# Botball Online Project Documentation Handbook

version 1.0.4



The KISS Institute is a 501c3 nonprofit organization. Our mission is to improve the public's understanding of science, technology, engineering, and math; develop the skills, character, and aspirations of students; and contribute to the enrichment of our school systems, communities, and the nation.

# Botball Online Project Documentation

Throughout the Botball season, your team will use the Team Home Base to post documentation activities detailing your progress. Botball Online Project Documentation(BOPD) activities are a required element of the Botball program and factor in to each team's overall score at their regional tournament. Additionally, awards recognizing excellent documentation will be given out in each region.

## Scoring

Each online documentation activity includes a scoring rubric that outlines what points may be scored and how. Your submissions will be evaluated to see if they meet each requirement given by the rubric. Your team will receive the indicated point value for each requirement met, or 0 points if the requirement is not fully met. Partial credit will not be given for any requirement.

Part of the scoring for the Period 2 software documentation is presented below.

| **Flow Chart or Pseudo-Code** | |
|---|---|
| The flow chart or pseudo-code has a meaningful title or caption. | 1 |
| The flow chart or pseudo-code presents each step in a clear order. | 2 |
| The flow chart or pseudo-code clearly indicates branches, conditional steps, and/or decisions. | 2 |

If your flow chart or pseudo-code meets the first two requirements, but fails to completely meet the third, then you would earn a total of 3 points: 1 point for the first requirement, 2 points for the second requirement, and no points for the third requirement. Again, partial credit will not be given for any requirement.

## Example Submissions

With each activity, we have included an example submission with a note on how that submission would score. These example submissions should be help you to gauge the appropriate length and thoroughness for your responses to each documentation activity.

## Documentation Periods and Deadlines

Between your Botball workshop and tournament, there will be **3 documentation periods**.

All documentation deadlines for all regions will fall on a **Wednesday, at 3pm Central Time**, unless otherwise noted in the Team Home Base. Documentation for any period may be submitted at any time prior to the deadline for that period.

**NOTE:** Each documentation period has a different number of required activities. Period 1 requires two submissions, Period 2 requires three submissions and Period 3 requires four submissions.

Check the Team Home Base (http://homebase.kipr.org) for documentation deadlines for your region.

# Documentation and Overall Score

Your score for each period will be the average of your percentage scores for each assignment in that period. This score will be used to calculate the overall winner of the Botball tournament. Scores from online documentation combine with your onsite presentation score to make up 1/3 of your overall score according to the following formula:

$$\text{Doc Score} = 0.2\left(\frac{n - P_1\ \text{Rank} + 1}{n}\right) + 0.23\left(\frac{n - P_2\ \text{Rank} + 1}{n}\right) + 0.27\left(\frac{n - P_3\ \text{Rank} + 1}{n}\right) + 0.3\left(\frac{n - \text{Onsite Rank} + 1}{n}\right)$$

| **Variable Definitions** | |
|---|---|
| $n$ | The number of participants in the tournament |
| $P_x$ Rank | Your ordinal rank in documentation for the xth documentation period |
| Onsite Rank | Your ordinal ranking for your onsite presentation |

# Submitting your Documentation

All online documentation should be submitted through the Botball Team Home Base

To submit your documentation:

1. Log in to the Team Home Base (http://homebase.kipr.org) using your team code and password.
2. Find the **2010 BOPD** area for your region.
3. Find the documentation activity for which you want to upload your response.
4. Click on the **Choose File** button to select the file you wish to upload.
5. Click on the **Upload the File** button to upload your file.

Once your file has been uploaded you should see a heading labeled **Submission**, with your file linked below it. We recommend that you click on the link to your file to make sure your submission uploaded correctly.

If you have any problems submitting your documents contact the KISS Institute office at 405-579-4609 or support@kipr.org for advice and assistance

## Deleting a Submission

If you wish to delete your submission and submit a different file, click the **X** next to the download link for your file.

# Learning From Experience

Judges and teams from previous years have noted the following recommendations for Botball teams. Learn from the past so you do not have to repeat it.

- Remember to check and follow the minimum requirements for each submission. Documents that do not meet the minimum requirements may not be scored (at the judges' discretion).
- When converting documents to PDFs, remember to open the resulting PDFs and double check that nothing has been lost or changed during the conversion process.
- Do not wait until the last minute to figure out how to create PDFs.
- Do not wait until the last minute to create and submit your documentation. Unexpected problems are likely to arise.
- If you have any problems creating or submitting the required documentation, contact the KISS Institute offices at 405-579-4609 or email support@kipr.org for advice and assistance.

# Making PDFs

All online documentation submissions must be submitted as PDF documents. PDF documents can be created in a variety of ways, depending on which software and operating systems you have available. Here are some of our favorites.

If you need help with creating PDFs, call the KISS Institute offices at 405-579-4609 or email support@kipr.org

Make sure that you have figured out how you plan to create the required PDFs before your online documentation deadline.

**Creating PDFs on MacOs X**

The ability to convert almost any printable file to a PDF document is built in to MacOs X. To do this:

Open the document.

1. Choose File > Print.
2. Find the PDF drop down on the bottom left of the Print menu
3. From the PDF Menu, choose Save as PDF. This creates a high quality PDF of your document.

If this PDF ends up being too large, you can usually create a smaller PDF by choosing Compress PDF instead of Save as PDF

Note: It's always a good idea to review the PDF that you've created to make sure it turned out as you expected. You may need to change some of the formatting and printing settings.

For more information see this article http://www.apple.com/pro/tips/saving_as_pdf.html

**Creating PDFs on Windows with PDFCreator**

PDFCreator is a free software tool for Windows that allows you to create a PDF version of almost any printable document. PDFCreator works by creating a virtual printer for your computer. Any documents printed to this virtual printer will be saved to your computer as PDFs.

You can download PDFCreator from http://www.pdfforge.org/pdfcreator

Once you've installed PDFCreator ( You may need to reboot), you should be able to save a document as a PDF by doing the following:

1. Open the document.
2. Choose File < Print.
3. You should now see a printer named PDFCreator. Select that printer and click OK.
4. That should bring up the PDFCreator dialog. Most likely, you should be able to just use the default options and click Save.
5. Choose an appropriate name and location to save the document, and you're done

For more information about PDFCreator, check out the download url given above.

**Creating PDFs on Windows with Office 2007**

Office 2007 can save documents as a PDF file, but you may have to install a free add-in from Microsoft.

In order to save a document as a PDF in Office 2007:

1. Open the document.
2. Select Save to PDF from the Office or File menu, depending on the program you are using.

**Creating PDFs with Google Docs**

Google Docs allows you to download your document as a PDF:

1. Open the document.
2. Select File > Download File As > PDF

**Creating PDFs with OpenOffice.org**

The open source office application OpenOffice.org includes an export to PDF feature:

1. Open the document.
2. Select File > Export as PDF

# Period 1

## Activities and Examples

# Period 1: Upload Your Project Plan

## Overview

This should be a document that covers the **entire** project **and** shows your actual progress up to the current date. The Botball Project Plan should be a living document that grows and changes over the course of the Botball season. As you progress, goals will be added and removed. Your schedule will change, tasks will get completed, etc. When the judges look at your plan, they won't be scoring how much progress you've made. Instead, they'll be focused on **how well you've documented** your progress. Make sure the judges can easily see your team's progress from period to period. Consider using color-coding, notes, or other elements to update your plan.

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

1. The submitted file must consist of a single file.
2. The submitted file must be a PDF document.
3. The submitted file must be named according the pattern **00-0000_P1_ProjectPlan.pdf**, substituting your team code for 00-0000.
4. The submitted file cannot cannot contain pages formatted to print more than 11" wide.

