Robot Design and Programming

Who are we?

- Debbie English
  - Computer Science/developer/connected & automated vehicle
  - Mentor of FLL – 7 year, FTC – 2 years, FRC(#4499) – going into 8th season!
  - FLL volunteer/assisting Liberty tournament – over 10 years
- John Wiens
  - Senior at Colorado School of Mines, Electrical Engineering
  - Mentor of FRC #4499, Student on FRC 3 years, FLL – 2 years,
  - FLL Ref – 5 years, FLL Judge 2 years
- Hailey Holman
  - Member of FRC #4499, FLL #? Years
  - FLL Volunteer experience

What makes a good robot

- Consistency
- Consistency
- Consistency
- If your robot cannot perform a task 10 times in a row without failing, it will probably not work properly in competition!
- Keep it simple (KISS)

Introduction to Robot Design

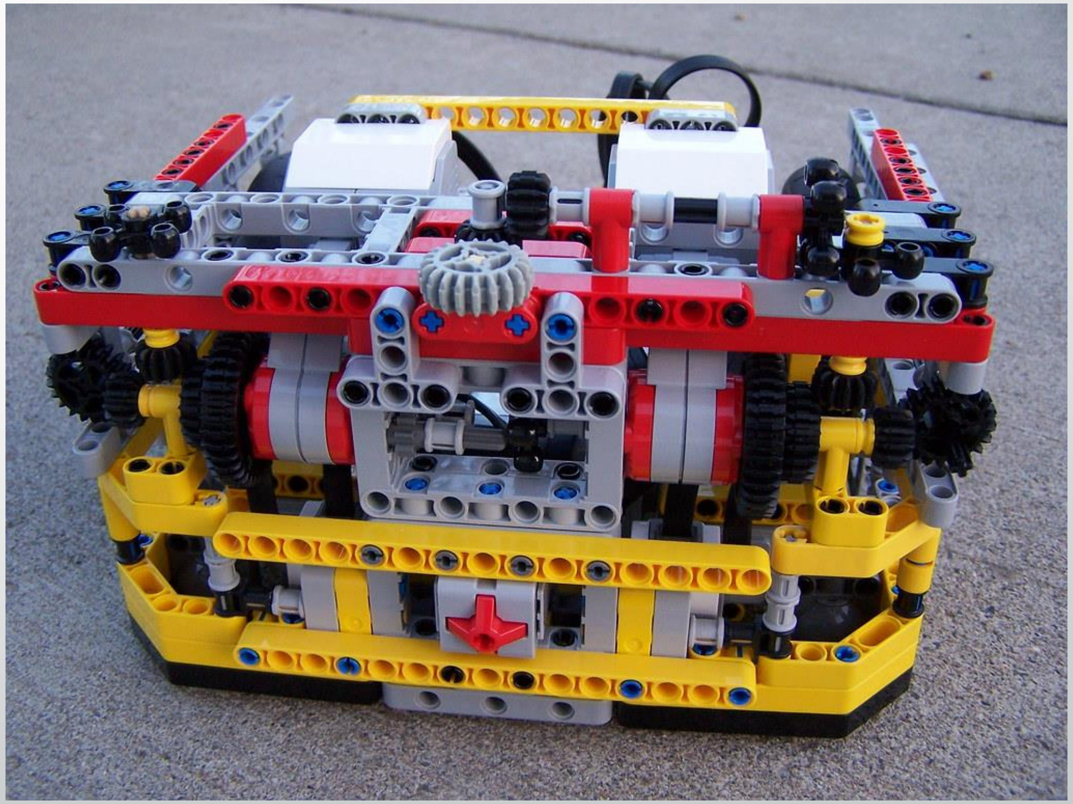
- As stated previously the best robots are the ones that are the most consistent. Total point potential doesn't matter if you can never achieve it.
- Things to help with robot consistency:
  - Quality robot construction
  - Sensors
  - Thoroughly test robot code
  - Build Mission Models correctly
  - Practice running your robot in tournament conditions
  - Do not overextend your team's capabilities

Robot Design and Construction - Strategy

- There is no one design that is correct in FLL, many different designs and strategies are effective.
- Things to consider when constructing a robot:
  - Make sure that your robot is rigid. Wobbly or flexible frames hurt overall robot consistency.
  - Design your robot to be unaffected by map defects such as wrinkles or warped baseboards.
  - Do not underestimate the difficulty of traversing the field. Easy, High Scoring missions that are far away require a lot of time and make your robot more likely to get stuck.

