The Terminators: Challenge Design Outline

1.Overview:

We settled on a simple but elegant approach in solving the challenge at hand. This does not, however, mean that we aim to simply meet the bare specifications. We will focus more on solving the problem at hand rather than spending a significant amount of time on complicated approaches that have minimal gains. From the Mars Curiosity rover, with a measly 132 MHz CPU and 128 MB RAM, we see that a lot can be achieved even with minimal resources.

2.Problem Statement:

The grand challenge is to create a robot which can construct a shelter in an unknown environment using construction materials which are blocks in previously specified sizes and shapes and given an uncertain map.

3.Assumptions:

There are a number of assumptions made in order to make designing the robot more straightforward as well as more feasible while accomplishing the challenge stated above. The assumptions can be grouped generally as relating to the environment, the context, and the blocks, and the robot itself.

Environment

The environment is assumed to be navigable by a robot without any leg-like appendages being necessary as well as being static. Additionally, it is assumed that a significant amount of the total terrain has blocks. The obstacles are also assumed to all be taller than our highest sonar sensor.

Context

The robot is being constructed in the context of this class meaning that there are limited fiscal resources and more importantly time restraints. Because this robot is going to be constructed during the class, some designs like using expensive hardware, while potentially improving the function, cannot be implemented. Lastly we assume that there will be enough power and time for the robot to complete the task.

Blocks

The blocks are assumed to all be of the same height which will allow for the bot to use sonar sensors to detect where blocks are. The blocks are also assumed to be able to fit underneath the robot.

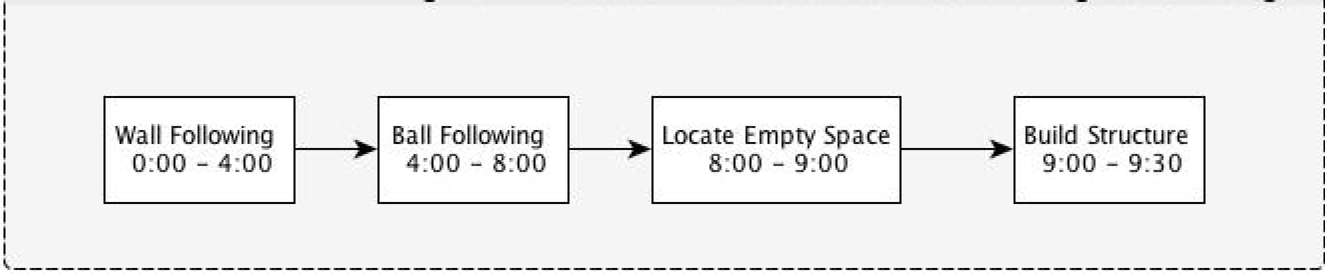
Robot

The robot's body, excluding some moving parts, is rigid. With this assumption we can use fixed numbers such the wheelbase distance and wheel radii consistently across all calculations

4. Technical Approach:

The robot will have three high level states: *Wall Follow*, *Ball Finding*, and *Construction*.

Figure 1: The State Machine of the Robot During the Challenge

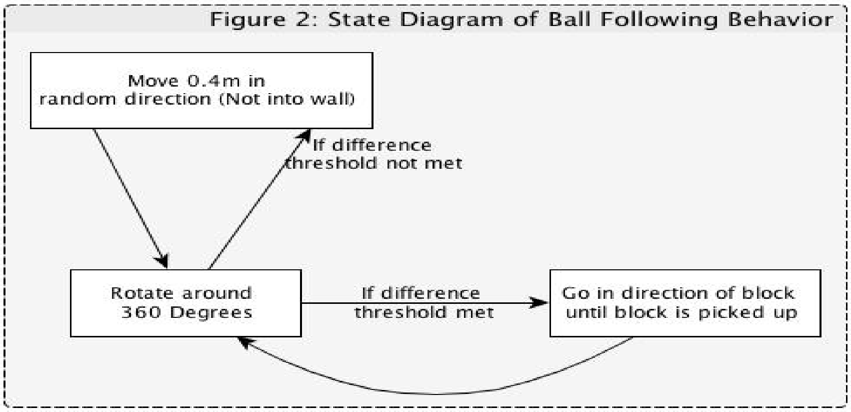


Wall Follow: [member]

When in *Wall Follow,* the robot will move forward until it finds itself at some to be determined distance from the wall where it will turn left and then move forward using a proportional controller to same a relatively constant distance from the wall.