## Scoring

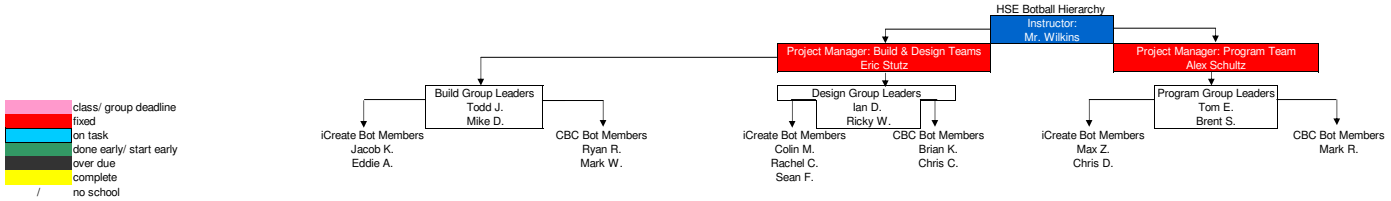
| **Project Plan Basic Requirements** | |
|---|---|
| 1. The submitted document meets all minimum requirement. | 1 |
| 2. All pages include any relevant labels and headers. | 2 |
| 3. The plan includes a comments area for status updates, explanations, and general comments. | 2 |
| 4. The comments area is easy to identify and clearly related to the subject of the comments. | 1 |

| **Project Milestones** | |
|---|---|
| 5. The plan includes project milestones that are measurable. | 1 |
| 6. Project milestones span the **entire** project from start to finish. | 2 |
| 7. Project milestones are clearly distinguishable from their related tasks. | 1 |

| **Project Task Lists** | |
|---|---|
| 8. The plan includes a list of tasks that clearly relate to a specific milestone and show how that milestone will be achieved. | 3 |
| 9. Task lists span the **entire** project from start to finish. | 1 |
| 10. Task lists include due dates for **everything** even if dates change from period to period. | 1 |
| 11. Task lists indicate which team member(s) is assigned to each task. | 1 |
| 12. Task lists show the status of each task (completed, behind schedule, ahead of schedule, etc.). | 2 |

## Example Scoring Notes

The following example would score points for all of the above requirements.




# 2009 Botball Project Plan

| | Task/ Event | Person in Charge | Prereq. | Week 1 | | | | | Week 2 | | | | | Week 3 | | | | | Week 4 | | | | | Week 5 | | | | | Week 6 | | | | | Week 7 | | | | | Week 8 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 19-Jan | 20-Jan | 21-Jan | 22-Jan | 23-Jan | 26-Jan | 27-Jan | 28-Jan | 29-Jan | 30-Jan | 2-Feb | 3-Feb | 4-Feb | 5-Feb | 6-Feb | 9-Feb | 10-Feb | 11-Feb | 12-Feb | 13-Feb | 16-Feb | 17-Feb | 18-Feb | 19-Feb | 20-Feb | 23-Feb | 24-Feb | 25-Feb | 26-Feb | 27-Feb | 2-Mar | 3-Mar | 4-Mar | 5-Mar | 6-Mar | 9-Mar | 10-Mar | 11-Mar | 12-Mar | 13-Mar |
| 1 | **Doc Submission 1** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | **Doc Submission 2** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | **Doc Submission 3** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | **Game** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4a | Go over Rules | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4b | Test over Rules | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4c | Break up into groups | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4d | Strategy | Both Design Teams | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | **Design** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5a | iCreate Design | Ian; Colin, Rachel, Sean | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5ai | Claw Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5aii | Bin Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5b | CBC Design | Ricky; Brian, Chris | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5bi | Chassis Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5bii | Claw Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5biii | Puff Ball Holder Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | **Construction** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6a | iCreate Build | Mike; Jacob, Eddie | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6ai | Claw Construction | | 4a, 5ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6aii | Bin Construction | | 4a, 5aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6b | CBC Build | Todd; Ryan, Mark | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6bi | Chassis Construction | | 4a, 5bi | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6bii | Claw Construction | | 4a, 5bii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6biii | Puff Ball Holder Construction | | 4a, 5biii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | **Program** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7a | Test programs | Program team | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7ai | Program basic movement | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aii | Program Percision turns | | 4a, 7ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aiii | Program Color tracking | | 4a, 7ai, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aiiii | Program sonar | | 4a, 7ai, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7b | iCreate Program | Brent; Max, Mark | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7bi | Program create movements | | 4a, 7aiii, 7aiiii, 6 | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7bii | Program claw movements | | 4a, 7aii, 6ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7c | CBC Program | Alex; Tom, Chris | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7ci | Program CBC movement | | 4a, 6bi, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7cii | Program Claw movement | | 4a, 6bii, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7d | Robot Testing and Debugging | Class | 4a, 7b, 7c | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | Weekly Comments: | | | | | Weekly Comments: | | | | | Weekly Comments: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | This week we were excited to start this new competition. The ideas in all the groups were flowing a lot. | | | | | This week we started with the strategies of both the iCreate and CBC robots. We lost a little bit of time due to school delays but managed to get ever thing squared away | | | | | This week the class started on the design of the iCreate bot early due to an unanimous decision amongst the Design Team. The iCreate bot got a much needed head start. Program Team was able to learn the CBC program and start on the programming of simple tasks. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

# Period 1: Upload Design Concepts

## Overview

Upload a design document focusing on your early designs. Supply a list of three design concepts discussed at early Botball planning meetings. One of the design concepts should be a rejected one.

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

1. The submitted file must consist of a single file.
2. The submitted file must be a PDF document.
3. The submitted file must be named according the pattern **00-0000_P1_Mechanical.pdf**, substituting your team code for 00-0000.

## Scoring

| Basic Requirements | |
|---|---|
| 1. The submitted document meets all minimum requirements | 1 |

| Evaluation Section | |
|---|---|
| 2. The section begins with the title *Evaluation,* clearly separated from text. | 1 |
| 3. The section includes how your team evaluated your design concepts. | 3 |

| First Initial Design Concept Section | |
|---|---|
| 4. The section begins with the title *First Initial Design Concept,* clearly separated from text | 1 |
| 5. The section includes a description on what points your robot(s) will score. | 2 |
| 6. The section includes the key features your robot will have for scoring points. | 2 |
| 7. The section explains the most difficult aspects of implementing this design. | 2 |

| Second Initial Design Section | |
|---|---|
| 8. The section begins with the title *Second Initial Design,* clearly separated from text. | 1 |
| 9. The section includes a description on what points your robot(s) will score. | 2 |
| 10. The section includes the key features your robot will have for scoring points. | 2 |
| 11. The section explains the most difficult aspects of implementing this design. | 2 |

| Rejected Design Section | |
|---|---|
| 12. The section begins with the title *Rejected Design,* clearly separated from text. | 1 |
| 13. The section includes a description on what points your robot(s) will score. | 2 |
| 14. The section includes the key features your robot will have for scoring points. | 2 |
| 15. The section explains why your team rejected this design. | 2 |

## Example Scoring Notes

The following example would score points for all of the above requirements.

# Period1: Design Concepts Example

## Evaluation:

To have ideas to evaluate, first you need to have ideas. To generate possible ideas, the team sat in a circle and people just started coming up with ideas. As each idea was suggested, we added it to the dry erase board. After we had several ideas on the board, we then sat and looked at them for 3 minutes in silence. After that, if we had a suggestion about an idea we raised our hands and offered our suggestion, often about combining ideas or adding/dropping ideas. After that we voted on all of the ideas until we had just two left. Our final two ideas were the all encompassing robot that can score all of the points and the simple plow robot. One of my favorite rejected ideas involved projectiles.

## First Initial Design Concept:

Our first accepted design concept was the "omnibot" that will score all of the points. First it will put sorbents on slicks while collecting ducks. Then it will deposit the dirty ducks, pick up clean ducks and put the clean ducks on the line. Finally it will collect Botguy and the frogs for deployment next to the ducks. The robot's key scoring feature will be the arm with 6 degrees of freedom with a claw at the end. The most difficult aspect will be building the arm so that it will work and have enough strength to lift Botguy.

## Second Initial Design Concept:

The second accepted design concept was a plow robot. This robot will drive around pushing the game pieces around. The robot will try to push the sorbents on to the slicks and the ducks off all at the same time. Towards the end of the round the robot will just try to push the dirty ducks into the cleaning center. The key feature of this robot will be the plow. It will need to be strong and set to just the right angle. The most difficult aspect though, will be the programming. Our programmer said he could handle it, so we are keeping this concept for now.

## Rejected Design Concept:

One of our rejected designs involved shooting a projectile to lasso Botguy. The lasso would be shot a Botguy and we would then drag him into the scoring area. The key feature would be the lasso and lasso shooting mechanism. We rejected this idea for two reasons. The first was the difficulty to lasso Botguy every time without fail. We used a tethered projectile last year and it turned out to be to hit and miss to use at the tournament. The second reason we rejected this idea was for how few points this robot would score. We felt that we needed to score more points than just Botguy to be competitive at the tournament.

