The Most Excellent and Lamentable Tragedie of

Snarfy the Bunder Wot

6.270 2000
Team 41 Journal

Seth Purcell
Chris Osborn

1/4/2000 Day 1:

Attended first lecture, received kit. Checked kit for completeness and began thinking about strategies.

Proposed strategies:

"Local Circuit" (24 points max): - do a loop going up from jail, down the row of blocks on our campus, then go back up to jail, sorting and depositing blocks appropriately. Advantages: blocks opponent from putting profs in our jail, fast, simple.

Put all professors in opponent's jail (36 points max). Advantages: lots of points, possibly very fast. Disadvantages: might be popular, fairly obvious, high probability of encountering opponent, logistically more difficult than Local Circuit.

Aims to achieve: speed, reliability, keep profs out of our jail.

We want to be in control of as many blocks as possible as soon as possible, so speed is important.

Did some thinking about snarfer (part of robot that picks up blocks from the table when we drive over them).

1/5/2000 Day 2:

Thought about snarfer some more, went to second lecture, received and glued new box, thought about strategies some more. Came up with a first draft strategy: sweep opponent's blocks with a plow (we don't get the points, but neither does the opponent) then return to our blocks and snarf, sort and store hacker blocks, then go sweep opponent's jail in case he has put any hackers there, then go sweep our jail in case the other robot has put professors there or is attempting a jail break, deposit hackers in jail.

Preliminary list of hardware subsystems required for current strategy:

- plow - maybe with shock absorbers/bump detectors, maybe with scoop sides
- scoop in front of snarfer
- block snarfer
- block sorter
- type detector
- block mover

• block storage/dumper

Preliminary software flowchart:

• initial opponent's blocks attack
• main snarfing run
• possible secondary opponent attacks
• dump students/profs
• clear jail
• dump hackers

Determine what the next action should be based on how much time remains.

Preliminary software subsystems:

• initial orientation
• driving/plowing
• navigation
• sorting/loading blocks
• dumping blocks

Discovery of the day: snarfing cubes reliably will be difficult.

Storage bin ramblings:

We want the blocks stored off the table so that they don't drag.
We want an easy way to drop the blocks.
If we drop the blocks off the back, we can drive forward to clear them.
If we store the blocks vertically, they take up little area on the robot.

Chris creates a snarfer which resembles the snout of the wild tapir.

1/6/2000 Day 3:

Team goes to lab and begins attaching gears to motors.

Went to first recitation, built gear boxes, learned about sensors.

Return to lab, finish attaching gears to motors, test attachments. Tapir-proboscis snarfer debuts on the table for its first test, is met with wild acclaim, but needs much improvement in traction and overall reliability.

Team goes home, Chris takes rest of the evening off, Seth takes most of the evening off, works on battery pack/plow attachment schemes from about 12 a.m. to 4 a.m., decides to store batteries vertically between main axle and plow to have a lot of weight right on the axle for good traction, and right behind the plow for decisive plowing abilities.

1/7/2000 Day 4:

Attended optional Java lecture, 99% useless. Only benefit derived from aforementioned lecture was the refreshing of the "extends" keyword in Seth's memory. Regretted not sleeping in instead.

Went to lunch. Seth ate little.

Went to lecture 3, disappointed by a notable lack of controller-boards. In a semi-conscious stupor while listening to Edwin Foo drone about what he thought control theory was, Seth came up with what he thought to be a good idea for an alternative method for table navigation. Previous ideas were: dead reckoning with shaft encoders, checking the distance from and direction to the bright polarized table lights and wall following. Foo was talking about line following. Seth thought as follows: wouldn't it be nice if an optical sensor could track the table as it moved beneath the robot? -> optical mice do that, but they need special mousepads, but we don't have a special mousepad, we just have a table -> mechanical mice don't need special mousepads, they just need a surface -> we could hack a mechanical mouse to be a great high resolution 2-D shaft encoder, provided we can get it for under $20.

Seth proposes idea to Chris, who likes it but notes it can only detect translational movement, not rotational movement. Seth suggests two hacked mice. Chris says an off-axis mouse can be used as a forward motion/rotational motion detector (essentially polar coordinates). Seth approves good idea, and notes also that we have a USB port, so we can simply use a USB mouse instead of hacking a PS/2 or serial mouse.

Team goes to lab and constructs battery packs. Seth works on main structure for mounting batteries and wheels he started last night. Chris works on snarfer. Improved snarfer lauded by critics. Team goes home, Chris improves drivetrain and snarfer while Seth kicks things.

Chris proposes an unseemly idea for an alternative shaft encoder, perhaps because he has been working for 14 hours straight. Idea is put to a vote and quickly rejected. Chris descends into further silliness. Motion is put forth to discontinue all further brain activity for the remainder of the evening, motion passes, so we work on updating the journal, which brings us to right now. Seth types this sentence.

