PROGRAMMING AGENTS TO LOCATE ITEMS IN BUILDINGS

New Mexico
Super Computing Challenge
Final Report
April 1, 2009

Team #: 75

School
Melrose Junior High

Team Members
Randall Rush
Kaleb Broome

Teachers
Alan Daugherty
Rebecca Raulie

TABLE OF CONTENTS

PROGRAMMING AGENTS TO LOCATE ITEMS IN BUILDINGS

Executive Summary

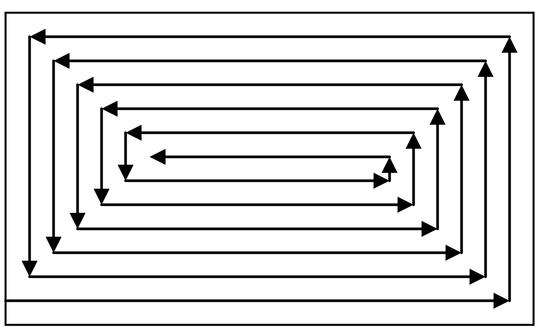
Searching unfamiliar places for explosives is, as you might have guessed, not very safe, and radio signals from a remote control to a robot can possibly cause the explosives to detonate. To solve this problem, we decided to create a program that would allow an agent to search a building while being completely unmanned. We used StarLogo TNG because it is agent based and is easy to work with. To measure our results, we will record the number of time clicks that it takes to completely cover the entire area. We will take the average number of time clicks for one floor plan, and compare it with the rest of the the data from different floor plans.

Research

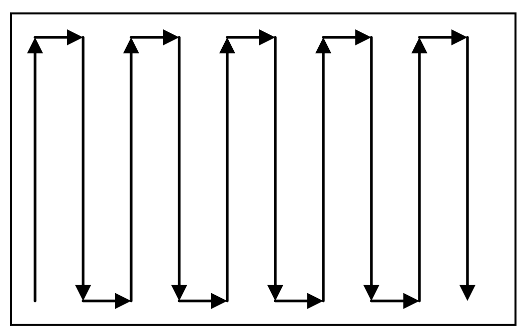
Unmanned machines have been a source of mythology dating back to ancient Greek writings, and Leonardo Da Vinci made a mechanical knight using springs, strings, and pulleys. But until recently, unmanned machines couldn't actually perform specific tasks. Even now, the machines that are completely unmanned are still not working like people want them to. For example, some experimental sentries and patrol robots will not open fire upon someone hiding behind something, but will open fire upon anything that looks like an AK-47. Also, so far as the research available to us goes, there are no completely unmanned machines used for finding and destroying explosives, although there are varieties of remote controlled ones.

Description

In our project, we are trying to program an agent "robot" to search a building and find and destroy all explosives in the building. We will have the floor plan of a building in which we will place the robot at the supposed entrance. We will time the agent to see how long it will take to cover the entire building using a spiraling pattern to "sweep" the entire building in an exhaustive search covering every single patch in the building. We tried a "lawn mower" type pattern where the agent would go straight in one direction until it comes to a wall, then turn right, take a step, turn right again and continue on until it comes to another wall where it would turn left instead of right. Trial runs of this type were unsuccessful, for the robot would get to the end of a wall and stop. In the sweeping mode, the agent will go until it comes to a wall. It will then turn left and then continue on until it comes to another wall. It will also treat places that it has already been as a wall in the sweeping mode.



Spiral mode



Lawn Mower type mode

Once the agent has gone as far as it can go using the sweeping mode, it will retrace its steps back to where it started the sweeping pattern which will be beside a wall. It will then follow the wall until it gets to a place where it has not been and can resume the sweeping pattern. The agent will follow this process until the entire building is covered.

Programming

For the most part, we used a simple algorithm to cover the entire floor plan, where, as I stated previously, it would turn left every time it came to a wall or somewhere that it had already been, and every step that it took it would stamp the patch yellow, and stomp it down 0.01 of the patch height that it was previously on. It would continue to do that until there was nowhere that it could go using that algorithm that would not have been already covered, it would then look for the patch around it that is 0.01 higher than it was, step onto it, and then it would then pull the patch that it had just left up to 0, and then it would begin looking for another patch that was 0.01 higher than it and repeat the process until it was where it had begun the sweeping mode, and then it would find a wall and follow it until it had gotten to somewhere that it had not been yet and it would start all over again.

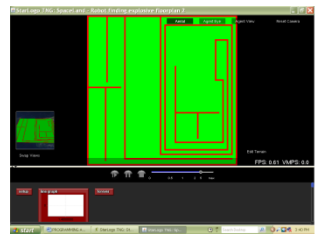
Results

To help measure our results, we have put the approximate number of time clicks for 5 different floor plans, including a control with no walls. We are hoping that the data will help us find out what kind of floor plan the agent will work best in.