Ball Following:[member]

When in *Ball Following*, the robot will rotate around until there is a difference bigger than a predetermined threshold between the upper front, and upper lower sonar. Once the robot stops rotating it will move in the direction until it picks up the block and then begin the process of rotating again. This is a Braitenberg behavior because it will only require simple coding but the robots movements will seem complex since it will be finding a ball using only two sonar sensors.



Scooper: [member]

The robot will have an uncomplicated block collection mechanism, one primarily consisting of a ‘funneling’ as well as block arrangement components. The idea is that as the robot traverses in the environment, it will accordingly scoop up the arbitrarily-located blocks via means of a ‘Y-shaped’ channel extension located on the anterior part of the robot and underneath the robot chassis as well. The ‘Y-shaped’ channel, then directs the blocks into a narrow rectangular channel running underneath the robot’s chassis such that a single stream of blocks is obtained as the robot ‘scoops’ more of them. The motivation for the ‘Y-shaped’ channel is that it helps extend the robot’s ability to gather more blocks as it explores its environment since it is able to sweep over a greater area. The other end of the rectangular channel terminates with an ingeniously designed block arrangement system that will force the stream of single-filed blocks into a semi-circle. The semi-circle arrangement of blocks will form our robot’s shelter. The figures below illustrate the described system.