# Period 2
## Activities and Examples

# Period 2: Upload Your Project Plan

## Overview

This should be a document that covers the **entire** project **and** shows your actual progress up to the current date. The Botball Project Plan should be a living document that grows and changes over the course of the Botball season. As you progress, goals will be added and removed. Your schedule will change, tasks will get completed, etc. When the judges look at your plan, they won't be scoring how much progress you've made. Instead, they'll be focused on **how well you've documented** your progress. Make sure the judges can easily see your team's progress from period to period. Consider using color-coding, notes, or other elements to update your plan.

## File Submission Requirements

Your submission will score 0 points if it does not meet the following requirements.

- The submitted file must consist of a single file.
- The submitted file must be a PDF document.
- The submitted file must be named according the pattern 00-0000_P2_ProjectPlan.pdf, substituting your team code for 00-0000.
- The submitted file cannot be more than 11" wide.

## Scoring

| Project Plan Basic Requirements | |
|---|---|
| 1.The project plan will meet all the basic requirements outlined in the first period. | 1 |
| 2.The plan will include an updated comments section relevant to the current period that explains/updates status and/or for general comments. | 4 |

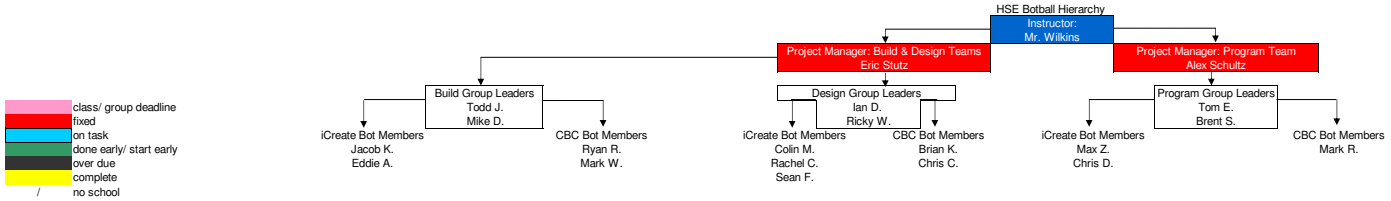
| Project Milestones | |
|---|---|
| 3.Project milestones will meet the requirements outlined in period one. | 1 |

| Project Task Lists | |
|---|---|
| 4.Project task lists will meet the requirements outlined in period one. | 1 |
| 5.Task lists will include updates relevant to any changes in milestones or the steps needed to achieve milestones. ** | 2 |
| 6.Task lists will include updated due dates for everything.** | 2 |
| 7.Task lists will include updated team member assignments. ** | 2 |
| 8.Task lists will include updated status of each task. | 2 |

**If no updates were needed between period 1 and 2, that must be clearly stated in the comments section.

## Example Scoring Notes

The following example would score points for all of the above requirements.




# 2009 Botball Project Plan

| | Task/ Event | Person in Charge | Prereq. | Week 1 | | | | | Week 2 | | | | | Week 3 | | | | | Week 4 | | | | | Week 5 | | | | | Week 6 | | | | | Week 7 | | | | | Week 8 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 19-Jan | 20-Jan | 21-Jan | 22-Jan | 23-Jan | 26-Jan | 27-Jan | 28-Jan | 29-Jan | 30-Jan | 2-Feb | 3-Feb | 4-Feb | 5-Feb | 6-Feb | 9-Feb | 10-Feb | 11-Feb | 12-Feb | 13-Feb | 16-Feb | 17-Feb | 18-Feb | 19-Feb | 20-Feb | 23-Feb | 24-Feb | 25-Feb | 26-Feb | 27-Feb | 2-Mar | 3-Mar | 4-Mar | 5-Mar | 6-Mar | 9-Mar | 10-Mar | 11-Mar | 12-Mar | 13-Mar |
| 1 | **Doc Submission 1** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | **Doc Submission 2** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | **Doc Submission 3** | Project Managers | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | **Game** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4a | Go over Rules | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4b | Test over Rules | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4c | Break up into groups | Class | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4d | Strategy | Both Design Teams | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | **Design** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5a | iCreate Design | Ian; Colin, Rachel, Sean | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5ai | Claw Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5aii | Bin Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5b | CBC Design | Ricky; Brian, Chris | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5bi | Chassis Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5bii | Claw Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5biii | Puff Ball Holder Drawings | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | **Construction** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6a | iCreate Build | Mike; Jacob, Eddie | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6ai | Claw Construction | | 4a, 5ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6aii | Bin Construction | | 4a, 5aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6b | CBC Build | Todd; Ryan, Mark | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6bi | Chassis Construction | | 4a, 5bi | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6bii | Claw Construction | | 4a, 5bii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6biii | Puff Ball Holder Construction | | 4a, 5biii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | **Program** | | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7a | Test programs | Program team | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7ai | Program basic movement | | 4a | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aii | Program Percision turns | | 4a, 7ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aiii | Program Color tracking | | 4a, 7ai, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7aiiii | Program sonar | | 4a, 7ai, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7b | iCreate Program | Brent; Max, Mark | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7bi | Program create movements | | 4a, 7aiii, 7aiiii, 6 | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7bii | Program claw movements | | 4a, 7aii, 6ai | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7c | CBC Program | Alex; Tom, Chris | | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7ci | Program CBC movement | | 4a, 6bi, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7cii | Program Claw movement | | 4a, 6bii, 7aii | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7d | Robot Testing and Debugging | Class | 4a, 7b, 7c | / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Weekly Comments:

This week we were excited to start this new competition. The ideas in all the groups were flowing a lot.

Weekly Comments:

This week we started with the strategies of both the iCreate and CBC robots. We lost a little bit of time due to school delays but managed to get ever thing squared away

Weekly Comments:

This week the class started on the design of the iCreate bot early due to an unanimous decision amongst the Design Team. The iCreate bot got a much needed head start. Program Team was able to learn the CBC program and start on the programming of simple tasks. program team is waiting

Weekly Comments:

This week we had a set back when the designs we origianlly designed for the CBC did not work. We had to start trouble shooting the problem and come up with a new design that everyone agreed would work. The iCreate bot should be done soon next week along with the CBC so that we can hand it over to the program team. I felt that it would be best to move the deadline for the iCreate in hopes that it would get done. Program team has designed the programs but do to the build and design teams using the robots, the program team is unable to fully test their work.

Weekly Comments

This week the iCreate was put on hold for a bit so that the Build Teams and Design Teams could focus on agreeing on a design that will work. The Build Team got started on the CBC bot and everyone feels it will be strong. The iCreate bot got done once the teams were balanced again. I talked to Alex along with Mr. Wilkins and we decided that it would be a good idea to move the deadline back for the CBC. It will be done by the beginning or middle of the week. Program Team has completed the sonar and color camera programs. Unfortunatly the Program Team was unable to test any of the programs developed.

# Period 2: Upload Mechanical System Designs

## Overview

Upload a design document focusing on the mechanical systems that you have built for your robot. Your document must include four sections as described below. Each section should include both an image and and at least a paragraph of text describing the mechanical system. Systems may be from different robots.

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

1. The submitted file must consist of a single file.
2. The submitted file must be a PDF document.
3. The submitted file must be named according the pattern **00-0000_P2_Mechanical.pdf**, substituting your team code for 00-0000.

