THE ROBOT COMPANION

the newsletter of the
Dallas Personal Robotics Group

July, 1988

Reminder to DPRG members: You are already signed up on the BBS. Log on with your first and last name as they appear on the mailing label. Your password is your zip-code (5 digits). Once you log on, I suggest that you change your password, to make the BBS more secure. To change your password, use the C)hange Setup option from the main menu.

JUNE MEETING MINUTES

June 11, 1988 2:00 p.m.

Ed Rivers demonstrated his demo program for the HERO 1, which was shown to the DALTRUG members. He also gave a general report of the demo we did for DALTRUG. Their response was very favorable.

Bev Bryant attepted to show the BLOCKS program, but it had been wiped out when she ran one of the built-in demos. Wouldn't a floppy disk drive be nice, Bev?

JULY MEETING AGENDA

The July meeting was held at 2:00 P.M., July 9th at the Infomart in Dallas.

Since the newsletter is being produced AFTER the July meeting, I'll just mention briefly what DID transpire:

We finally had the BLOCKS demo at the meeting. It went pretty well, even though the robot's gripper was uncalibrated

We discussed the Checkers proposal (see below).

We had a pseudo-treasurer's report (the treasurer was not present).

PRESIDENT'S CORNER

by Bev Bryant, President

How many of you out there have a house with a floor plan exactly like Walter's and mine? How many of you out there have two HERO 2000's? Well, without taking an actual survey, my gut feel is that the count would be near zero. This observation may have something to do with the lack of enthusiasm over the Data Communications program and the Home Navigation program. So, Stan, Walter and I together come up with a new idea for a program. How would like for your HERO 2000 to play checkers? Yes, we thought of chess, but we decided the algorithm might be too much for the HERO 2000 (and also, too much for us). Stan's BLOCKS program has inspired us. (If you haven't seen this outstanding program, you can download it (BLOCKS.H2) from the BBS.) The robot puts the blocks at such precise locations, we decided it would probably be able to play checkers on a standard checkers board. I have thought about it for a few days and I came up with the routines I think will be necessary. If you see any cracks for us to fall through, please let me know. (Routines requiring movements are marked with an asterisk and programmers will have to access the same database for block locations.)

```
*INIT       - Places board (or has human place board at a  specific point and orientation).
            - Determines which player moves first.
            - OUTPUT - GCUR_PLAYER (IF = 0, self, IF = 1, opponent)
 CALC_MOVE  - Calculates best move for robot
            - Makes appropriate comment (if any)
            - OUTPUT - PIBN (Pickup Block Number)
                     - PLBN (Place Block Number)
                     - KBN (King Block Number)
                     - RMBN (Remove Block Number)
*PICKUP     - INPUT - PIBN
            - Picks up piece at PIBN
*PLACE      - INPUT - PLBN
            - Places piece at PLBN
*REMOVE     - INPUT - RMBN (IF RMBN = 0, REMOVE is not called)
            - Removes piece at RMBN and places it off the board
CHK_FOR_WIN- If Number of Opp's pieces = 0 and GCUR_PLAYER = 0, robot wins
             If Number of Opp's pieces = 0 and GCUR_PLAYER = 1, robot loses
*KING       - INPUT - KBN (IF KBN = 0, KING is not called)
            - Kings piece  at KBN by rotating until King symbol is on top (pieces have King
              symbol on bottom when play begins).
ACC_OPP_MV - Accept keypad input
            - Check to ensure it is a valid move
            - (Optional) - Call  Calc_Move to  determine wisdom of  opponent's  move  and
              comment (if appropriate).
```

PERSONALITY

(This is where anybody can help. Tell us what phrases you think would be appropriate, cute, intelligent-sounding, etc.)

We need the aid of someone who has experience at estimating lines of code. To fan these subroutines out to club members, we need to know about how many lines of code each subroutine will take. Then we can assign specific line numbers to each subroutine. For instance, I have flowcharted the executive and estimated it will take approximately 50 lines of code. (We want to overestimate a little to ensure we do not overwrite someone

else's subroutine.) Therefore, EXEC will be assigned lines 100-600 (numbering in increments of 10). INIT will probably require some movement, so I will need the help of someone like Stan to estimate those lines of code. CALC_MOVE and ACC_OPP_MOVE will probably be the heftiest of the subroutines (at least, the ones requiring the most effort and at this point). We may decide to put them at the end of the program because of this and because of the uncertainty involved.

