Reflection and Synthesis -- Code Review
Firecracker
Joey Maalouf, Tom Heale, Keenan Zucker

## Feedback and decisions

Based upon your notes from the technical review, synthesize the feedback you received addressing your key questions. How do you plan to incorporate it going forward? What new questions did you generate?

During our code review, our team received helpful feedback in both the technical aspects and with a live demo. We reviewed our code's current architecture, and most of that ran smoothly. We asked for suggestions regarding a long list of if statements that we had. It started with only a few if statements, but then we kept adding more and more features and more if statements as a result.  The feedback made us realize that a dictionary would be more effective.

We also asked for coding suggestions on aspects and features that we hadn't implemented yet. One was a feature that would bring all widgets in front of the current windows to see the time, weather, etc without minimizing the current windows. Another group was also using PyGTK for their project, and told us there is a built in function into PyGTK for that purpose, which was quite helpful.

Finally, we had the audience clone into our repository and test out the Graphical User Interface for making config files for widgets. We wanted to see how easy it was to run, if there were any major problems, and play test to see what they could create. It was a very rewarding exercise for us because we not only got to see what features we lacked or gave errors, but we got to see how people who hadn't been involved in the programming of the project interact with it. We now have a list of improvements we want to implement moving forward.

## Review process reflection

How did the review go? Did you get answers to your key questions? Did you provide too much/too little context for your audience? Did you stick closely to your planned agenda, or did you discover new things during the discussion that made you change your plans? What could you do next time to have an even more effective technical review?

We thought overall the review went pretty well. Since the audience was the same as before, they already had context into what we were making. We also sent out some reading ahead of time, which we think helped the audience understand some of our

code structure. We also learned from last time that extending the displays using the projector is difficult to manage, so we duplicated the displays so that the projector showed the same thing as the laptop.

We learned last review that an interactive presentation is more effective, which led us to having the audience clone into our repo. This was definitely a good decision, but we also asked the audience to write down comments, suggestions, and confusions on paper about our project while they tested our program and GUI. Unfortunately, this part of the review didn't go great, as nobody really wrote much on the paper since they were playing with the code and widgets the whole time. We did ask for verbal suggestions later in the review, but I think some feedback could have been lost between the time of testing and then talking about it. Next time, we should maybe split people into group and use the whiteboards to write down feedback, and stress more about writing down the ideas rather than waiting to say them verbally.