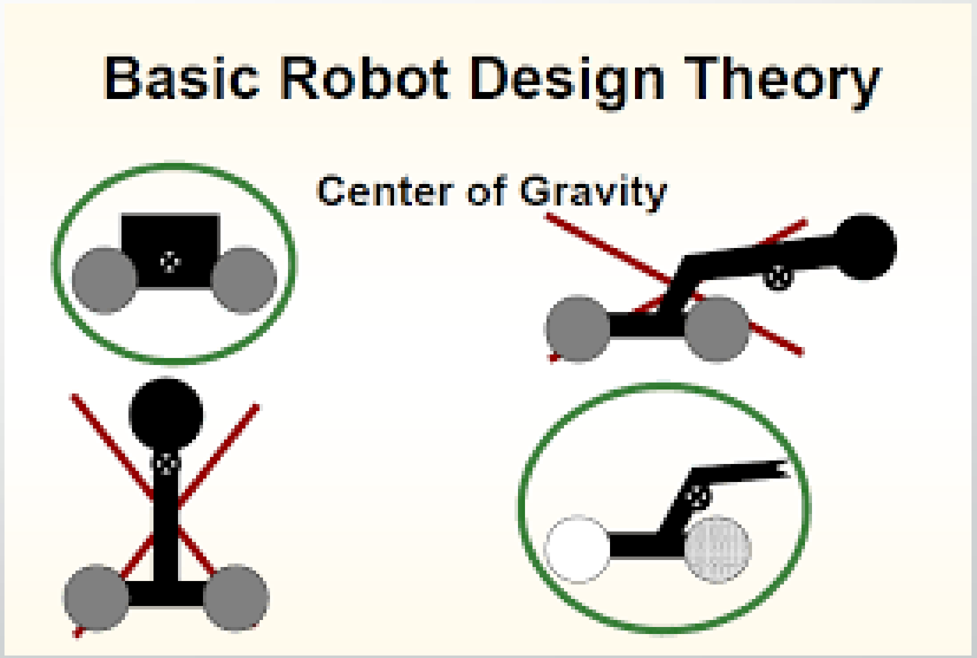
Robot Design and Construction - Modularity

- Many Robots use different attachments for different missions. This is a great way to make your robot perform a variety of missions more effectively.
- Avoid needing to snap or click new parts onto your robot. Even though Legos come apart, attachments that connect this way are often slower to swap in an out and are more commonly installed incorrectly.