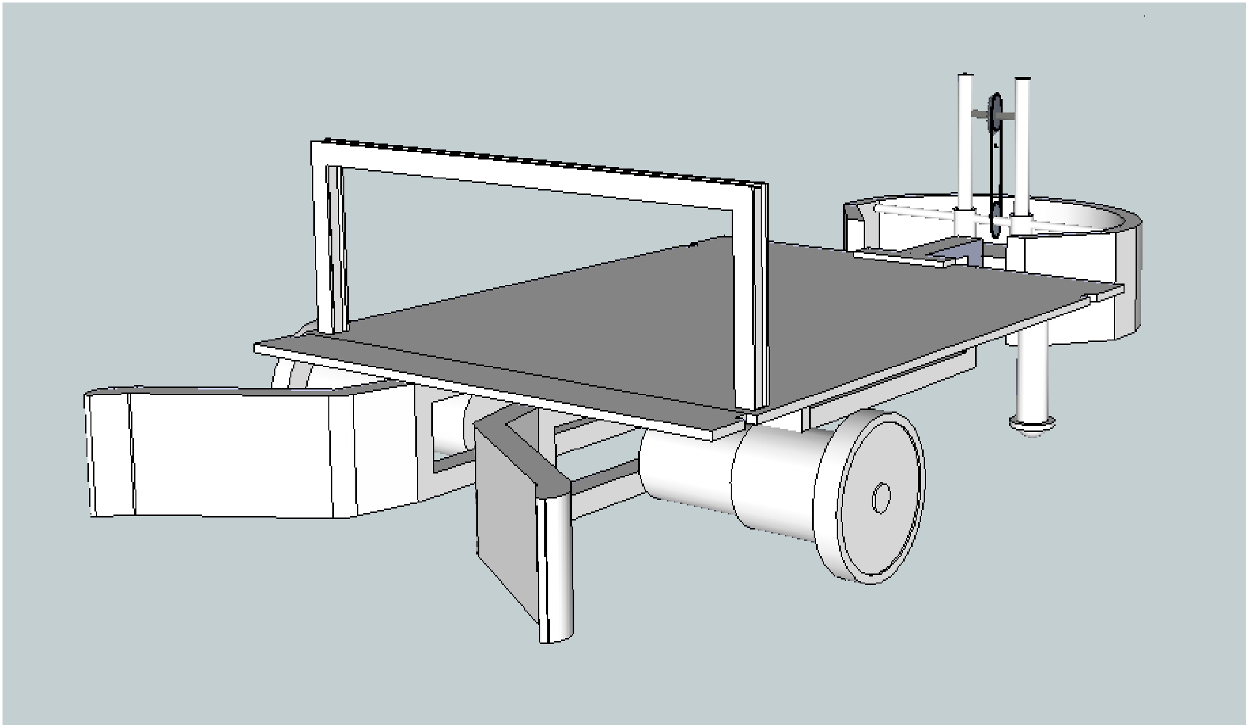


Figure 3: Front View of the robot with the block ‘scooping’’ mechanism attached

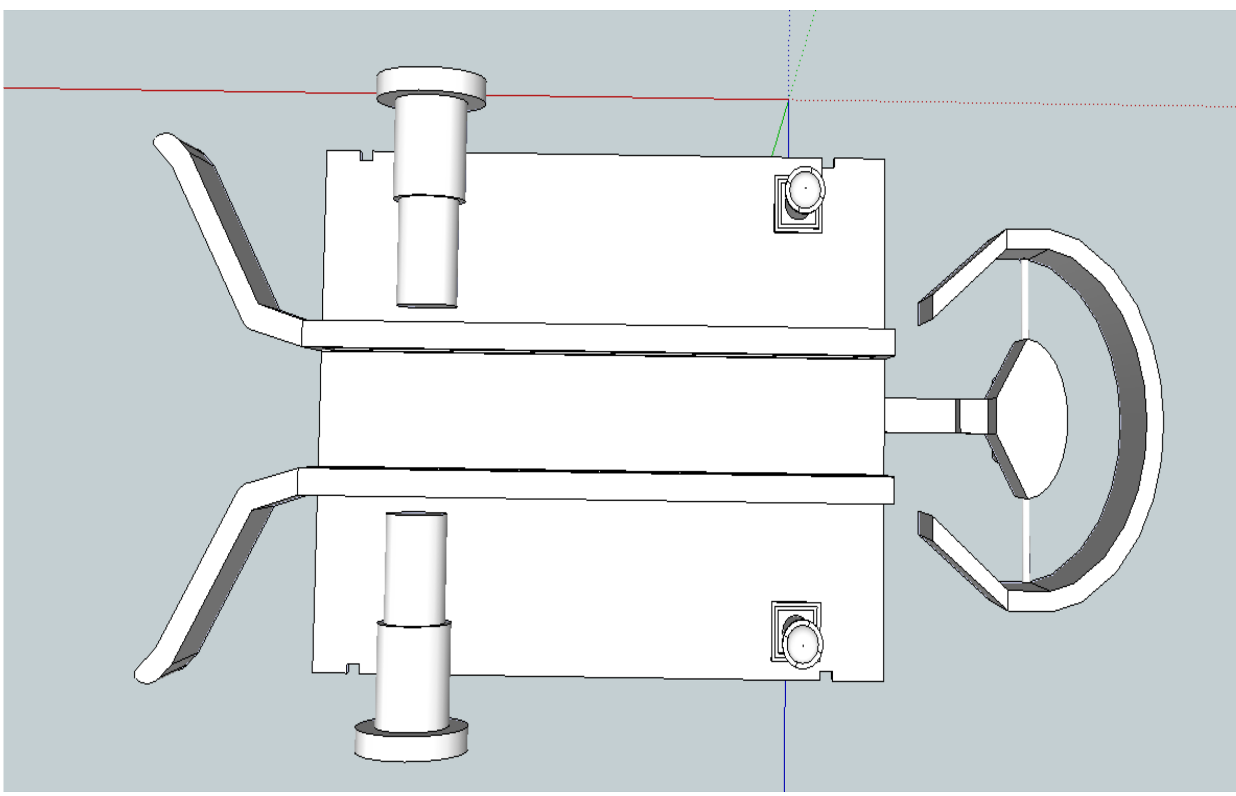


Figure 4 : Under-chassis view of the robot showing the ‘Y-shaped’ channel, the rectangular channel and the semi-circular block arranging component