## Scoring

| **Overall Requirements** | |
|---|---|
| 1. The submitted document meets all minimum requirements | 1 |

| **Drive Train Section** | |
|---|---|
| 2. The section begins with the title *Drive Train,* clearly separated from text. | 1 |
| 3. The section includes a clear picture of the drive train of a robot. | 1 |
| 4. The section includes text describing your drive train. | 2 |
| 5. The section explains why your team chose this type of drive train. | 2 |
| 6. The section describes how your team has/will evaluate your drive train. | 2 |

| **Effector Section** | |
|---|---|
| 7. The section begins with the title *Effector,* clearly separated from text | 1 |
| 8. The section includes a clear picture of the effector of a robot | 1 |
| 9. The section has includes text describing your effector | 2 |
| 10. The section explains why your team chose this type of effector | 2 |
| 11. The section describes how your team has/will evaluate your effector | 2 |

| **Sensor Mount Section** | |
|---|---|
| 12. The section begins with the title *Sensor Mount,* clearly separated from text. | 1 |
| 13. The section includes a clear picture of the sensor mount on a robot. | 1 |
| 14. The section includes text that describes your sensor mount. | 2 |
| 15. The section explains why your team chose this type of sensor mount. | 2 |
| 16. The section describes how your team has/will evaluate your sensor mount. | 2 |

**(Scoring continued on next page)**

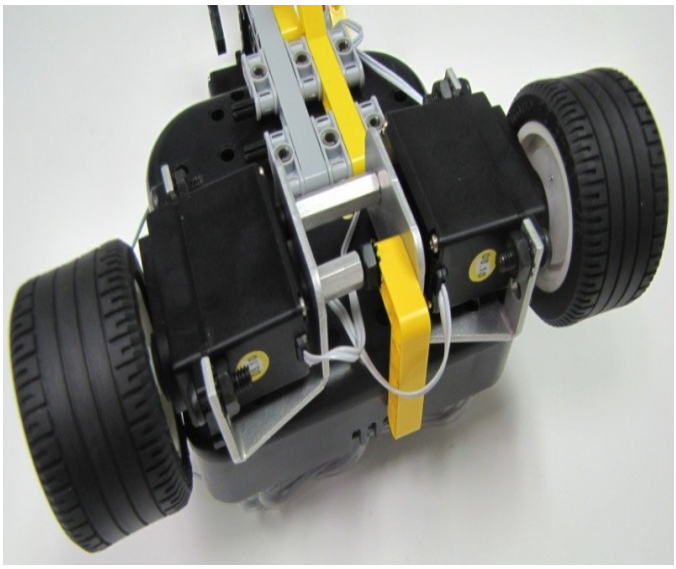
| Testing Section | |
|---|---|
| 17. The section begins with the title *Testing,* clearly separated from text. | 1 |
| 18. The section includes a clear chart or graph of your testing results. | 1 |
| 19. The section includes text describing how you gathered empirical data from testing. | 2 |
| 20. The section explains how your design has/will change based on gathered data. | 2 |

## Example Scoring Notes

The following example would score points for all of the above requirements.

# Period 2: Mechanical Systems Designs Example

## Drive Train:

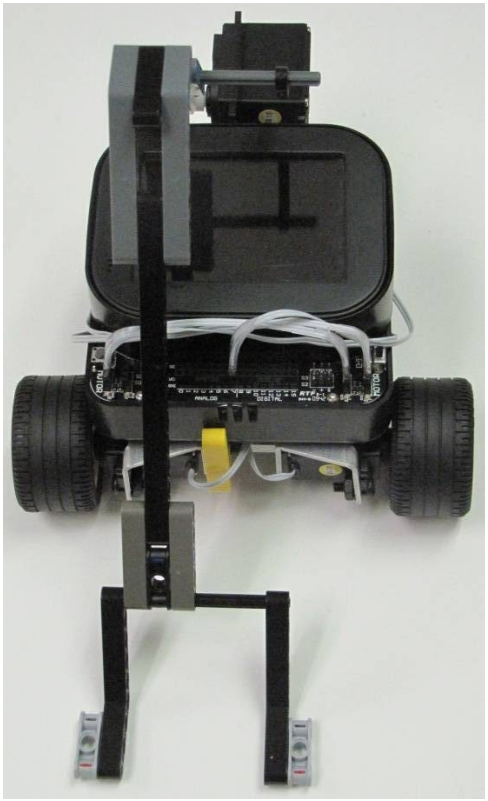


This is a picture of our drive train. We are using a direct drive between the wheels and the motors. The motors are mounted directly to the CBC. We chose to use this direct drive system because we wanted a small compact robot, and there are fewer things that could go wrong.

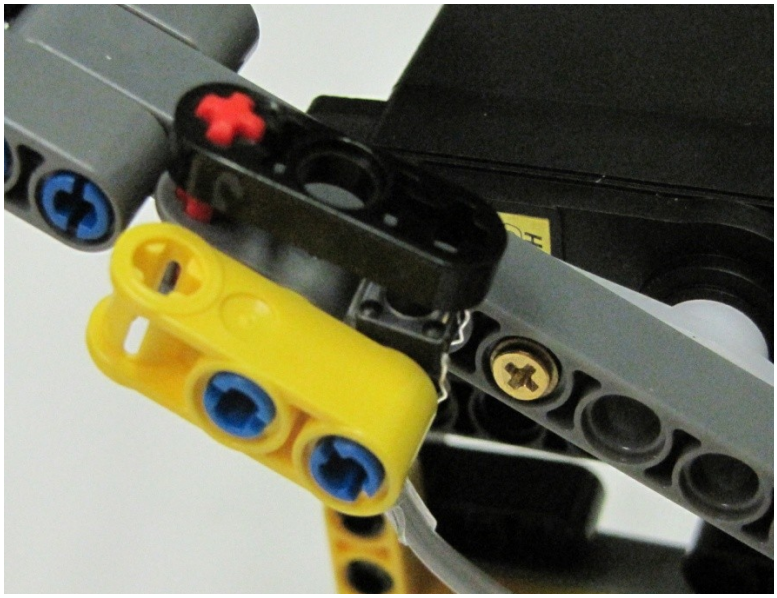
To evaluate our drive train we will run the robot over and over for testing on the game board. We will also see what obstacles the drive train will be able to clear, like PVC and the game pieces. If things happen just once or twice we will ignore those, but if we start to see trends we will redesign the drive train to eliminate those problems.

## Effector:

The only effector that we are using is our fork lift. It is two tines tied to a large arm. The lifting motor for the arm is at the back of the robot to help distribute the weight. We were still going for a compact robot, so everything is s close to the robot body as we can make it. We chose an arm like this because it is simple and light weight. Lots of teams like to use arms and claws, but the forks can be just as effective as a claw without the extra weight of a motor or servo to open and close the claw. This design will be evaluated on how well it can pick up the ducks and Botguy, since those are the points we would like to score.

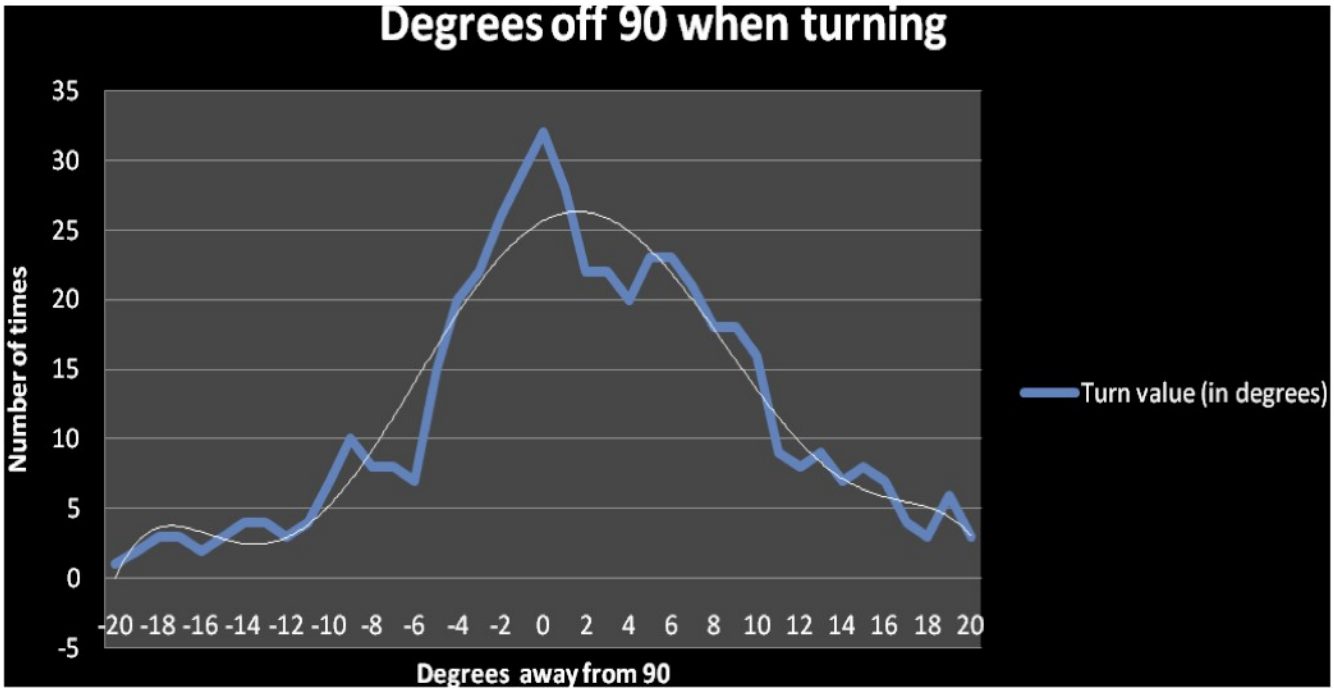


## Sensor Mount:



The picture to the right shows how we mounted a small touch sensor to our arm. We secured it using pins and captured it with the long yellow connector. We added the black three long piece to make the sensor longer and to hold it in from the top. We chose this mount because we believe that the LEGO pieces hold better then the UGlu for most applications. So far our evaluation process has been testing moving the arm over and over again and seeing if it stops when the sensor hits the side of the CBC. So far we have had a 100% success rate with this mount!