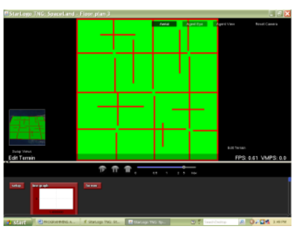
Floor Plan 1:

Floor Plan 2:

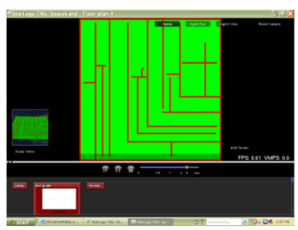
Floor Plan 3:



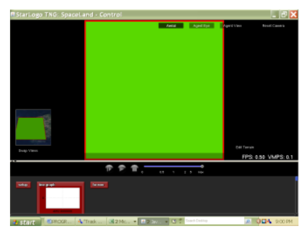
Floor Plan 4:



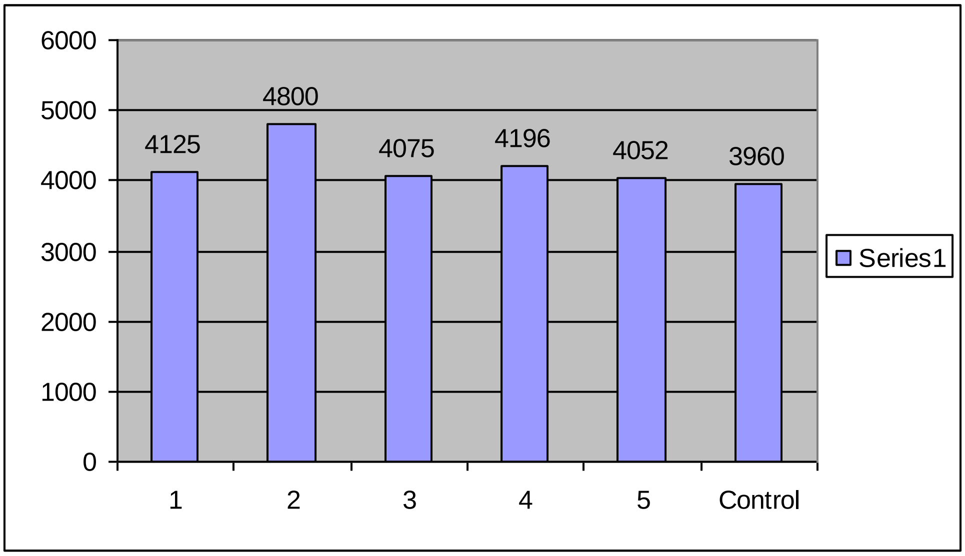
Floor Plan 5:



Control:



Approximate number of time clicks for each floor plan



We found that the simplest appearing floor plan, (1) took about 4125 time clicks to complete, the second simplest appearing, (4) took about 4196, the third simplest (2) took 4800 time clicks, the fourth simplest appearing, (3) took about 4075, and the fifth simplest appearing, also the most complex, (5) took about 4052 time clicks to complete. The control took about 3960 time clicks.

TOTAL TIME CLICKS PER FLOOR PLAN				
Least				**Most**
Floor Plan 5	Floor Plan 3	Floor Plan 1	Floor Plan 4	Floor Plan 2
4052	4075	4125	4196	4800

Conclusion

From the graph we can conclude that the complexity of the floor plan has slight effect on the time it takes to cover the entire floor plan, however, the type of floor plan does make a difference, because using the graph we could tell that the most complicated floor plan took the least amount of time excluding the control, and that the least complex took the third longest time to cover. The amount of time that it takes depends on how much the agent has to go back over places that it has already been. It seems to work better in a building with long halls instead of just going from room to room. We can also conclude that it will only work in buildings with no free standing walls, because the agent has to follow the walls to make sure that it goes through the entire building. We learned this because I made a mistake while making the floor plan and left one wall free standing, the agent then missed about 100 patches because of it.

Missed patches in floor plan 2

Post Actions

Now that we have designed a program that will cover the entire floor plan of a building with no free standing walls in an exhaustive search, we hope to improve the design to where it will work in a building that has free standing walls, and will still cover the entire area. We also hope to get to where we can use the “smell” command to give it a sense of smell, or we can limit it to give it some sight, so that it can just go into a room and then test it to see if there are any explosives in the room without actually going over every single patch.

Sources

Ancient Discoveries. History Channel,

"Robot." Wikipedia. 2009. 30 Feb 2009 <http://en.wikipedia.org/wiki/Robots>.