1/8/2000 Day 5:

Supplies are running out. Morale is low, there has been no sight of land for weeks. I fear the crew may mutiny.

1/9/2000 Day 6:

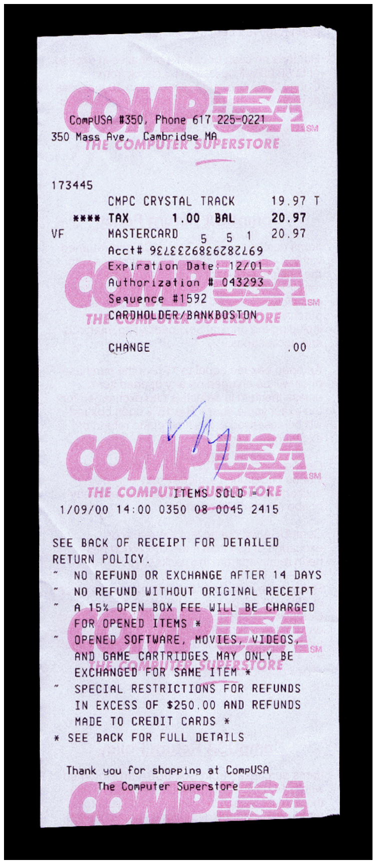
On a bright and disturbingly un-Januaryesque morning, we set out for CompUSA to find a mouse. CompUSA has many different kinds of mice, some for as low as $4.99. Unfortunately, none of the mice have what we would consider good ground clearance. We worry about how a mouse designed to slide around on a nylon mousepad (and manufactured, distributed, and marketed for less than the cost of lunch at Lobdell) would handle on a plywood table. The table has unintentional features that are almost as large as the intentional ones, and they could easily snag a mouse as it drags beneath the robot. Luckily, CompUSA also has trackballs, which we find are much better than mice in terms of clearance. The least expensive model is the Crystal Trackball, CompUSA's answer to the iMac-induced transparency craze. It is big, sturdy, pleasantly clear, and fits under the budget with three cents to spare. We don't know if tax counts toward the $20 budget, so we hold off buying until we can check.

1/10/2000 Day 7:

We have succeeded in finding a suitable trackball, but it costs $19.97, meaning that tax would bring it well over the $20 electronics budget. We go to lab and ask an organizer (Adrian) if tax must be taken into account. The answer is no. If the base price fits in the budget, the sensor is fair game. We go to CompUSA and buy the trackball.

Upon returning to the lab, it is brought to our attention that the trackball idea is probably in violation of the rule against sensors as structural members. We send an email to 6.270-rules

requesting a ruling on this.



CompUSA #350, Phone 617 225-0221
350 Mass Ave, Cambridge MA
THE COMPUTER SUPERSTORE

173445
CMPC CRYSTAL TRACK 19.97 T
**** TAX 1.00 BAL 20.97
VF MASTERCARD 5 5 1 20.97
Acct# 9ƐLƐƐƐ268Ɛ6282L69
Expiration Date: 12/01
Authorization # 043293
Sequence #1592
CARDHOLDER/BANKBOSTON

CHANGE .00

ITEMS SOLD 1
1/09/00 14:00 0350 08 0045 2415

SEE BACK OF RECEIPT FOR DETAILED
RETURN POLICY.
- NO REFUND OR EXCHANGE AFTER 14 DAYS
- NO REFUND WITHOUT ORIGINAL RECEIPT
- A 15% OPEN BOX FEE WILL BE CHARGED
FOR OPENED ITEMS *
- OPENED SOFTWARE, MOVIES, VIDEOS,
AND GAME CARTRIDGES MAY ONLY BE
EXCHANGED FOR SAME ITEM *
- SPECIAL RESTRICTIONS FOR REFUNDS
IN EXCESS OF $250.00 AND REFUNDS
MADE TO CREDIT CARDS *
* SEE BACK FOR FULL DETAILS

Thank you for shopping at CompUSA
The Computer Superstore

1/11/2000 Day 8:

By morning, we have received no reply to our email, and go to lab in the hopes of getting things moving again. We first find Andy, who says that the organizers are still debating our question. A few minutes later, Anthony tells us that they have decided, and that the trackball is legal. This is Good News.

In the evening we finally receive a formal email reply to our extra sensor question. Anthony states that it is okay if the trackball bears some weight, as long as the robot does not simply collapse if the trackball is removed. This is not quite consistent with his first answer, but we will go along with it anyway. Our best idea is to have the trackball accompanied by a Lego skid (perhaps a hemispherical foot made out of the baseplate) that will slide along the ground and provide structural support independent of the sensor.

1/12/2000 Day 9:

We study trackball literature on the web and learn how trackballs determine direction. Having been told that the board can sample sensors at nearly a million times per second, we decide that the best way to interface the trackball with the Skiff is simply to read the four shaft encoders using sensor ports, and emulate the trackball circuitry in software. Chris opens up the trackball and installs resistors to limit current to the LEDs and to adjust the output signal from the detectors.

