Our Mickey Mouse bot:

* The same as last year
* Except: arm-and-box replaced pom-grabber
* Added: reflectance sensors

Sensors:

* Idea: use reflectance sensors to find black divider and black tape
* Trouble: large variance in sensor data
  + We ran trials to find the cause – collected data (printf) of the sensor values.
  + Large variance in data
  + Cause: wobbly chassis
* Trouble: not finding tape very often
  + Single sensor finding divider 2nd time.
  + Fix: add 2nd sensor.
    - Source code:
    - One: Trigger on one sensor – finding divider
    - Two: Trigger on two sensors – find tape

Arm/box – Effector:

* One difficult problem: turning around
  + The bot was too big when the arm/box was down.
  + Solution: throw poms over PVC.
* An elegant solution:
  + Simply raise box back over the bot’s body to make it back-heavy, so it had traction to hit the bacon-bot.

Our mm bot was a recycled robot from last year and, instead of having a pom grabber, the same servo brought down a box to drag the gold poms out of the cave. Another mod from last years deisgn was that we needed to have a reflectance sensor to find the strip of tape following the cave. We decided o use two instead of one so we could detect if we were traveling along vs. running into a black line. Which was giving us a lot of grief as we ran trials.

Effector

Our bot was far too wide to reliably turn around with the arm down. So, instead of having it turn around, we decided to modify the arm to push the poms over the PVC and into the starting zone.

The majority of our trial and error session was to find out the proper sensitivity of the reflecdtance sensors to degtect the black lines. There was a lot of variance in the reflectance as we moved along the black line. Even though 2nd helped, there was still too much variance and our program failed. The measurement that we got had a mean of 515, but there were outliers

The measurements we got was about 200 and black was 525, but as the bot would run, black would trigger randomly over the white area. After many trials, we discovered the bot wobbled (like a teeter-totter), which lead to the large variance we were seeing. We made the bot front-heavy to keep it from lifting.

Not onlhy was the bot wobbly ;

We balanced it to make it front heavey.

Tough problem

Wobbly bot

Elegant solution

Folding the square backward so the weight distribution on the mm bot was correct, so it could trigger the bacon bot.