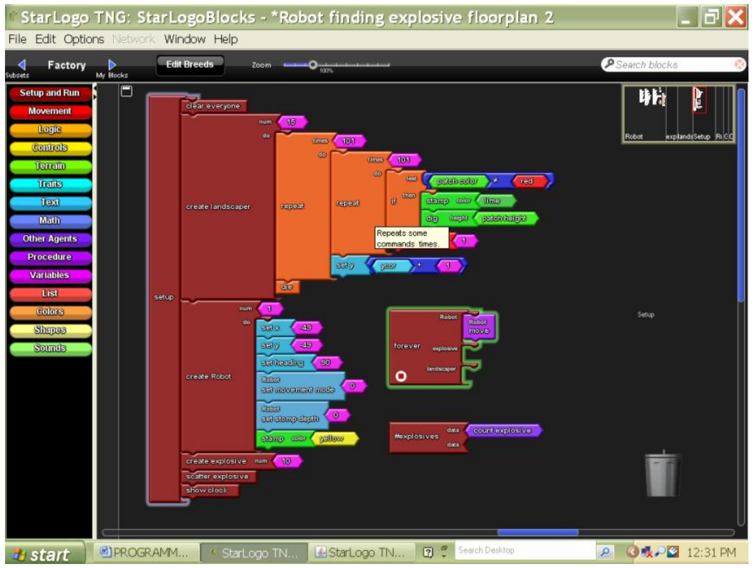
"Robots in Warfare." Wikipedia. 2009. 23 Mar 2009 <http://en.wikipedia.org/wiki/Robots_in_warfare>.

Acknowledgements

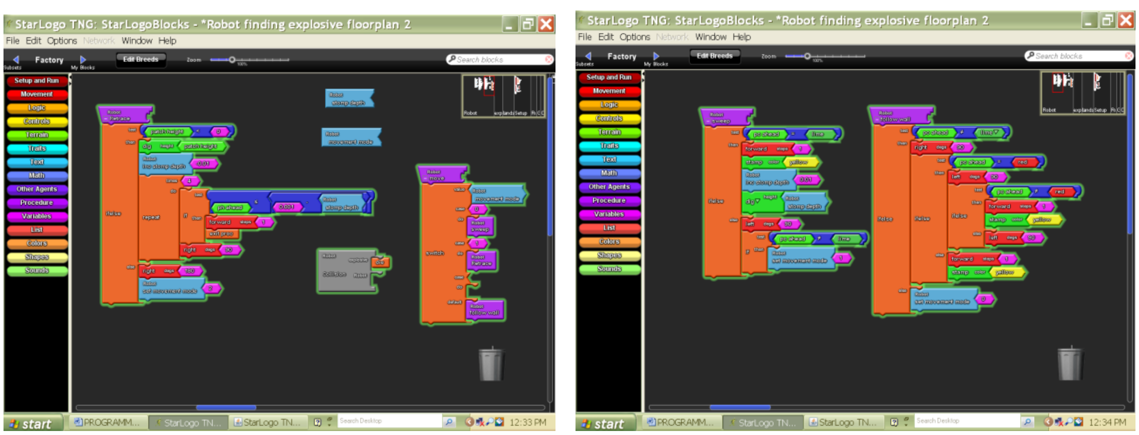
Nick Bennett, systems architect

Appendix

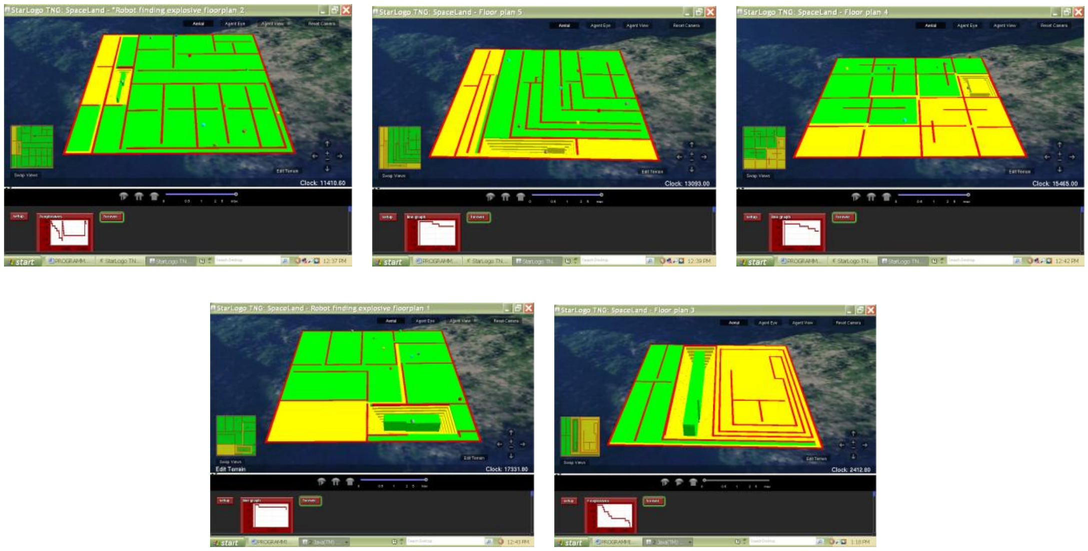
Setup Code



Run Code



Running Code



Number of Explosives