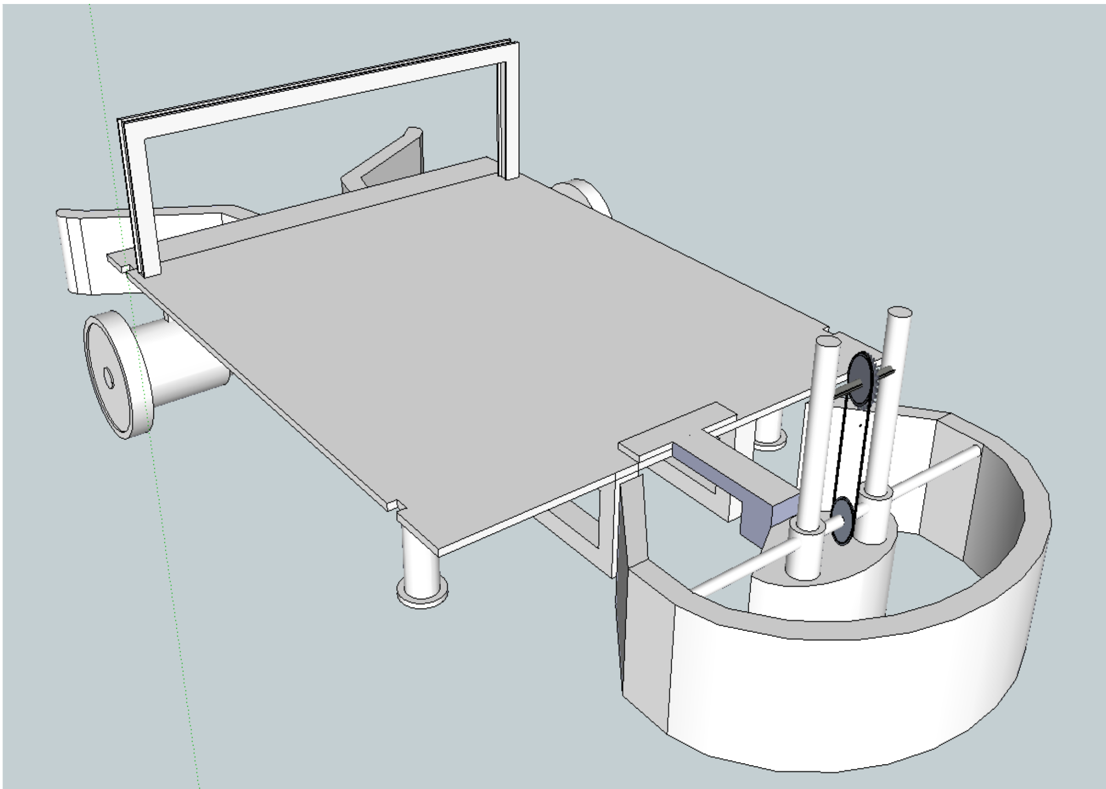


Figure 5: View showing a close up of the semi-circular block arranging component

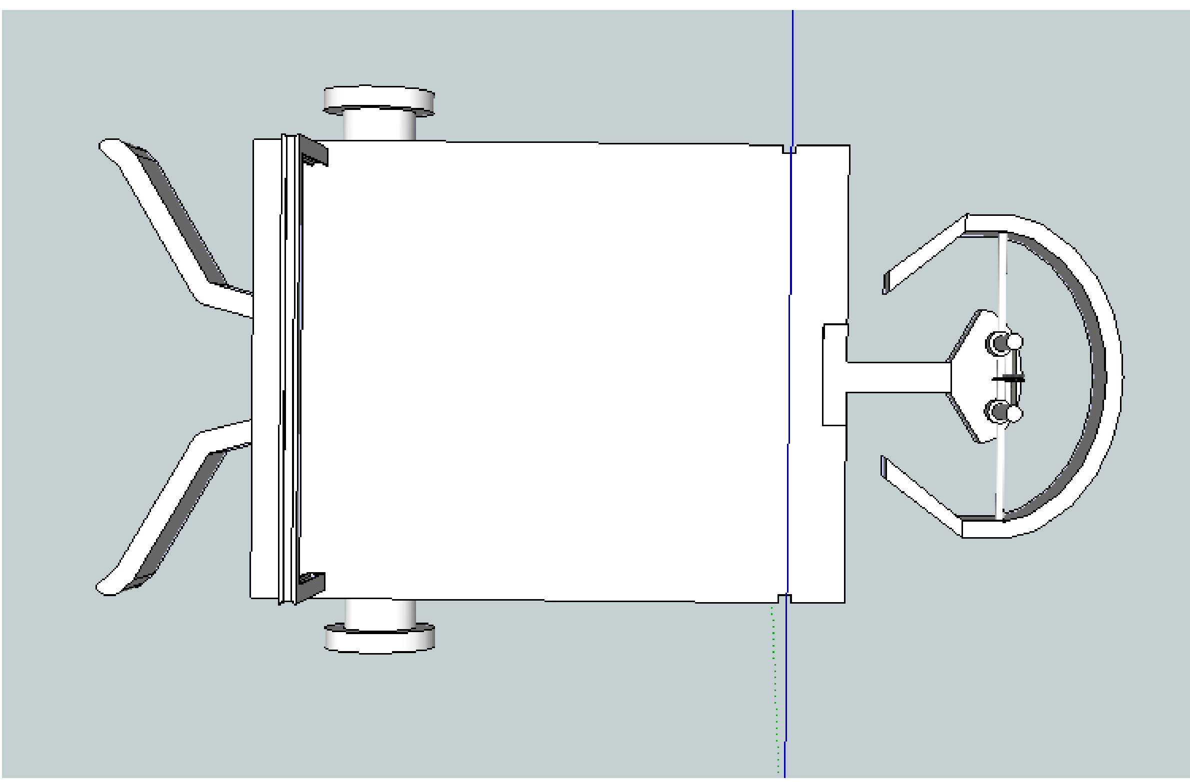


Figure 6: Overhead view of the robot

Sensors: [member]

We will incorporate the use of sonars (4 in total) for sensing the robot's environment. Our robot will use these sonars to detect obstacles and building blocks. The installation of a pair of sonars at the front of the