Agreement

This project assignment can be completed in groups of two or three or individually. If working in a team, each team member must follow the robotics team member guidelines developed by the students in this class, included in the next section. Instructor will reduce the grade of the team member who fails to follow the team guidelines generated in our robotics group.

By working on and completing this laboratory assignment you agree to use the hardware given to you in a responsible manner. Each team is responsible for the safety and security of the assigned robot(s) while the hardware is in their possession. Malicious and unsafe operation of the robot(s) by any team member will result in the inability of that team member to work with the robot(s) provided by the instructor.

You must check out and check back in the robot(s) every time you take it outside of Doane. To do so, one team member must fill out the sheet located on the door of the cabinet with the robot type and number and its status (checked in or checked out).

Lab Team Work Guidelines

1. Play to your strengths: each team member is assigned specific task(s) that tailor to their strength.

2. Try every task: despite possibly concentrating on a particular task more heavily, each team member will also participate in some degree in every aspect of the lab completion tasks, including robot design, robotic software development, testing, and demonstrations.

3. Communicate any concerns: each team member will continuously communicate with other team members, especially when any concerning issues arise.

4. Don’t abandon your team members: under no circumstances a team member is to stop completing their assigned tasks and attending team work meetings.

5. Prioritize time and set boundaries: team members are to discuss the feasibility of the project at the beginning and fill out the “Timeline” section of the report document before beginning the lab work.

6. Plan the robotic design and team work: team members should discuss and agree on the robotic design and its task and also how the work for the lab is to be completed/distributed in the team.

7. Teams on GitHub: teams are to use GitHub to commit their work regularly and to submit their completed code and the report. All team members can contribute to their lab team repository as necessary.
Robot Assignments

- **Carson, Mitch, Jon**: turtlebot 1, bebop 1, laptop turtlebot4
- **Thomas, Chris, Cayla**: EV3 #9
- **Jacob M. Alex K.**: turtlebot 3, bebop 2, laptop turtlebot5
- **Simon, Jake S. Zach**: EV3 #10, EV3 #13
- **Chris, Spencer, Matt**: turtlebot 4, EV3 #8, laptop turtlebot6
- **Devin, Mikey**: bebop 6, bebop 8
- **Trent, Jake VS**: EV3 #2, EV3 #11, EV3 #12
- **Michael, Linh**: EV3 #3, EV3 #4

Configuring Git and GitHub

To access the template directory for the project assignment, you should go into the `#labs` channel in our Slack team and find the announcement that provides a link for it. The team leader should accept the laboratory assignment first, create a team and set up the GitHub repository for the team to access the assignment’s starting materials and to store the completed version of your assignment. Once the team has been created, the other members of the team can click on the given GitHub Classroom link and join the team. Every team member can clone the team’s lab03 repository, and use regular Git commands, such as `git commit`, `git push` and `git pull`. Please ensure your team practices standard GitHub practices to avoid merge conflicts.

Project Summary

This course has a project component valued at 30% of your grade. Your project should design and implement an autonomous and intelligent multi-agent/robot system for some application of your choice. This implies that your chosen system must either utilize 1) two or more autonomous robots of your choice (Lego, Bebop, Turtlebot) or incorporate one or more software agent with one or more robot of your choice. The agents/robots in your system can be heterogeneous or homogeneous, competing or cooperating, and they should contain an intelligent behavior. You may implement your system in a language of your choice.

I encourage you to be creative and to choose something that is interesting to you! If you are interested in something and would like some help locating the resources (references, software, or hardware), please let me know.

Core Requirements

1. Your project must be approved by me before you start working on it. I will assess the viability of proposed projects during class on November 12th.
2. Your project must be developed for a specific application using an intelligent multi-robot or a robotic and software agent system. You need to research the problem you select to get an idea of what has been already done. You must include references to existing work in your final report and justify why using your particular system is appropriate.

3. Your project must have an implementation component. You may write your code from scratch or reuse and extend some existing code. Obviously, anything you use that is not yours must be documented (in the source code and in the report).

4. Your project must be extensive enough to qualify as a project (think of work for about two lab assignments), but not too extensive so that you can not finish it by the end of the semester.

**Project Ideas: Extensions of Existing Projects**

There are a few existing projects that have been developed by former students at Allegheny and several projects that have been done by individuals at other institutions. You are welcome to start with the idea given below and utilize existing code and extend it further if it is available.

1. Search and rescue using drones. Can be implemented using pyparrot with Bebop drones and a search algorithm, such as A*.

