Preparing for Competition

Colton Cappa

Pembroke High School

Preparing for Competition

1 Introduction

Picture this: you and your team have been working hard, slaving away for hours on end to prepare for a major robotics competition, for example, Botball, and you are down to the final preparations. You obviously want to be in the best possible condition as a team before the tournament, and you want to have a shot at winning, or at least doing well. How well prepared are you? How can I get as prepared as possible before competition day?

For another example, imagine yourself designing a new robot which you plan on selling to the public. Is it safe mechanically and does the code run smoothly? Do you and/or your team know if you are in good shape? You owe it to your team, or whoever your creation is designed for to be able to respond to any and all of those questions with an immediate and affirmative response.

2 Preparing Code

One major part of preparations for robotics and even just coding in general, is testing. In all segments of your work one should be constantly testing and observing both programming and mechanical aspects of your design. The first major part of this, would be testing your programming. It is of great importance to make sure your program is sound, and can replicate itself in most scenarios.

2.1 Testing

Testing your code is one of the most important aspects of robotics, or of any program in general you may develop in your future. For instance, let's say at some point in the future you plan on designing an application for the Apple App Store. You need to make sure your code works, or your application will not run. You also need it to be able to repeat itself in a vast majority of scenarios.

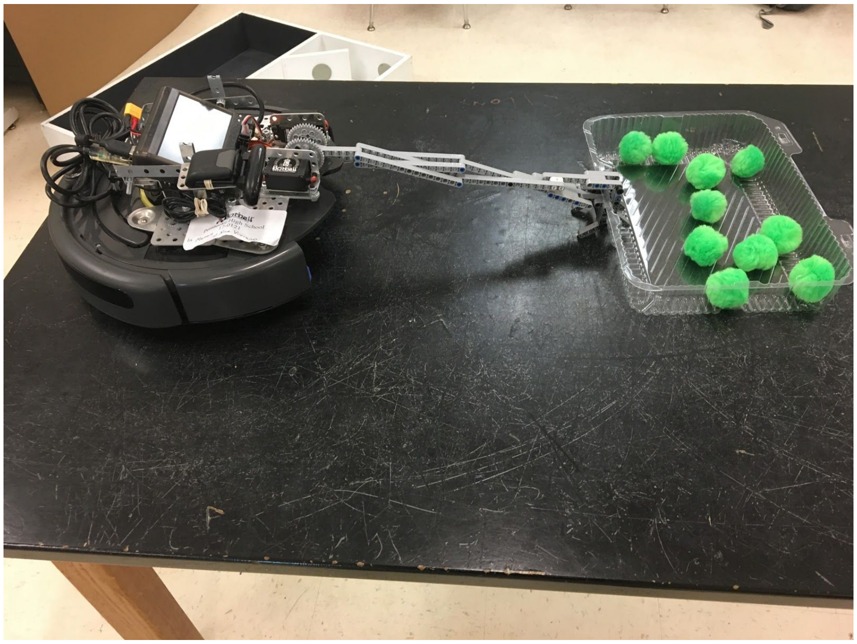
2.2 Reproducibility

A fundamental ideal in programming is reproducibility. This is defined as the ability of an entire analysis of an experiment or study to be duplicated, either by the same researcher or by someone else working independently. In the medium of robotics, this means being able to reproduce the same exact desired event that the written code told the robot to do.

The robot must be able to do the same action by using the same written code in multiple instances. This is one of the backbones of coding that can ultimately only be proven or disproven through testing of the code. If the code of your robot works to perform the exact same functions in every instance of testing, then its reproducibility is sound and one major preparation before completing any code or robot.