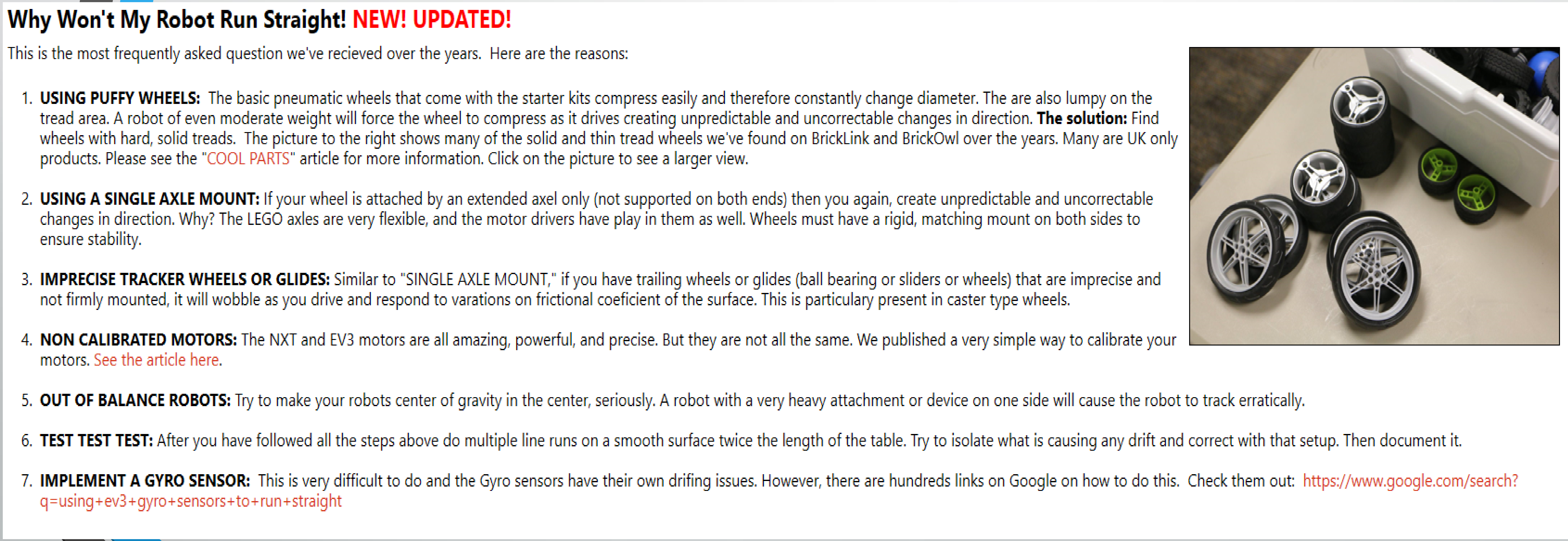


Robot Design and Construction – Drive Train

- Robots are not stable on 2 wheels, assume that you will need at least 3 or more points of contact with the floor.
- Put the Center of gravity of your robot between your wheel-base, this prevents a tippy robot and increases consistency.
- Robots with Tires can accelerate and stop faster. However, tires wear down over the season which hurts consistency.
- Faster doesn't always mean better.



Should we add this?



Why Won't My Robot Run Straight! NEW! UPDATED!

This is the most frequently asked question we've recieved over the years. Here are the reasons:

1. **USING PUFFY WHEELS:** The basic pneumatic wheels that come with the starter kits compress easily and therefore constantly change diameter. The are also lumpy on the tread area. A robot of even moderate weight will force the wheel to compress as it drives creating unpredictable and uncorrectable changes in direction. **The solution:** Find wheels with hard, solid treads. The picture to the right shows many of the solid and thin tread wheels we've found on BrickLink and BrickOwl over the years. Many are UK only products. Please see the "COOL PARTS" article for more information. Click on the picture to see a larger view.

2. **USING A SINGLE AXLE MOUNT:** If your wheel is attached by an extended axel only (not supported on both ends) then you again, create unpredictable and uncorrectable changes in direction. Why? The LEGO axles are very flexible, and the motor drivers have play in them as well. Wheels must have a rigid, matching mount on both sides to ensure stability.

3. **IMPRECISE TRACKER WHEELS OR GLIDES:** Similar to "SINGLE AXLE MOUNT," if you have trailing wheels or glides (ball bearing or sliders or wheels) that are imprecise and not firmly mounted, it will wobble as you drive and respond to varations on frictional coeficient of the surface. This is particulary present in caster type wheels.

4. **NON CALIBRATED MOTORS:** The NXT and EV3 motors are all amazing, powerful, and precise. But they are not all the same. We published a very simple way to calibrate your motors. See the article here.

5. **OUT OF BALANCE ROBOTS:** Try to make your robots center of gravity in the center, seriously. A robot with a very heavy attachment or device on one side will cause the robot to track erratically.

6. **TEST TEST TEST:** After you have followed all the steps above do multiple line runs on a smooth surface twice the length of the table. Try to isolate what is causing any drift and correct with that setup. Then document it.

7. **IMPLEMENT A GYRO SENSOR:** This is very difficult to do and the Gyro sensors have their own drifing issues. However, there are hundreds links on Google on how to do this. Check them out: https://www.google.com/search?q=using+ev3+gyro+sensors+to+run+straight

Final Tips for Robot Construction

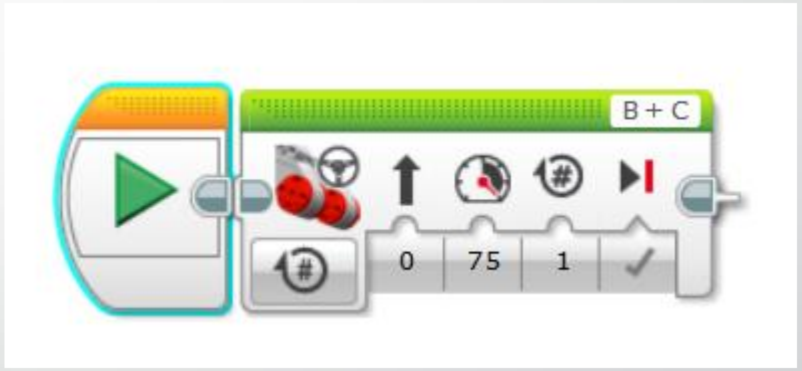
- Make sure your EV3 Brick can be removed. Your batteries will need to get changed eventually!
  - Use a rechargeable battery pack if possible
  - Make sure that both the programming port, charging port, and buttons on your EV3 are accessible. Taking apart a robot for any reason will hurt your consistency.
- Build a robot that can safely crash into walls and align itself.
- Think about sensors early in robot design. Placing your sensors in the right places will really help with using them as effectively as possible.