## Testing:




We wrote a program to turn the robot to what should be 90 degrees. We put an arrow on the front of the robot and set it on a surface marked with degree angles. We would run the program and see how much the robot turned. We ended up doing over 400 runs, so the data should be a good average representation of an average run. We noticed that our robot tends to overturn rather than under turn. In our programming we will shoot for under turning each corner by 1 or 2 degrees, just to make sure that we do not over rotate and on an average run will rotate exactly to 90 degrees.

# Period 2: Upload a Software Design Document

## Overview

Upload a design document that outlines the functional requirements of a program that you plan to write to control your robot.

Your design document should include 2 sections:

- a written specification of how your software will function
- a flowchart or pseudo-code outline of your software

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

- The submitted file must consist of a single file.
- The submitted file must be a PDF document.
- The submitted file must be named according the pattern 00-0000_P2_Software.pdf

## Scoring

| **Overall** | |
|---|---|
| 1. The submitted document meets all minimum requirements. | 1 |
| 2. The specification is broken into paragraphs and uses meaningful headings. | 1 |
| 3. The specifications uses standard grammar and punctuation. | 1 |

| **Written Specification** | |
|---|---|
| 4. The specification identifies and **briefly** describes the robot for which the software will be written | 2 |
| 5. The specification identifies the major goals that the software will direct your robot to achieve. | 6 |
| 6. The specification presents the tasks the robot must perform to accomplish 3 of the identified goals. | 6 |
| 7. The specification presents the tasks the robot must perform to accomplish 4 of the identified goals. | 2 |
| 8. The specification presents the tasks the robot must perform to accomplish 5 of the identified goals. | 2 |
| 9. The specification includes test procedures and reliability requirements for 3 of the goals. | 6 |
| 10. The specification includes test procedures and reliability requirements for 4 of the goals. | 2 |
| 11. The specification includes test procedures and reliability requirements for 5 of the goals. | 2 |
| 12. The specification requires waiting for the starting light and stopping before the end of the round. | 1 |

| **Flow Chart or Pseudo-Code** | |
|---|---|
| 13. The flow chart or pseudo-code has a meaningful title or caption. | 1 |
| 14. The flow chart or pseudo-code presents each step in a clear order. | 2 |
| 15. The flow chart or pseudo-code clearly indicates branches, conditional steps, and/or decisions. | 2 |
| 16. The flow chart or pseudo-code clearly indicates program inputs ( i.e. sensor inputs, button presses) | 2 |
| 17. The flow chart or pseudo-code clearly indicates program outputs ( i.e. robot actions ) . | 2 |

## Example Scoring Notes

The following example would score points for requirements 1,2,3,4,5,6, 9,12,13,14,15,16, and 17.
It does not meet requirements 7,8,10, or 11.

# Period 2: Software Design Document Example
## Team 00-0000

## Introduction

This will be our first robot program and it is designed to run on our iRobot Create based robot.

## Program  Goals

This program just has three goals:

1. Start with the starting light.
2. Leave the starting box.
3. Stop and shut down.

Because this is our first program, and we are still working on the effectors that our robot will use to move the game pieces, our primary goal is simply to leave the starting box. We will not attempt to score any points.

**1. Start With the Starting Light**

Our robot will need to detect the starting light. In order to do this, we will be using the wait_for_light() function to calibrate for the background light level and detect that the starting light has turned on. Using this function requires that the robot have a properly shielded light sensor mounted to face the starting light.  No servos or motors are required to accomplish this goal.

**Testing**

We will test that our robot can reliably meet this goal by repeatedly starting our robot with the starting light on our game table. Our reliability requirement for this goal is that the robot will start correctly 5 times in a series of 5 tests.

**2. Leave the starting Box**

To leave the starting box, our robot will need to drive forward for several seconds so that it completely clears the starting box. Because our robot is Create based, it will rely on the Create's drive train to move it forward.

**Testing**

We will test that this goal is reliably met by running our program repeatedly in the simulator to see if our robot leaves the starting box. Our reliability requirement for this goal is that the robot will leave the starting box at least 90% of the time.

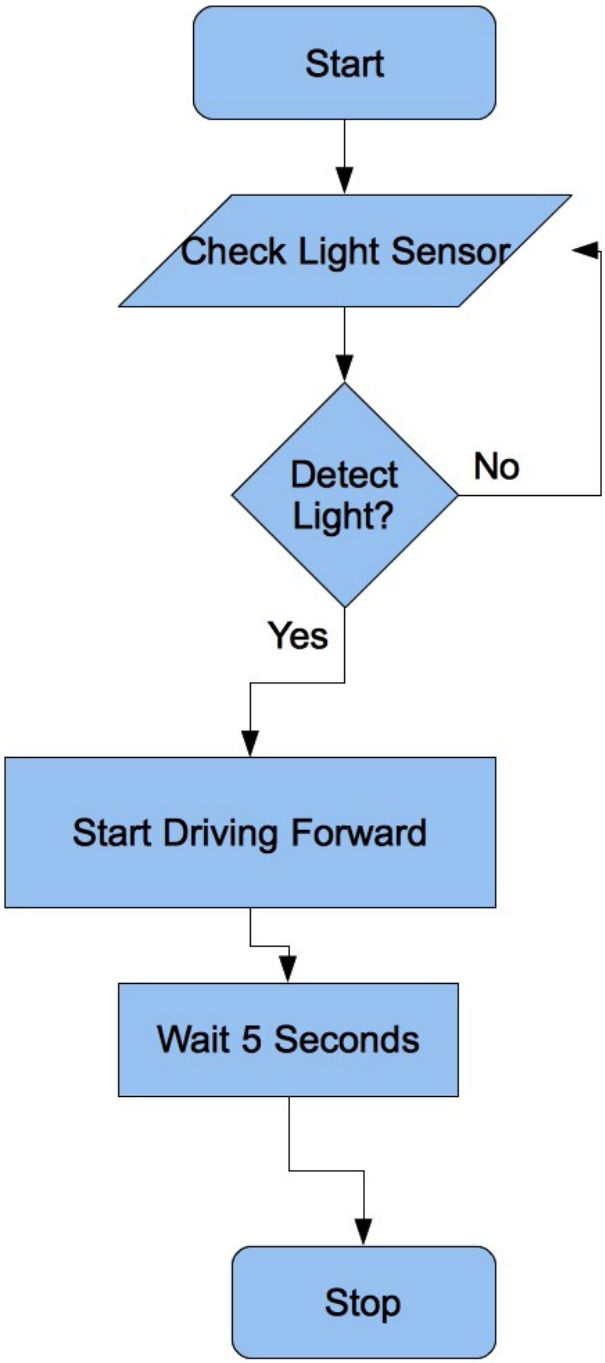
**3. Stop and shut down**

After leaving the starting box, our robot will stop and perform no further actions.

**Testing**

We will test that this goal is reliably met by running our program repeatedly in the simulator to see if it shuts down and stops after leaving the starting box. Our reliability requirement for this goal is that the robot will stop and shut down 5 times in a series of 5 tests.

## Program Flow Chart



# Period 3
## Activities and Examples

# Period 3: Upload Your Project Plan

## Overview

This should be a document that covers the **entire** project **and** shows your actual progress up to the current date. The Botball Project Plan should be a living document that grows and changes over the course of the Botball season. As you progress, goals will be added and removed. Your schedule will change, tasks will get completed, etc. When the judges look at your plan, they won't be scoring how much progress you've made. Instead, they'll be focused on **how well you've documented** your progress. Make sure the judges can easily see your team's progress from period to period. Consider using color-coding, notes, or other elements to update your plan.

