Joey Maalouf, Tom Heale, Keenan Zucker

FIrecracker

1. **The Big Idea:** What is the main idea of your project? What topics will you explore and what will you generate? What is your **minimum viable product**? What is a **stretch goal**?

The main idea of our project is to create a desktop widget for Ubuntu. It will have a transparent window with several widgets, such as displaying time, weather, a calendar, as well as possibly computer information like RAM usage, etc. The minimum viable product is a window displaying a couple widgets that is run through the terminal. A stretch goal will have it run automatically upon startup of Ubuntu and be able to be manipulated with right clicking / dragging and remembering those settings. One of the biggest features of this widget system will be the customizability; users will be able to create their own configuration files to tweak the details of the widgets.

2. **Learning Goals:** Since this is a team project, you may want to articulate both shared and individual learning goals.

I think we all want to learn about creating widgets that are started automatically and can run within a transparent window. Moreover, I think we all have a desire to improve our design and interface skills by making a widget that is simple and pretty! We would also like to learn more about the gtk library that we are planning on using, since none of us have any experience with it.

3. **Implementation Plan:** this will probably be pretty vague initially. Perhaps at this early juncture you will have identified a library or a framework that you think will be useful for your project**.** If you don't have any idea how you will implement your project, provide a rough plan for how you will determine this information.

At the moment, we are leaning towards using PyGTK for the graphics implementation. We already have configuration file parsing completed, and we will begin working on the window widgets next. Initially, we will create basic windows and as we progress, we will add more tweakable features. As we progress on making widgets, we will work hard to design them in a simple and beautiful way. Depending on how difficult it is to make the transparent windows and widgets, we can adjust how many widgets we want to make based on how much time we have. If we figure out the setup and background more quickly, then we can spend more time working on different types of widgets and features.

4. **Project schedule:** You have 6 weeks (roughly) to finish the project. Sketch out a rough schedule for completing the project. Depending on your project, you may be able to do this in great specificity

or you may only be able to give a broad outline. Additionally, longer projects come with increased uncertainty, and this schedule will likely need to be refined along the way.

Week 1 - Display information on transparent figure window & setup config files
Week 2 - Add to displayed information & finalize aesthetics of windows
Week 3 - Implement config files into display options & attempt to initialize windows in set positions
Week 4 - Play catch-up & add to config and window options (add widgets if time)
Week 5 - Attempt to launch app at startup (add widgets if time)
Week 6 - Finalize Project (add widgets if time)

5. **Collaboration plan:** How do you plan to collaborate with your teammates on this project? Will you split tasks up, complete them independently, and then integrate? Will you pair program the entire thing? Make sure to articulate your plan for successfully working together as a team. This might also include information about any software development methodologies you plan to use (e.g. agile development). Make sure to make clear why you are choosing this particular organizational structure.

Since there are three of us, pair programming is not going to be the most effective way to complete this entire project. We think we should do some, especially for parts that are unfamiliar, so that we all learn new strategies and practices. I think after we get the basic structure and background figured out, we could probably work more individually into developing certain widgets, like a clock, calendar, weather, cpu statistics, etc. This way, it should be easier to integrate, since it will simply be a matter of adding and positioning that additional widget to the main background.

6. **Risks:** What do you view as the biggest risks to the success of this project?

Unfamiliarity with PyGTK, and possibly the fact that we don't know how many configurable features we'll have. Also, we don't really have much knowledge of getting programs to run upon startup and running in the background. It might take a lot of processing power, which we will hopefully try to avoid.

7. **Additional Course Content:** What are some topics that we might cover in class that you think would be especially helpful for your project?

Graphics modules. PyGame vs. TkInter vs. PyGTK vs. ??? Transparent windows?! Configuration Files