Robot Programming

- Robot programming varies greatly between teams in FLL. There are many different tools, languages, and environments that EV3 robots can be programmed in. Pick the one that matches your team's capabilities.
- The key to effective robot programming is using your sensors. It is impossible to be consistent on a FLL field without them.
- The difference between a good program and a bad program is that a good program always works, a bad program mostly works.

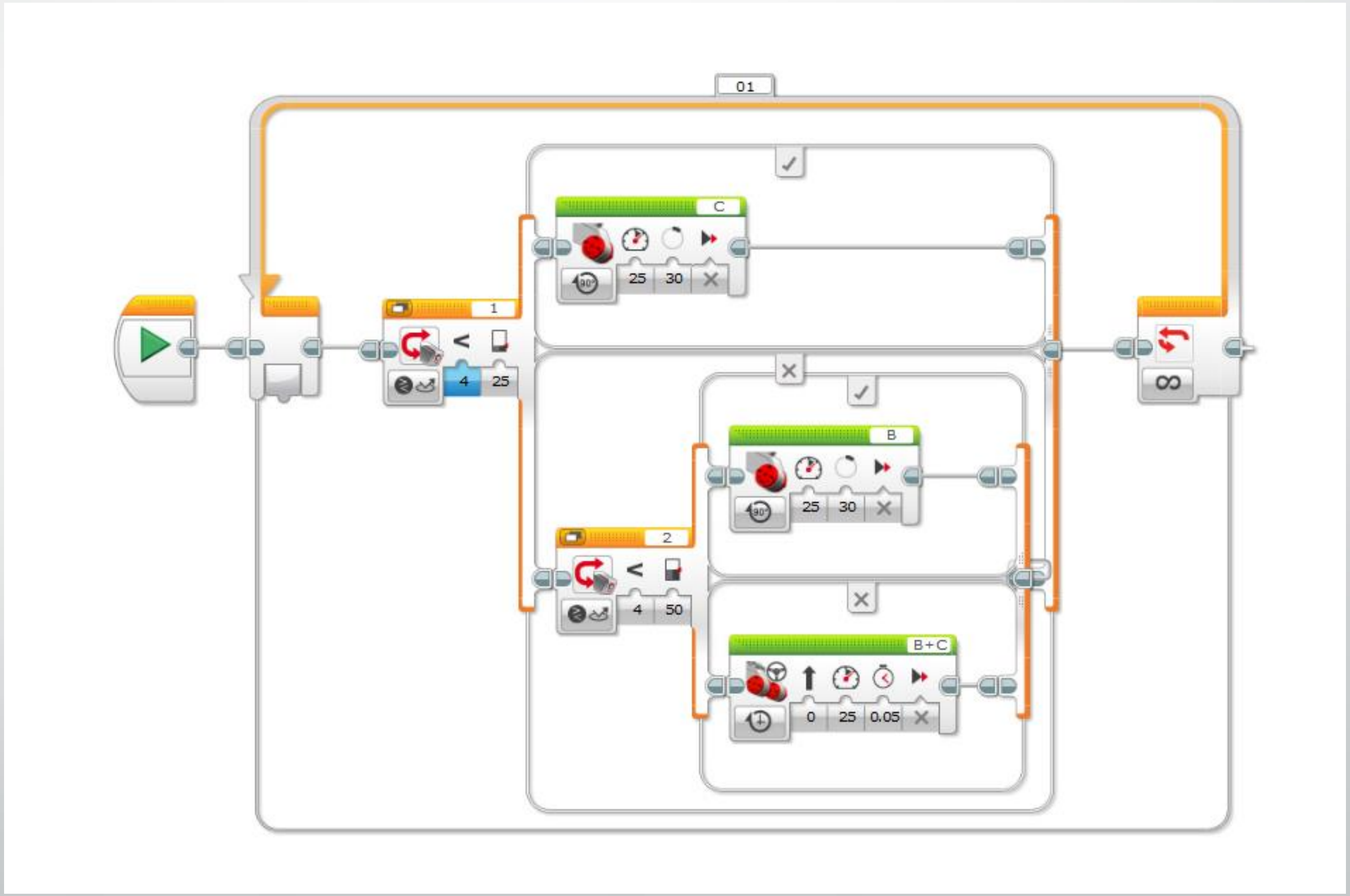
Getting Started with EV3 Programming

- The EV3 programming environment is a graphical program interface that is easy to use and learn.
- The default EV3 Environment is commonly referred to as EV3-G and is built on top of a NI LabView backend.
- The EV3 Software can be downloaded from the Lego Group here: https://www.lego.com/en-us/themes/mindstorms/downloads



Basic Line Following Program

- Complex Behaviors do not require complex programs!
- The program to the right is a simple line following program. This program only has 6 blocks.
- If you have trouble explaining a program it is to complex!
- Complex programs are hard to maintain and cause issues as the season progresses.