robot, at the same lateral position but at different heights, will enable our robot to detect building blocks by comparing the two sonar readings. The height of the higher sonar will be greater than the height of building blocks. This mechanism is illustrated below.

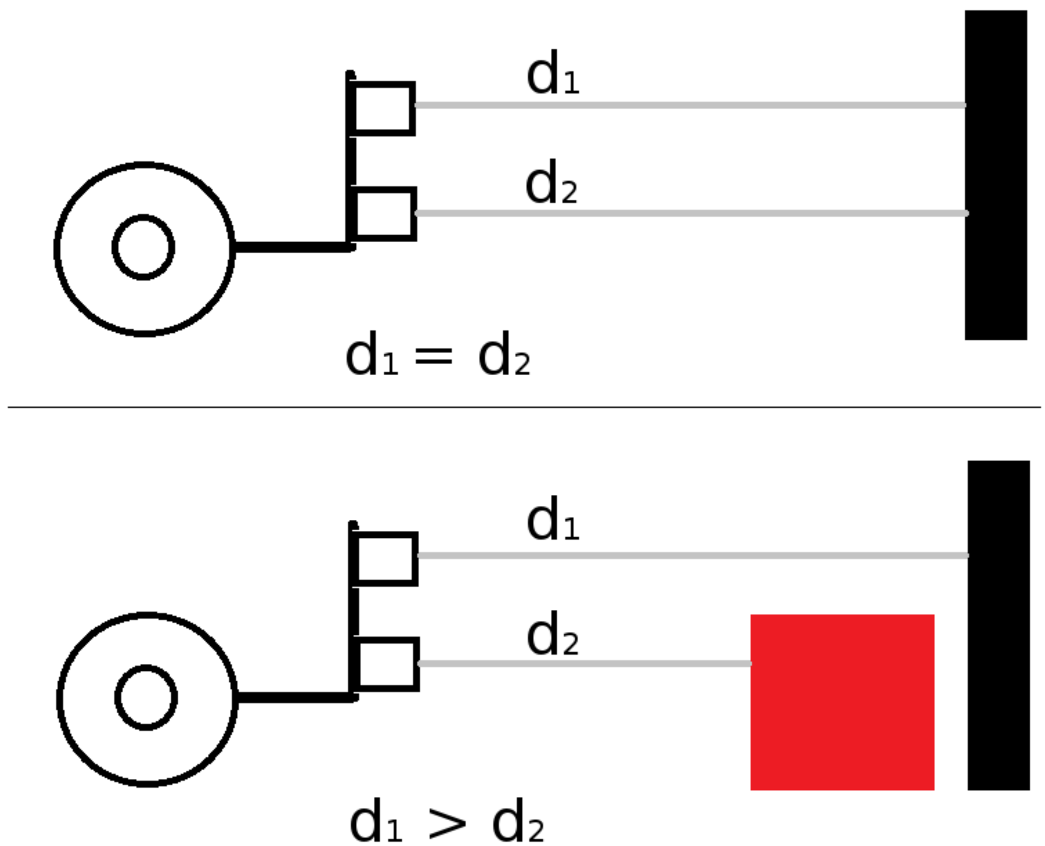


Figure 7. The robot's front sonar readings (d_1 and d_2) with no building block in sight (top) and with a building block in sight (bottom).

