CS 431
Kenny Hunt
2017/05/06

ThunderDome

Preliminary

Strategy:

The strategy for the preliminary was to use adhesive as a way to gather the balls. This strategy involved following the black line in a pattern that traversed a majority of the map. It used back-tracking to account for bricks, that is, it would turn around and follow the line to another (hopefully brick-less path). By following the line my robot would gather all balls within a 15cm radius of the black lines, avoid bricks, and return home in under 2.5 minutes.

Implementation:

The construction of the robot was a simple cylinder about 12cm in height with a radius of 15cm using only two `NXT Large Motors`, two `NXT Light Sensors`, and an NXT Touch Sensor. The large motors were used to traverse the map, The light sensors were used to follow the black line, and the touch sensor was used to sense bricks in front only. The light sensors were placed symmetrically on the right and left sides of the bot just in front of the wheels. This placement helped achieve the acute angles that the black lined path had. The touch sensor was placed directly in front of the robot to be activated when touching a brick.

The software had two `Behaviors` prioritize in the `Arbitrator` as: `TurnAround`, `LineFollower`. The `LineFollower Behavior` consisted of a basic PID controller to travel the straight black lines at a reasonable and controlled pace. Because the path consisted of 5 black lines, each about ~90cm in length, speed was important and had to be adjusted with battery voltage depletion. The sharp turns that the path contained made using the basic PID controller very challenging, so I decided to add the second light sensor. This light sensor was used as an alternating line following sensor that activated in an every-other-corner pattern. My final line following algorithm went like this:

```
NXTLightSensor rightLight = new NXTLightSensor();
NXTLightSensor leftLight = new NXTLightSensor();
NXTLightSensor currentLight;
     while(!suppressed){
          for(int corner = 0; corner < 6; corner++){
               if(corner % 2 == 0)
```

```
                        currentLight = rightLight;
                        rightMotor.forward();
                        leftMotor.forward();
                    else
                        currentLight = leftLight;
                        rightMotor.backward();
                        leftMotor.backward();
                turnCorner();
                followStraightLine();
            }
        }
```

This code uses a `currentLight NXTLightSensor` for the `LineFollower` at all times but swaps between the `rightLight` and the `leftLight` every other line to achieve the sharp turns. The `turnAround Behavior` was activated when the touch sensor was being touched. It caused the robot to simply turn around and continue with the `LineFollower Behavior`.

Analysis:

There were many problems with my robot, the most apparent of which was the effectiveness of the adhesive. The balls did not stick to the robot often, when they did, even a slight change in direction would shake them off. Another problem was the error in the design of my `LineFollower`. While it was not demonstrated during any of the rounds, when my robot encounters a brick it would not always guarantee getting home, as my algorithm does not localize the corners in a good way: Whenever it encounters six corners it stops, I had not considered this problem until deciding to restart for the finals. I feel that I succeeded in making a unique design for my robot. This achievement fell short with my inability to affix the dome in time during the second round, along with it's general performance in all three.

Finals

Strategy:

With the outcome of the finals I was determined to completely change my approach. The final strategy tried to tackled all of the problems in the project: Eject wooden ball, gather pink balls, and return home. I first focused on the

wooden ball. My strategy was to isolate it with size, and eject it at the end of the round. After figuring this out, I worked on getting home using a beacon as a landmark. The final thing i improved was gathering the pink balls, relying solely on hardware that allowed balls in but not out. The strategy was to drive around randomly avoiding walls and collecting balls at the 1.75 minute mark I have the robot head home and dump the wooden ball out of the arena if it was captured.

Implementation:

The mechanical design of the robot was completely different. It Shared with the first, only maneuverability, as the two EV3LargeMotors were almost required for all robots. The new motor that was introduced was a single EV3MediumMotor. The sensors this time consisted of two IRSensors, a HiTechnicCompassSensor and a HiTechnicColorSensor. The color sensor and the new motor worked in close relation as a way of filtering the wooden ball, forming a sort of basket in the back. One of the IR sensors was placed low to the ground on the front of the robot. The other IR sensor was placed atop the robot along with the compass sensor that was attached as a sort of antenna.

The abilities of my robot was greatly improved resulting in the following Behavior priority order: AvoidWall, GetGolden, Drive. The AvoidWall Behavior's goal was to help the robot maneuver throughout the arena. It used the low frontal IR sensor to measure the distance it was from something, when the distance reading was lower than a threshold (varied depending on battery voltage), the robot would back up, and turn left or right (depending on whether the system clock milliseconds reading was odd or even). The GetGolden Behavior was used to filter and capture the wooden ball. It used a color sensor to sense whether the wooden ball was underneath it (reading > 45) and the medium motor to spin the basket upward 90 degrees. The basket filtered out pink balls with a slot that only the wooden ball could fit into without falling through. The Drive Behavior was split into two actions that depended on how long the

robot had been driving around. The Drive Behavior changes when the robot's clock has reached 1.75 minutes, after which the home ward driving action kicks in. To get home my robot would drive randomly until it found the IR beacon, then using a PID controller it would follow the beacon to the goal. After achieving a certain distance from the Beacon (getting into the goal), if the wooden ball is sensed to be captured then it would activate the wooden ball dump. This consisted of backing up, turning around 180 degrees, backing up again (towards the wall) and spinning the medium motor another 90 degrees to dump out the captured ball, if it is there.

Analysis:

The finals went much better than the prelims but there were still many troubles. The biggest trouble was getting back home, it was not demonstrated during the first two rounds because my robot was getting stuck on the walls surrounding the goals before the 1.75 min mark. The third round with no bricks did not go well at all as my robot had a higher chance of getting stuck in a goal when there were more obstacles, allowing it to change direction more often. The other problem was that my wooden ball catching system was never truly demonstrated because the first two rounds I had placed my robot in a way that when driving forward would always just push the ball into the goal, making it unreachable for the rest of the round. my main success I believe was the usage of drapes to keep balls in.

Conclusions

To improve my robot I would start with the drapes. As well as they did, they still allowed some balls to get out. Another thing I would improve upon would be the homeward Drive Behavior, taking inspiration from Tyler's Cubone, I would try to use the compass sensor as a better bearing to find where the beacon is, in the cases where it cannot see it due to distance. I also believe that the placement of the front IR sensor could be changed to allow for capturing the wooden ball more effectively.

I very much enjoyed this project and wouldn't change a thing.