New this year – Programming in Other environments

- For the first time FLL is now allowing teams to program in other environments besides EV3 –G
- Judges have been instructed not to award any additional benefit to Teams who program in a different language.
- If you choose to use another environment it is purely for the advantages of the environment and to help students learn new skills.

Other Programming Environments

- **EV3Python** (EV3)
  EV3Python helps users familiar with Python programming to use this language to control an EV3 robot. This is done with the assistance of the Microsoft Visual Studio Code.
- **MakeCode** (EV3)
  Microsoft MakeCode is an online programming platform that can control the EV3 and other devices, such as Cue and the BBC micro:bit. Being able to use MakeCode on various devices might make it an attractive option. MakeCode uses blocks (like Scratch) or JavaScript (text) programming.
- **RobotC** (EV3, NXT, RCX)
  RobotC is a C-based programming language with a fully integrated software debugger that supports a range of different hardware platforms. Extensive documentation and online support is available. For more information, visit: http://www.robotc.net/.
- **leJOS** (EV3, NXT, RCX)
  LeJOS (pronounced like the Spanish word "lejos" for "far") is a tiny Java Virtual Machine that supports Java. For more information, see: http://www.lejos.org/ev3.php.
- **OpenRoberta** (EV3, NXT)
  Open Roberta is a free, drag and drop, cloud-based platform for programming LEGO EV3 and NXT robots. For more information, see: Open Roberta: A Review.
- **LabVIEW** (EV3, NXT)
  LabVIEW for LEGO MINDSTORMS (LVLM) and LabVIEW for Education (LV4E) are visual programming environments. The EV3 Software was built in LabVIEW, so LVLM provides a great next step for students who are familiar with that programming language and ready for something more powerful and versatile. Browse all LabVIEW posts on this site.

Know the strengths of your coding platform

- EV3 –G Strengths:
  - Simple – easy to use platform
  - Single all inclusive software package works out of the box
  - Graphical User interface is accessible for all students
- EV3 – G Weaknesses:
  - Arithmetic Operations
  - Limited Sensor Functionality
  - Programs are hard to read and debug

Programming in Python

- Python is an easy to learn full featured language that is commonly used for a variety of applications in both industry and academia.
- Setting up an EV3 for Python programming is trivial.
  - Format a microSD card with the EV3 python image
  - Insert the SD Card into the EV3
  - To revert to the default EV3 image simple remove the SD Card
- Advantages of Python Programming
  - Easier to handler arithmetic operations
  - Better control of EV3 Sensors
- Python for EV3 runs on Debian. Use your EV3 just like any other Debian system.

Robot Design Judging

- New this year…there will not be a table in the robot design judging room
- Be able to explain robot designs, programming
- Engineering journal
  - journal is not required for Judging. Judges have little time to review each team, not enough time to go through the Journals. Successful teams will make a short concise document for each judged category. Teams should still bring their robot and attachments as well as their code to the judging room
  - Sketches, designs, ideas, designs that failed, pictures/drawings, calculations, code examples

Advice

- Keep things simple!
- Its ok for team members to fail
- Teach debug/troubleshooting techniques than solving problems for them
- Know your abilities and what to take on with what you have
- Set a schedule and stick to it
- Read the rules - multiple times
- Practice, Practice, Practice! (well enough in advance)
  - Practice running robot under time
  - Practice presentation in front of other people
  - Review rubrics (https://firstinspiresst01.blob.core.windows.net/fll/2020/first-lego-league-rubrics.pdf)

Click to add text

For a copy of this presentation Scan the QR code below

References

- http://www.legoengineering.com/alternative-programming-languages/
- https://docs.google.com/spreadsheets/d/1OxyXEIB5pHzOSpAuOKY_2XeK1Uomv7lzHLVG1ZKLFH8/edit#gid=0
- http://flltutorials.com/
- https://techbrick.com/fll-resources/fll2019