## File Submission Requirements

Your submission will score 0 points if it does not meet the following requirements.

- The submitted file must consist of a single file.
- The submitted file must be a PDF document.
- The submitted file must be named according the pattern 00-0000_P2_ProjectPlan.pdf, substituting your team code for 00-0000.
- The submitted file cannot be more than 11" wide.

## Scoring

| **Project Plan Basic Requirements** | |
|---|---|
| 1. The project plan will meet all the basic requirements outlined in the first period. | 1 |
| 2. The plan will include an updated comments section relevant to the current period that explains/updates status and/or for general comments. | 4 |

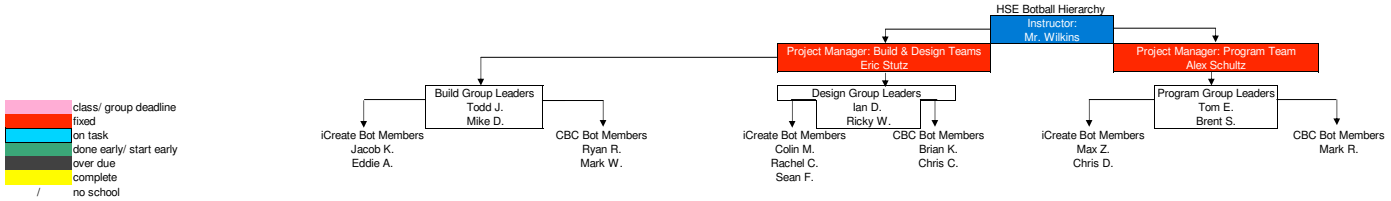
| **Project Milestones** | |
|---|---|
| 3. Project milestones will meet the requirements outlined in period one. | 1 |

| **Project Task Lists** | |
|---|---|
| 4. Project task lists will meet the requirements outlined in period one. | 1 |
| 5. Task lists will include updates relevant to any changes in milestones or the steps needed to achieve milestones. ** | 2 |
| 6. Task lists will include updated due dates for everything.** | 2 |
| 7. Task lists will include updated team member assignments. ** | 2 |
| 8. Task lists will include updated status of each task. | 2 |

****If no updates were needed between period 2 and 3, that must be clearly stated in the comments section.

## Example Scoring Notes

The following example would score points for all of the above requirements.




# 2009 Botball Project Plan

| | Task/ Event | Person in Charge | Prereq. | Week 1 | | | | | Week 2 | | | | | Week 3 | | | | | Week 4 | | | | | Week 5 | | | | | Week 6 | | | | | Week 7 | | | | | Week 8 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 19-Jan | 20-Jan | 21-Jan | 22-Jan | 23-Jan | 26-Jan | 27-Jan | 28-Jan | 29-Jan | 30-Jan | 2-Feb | 3-Feb | 4-Feb | 5-Feb | 6-Feb | 9-Feb | 10-Feb | 11-Feb | 12-Feb | 13-Feb | 16-Feb | 17-Feb | 18-Feb | 19-Feb | 20-Feb | 23-Feb | 24-Feb | 25-Feb | 26-Feb | 27-Feb | 2-Mar | 3-Mar | 4-Mar | 5-Mar | 6-Mar | 9-Mar | 10-Mar | 11-Mar | 12-Mar | 13-Mar |
| 1 | **Doc Submission 1** | Project Managers | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 2 | **Doc Submission 2** | Project Managers | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 3 | **Doc Submission 3** | Project Managers | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| | | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 4 | **Game** | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 4a | Go over Rules | Class | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 4b | Test over Rules | Class | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 4c | Break up into groups | Class | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 4d | Strategy | Both Design Teams | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| | | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5 | **Design** | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5a | iCreate Design | Ian; Colin, Rachel, Sean | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5ai | Claw Drawings | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5aii | Bin Drawings | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5b | CBC Design | Ricky; Brian, Chris | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5bi | Chassis Drawings | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5bii | Claw Drawings | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 5biii | Puff Ball Holder Drawings | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| | | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6 | **Construction** | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6a | iCreate Build | Mike; Jacob, Eddie | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6ai | Claw Construction | | 4a, 5ai | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6aii | Bin Construction | | 4a, 5aii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6b | CBC Build | Todd; Ryan, Mark | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6bi | Chassis Construction | | 4a, 5bi | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6bii | Claw Construction | | 4a, 5bii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 6biii | Puff Ball Holder Construction | | 4a, 5biii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| | | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7 | **Program** | | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7a | Test programs | Program team | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7ai | Program basic movement | | 4a | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7aii | Program Percision turns | | 4a, 7ai | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7aiii | Program Color tracking | | 4a, 7ai, 7aii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7aiiii | Program sonar | | 4a, 7ai, 7aii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7b | iCreate Program | Alex, Chris | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7bi | Program create movements | | 4a, 7aiii, 7aiiii, 6 | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7bii | Program claw movements | | 4a, 7aii, 6ai | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7c | CBC Program | Brent, Tom, Mark, Max | | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7ci | Program CBC movement | | 4a, 6bi, 7aii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7cii | Program Claw movement | | 4a, 6bii, 7aii | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |
| 7d | Robot Testing and Debugging | Class | 4a, 7b, 7c | / | | | | | | | / | | | | | | | | | | | | | / | | | | | | / | | | | | | | | | | | | |

Weekly comments:
This week we were excited to start this new competition. The ideas in all the groups were flowing a lot.

Weekly Comments:
This week we started with the stratedgies of both the iCreate and CBC robots. We lost a little bit of time due to school delays but managed to get ever thing squared away.

Weekly Comments:
This week the class started on the design of the iCreate bot early due to an unanimous decision amongst the Design Team. The iCreate bot got a much needed head start. Program Team was able to learn the CBC program and start on the programming of simple tasks.

Weekly Comments:
This week we had a set back when the designs we origianlly designed for the CBC did not work. We had to start trouble shooting the problem and come up with a new design that everyone agreed would work. The iCreate bot should be done soon next week along with the CBC so that we can hand it over to the program team. I felt that it would be best to move the deadline for the iCreate in hopes that it would get done. Program team has designed the programs but do to the build and design teams using the robots, the program team is unable to fully test their work.

Weekly Comments:
This week the iCreate was put on hold for a bit so that the Build Teams and Design Teams could focus on agreeing on a design that will work. The Build Team got started on the CBC bot and everyone feels it will be strong. The iCreate bot got done once the teams were balanced again. I talked to Alex along with Mr. Wilkins and we decided that it would be a good idea to move the deadline back for the CBC. It will be done by the beginning or middle of the week.Program Team has completed the sonar and color camera programs. Unfortunatly the Program Team was unable to test any of the programs developed.

Weekly Comments:
This week we started off ready to get the CBC done. We had to redesign the CBC for the fifth and hopefully last time to instead of going for the recyclables it just picks them up out of the way. The CBC was late in completion for this week due to the redesign. hopefully it will be completed by early next week. We are very confident with this design to do the best job for us. the Program Team finally got the iCreate, so they were excited to test all their programs on it. it seems that currently they are having problems with the color camera but hopefully they will get that problem solved.

Weekly Comments
This week on wendsday the Build Team finished the CBC and it works awesome. The CBC has a new slimmer design maximizing efficiency with the least amount of parts. The Build Team's only responsibility now is working on documents and helping where ever they are needed. The Program Team is working on program combinations and trouble shooting the programs, also they decided to move back the deadlines a week in order to perfect the programs and to improve the overall robot. few minor tweaks to the construction of the robot have been made but nothing to huge. we have one week to go and hopefully we'll make it to Michigan with some robots.

Weekly Commnets:
This week so far program team has been scrambling to get the programs all nice and completed. Build Team is helping where ver needed and revising their documents. Firday is the big day and we hope to have a competed team of robots.

# Period 3: Upload Design Changes

## Overview

Upload a design document focusing on your design changes. Supply three pictures, diagrams, or models of your robot(s) that show off the changes your have made, along with supporting text.