I hope everyone is as excited as I am about this program. It's something everyone can participate in and every HERO 2000 owner will benefit from (and we can all point to some part of the finished product and say, "I thought of that"). I have a book with an algorithm for playing checkers that I will be bringing to the user's lab and to the regular meeting. If you have an algorithm somewhere, bring it and we'll try to decide on which one we want to use. Be ready to sign up for a subroutine. It should make for a great brainstorming session!

HACKERS AND HOMEBREWERS

by Stan Spielbsuch

I received a fantastic letter from Loren Heiny, a member in Arizona. He has taught his "robot" to play tic-tac-toe using vision as its only input! Here is a brief re-cap of his letter:

Dear Stan:

I saw in the March issue of The Robot Companion that several of you are interested in programming your robots to play games. I think the idea is great! As I see it, as long as we can't get our robots to wash the dishes, we might as well have some fun with them. I thought your club members might be interested in one of my attempts at a game-playing robot -- a robot that plays tic-tac-toe. The "robot" is actually nothing more than a camera, a Votrax speech synthesizer and an IBM PC.

A camera is the robot's primary sensor. It is used to sense all of the activity in the game. Because of this, the opponent doesn't have to press any buttons or switches to tell the robot what is going on. The vision system is built around a low-resolution (128 by 128 pixels) solid-state camera mounted on a tripod to the side of the game board. The board is a 15" square piece of white cardboard with a black grid. The game pieces are black construction paper about 3" in size. The speech synthesizer announces appropriate words of encouragement to the opponent. Almost all of the code is written in C and consists of about 2000 lines of code.

The vision system doesn't operate like you probably think it does. It can't actually see or recognize objects. This would take far too much time. Instead, the vision system is used in conjunction with expectations about the tic-tac-toe game in order to perform very specific tasks.

For instance, to tell when a person has made their move, the vision system merely waits for motion to come and go. It doesn't check to see what is responsible for the motion. It simply subtracts successive image frames and tests to see if there is a significant amount of change between them. If so, motion is assumed.

Similarly, to tell where a person has placed their token, it compares before and afetr images of the board. Combined with pre-determined knowledge of where valid moves

can be made and what tokens look like, it can usually tell where the person has made their move.

Of course, all of this is not fool-proof. For example, the person may have taken tokens away rather than laid one down, or he could have moved the camera to induce the motion, or not moved at all. Part of the fun in developing a robot like this is to program it for these types of eventualities. For a simple game like tic-tac-toe this hasn't been very difficult. A more complex game, or a game with many small intricate parts would be another matter.

Someday, I hope to integrate an arm into my game playing robot and enable it to move its own tokens. Besides making the robot look more like a robot, this should make it even more autonomous and increase the illusion that the robot really knows what it is doing.

If anyone has any questions or comments about my "robot" I'd be glad to hear them. Just drop me a note at: Loren Heiny
920 Terrace Rd. #211
Tempe, AZ 85281

Wishing you and your robots the best,
Loren Heiny

Editor's note: Thanks, Loren, for the great input. If anyone would like to see the original letter and a picture of his "robot", come to the next meeting!

HERO 2000 NEWS

by Stan Spielbusch

Thanks, Bev for your effort toward a realistic club project! I hope this project finally gets the enthusiasm we need in the club. I, for one, am very interested in the project. It's something I was planning to do myself "eventually", but I would be happy to have others help. I will volunteer to do the movement sections, and of course will contribute to personality. Ed Rivers has uploaded a checkers game, written in GWBASIC, to the BBS (see area 6, HERO 2000 and related programs). I have tried, it and although it is pretty weak, it is plenty good and fast enough for an initial project. Once we get everything coordinated, we can work on a better algorithm.

Higher-level languages:

I would like some input from HERO 2000 owners (and potential owners) -- I am going to develop a set of subroutines in either Turbo Pascal or Microsoft C to drive the robot's motors and sensors. This will enable those with a disk drive in their robot to program in a real language instead of BASIC. My main question is this: which language to start with? My strongest inclination is Pascal, because more people probably know it or can learn it easily, and Turbo Pascal is slightly more available than Microsoft C. If anyone has another opinion, let me know. The other question is: Turbo Pascal 3.0 or 4.0? i have both, and much prefer 4.0 (naturally), but what does everyone else have? Should I even care, since I only know 2 other members with disk drives? Has anyone else written such subroutines? Should I stop asking questions and just start programming?

