Team Agreement

Project: Robotic Transplanter for Hydroponic Grown Lettuce

Workflow

The 10_6 Commandments of Development Standards

1. Thy branches shall almost always reference a github issue
2. New branches shall be created with the following naming convention:
    a. <initials>-<name-of-feature> e.g. sb-development-standards
3. Thou shalt receive the approval of at least one teammate before merging
4. Thou shalt not push to main
5. Thou shalt use the proper labels to describe the feature in progress
6. Thou shalt not delete thy branches after merging

Feature Development Process

1. Open an issue on github (use the appropriate labels!)
2. If the feature is sufficiently orthogonal, start a new development branch
3. Merge new changes from the main branch into active development branches at least every 72 hours
4. Once the feature appears to solve the issue, open a pull request
5. If at least one other team member signs off on the pull request, the dev branch may be merged. Otherwise, repeat from step 3
6. Close the issue, get some ice cream

Team

Info

Name/Member	Role	Contact	Expected Response Time
Mira Welner	Project manager	mewelner@ucdavis.edu	< 12hrs
Scott Ballinger	Communication and contact manager	sballinger@ucdavis.edu	< 24hrs
Liam Carr	Minutes manager	ltcarr@ucdavis.edu	< 24hrs
Martin Orosa	Revision control manager	morosa@ucdavis.edu	< 24hrs

Team Meeting Times

In order of preference:

1. Friday, 1:30–2:30
2. Saturday, 11:30–12:30
3. Sunday, 1:00–2:00

Team Communication Platforms

Contact the communication and contact manager via email: sballinger@ucdavis.edu

Team inter-communication occurs on Discord, and communication with the client is via email

Task Allocation

Preliminary task allocation

Task	Name/Member
Investigate possible frame/structure designs	Martin Orosa
Investigate possible driven mechanisms	Scott Ballinger
Investigate possible microcontrollers and associated control languages	Liam Carr
Survey people to determine possible human interface designs	Mira Welner

What is task completion?

A task can be considered complete when the requirements of the task have been satisfactorily met, as agreed upon by other members of the team.
Improvements upon the original task would be considered new tasks, but if the process of finishing the new tasks breaks the original task, then the original task would no longer be completed.

Code Conventions

Style

TBD pending more detail on language choices from first meeting with client.

Likely going to follow the Google style guides, where applicable.

The language will likely be C or some similar neighboring language, like the Arduino language.

Documentation

Documentation will be created and maintained as the project progresses.

Initial documentation will serve as a guide for developing the hardware and software, and documentation will be updated as we release that design changes need to be made

Team Agreement

- Team meeting time
- Team communication platform/protocol
- How to allocate tasks?
- Roles and role rotations (if chosen to)
- What does it mean for a task to be complete?
  - Ex: pushed to github without conflict
  - Ex: deadline
- Team Development Standard (example on canvas)
  - Workflow, style, etc.
- Start working on it during the breakout session
- Due this Sat. 11:59pm

Deliverable:

Robotic arm with attachments as needed, work surface where operator can place a full raft and empty raft, and controller. Basically the complete package customized for a particular system (subject to discussion) designed and build with flexibility to allow it to be modified for other systems that exist.

"Todo" list
https://docs.google.com/spreadsheets/d/1tb-ENvJszuwXAJhiivMoHDwI5SLr1VnE3hpg8yDL0gY/edit#gid=555458189

Example of fully automated lettuce farm
https://www.youtube.com/watch?v=q9OuC6_Tc8g

Attachment to our project, probably relevant
https://drive.google.com/file/d/1u0UsIFcKr7ICkoyBCivw6sfUZ3OYgkR3/view