.

The second pair of sonars will be installed at the front and back of the left edge of the robot. These sensors will detect walls for *wall following*.

Map: [member]

As backup if our original plan does not work out, we will be introducing mapping to help us localize and orient our robot in the environment. Since we will be utilizing sonars, we can generate a map of the environment that models the robot's location in relation to the obstacles that are in its vicinity. Based on what we learned and work done in lab 5, we can have software generate a rough draft diagram of the environment with the obstacles by drawing line segments where the sonar indicates an obstruction. This methodology is akin to work we've done in 6.01 so we anticipate generating maps that are similar in nature to those shown below.

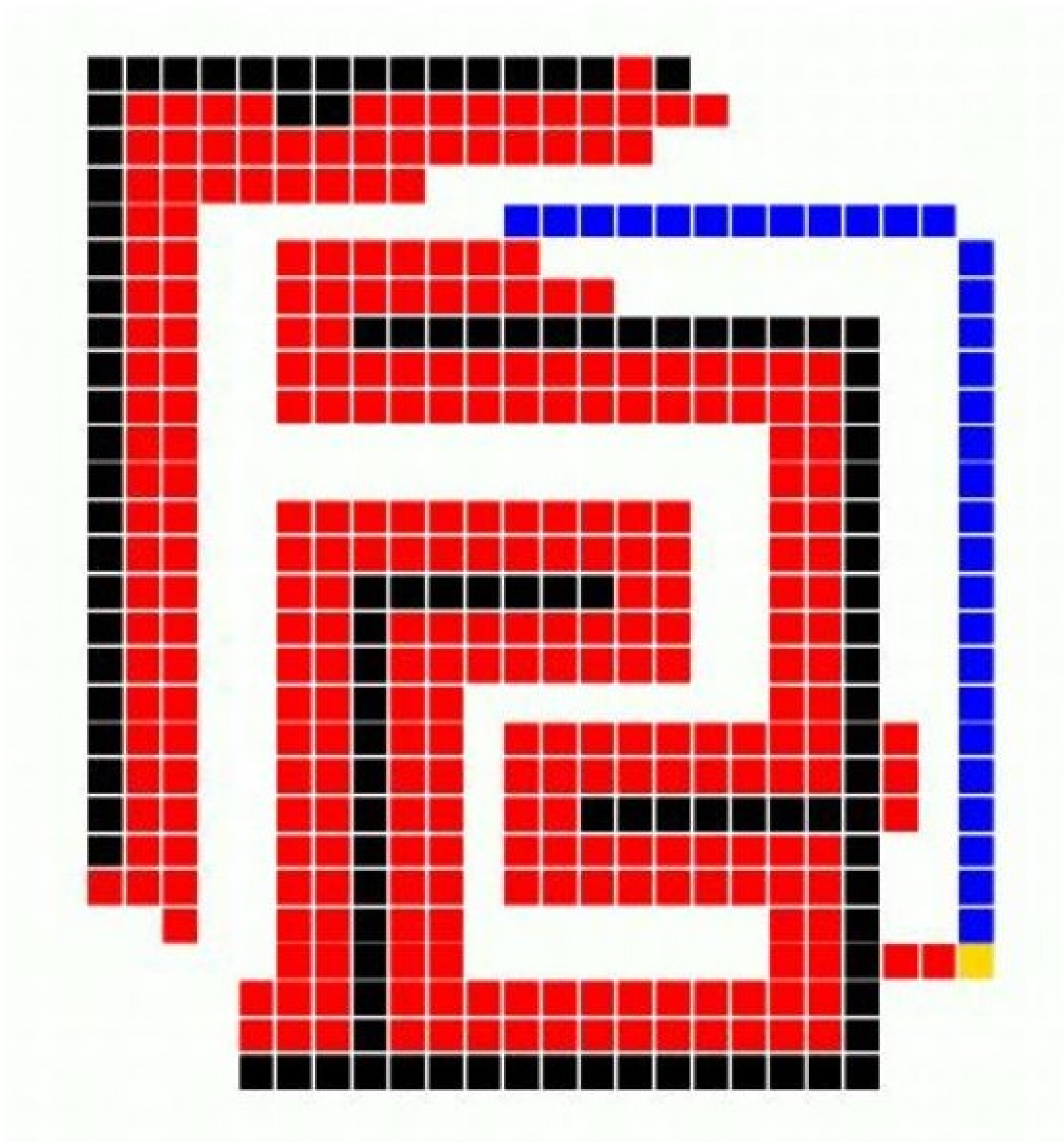


Figure 8: Maps generated for a formerly unknown environment

The black tiles are regions that we are certain are obstacles while the red are less certain. The difference in the certainty levels can be judged by a threshold that we set based on the perceived accuracy of our sonars. The yellow tile is the goal location/destination while the blue tiles form the traced out path computed by our mapping software as the best path to the goal.

5. Concrete capability

We have completed the basic chassis and structure of our robot as part of work we did in Lab 2. We will be utilising the motor controller solution code from Lab 2 as well. We will also be utilising the solution code to lab 5 (if there is one, else we will have to complete the implementation ourselves). All other software that we will need to complete the implementation of our robot will be written from scratch (using the libraries that we've been working with so far).

Task	Start Date	End Date
Constructing of shovel mechanism and mounting onto the robot chassis	04/01	04/08
Mounting and configuring all the sonar sensors we will be using	04/01	04/08
Configuring integrated autonomous behaviors	04/08	04/12
Testing & Debugging	04/12	04/15
Backup (if Testing & Debugging yield negative results) implementation of mapping and localization	04/15	04/22
More testing & debugging	04/22	05/02
Challenge Dry Runs	05/06	05/06
Grand Challenge	05/08	05/08

Table 1: Deadlines for individuals tasks for the grand challenge

