

Press Choose To Start. . .

*The Continuing Saga
Of
Jeremiah "Kill Killer" Johnson
And Armageddon*

I. Four Happy-Go-Lucky 148 Students Dream About Their Robots "Intelligently" Playing Tag

The initial outlandish goal of our project was for our robots to play a complex game of tag. This goal even after being stripped down to a simpler version of our initial premise was still somewhat unreachable but was successful in short trial runs.

This initial plan made it such that the robots would need to be able to distinguish between three different colors of light. We found that this was a wholly impossible plan.

We found that by properly shielding a robot from its own light we could have each robot have the same light "color" and thus two "colors" would be enough. This is when we devised our next failure to use horizontally and vertically polarized light as different colors. This initially seemed like a promising plan and in preliminary tests with small polarizing squares it seemed feasible. It was with this false sense of hope that we moved from the planning process of our project to implementation.

II. "Wait, This Is Actually Supposed To Work? NOTHING Works In CS 148!" - Dan Morris

IIA. Lights of All Shapes and Sizes

For tag to work in any respect each robot had to be able to detect the other. Thus each robot had to have a light source on it. We found that 6" florescent tubes gave the best light output to power consumption ratio. Thus began our ill-fated quest to start florescent tubes off of DC power. However, after multiple calls to Sylvania and Phillips and a discussion with Professor Paterson, we had tried with all our might but were unsuccessful in finding any way make multiple tubes start on DC power. There was a brief glimmer of hope when Mike suggested starting the tubes with a Tesla coil, but we decided against this since we desired not to kill people during our demo and that we didn't have multiple 50-gallon drums of mineral oil.

So we moved to our back-up plan of using AC power. We still opted to go with florescent fixtures since they wouldn't produce too much heat as to disrupt the fine and exacting discrimination of our polarizing materials.

Now it might sound like we were kind of down on polarizing material, but we in-fact spent a significant amount of time attempting to get them to work before we realized that they sucked. We were quite a ways into our project before our sheet of polarizing material arrived in a gratuitously large box (that I was able to carry because I'm so buff). We cut two cylinders horizontally polarized for each robot and one vertically polarized for the safe zone light. (We had completely abandoned the idea of the starting zone by now because we thought it would be too easy to implement.) We wrote some code to use two pairs of oppositely polarized light sensors on the robots where we could switch between the pair that was being used so that we could evaluate the ability of the robots to follow the correct polarization. As you may be able to guess this worked approximately like simulated crap.

Once we came to our senses and realized that polarized light was a lost cause, we threw out the safe zone concept (again it was just too simple to implement) and changed our game.

Light sensing was another issue we had to deal with; however, it was not as much due to our overly ambitious and insane goals. It was more a result of the fact that light sensors work in a somewhat useless fashion at times. It is known to all that light sensors are flaky, but in fact there are very distinct qualities of flakiness that only discriminating tastes can appreciate. After culturing our light sensor assessment skills we were able to rate particular light sensors on their ability to contribute to our crap simulation.

With a select crop of decent light sensors we were able to mount them on our robots and began to use them to guide their movements. We found that it was necessary for our robots to have light sensors in the back so that one robot could “intelligently” run away from the other robot.

IIB. - Our Robots Move Too Fast, So Let's Make Them Really Heavy!

So you may be asking “How were you going to mount florescent tubes, light fixtures, bump sensors (I'll mention that debacle later), 68011's processor boards, motor batteries, and doorbell circuits (again that magic will be analyzed in detail later) on top of your robots?” Our answer “Screw Legos! We want to mount big-ass-pieces-o-wood™ on our robots!” These took on the form of octagonal 1' diagonal pieces of 3/8" plywood. We had some strange ideas about constructing our own bump sensors that we would mount on this rim. This, of course, also was a bad idea, but at least we spent a lot of time thinking it might work. Holes were drilled in the wood to allow the wires to be routed properly from motors and light sensors. Thus our robots didn't look like a bunch of disorganized wires everywhere that were potentially shorting random things out. Or at least this is what was intended.

The other issue was that Lego's are simply not designed to carry the kind of load we put on them. Both robots to varying degrees experienced problems with warped axles and ground gears not to mention the grinding away of the Legos themselves by the axles. This conveniently made our robots pretty much non-drivable for the talent show. It is unfortunate that this caused us to be unable to stun the audience with a live showing of our amazingly strategic and intense game of tag.