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

1. The submitted file must consist of a single file.
2. The submitted file must be a PDF document.
3. The submitted file must be named according the pattern 00-0000_P1_Mechanical.pdf

## Scoring

| Basic Requirements | |
|---|---|
| 1. The submitted document meets all minimum requirements | 1 |

| First Change Section | |
|---|---|
| 2. The section begins with the title *First Change,* clearly separated from text. | 1 |
| 3. The section includes a clear picture emphasizing the first changed part of your robot. | 1 |
| 4. The section includes descriptive text describing the first change. | 2 |
| 5. The section explains what caused your team to make this change. | 2 |

| Second Change Section | |
|---|---|
| 6. The section begins with the title *Second Change,* clearly separated from text. | 1 |
| 7. The section includes a clear picture emphasizing the second changed part of your robot. | 1 |
| 8. The section includes descriptive text describing the second change. | 2 |
| 9. The section explains what caused your team to make this change. | 2 |

| Third Change Section | |
|---|---|
| 10. The section begins with the title *Third Change,* clearly separated from text. | 1 |
| 11. The section includes a clear picture emphasizing the third changed part of your robot. | 1 |
| 12. The section includes descriptive text describing the third change. | 2 |
| 13. The section explains what caused your team to make this change. | 2 |

| Elegant Solution Section | |
|---|---|
| 14. The section begins with the title *Elegant Solution,* clearly separated from text. | 1 |
| 15. The section includes a description of a fiendishly difficult problem your team encountered. | 2 |
| 16. The section includes a description of the particularly elegant solution to described problem. | 3 |

## Example Scoring Notes

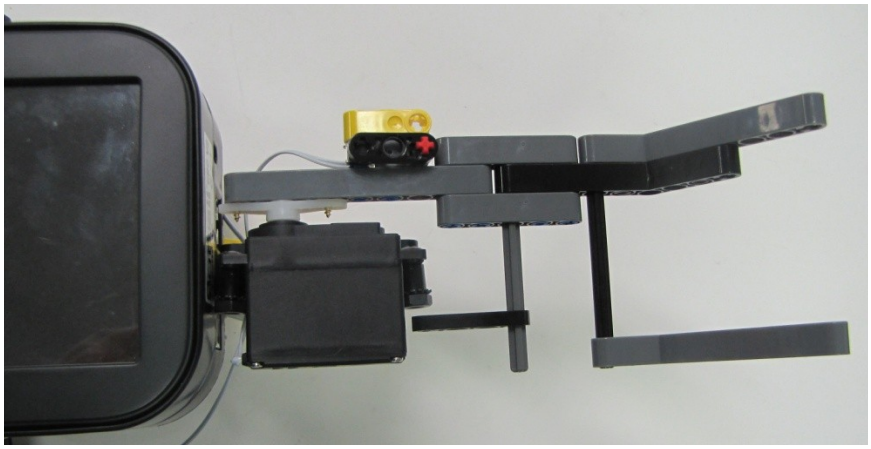
The following example would score points for all of the above requirements.

# Period 3: Design Changes Example

## First Change:

One of the first things we ended changing on our robot as we tested it were the tires. The new tires are the balloon tires. The square, low profile tires we started with were unable to get us over the PVC and were giving us problems with turning in place. The new tires have less surface in contact with the game board and allow us to turn more precisely. They are also taller and allow us to get over the PVC.

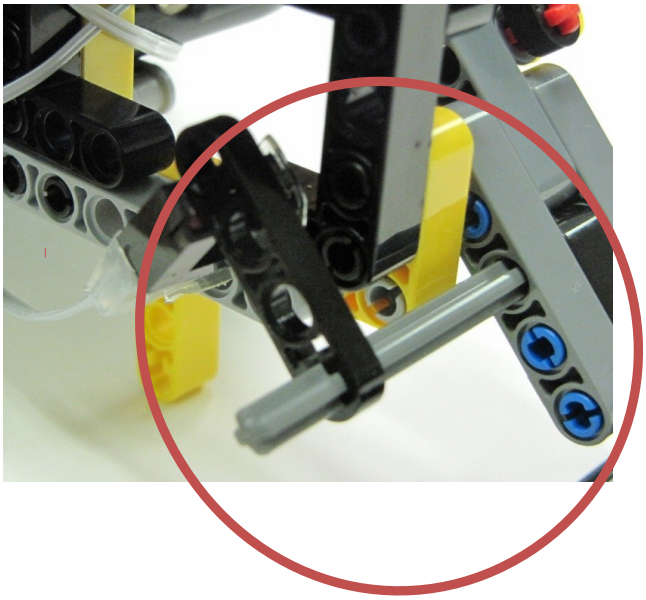
## Second Change:



The second change we had to make was to the lifting arm. As you can see in the picture, our method is the same and the mount of the motor is the same, but we made the arm shorter. Consequently, we now are driving backwards. Due to the forces exerted on the longer arm when lifting, we had to shorten the arm. Since torque is equal to the force you are lifting times the length of the arm, we can lift more weight (like Botguy) with the same motor and a shorter arm. This also had the benefit of making the overall dimensions of the robot smaller.

## Third Change:

Another change that we made was adding a slot sensor to act as a limit switch. The sensor is just UGlued in place. I know we prefer LEGO, but this is a quick fix that has not given us any trouble yet. Once we shortened the arm and turned everything around (see change 2), we had no way of knowing when the arm was on the ground. We added the slot sensor circled in the picture to the left to check when a LEGO part from the arm is in the sensor. If it is we know that the arm is on the ground. The main reason we had to do this was that we were having trouble collecting the ducks because we could not tell if the arm was low enough to get them. If we over rotated the arm, we would end up tipping our robot over.



## Elegant Solution:

The most elegant solution to a difficult problem that we had was the direction that the robot drives. We were having problems with the skid catching on the FRP divider on the table and messing up our program. We found that a combination of driving backwards and using a mrp motor command fixed that problem. So with a quick swap of the motor cables and some programming changes, we prevented a complete rebuild of our robot!

# Period 3: Upload Example Commented Code

## Overview

Upload a single example of your teams source code that shows how you have documented your robot code. Choose an example that contains at least one variable and at least one function. The judges would like to get an idea of how your robot will behave in the game, so they prefer source code that implements game strategy over libraries and utility functions.

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

- Your submission must consist of a single file.
- The submitted file must be a PDF or the source code for a C program.
- The submitted file must be named according the pattern 00-0000_P3_Software.c

## Scoring

| File Header | |
|---|---|
| 1. File header includes program or library name. | 1 |
| 2. File header includes author or maintainer information. | 1 |
| 3. File header includes a short description of the program. | 2 |
| 4. File header includes information on major revisions of the program ( change history). | 2 |

| Function Definitions | |
|---|---|
| 5. Each function definition is preceded by a short description of the purpose of the function. | 2 |
| 6. Each function definition is preceded by a comment giving the arguments of the function, if any | 2 |
| 7. Each function definition is preceded by a comment describing the return value of the function, if any. | 2 |
| 8. Each function definition is preceded by a comment giving side effects of the function or stating that there are none. | 2 |

| Functions and Variables | |
|---|---|
| 9. Function definitions and variable names are consistently formatted. | 2 |
| 10. Function definitions and variable names are descriptive and convey their use in the program. | 2 |
| 11. The program uses functions to organize blocks of related code and to perform repeated tasks. | 2 |

| Overall Program Documentation | |
|---|---|
| 12. The program uses white space consistently. | 2 |
| 13. The program is indented to show flow of control ( i.e. curly braces line up). | 2 |
| 14. Comments use standard spelling and grammar. | 2 |
| 15. Comments focus on the purpose of the code and the robots actions. | 2 |
| 16. Overall, the comments do not contain large blocks of code that is no longer in use. | 1 |

## Example Scoring Notes

The following example would score points for all of the above requirements.

```
/*
  Program: FindBall.ic
  Author: Cwafty Wabbit

  Desciption:

  This program finds a colored object of the color
  defined in color model 0, and then approaches it using
  the sonar sensor on port 10

  30 Jan 2010  -- CW modified program to use the sonar sensor to detect distance
  30 Jan 2010  -- CW reorganized main with utility functions for robot movment
  19 Jan 2010  -- CW created original version.
*/
```