HERO 1 / HERO JR NEWS

John Sprague (not a member) has a HERO Jr. robot for sale. It has:

- 24K memory
- I/R motion detector, voice, etc (standard features)
- BASIC cartridge!
- RS232 interface!
- Programming Language cartridge!
- Misc. cartridges (Songs, games, etc.)

He is asking $300.00 for it. I hope someone in the club buys it, so we can duplicate the BASIC cartidge, Prog. Lang. cartridge, and RS232 interface and make them available to other HERO Jr. owners. Or, if you are an enterprising sort of person, you could do this yourself and make some profit (if I had more time, I would do it).

Anyway, call John Sprague, 484-8270 evenings.

FROM THE LIBRARY

by Stan Spielbusch, Librarian

ALL LIBRARY PROGRAMS NOW AVAILABLE ON THE BBS!

HERO 2000---

I have revised BLOCKS.H2 to be more foolproof and provide more entertainment. Specifically, if you steal one of his blocks and won't put it back, he gets rather upset now (Try it!). Also, he will no longer allow erroneous inputs (if you told him to put block 1 on top of block 1, he would actually try to do it, then he would get very confused!). The new version (same name) is on the BBS and in the library.

.....

If you have a program to submit, the easiest way is to upload it to the BBS. Another way is to put it on an MS-DOS format disk (double sided, double-density standard format) and bring it to the meeting or send to:

Stan Spielbusch, 2404 Via Barcelona, Carrollton, TX 75006

****** Please ***** include a description of the program, either as comments in the program or as a separate .DOC file. I don't have the time to study each program to figure out what it does!

When you submit a disk, you receive credit for 1 disk in return. Let us know which one(s) you want, or if you just want your original disk back.

We currently have 3 disks in the library -- a HERO-1 BASIC disk, a HERO-2000 BASIC disk, and a HERO-1 Assembler disk.

If you want a copy of a disk, the best way is to bring a blank, formatted PC-DOS/MS-DOS disk to the meeting and trade with me there. (Or, you can download the programs you want, or the entire library, from the BBS.) If you forget to bring a disk, we will have to collect $2.00 per disk. Mail-order -- $3.00 per disk -- no need to include a disk with order. Send orders to Stan (address above).

CLUB INFORMATION

The Dallas Personal Robotics Group is a non-profit organization of individuals interested in learning about personal robots, sharing ideas, working on projects, and informing the public about the world of personal robotics. We are open to anyone who has an interest in personal robotics, whether or not they currently have a robot, and whether or not they have any knowledge of robotics.

To become a member and receive the newsletter, have access to our BBS and program library, and be involved in our monthly clubs and user's labs, simply fill out the form below, and send it with $10.00 to Walter Bryant, Treasurer (address below).

If you are interested, but not sure you want to be a member, feel free to visit our meetings. If you like, we can send you a sample issue of the newsletter. And please feel free to call our robotics BBS (number below).

Tentative Meeting Schedule (1988):
July 9 Aug 13 Sep 10 Oct 15 Nov 19 Dec 17

Meeting times and location: 2:00 PM at the Dallas Infomart.

Club officers:

President:	Bev Bryant	Vice-president:	Ed Rivers
Treasurer:	Walter Bryant	Secretary:	Brian Vaceluke

Back Issues

A complete set of back issues, from the formation of the club in 1984 to the present, is available for $3.00. Add $2.00 postage & handling if ordering by mail. Contact Stan Spielbusch, Editor, 2404 Via Barcelona, Carrollton, TX 75006.

BBS NUMBER: (214) 231-2836 300/1200/2400 baud. 8 bits, 1 stop, no parity

---

M E M B E R S H I P A P P L I C A T I O N
Dallas Personal Robotics Group
c/o Walter Bryant, 814 Mockingbird Circle, Lewisville, TX 75067

Check one: ( ) Renewal ( ) New Member ( ) Info Change ( ) Sample issue request

NAME (please print)_____

ADDRESS_____

CITY_____STATE_____ZIP_____

HOME PHONE (_____) _____-_____ WORK PHONE (_____) _____-_____

TYPE OF ROBOT (if any)_____

TYPE OF COMPUTER (if any)_____ MODEM? _____BAUD

Do you want the above information available to other members? _____