Meanwhile, Seth cuts up the trackball case to make it a more reasonable size, and develops an incredibly strong, if not topologically impossible, Lego chassis around it. The chassis survives both drop tests and mighty heave against the wall tests, and we are happy.

1/13,14/2000 Days 10 and 11:

More work on the trackball and chassis.

1/15/2000 Day 12:

We finish the chassis and glue the trackball in place. With the battery assembly installed, the machine is quite a beast. The chassis and drive system rises no more than four inches off the table, yet uses a substantial portion of our Lego supply. It is very heavy and very strong. Unless it falls off the table, it will be hard to break.

1/16/2000 Day 13:

Chris adds the vertical lifting snarfer (as prototyped earlier) to the chassis. After several revisions and adjustments, it seems to work quite well with the blocks. Work begins on the horizontal sweeping conveyors that will draw blocks toward the snarfer. We decide to power the sweepers using the main drive motors rather than independently. This will save motor ports and cause the sweepers to run at a rate proportional to the robot's motion. This may or may not be useful, but it should be nice looking. Chris builds the right hand sweeper and arrives at a mechanism that works., but is too weak. Seth attempts a different approach for the left-hand sweeper, but it is also too flimsy.

1/17/2000 Day 14:

We spend the day developing and refining the critical point-scoring portions of the robot, including the front horizontal sweepers, snarfer, kicker servo, block storage, and release door.

1/18/2000 Day 15:

We receive our first RoboSkiff, which promptly explodes, triggering a recall of all boards. Fun. We spend the rest of the day on software, which may or may not be in the correct language.

1/19/2000 Day 16:

Chris revises the sweeper design, making it much stronger. After copying the design on the other side of the robot, we move on to other systems.

1/20/2000 Day 17:

We finally receive our RoboSkiff, and the first thing we do is to test the trackball. This is a big disappointment. The trackball works perfectly, and the software that Seth wrote interprets the signal nicely. However, it only works at incredibly slow speeds. We test the sensor polling speed and find it to be in the area of 250 Hz. This is a far cry from the near MHz range that we were told was possible. We check with tech support, and find out that the board is indeed capable of something like a million samples per second, but there is no way to achieve that with software, so technically, they weren't lying in the first place. Silly us.

1/21/2000 Day 18:

Chris attempts to build some custom hardware that will do the more demanding processing

on the trackball independent of the RoboSkiff. Seth completes the block release door mechanism, allowing the robot to release stored blocks on command. He also mounts the kicker servo and redesigns the storage tunnel to prevent blocks from getting jammed.

1/22/2000 Day 19:

Chris continues work on the circuit to determine trackball speed and direction circuit while Seth writes code to support the release door and kicker servo mechanisms, as well as the still hypothetical tracking system.

1/23/2000 Day 20:

After a lot of soldering, we have a circuit that works well on an external power supply, but it too sensitive to voltage to work off the battery with the RoboSkiff. By this point, Chris has more or less reached the limits of his analog electronics expertise, and there is not much more that can be done.

1/24, 25, 26/2000 Day 21:

The Final Push: One last attempt to make the direction counter circuit work is unsuccessful for unknown electronic reasons, and it is ditched in favor of code that will infer direction based on drive motor states. We try to hook up the trackball directly to the shaft encoder ports, because the RoboSkiff is so slow that it can't keep up with the signal on the sensor ports. Unfortunately, the pullup resistor skews the signal toward Vcc so that it never drops below the threshold voltage. Matt informs us that the threshold setting function described in the API has been secretly removed. Seth narrowly resists throttling Matt.

Later, Chris succeeds in readjusting the trackball signal so that it matches the set threshold, and we test the trackball. The trackball misses counts, which we attribute to slippage. In one last desperate attempt to get good tracking, we buy a mouse, hoping that the rubberized ball will grip better. Again, the input current on the shaft encoder ports screws up the signal, and it can't be adjusted this time. We give up and go back to normal shaft encoders. We have noticed that the normal shaft encoders miss counts sometimes, so we decide to build some of our own out of the dependable LED/phototransistor pairs. Unfortunately, the sensor store is all out of them, and Matt tells Chris that it wouldn't make a difference anyway because an unpublished bug in the Skiff causes the shaft encoders to miss beats one sixth of the time. Chris narrowly resists throttling Matt.

With only an hour left before impounding, we decide that a random walk around the table with the snarfer running and the kicker sorting will possibly score a point or two, so we try it. We qualify, and relax with the knowledge that we have done the best we can despite the monumental forces aligned against us. Reflecting our mental state shortly after impounding, and with a big fat Crayola marker in each hand, we name our robot: Snarfy the Bunder Wot, whose Beautiful Harware was for Nought.