We are approaching the initial aspects of putting the robot together in parallel. Patrick has shop access and will be responsible for designing fabricating and mounting the shovel; at the same time Todd and Caleb will be mounting configuring and calibrating the sonars. Once these tasks are done we will be able to carry out a first time run to see the efficacy of our plan. Then we will proceed to program the behaviors we intend to achieve. Our timeline is such that we are aiming to be able to make a decision on whether or not we have to resort to our Mapping and Localization backup plan early enough so that we have time to implement and test it.

6. Design- Making Process

Our team decided we wanted to solve the challenge using as little technology as possible. We watched videos of other teams who used really complicated designs and thought there were much simpler ways to solve this problem. At first we thought about only using bump sensors and sonar but as discussions went further we hypothesised the bump sensors would not even be needed. From that point we had other decisions to further clarify our design.

After making the decision to use only sonar, there were two implementation details which we had to focus on. They were the high level movement of the robot, and the the design of the shovel. We quickly realized that the biggest potential problem was the robot not traversing enough terrain to obtain the desired number of blocks to build the shelter. One of the most likely ways that this could occur was the robot being stuck in an infinite loop so we were sure to avoid any high level motion plans which could potentially cause this. Another way was simply not visiting the majority of the map. We then realized that even with our rudimentary technology, we would be able to make imprecise guesses as to where the blocks were by having two sonars at different heights and checking if there was a large disparity in the range measurements (indicating a block).

We knew that wall following would be a simple behavior to implement so we were bias towards using it but we also realized that only implementing that behavior had the potential to not visit a large portion of the navigable terrain. As a result, we decided to also implement a braitenberg like behavior to assure that we did not only check the edges of the terrain. We decided to implement four minutes of the wall following behavior and then four minutes of the Braitenberg like behavior since both of these behaviors are effective on different kinds of maps.

After deciding with the movement of our robot, we agreed pretty quickly on the shovel like feature explained above. This driver for this idea is our robot will not have the exact location of the block so the shovel will give us a larger margin of error than any other method to pick up the blocks.