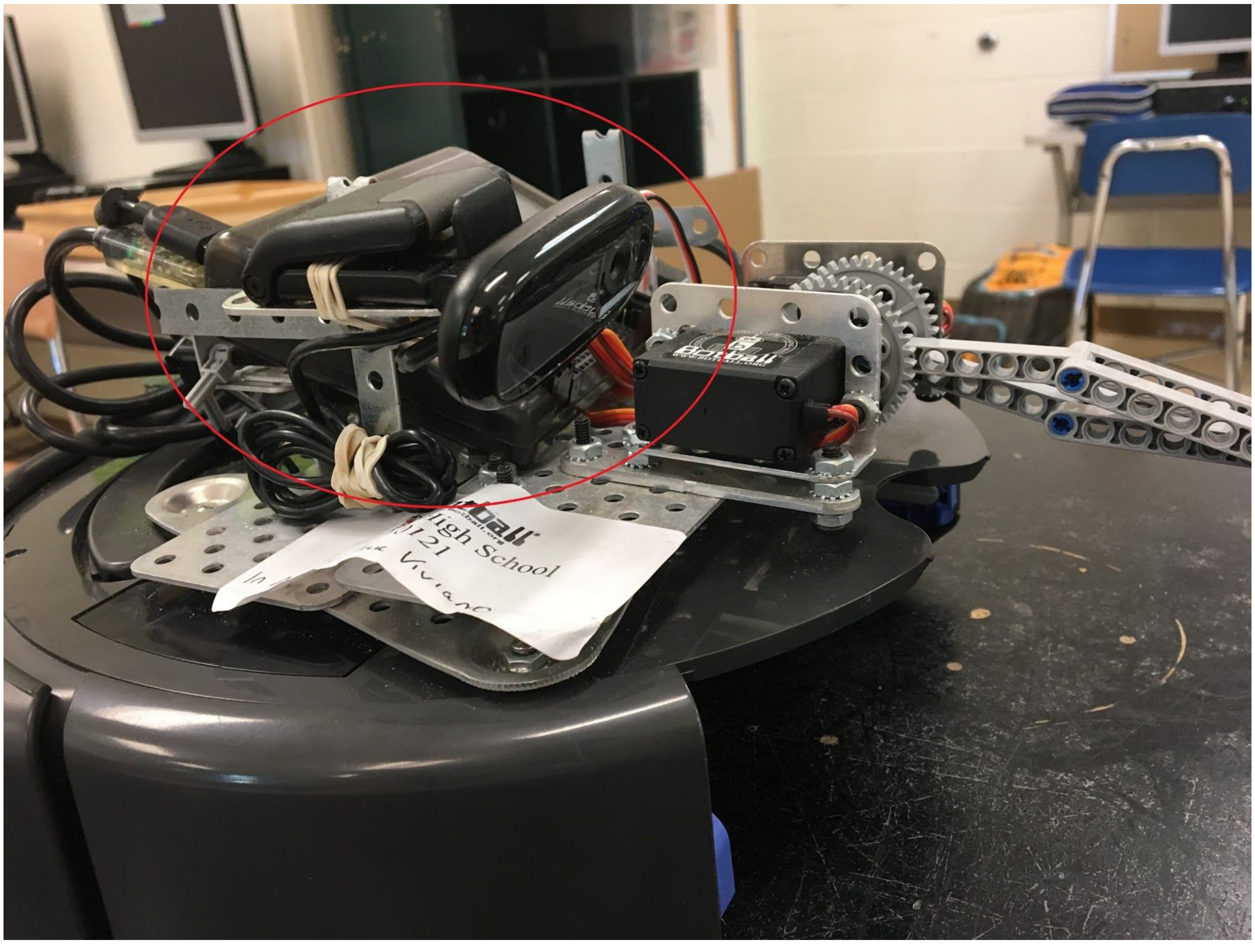
2.3 Murphy's Law

Another fundamental ideal of programming, especially for robotics, is to prepare for many possible outcomes. I was always told how it's important to "Be Prepared" as a Boy Scout, and this same principle carries right over into Robotics. When programming robots, you need to expect the worst possible outcomes, and plan for them in succession. You must prepare for a multitude of different things that may go wrong, as in many cases, as stated by Murphy's Law, "whatever can go wrong, will go wrong," (1). So it can be expected that throughout your different tests you may encounter various problems. The true way in which robotics works is to solve each problem you're faced with, and once perfected, move on to the next challenge, and so on.



Let's say for example you are programming a robot to perform a menial task such as in the Botball competition this year, like picking up a bucket, and moving it from one place to another. This comes with a margin of error, we can vary with a few different factors, and many of which are out of your control. In order to account for this, you must create some fail-safe procedure or mechanism to prepare for and counteract any of these factors.

