Creating a Web Application to Handle Battery Scheduling

Tiffany Valitis & Raychel Beasley

Introduction

We seek to create a web application that will handle the statuses of all of the robots within the BWI Project. It will display the robots' tasks, battery life, a brief description of its capabilities, and link to a Google Calendar that will allow users to schedule when the robots should be on or off. In addition, there will be a feature that alerts us when a robot is low on battery and should be plugged in, and which robot should be turned on next. We will also include instructions on how to turn on and off the robots correctly.

Proposed Approach

Our plan is to write a web application in Python that essentially serves as a GUI for all of the robots. We will write a ROS node to subscribe to the right topics, in order to collect information for our site. The node will then send the information to our application, which will present it in an aesthetically pleasing manner. The information will include topics such as battery life, whether the robot is on or off, if it is charging, if it is busy and if so, what task(s) it is completing, and approximately how much longer the robot will last until it needs to be recharged. We are aware that we will have to handle multiple, rather than a single, ROS Master and interface with ROS in order to receive the information we need. There is an existing email service that we will utilize to add even more functionality to our web application. In addition, we may have to write nodes that publish about information that there is not currently a node for. Ideally, we will also interface with Google Calendars in order to have a more dynamic scheduling system, but regardless, there will be a schedule for regular school weeks that will notify us when the robots should be turned on/off. If there is a FRI page to link to we will; however, there will be a URL provided to us by one of the postdoctoral researchers of the BWI project.

Prior Experience

- Proficiency in Java
- Some experience creating GUIs in Java and C++
- Some experience writing ROS Nodes

Related Work in Robotics

- Sagehorn, Walter, and Jonathan Butler. "lg2s4tg/ROS-Web-Interaction_web." GitHub. N.p., 14 May 2016. Web. 31 Mar. 2017.
- Veloso, Manuela, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. CoBots: Robust Symbiotic Autonomous Mobile Service Robots. Proc. of Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires. N.p., n.d. Web.
- Sagehorn, Walter, and Jonathan Butler. Robot Interaction and Display via The Internet. N.p., n.d. Web.
- Sagehorn, Walter. "Utexas-bwi/bwi_common." GitHub. Utexas-bwi, 24 Mar. 2017. Web. 31 Mar. 2017.

- Utexas-bwi. "Utexas-bwi/segbot." GitHub. Utexas-bwi, 04 June 2016. Web. 31 Mar. 2017.

Proposed Evaluation

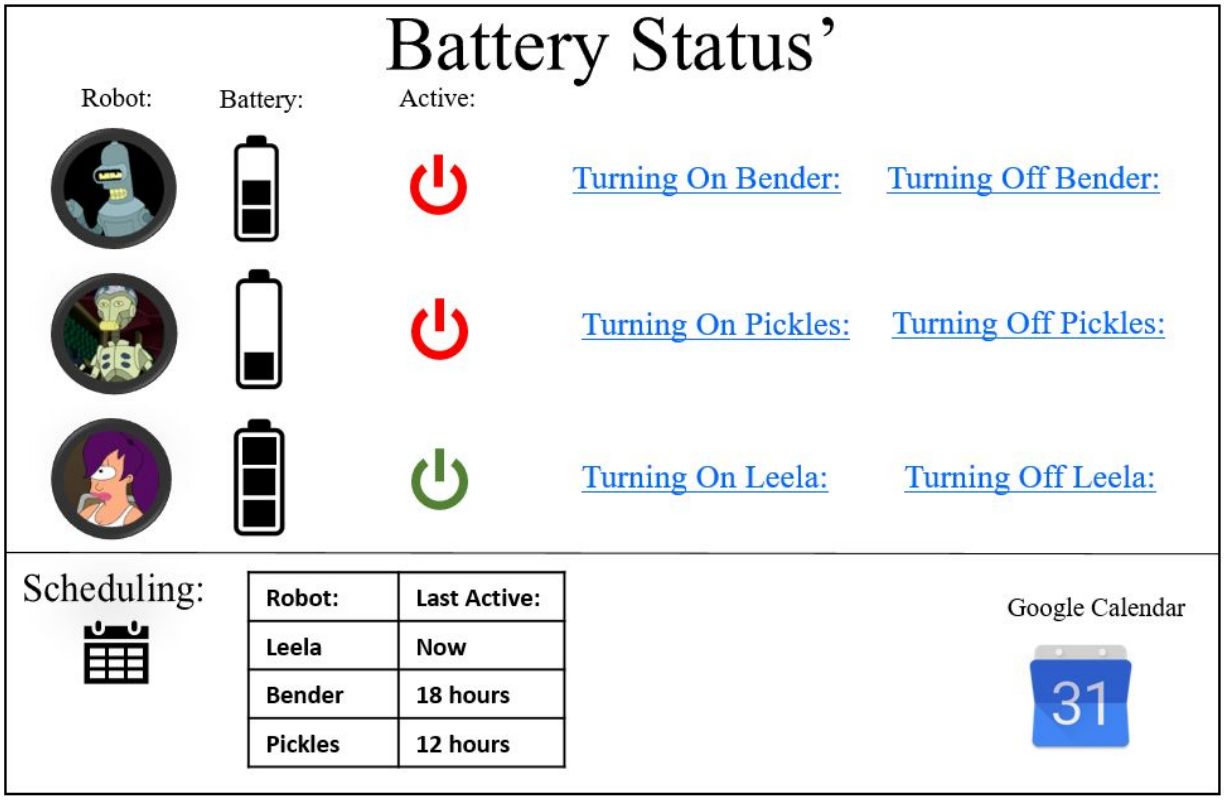
In order to test our final project, we will conduct a few experiments.

1. Turn off all the robots and check the web application to see if it presents the correct information.
2. Turn on all the robots and check again.
3. Try multiple combinations of some on and some off and check the web application.
4. During a regular day (when people are using the robots, and they will be in various states), we will record what states the robots are in and compare it to our web application.
5. Finally, we will ask some of our peers and mentors to try using our web application and have them give us feedback.

Anticipated End Result

We hope to create a fully functional web application that will handle and display the battery life, schedule, and additional data of the multiple robots in the BWI Project. Our application will allow users and other people involved in the project to easily access important status information about the robots via an intuitive graphical user interface. Ideally, it will also interface with Google Calendars to allow for a more dynamic scheduling system that will enable flexibility in event scheduling.

Mock-Up of Web Application



Schedule:	**Task:**
Saturday, 3/25	Write rough draft of proposal
Monday, 3/27	Refine proposal with TAs/mentors
Friday, 3/31	Final editing of proposal
Monday 4/3	Final checking & submission of proposal; outline program
Wednesday, 4/5	Code method of receiving and processing data
Friday, 4/7	Code basic Web App & pub/sub (battery info)
Monday, 4/10	Continue coding basic Web App & testing it
Thursday, 4/13	Add image support to Web App & test
Wednesday, 4/19	Improve App aesthetic
Saturday, 4/22	Outline scheduling pub/sub
Monday, 4/24	Code scheduling & test
Saturday, 4/29	Write final paper
Monday, 5/1	Conduct final evaluation & record video
Friday, 5/5	Continue writing final paper & edit video
Sunday, 5/7	Submit project