Bluestamp Engineering Build Plan

Name: Omar Garcia
Location: Houston
Instructor: Rachel Alexander
Main Project: #15 A watch that can interface with computers or smart phones.

Major Steps: Before Coding

1. Verify that all parts have arrived.
2. If have not already, make a Pebble account. This is needed for logging in the Pebble app as well as Cloudpebble (more on Cloudpebble that later).
3. Turn on and set up the watch. Also install the app on iOS or Android since the watch requires the Pebble app to turn it on right out of the box.
4. Install the necessary software on both the computer and smartphone that the watch will be paired with. This will either be a development program installed on the computer or programming online in the cloud. Foroperating systems, an SDK can be installed. This isn't available for Windows, so Cloudpebble will be used which requires a pebble account and you can program in the cloud, no downloading required.
5. For smartphones, you will need to go into your phone settings if on iOS and for Android go to developer in the Pebble app. In iOS turn on developer mode and return to the app. In both platforms you go to the developer option and enable the developer connection. This is to upload apps you code from Cloudpebble.
6. Test the connections between the Pebble smartwatch, smartphone, and computer to make sure they are working properly. This is for uploading apps you make to the watch.
7. Begin coding the app and making sure it works properly.
8. Throughout the project, I will document my project on the blog that will include various steps of app development.

Ideas for Apps:

- Chess clock- one person monitors the time for both players in a chess game which can be a player or observer. The clock will have be from 5 minutes to an hour in 5 minute intervals.
- Bing watch face- A regularly updated photo (weekly or daily) that comes from the daily picture on Bing.com or one from 500 px
- Music changer- An app that uses the 3D sensors on the watch to change the music based on motion. (This idea likely has other applications especially with controlling a computer)
- Wireless Mouse- Using the 3D sensors on the watch, the watch will serve a mouse. If the watch moves up, then the cursor on the computer will move up as well.

- Skeeball- Using the 3D sensors on the watch, a user can swing the watch and the movement will be mirrored on the screen.
- Fact of the day- A random fact everyday

Major Steps: Coding (Chess Clock)

It is important to note that for this project, the programmer knows C since Pebble uses C-based apps and watchfaces.

9. Once you have set up CloudPebble and the connection between the watch, phone, and computer, begin programming with Hello World. This option can be found under create and choosing Hello World under template. Don't forget to name it. The code won't show up right away. Instead you will need to select helloworld.c to show it on the sidebar.
10. Once you have modified the code, you can compile it which means to run the code. This is done using the play button on the right side of the screen. From there CloudPebble saves and compiles the code to see if it works. You will either get Successful or Failed. If successful proceed to step 10.

- If a build of the app has failed, you can click Build Log to see the errors in the code. Correct the code and repeat this step.

11. If the code has been successful, you can install the app onto your watch. First you want to make sure that your phone turns up on the list of devices which requires the Pebble app on your phone to be in developer mode. If you don't see it, go back to the develop option in the Pebble app and check the IP address of your phone. Type that IP address onto Cloudpebble after clicking the option to type the phone's IP address manually.
12. Since this project will build upon an existing timer app to make a chess clock, find the code for a timer app. I found mine here: http://github.com/mwylde/pebble-timer
13. Copy and Paste that code onto Cloudpebble as a new project. Once that it done, you can begin modifying the code.
14. For my code I copied some functions in order for the computer to know that there are two timers.
15. In order to help developers, Pebble provides some help on its website showing how to program an image into a watchface for example.

My code is here: https://github.com/31422