2. Image processing using OpenCV and Bebop drones. OpenCV’s the aruco marker library can be used for finding specific objects. OpenCV’s object detection libraries can be used for a more in-depth detection of an object.

3. Product delivery to a specific location using drones and pyparrot library.

4. Capture the flag using drones and AR Code reader. Can be implemented using pyparrot and ar-markers library (ref: https://pypi.python.org/pypi/ar-markers/0.4.1.)

5. Multi-robot security patrol using multiple Lego EV3 robots and leJOS. Ref: code available upon request from the instructor.

6. Multi-robot information exchange using ROS and scripting. Ref: code available upon request from the instructor.

7. Gesture mimicking with robotic hand (3D printed using Ultimaker) and leapmotion using MyRobotLab (MRL). Ref: hardware available upon request from the instructor.

8. Parrot Bebop Teleoperation suite that features a graphical user interface (GUI) and an autonomous “patrol” flight feature. Implemented in C++, using ROS the C++ SDL library. Ref: https://github.com/Michionlion/bebop_teleop

9. QR code reading and analysis using drones. Implemented in Python and Go. Ref: code available upon request from the instructor.

10. AR code reading and analysis using drones. AR code was attached to a mobile Turtlebot, which the drone followed. Implemented in C++ using ROS. Ref: https://github.com/Michionlion/bebop_track_tag
11. Robot follower (robots that follow each other autonomously). Turtlebot and Lego EV3 robots were used with ROS and CPP and leJOS and Java. Ref: code available upon request from the instructor.

12. Android Application to Control Multiple EV3 Robots using leJOS and Java. Ref: code available upon request from the instructor.
Timeline: Deliverables

1. Proposal (at least 450 words) Deadline: 13 November, 2019 by 2:30 pm:
   Develop an idea for your project, research its feasibility and have it approved by me. After receiving a verbal approval, write a description of what you propose to do for your project and submit it to your project GitHub repository by the due date. Your proposal should describe what you want to do (the application you will tackle), provide motivation for your chosen application and describe ideas for how you plan to implement your proposed solution, including the hardware requirements for your project. You should also include at least two references to demonstrate that you have done some research about the tools you plan to utilize and the application you plan to tackle. Finally, you should describe how you will consider engineering ethics into your proposed system.

2. Progress report (at least 500 words) Deadline: 21 November, 2019 by 9:30 am:
   Describe everything you have done so far in your progress report. By this point, you should have decided on how you will design your system and what algorithm(s) you will use, and made some progress towards implementing the solution to your proposed project. Describe anything new that you have learned so far and any unexpected challenges that you have encountered?

3. Presentation 5 December
   By the presentation session, you should have mostly finished implementation, and have run some preliminary tests. During the presentation, you should describe what you have done for the project and provide a demonstration if applicable. Use diagrams, images and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Prepare for a ~ 5 minute presentation. Every member of the group needs to contribute to the presentation talk. At the end of the presentation give a demonstration of your project or show a video with your demonstration.

4. Final programs, output and final report (at least 1,200 words) Deadline: December 10, 2019 by 9:30 am:
   Your programs should be well documented and conform to Google styling conventions. Your repository should have a modified README that contains detailed instructions on the tools and steps needed to run your system. Submissions without reproducible set up instructions will receive a grade reduction.

   Your final report should highlight the key contributions of your work and consist of at least 1200 words. The report should include a description of why the chosen application and solution is important and discuss the implementation that you undertook. The written material should be precise, formal, appropriately formatted, grammatically correct, informative, and interesting. In summary, your report should include:

   - The motivation for your project. Why is the application you chose important/useful?
   - Background for the proposed problem. What have others done for it already? Include references.
• Detailed description of the work you completed for this project. Without giving a snapshot of the code you wrote, provide technical description of what you implemented and how you implemented it. Include algorithms and diagrams when necessary.

• Analysis of your results. Make graphs, tables, comprehensible snapshots of output, or anything else that can help me understand your results.

• Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges and the biggest rewards.

• If you worked in a team, you should also include a paragraph that describes the team work and the contribution of each team member.

• Finally, describe the engineering of ethics into your implemented system concentrating on its positive design; for example what makes it ethical, how does it benefit the society, etc. Then, answer the five questions provided in the survey inside the report.md file.

Grading rubric

10 points: Proposal

10 points: Progress report

20 points: Presentation and demonstration

60 points: Final report and project implementation

Since you have a lot of flexibility in this project, please review the Honor code, and make sure to follow it.