

Project 3 - Stage 2

Try a bit of leJOS' implementation of subsumption architecture/behavior-based robotics.

Step 1

- Implement and run `BehaviorForward.java`, `BehaviorProximity.java`, and `TwoBehaviors.java`. Run `TwoBehaviors.nxj` on your team's robot.
 - Include all present team members' names in an `@author` comment in each class's javadoc comment!
 - Feel free to experiment with different values of the named constants as desired.
- Submit your resulting three files `BehaviorForward.java`, `BehaviorProximity.java`, and `TwoBehaviors.java` with a homework number of 32.

Step 2

Say that we were interested in measured trials on an obstacle course in BSS 315,

...this time with robots **using behavior-based robotics**,

...AND for which we would like them to make it from the beginning to the end of the course with **AS LITTLE human intervention** as possible.

(How would the times compare? Obviously the number of button pushes would be less -- would the number of collisions be less? The number of catastrophic falls?)

- Decide on and implement at least one additional behavior to **add** to those you've already implemented. Give it/each an appropriate name.
- Implement a "new" application for each additional behavior, combining it with others -- `ThreeBehaviors.java`, `FourBehaviors.java`, etc.
 - of course, always include appropriate javadoc comments for EVERY class and EVERY method! including all present team members' names.
 - You may use whatever sensors you would like.
 - You may modify your robot as desired.
 - Be sure to carefully consider: what **PRIORITY** should your new behavior be, compared to the others? You'll want to order the arbitrator's behavior array appropriately!
- NOTE: it is POSSIBLE that the "new" behaviors MAY be posted, because PART of the advantage to this approach is how relatively reasonable it can be for people to SHARE and REUSE and EXPERIMENT with different behaviors (and combinations of behaviors) in this architecture.
- Submit your resulting `.java` files for your resulting behaviors (at least three -- `BehaviorForward.java`, `BehaviorProximity.java`, and your at-least-one-additional behavior's `.java` file) AND an application setting up an `Arbitrator` and using those behaviors, with a homework number, again, of **32** (don't worry, they don't overwrite previous submissions!).

AND -- yes, I hope we'll run measured trials of the teams' favorite behavior-combinations.