File Header

```
#include "xbccamlib.ic"
// Sensor Definitions (we'll call the color model part of this)
#define OBJECT_MODEL 0
#define SONAR_PORT 10

// Motor definitions
#define LEFT_MOTOR 0
#define RIGHT_MOTOR 1

// Thresholds
#define SONAR_THRESH 50
#define OBJECT_MAX_OFFSET 20

// Maximum width of the camera in pixels
#define CAMERA_MAX_WIDTH 240


/*
  This is my main function. This will use the motors
  defined above to face the largest colored object
  that it sees with color model 0. It will then approach
  the object, re-centering if necessary
*/
void main()
{
        // Finds the object to begin with
        find_object(OBJECT_MODEL);
        forward();

        // Continues forward until sonar <= thresh
        while(sonar(SONAR_PORT) > SONAR_THRESH) {
                track_update();
                // Uh-oh, lost track of the ball, find it again
                if(abs(objet_offset(OBJECT_MODEL)) > OBJECT_MAX_OFFSET) {
                        find_object(OBJECT_MODEL);
                        forward();
                }
        }
        // I'm done! yay!
        stop();
}


/*
  This function turns the robot in place to face
  the largest colored object using the specified
  color model

  Arguments: int model, the color model for which to search
*/
void find_object(int model)
{
        int offset = 0;
        // Check where the blob is to begin with
        track_update();
```

Function Definition

```
        offset = object_offset(model);
        // It's either to the left, or not on the screen, so turn left
        if(offset < -OBJECT_MAX_OFFSET) {
                turn_left();
        }
        // It's to the right, turn right
        else if(offset > OBJECT_MAX_OFFSET) {
                turn_right();
        }
        // While it's still not centered, keep turning
        while(abs(offset) > OBJECT_MAX_OFFSET) {
                track_update();
                offset = object_offset(model);
        }
        // The ball centered, i'm done!
        stop();
}

/*
  This returns the offset in pixels from the center x
  pixel of the largest object on the specified color model

  Arguments: int model, the color model for which to search
*/

int object_offset(int model)
{
        return track_x(model, 0) - (CAMERA_MAX_WIDTH/2);
}


/*
    Given: an int a
    Returns the absolute value of a
*/
int abs(int a)
{
        if(a < 0)
                return -a;
        return a;
}

// Drives the robot forward
void forward()
{
        fd(LEFT_MOTOR);
        fd(RIGHT_MOTOR);
}

// Turns the robot to the left
void turn_left()
{
        bk(LEFT_MOTOR);
        fd(RIGHT_MOTOR);
}

// Turns the robot to the right
void turn_right()
{
        fd(LEFT_MOTOR);
        bk(RIGHT_MOTOR);
}

// Turns off the robot's drive motors
void stop()
{
        off(LEFT_MOTOR);
        off(RIGHT_MOTOR);
}
```

# Period 3: Upload Lessons Learned

## Overview

Provide a summary of the important lessons your team learned throughout the Botball project. You can follow the format used in one of the examples below or create your own as long as you use the proper headings and provide the required content. Remember, every team has a unique experience so the lessons learned presented in the examples will not be the same for every team!

## Minimum Requirements

Your submission will score 0 points if it does not meet the following requirements.

- The submitted file must consist of a single file.
- The submitted file must be a PDF document.
- The submitted file must be named according the pattern 00-0000_P3_LessonsLearned.pdf

## Scoring

| Overall | |
|---|---|
| First section begins with the header: **Experience Gained** | 1 |
| First section describes what your team gained from the Botball experience. | 2 |
| Second section beings with the header: **Documentation Process** | 1 |
| Second section describes what your team gained from the documentation process. | 2 |
| Third section begins with the header: **Surprises** | 1 |
| Third section describes what surprised you most about the Botball experience. | 2 |
| Final section beings with the header: **Advice for Future Teams** | 1 |
| Final section lists any advice you would pass along to future Botball teams. | 2 |

## Example Scoring Notes

The following examples would score points for all of the above requirements.

# Period 3: Lessons Learned Example #1

**Experience Gained**
We have learned a lot about the design process and how important testing is if one wants a reliable robot. We have also learned a lot about how programming works and how there can many ways to go about one coding task. Most importantly we learned that communication is key and that everyone must be on the same page in the plan. We weren't and the effects are now being felt.

**Documentation Process**
The documentation process helped us get organized and keep on task. It also helped us break our goals down into manageable sections, preventing us from being as overwhelmed.

**Surprises**
We were all surprised at how many things an autonomous robot can actually manage to do using the right sensors and mechanical systems. We also did not expect some of the seemingly easy building tasks to take nearly as long as they did. Just building a simple claw up to meet our preconditions took us three weeks of building, testing and redesigning.

**Advice for Future Teams**
OVERPLAN!!! Spend as much time as you need to get the perfect DETAILED plan. Then give a copy to every member of your team and continually check progress against the original plan. If we had done a better job of this our team would have worked together better and much less time would have been wasted.

# Period 3: Lessons Learned Example #2

**Experience Gained**

1. Problem solving and persistence
2. How to think creatively and to prototype things that may not work to find the solution to a problem (hardware)
3. How to look at and do things in many different ways, as opposed to doing something considered "normal".

**Documentation Process**

1. The (code) documentation process has pushed me to write cleaner, more easily understandable code with meaningful names for methods and variables.
2. Helped us understand how to write about and describe our creation to someone else.
3. Helped keep track of what has been done and what must be done.
4. Helped me be more organized and thorough.

**Surprises**

1. The good-natured-ness that the teams at competition have despite the competition.
2. That everyone is willing to help you whether they are your own mentors or a different team.
3. How much fun it is!
4. How important the documentation is.

**Advice for Future Teams**

1. Make sure that you test your code thoroughly before moving on.
2. Have Fun! It may be a competition but don't let that take the fun out of it
3. Yea, have fun; also, when making a chassis or some sort of standing square structure, remember (if you do not support it): square -> rhombus -> parallelogram -> line...
4. Do everything as soon as possible. The longer you hold things off, the less time you'll have for testing.

# Period 3: Lessons Learned Example #3

**Experience Gained**

- Joshua - I learned how to program with state programming and functions. I learned how to use loops, “ifs” and many other C commands. I also learned how gearing down loses speed and gains accuracy and how gearing up gains speed and loses accuracy. I learned how to use sensors, servos, and motors.
- Brian: - I gained more experience in CADing in SolidWorks.
- Lucas: - I learned that building a tripod helps aiming tethered projectiles. They help the aiming a lot.
- Parker: - I learned how to program the robot in an abstract way so it is easy to update your code using libraries and path planning that is based off of a real coordinate system.
- Kevin: - I learned better engineering and how to cooperate with VEX and legos.
- Steven: - I learned how to make libraries

**Documentation Process**

- Joshua: - Through documentation, I learned how to make a project plan. I learned how useful a project plan can be for keeping the team on track and planning ahead for what to work on next. I also learned that by running simulations and designing prototypes, the robot became more accurate and reliable. Designing a flow chart made the programming easier for it planned out the architecture of the code.
- Brian: - I learned that documentation can be very useful if you forget why you did things.
- Lucas: - Through documentation, I learned how to take good photos of our team robots’ mechanical designs and other important key features.
- Parker: - I learned that it is lots of fun!
- Kevin: - The documentation process helped me to understand that it could help us stay organized and focused.
- Steven: - By doing the project plan, I was better able to understand what was going on.

**Surprises**

- Joshua: - I was surprised at how useful the documentation was in the long-term! I was also surprised how the programming in C is hard compared to graphical languages.
- Brian: - The season was very short this year, and it surprised me.
- Lucas: - I was surprised how the building with VEX parts is hard compared to building in legos.
- Parker: - I have been doing botball for a number of years now so not that many things surprised me. However I was surprised that edge tracking while simple in concept is difficult to implement.
- Kevin: - The season was very short this year, and it surprised me.
- Steven: - I was surprised at how hard it is to be a team captain and how hard it was to keep the team focused.

**Advice for Future Teams**

- Joshua: - My advice to future Botball teams is to work hard on documentation for it is rewarding in the long run.
- Brian: - My advice to future Botball teams is that CAD models take longer than you might expect and that coding does too.
- Lucas: - My advice to future Botball teams is to use rubber bands for certain mechanisms for they are faster than motors and servos.
- Parker: - “Simplicity on the other side of complexity”
- Steven: - Come up with elegant mechanisms that do the job effectively.
- Kevin: - Learn how to program in C and discover how all the different motors work before the season starts.