* because of the “recognition rather than recall” heuristic as there are less gestures needed. This is also reminiscent of the controls in existing video players as clicking on a paused video will make it play and clicking on a video that is playing will pause it. This also allows users to quickly go from the play state to the pause stage very easily if they make the mistake which corresponds to the “user control and freedom” heuristic.

Gesture	Explanation
seek	In apps such as the Podcast app within the iPhone their seek button skips to 15 seconds ahead or behind the current position in the video. The gesture mirrors the icon used in the Podcast app and also skips to 15 seconds ahead when the gesture for seek forward is made and 15 seconds backwards when the seek backwards gesture is made. This relates to the “match between system and real world” heuristic as well as the “recognition rather than recall” heuristic as the user doesn’t need to think about how far ahead they want to skip to— they already have the option of 15 seconds backwards or forwards.
volume	The volume gesture reflects the icons often used for increasing and decreasing the volume. I chose the gesture also for the speed and ease in making them as well. The volume can be increased and decreased in increments/decrements of 0.2. The volume cannot surpass 1.0 and cannot go beneath 0.0 (mute) as neither would be beneficial to the user experience. These design choices relate to the “match between system and real world” heuristic and the “recognition rather than recall” heuristic.
playback rate	The playback faster and slower function adds and subtracts speeds by 0.25. I chose 0.25 because YouTube videos as well as other video players allow their users to speed up and down by 0.25. I do not allow the user to exceed 2 times the normal speed or fall under 0.25. Because the short length of the video, if the speed exceeds 2 times the normal speed, the video would be too fast and therefore no longer valuable to the user and if the speed is slower than 0.25 the normal speed, content would no longer be understandable. These design choices are reflective of the “match between system and real world” heuristic.

Gesture	Explanation
change height and width of video	This gesture allows the user to only increase the size of the video proportionally. I decided not to let the user determine height and width separately because user experience will be compromised if the video's dimensions are not proportional. The size can only be increased up until a certain size to simplify the ability to change the size of the video so that the user does not need to remember another gesture for decreasing the size of the video. I chose these sizes in video because the user experience would again be compromised if the size of the video exceeded the size of a standard laptop screen. The design choices were made to simplify the user experience.
mute	The mute gesture was originally too similar to one of the seek gestures which posed a problem as the actions are so different and the ability to recover from skipping forward rather than mute would be annoying for the user. The choice was made adhering to the "error prevention" heuristic.

I confined the multi video player to a **single page** as I envision the use case being that the user is able to enjoy multiple videos at the same time without having to scroll down the page. Those which are below the fold are not visible and therefore not enhancing the experience of the user.

For help in using the multiple video player, I included a **help button** at the top of the page that allows the user to easily access documentation that instructs them how to use the player. The button opens up a new window and therefore does not interfere with the experience and present state of the player if the user wants to look at **documentation** while using the video player. This addresses the "help and documentation" heuristic.

While using the video player, after each gesture text appears to show the user the gesture that was detected by the recognizer. If the gesture was unclear or misread then the user will be able to quickly see and fix the mistake. This addresses the "visibility of system status" and "help users recognize, diagnose and recover from errors" heuristics.