For example, in my Botball example, say for instance, the robot’s arm reaches out to grab the bucket. To be sure if the robot did truly pick up the bucket, we were able to use a color-sensitive camera to determine if the bot did actually pick up the bucket. Within the code we can accomplish this with a few simple procedures, and if the robot did not pick up the bucket, we returned it to the location of the bucket and had it try again, multiple times until it eventually has



success. It is through different challenges like these that we can truly prepare both robots, and ourselves for taking on or facing problems.

2.4 Cessation of One's Coding

As you continue throughout testing your code, you and/or your team should be sure to delve into the possible scenarios in which your code or the robot you coded faced any problems, and be able to solve these problems. In doing so you want your code to be as reproducible as possible and be able to handle any such problems it may face even if not seen in testing.

3 Teams

In robotics, more specifically in competitions, your team is essential to your success. It is very important to have good team chemistry with your team and to be on good enough terms to cooperate and to deal with the current tasks. "To account for this, team-building is now a frequent practice of many organizations and can include activities such as bonding exercises and even overnight retreats to foster team cohesion," (2). To the extent that these exercises are meaningful to members, they can be effective in improving team motivation and productivity.

3.1 Team Preparedness

When preparing for competition in robotics, it is essential that the group can function cooperatively and efficiently to complete any task set before them. In order to do this you must be able to have leaders whom the team can listen to and to make decisions for the well being of the team. These leaders must be a good example for the others and must be able to decisively choose jobs for different members of the team to continue on.

3.2 Team Structure

A team in robotics should generally be structured as follows:

- Leaders who distribute clear and concise instruction and are examples for others to follow.
- Coders who primarily focus on writing and editing code that should be used in the robot.
- Builders who create and design mechanical aspects of robots and put the pieces together.
- Testers who execute code on the robots to see if it does the desired task, and inform the other groups of changes to make or add.
- Strategists who can decide for the team what their code should be attempting to accomplish and what the goals of the robot are.(3)

3.2 Jobs

In a team, it is important that each member has a job to do and they are fully capable of completing it, or they must be taught to do this. It is the leader's job to make sure they assign tasks different team members are able to do, or to teach them how to fulfill a task that needs to be completed.

These jobs can include a variety of things such as the ones stated previously in team structure, but also more specific, need based jobs. A leader should specifically give each member a more specific task that allows for the robots to be manufactured in a way that they will function properly, by each separate phase of its creation being accomplished by a team member or a smaller team dedicated to that phase alone.

The creation of the robots should also be separated into phases by which different groups can work simultaneously and separately in order to keep both groups working as efficiently as possible, and when needed they come together to test their finished products. Once they complete their testing, once more they can break off to their own separate groups to make changes or new modifications and eventually come back together to test again.

4 Conclusion

Relating back to the example of a Botball team, once you have finally come together as a team, tested your robot, and prepared as best as possible, you would indeed be on the path to greater success in the robotics field. With these devices in mind you should be able to perform better in robotics, whether working individually, in a team, or in a competition. You should also hopefully have learned a thing or two about life as well, which is an important lesson to take out of robotics. You should be prepared for whatever life may send your way, and be prepared and readily able to solve each problem you are faced with, and then continue on to solve the next.

(1) "Murphy's law". *The American Heritage® Dictionary of Idioms by Christine Ammer*. Houghton Mifflin Company. 2 Jun. 2017. <Dictionary.comhttp://www.dictionary.com/browse/murphy-s-law>.

(2) Boundless. "Team-Based Structure - Boundless Open Textbook." *Boundless*. Boundless, 31 May 2016. Web. 01 June 2017. <https://www.boundless.com/management/textbooks/boundless-management-textbook/organizational-structure-2/common-organizational-structures-25/team-based-structure-149-3978/>.

(3) "Clipart Panda - Free Clipart Images." *Clipart Panda - Free Clipart Images*. N.p., n.d. Web. 01 June 2017. <http://www.clipartpanda.com/>.

Note* Most of the information is original and of my own individual knowledge.