The wooden hat was also decided upon for its usefulness for mounting bump sensors on. This brings us to. . .

IIC. Our Robots Are SO Structurally Sound That It's A Good Idea To Ram Them Into Each Other Repeatedly

For robots to tag each other they must be able to detect collisions. Naturally, we traveled down about a hundred more dead-end roads for this portion of our project. When we were

uninitiated snot-nosed robot jocks we had the grand idea that we could build complicated tilt sensors and many other varieties of bump sensors that in the end would be just as crappy as the solution we ended up using. Luckily we managed to nip a few bad ideas in the bud before we spent a lot of time and money on them, but we also cleverly picked out the really frustrating ideas and spent a lot of time on them. We had an idea to use push button switches from Home Depot, but we thought they would work too well and we were going to enjoy spending \$50 on them too much, so out of guilt and humbleness we opted for a more challenging and tortuous path.

We mounted the sensors so that the actual switches were on the top (something that unambiguously didn't work and was to be changed later). For the game it wasn't important to have information about the direction of a tag; we just needed to know when a tag happened. So we decided to wire them all up together. For this system to work it was also necessary to have a continuous surface to bump against. The initial (and longer lived than it should have been) solution was to glue plastic knives to the bump sensors to create a ring.

This system worked in theory, but in practice the brilliant idea of making our robots hit things at full speed was brought to light. We observed that crashing the robots into things made them break and made the sketchily glued knives break off even more. Overjoyed by this knowledge of the Achilles heel of our robots we decided we'd just keep gluing things back on. After a lot of glue and a change of heart we realized the knives had to go, but wait, don't worry they made a brief reappearance later before being trashed for a second time.

We decided that a more rigid solution would work better and thus chose to make a ring out of sheet metal. This was good plan because it added more weight to our robots and also we had the added bonus that our robots were dangerous and could slash a person's hand with their sharp metal edges. We realized that the bump sensors should be mounted with the switches on the side so that the weight of the ring would not push down on the switch; instead lateral movement would trip the switch. Once we made this discovery we decided to try to mount the metal ring this way and also dug plastic knives out of the trash (I said they'd be back) and tried to mount them on the other robot with the switches the same way. This actually never worked reliably at all since the bump sensors would often get stuck or they were internally already pretty broken and just plain weren't any good. The general unreliability of our bump sensing system and the wear and tear to the form of the ring during collisions with other objects proved to be a major weakness in our otherwise robust and ingenious design of crappy robots. For the game of tag to work even remotely it was necessary to supplement the detecting of bumps with a confirmation system. This is where the RF doorbell circuits came in to the picture.

IID. Unsolved Mysteries: Our Insanely Hacked Doorbell Circuit WAS The Most Reliable Thing We Had

As mentioned before this was one of the earliest functional parts of our project. It went through multiple incarnations of circuitry and managed to be usable in the end.

There was a short period of time when the receiver drew a lot of current from the batteries and we weren't really sure why. We think the receiver was just trying to help our project reach the level of dysfunction that we were shooting for. Somehow, though, the problem seemed to go away. I'm not sure if we actually fixed something, sorry I mean we actually just hacked it better, or if we got really good at changing batteries a lot and turning off our boards to conserve power. It may just have been some electrical tape put in the right places that made this a non-issue.

III. Rex Compile Errors And "Integrated Debugging Environments": Making All This Stuff Break Our Robots In An Intelligent Manner

All these failed ideas and ungodly sketchy hardware actually was meant to tie together in a meaningful way. Surprisingly enough the code to use these inputs and outputs was well designed enough to work in simulation. But as we all know, simulation is the devil's work, because it makes one think that things are actually going to work. Or if you're Rodney Brooks you can actually write your code so that it doesn't work in simulation, but works based on the uncertainty of your environment.

The robots were guided by three basic behaviors, represented as states (boolean wires), that guided their movement. These states consisted of "it" (in code "dansmom"), not it (notm dansmom), "paused", and "runway". In an effort not to blow (or bore) your mind with our inspired code, I will describe the finite state machine in a high level and anthropomorphic fashion. [EXCLUDED]

Upon inspection of the code one can see that coding this was not trivial. Since REX is such a funky language there were many special cases to be handled.

IV. Emergent Behaviors a.k.a. It's A Feature, Not A Bug

Some interesting things happened when this was all put together and these were unexpected, yet welcome additions to our game of tag. In the true spirit of Brown, we as a group tried to keep our minds open to new and inventive bugs that would make our lives more interesting and fulfilling. The first of said behaviors was the spinning in place behavior. There was also the situation in which the spin-inducing sensors would cause a spin and then the other set of sensors would take over, but wouldn't be close enough to the light so they also caused a similar uninterpretable behavior. We liked to think of this emergent behavior as the robots "taunting" state where hotdogs around in a circle while it waits for the other robot to come near it.

It turns out that having lights on the top of a robot causes some strange problems. What happened is that a robot would see the reflection of its own light as brighter than the other robot. This worked well as an obstacle avoidance maneuver, but was bad for tag. We were able to address this problem by taking our waste-of-money polarizers and using them as shades for the lights. This helped cut down on glare and reflected light to some degree.

Another somewhat unintended, though fully welcome into the family of weird stuff happening, was the behavior of the not “it” robot to attempt to orbit the “it” robot and to essentially wait until it got close to run away.

Lastly the robots inspired an emergent behavior in us humans by causing us to participate in the dance of the robots. With 15’ lamp cords connected to large extension cords we spent significant time coordinating our polished two-steps with the beautiful flow of our robots running into things.

V. Our Project Ain’t No Shakey, But At Least We Had A Kick-Ass Metal Video

We managed to stay gremlin free and didn’t have to use time-lapse photography and in the end our project worked somewhat well for very short periods of time. When the cycles were working they usually broke because the cord got tangled, we ran out of room, or the robots physically broke because they hit each other. When the robots didn’t work at all it was usually because a radio signal was missed or an extra one was received from a mystery transmitter. We found that our \$10 Home Depot doorbell circuits were not complexly encoded enough so that random radio, cell-phone, or negative vibes would disrupt them.

We learned a lot from this project regarding robotics in general and about two-robot coordination. Unfortunately, most of our lessons were taught by failure and we pretty much learned how hard it is to make very simple ideas work and why it is really good to get the military to pour money into your robots so that you’re well funded for your three trips a week to Home Depot. We also learned that most ideas that sound in theory will only have a chance of working when they are incredibly simple. Complex, clever, well reasoned solutions never work because the success of mechanics on a robot varies inversely-exponentially with the complexity of what you’re trying to do. We also learned specifically that two-robot coordination is very hard. First of all there is the simple problem that two robots running around means less space and more entanglements and snafus.

All in all the process was long and arduous but in a very enlightening and education way. It became clear to us that the integration of hardware and software in a mobile autonomous form is extremely challenging, mostly frustrating, but the cause of boundless exuberant excitement (usually profanity) when it works (case and point the only music-less clip in our video).

VI. The Cast And Crew Of The SS Overly-Ambitious-And-Essentially-Crappy Robot Project

- Soren “I flirt therefore I am” Spies was responsible for the construction, maintenance, and debugging of the Radio transmitter and receiver circuits
- Mike “Creativity” Fried was responsible for some of the structural modifications of the robot
- Dan “Internet Flirting Master & Rock God” Morris and Neel “I like to complain about women a lot & Bad Indian Numero Uno” Joshi worked wrote all of the code and made

significant contributions to all other parts of the project that involved being up really late and seeing the sun come up way too many times

- All other parts of the robot were worked on by all members to varying degrees

VII. Brought To You In Part By. . .

- Professor Robert Netzer & Professor William Patterson
- (= !'1b !0b)
- The variables: !dansmom and dansmomjustgotbumped
- Woven Wheats with their woven goodness
- The Bytches Club, you know who you are
- Crack
- El Kabong
- Boris
- The Ursas
- Bruce Bates – The guy who dropped the class and gave us an extra robot
- Sketchiness
- Whose Booty
- The Big Florescent Tube In The Sky
- REX ERROR: WRONG NUMBER OF OUTPUTS

Appendix A: Selected Code

```
(makem (maze*) :module 'bytch :ic t)
(setq INITIALDANSMOMVAL '1b)
(setq NUMER !6)
(setq DENOM !5)
(setq BACKADD !15)
(makem (maze*) :module 'danbytch :ic t)
```