Agreement

This laboratory assignment will be completed in groups of two or three (max). Each team member must follow the 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 Arduino board and any sensors, motors, cables and other electronics equipment utilized for their lab. Malicious unsafe operation of the hardware by any team member will result in the inability of that team member to work with the hardware provided by the instructor.

By midnight on Monday, January 27, you must specify all hardware components you are planning to use in the appropriate section of your report document. At the completion of the lab, the instructor will check all of your specified equipment back in. All equipment must be returned before grades can be released.

Lab Team Work Guidelines

1. Organization and function:
   - Team will delegate one member to be a team manager. In teams of two, the role of the team manager will be restricted to initiating the project on GitHub and resolving merge issues.
   - The project work will be distributed evenly among the team members. The work of being a team manager should count toward work distribution.
   - The team will work collaboratively to create a detailed project work plan with clear deadlines.
   - Teams must ensure that the project’s scope is feasible and that every member of the team is comfortable with the projected scope.

2. Communication:
   - All team members must promptly respond to the other team members’ communication.
   - All team members must regularly check and respond to Slack messages and GitHub issues.
   - Each team member should keep their communication clear and concise.

3. Interaction:
   - Conflicts within the team should be first brought to the team leader first. If unresolved, they should be elevated to the instructor.
• All team members must recognize and avoid biases.
• Each team member must act in a professional and in a respectful manner.
• All team members must attend the scheduled team meetings and be punctual.

4. Project Problem Solving and Decision Making:
• During problem solving teams should utilize collaborative brainstorming and researching tactics.
• During brainstorming each team must ensure that every member is able to express their ideas, solutions, concerns, etc.
• The decisions about the project are to be made collaboratively. If a decision can not be reached, a majority vote should be the deciding factor, or, in case of a draw, the instructor’s input should be the deciding vote.

Objectives
To apply concepts of agent types and agent environment to a practical application. Specifically, you will use an open-source electronics platform called Arduino to develop a simple agent. You will learn how to use Arduino programming language, its IDE, Arduino board and other hardware components such as sensors and motors. Finally, you will reflect on the design of your agent, its attribute and type, its decision making within the environment, and its social responsibility.

Reading Assignment
To review what you have learned about agents, their attributes, types, and environments, please read sections 1.3-1.5, 2.1-2.2 from the “Artificial Intelligence: Foundations of Computational Agents” textbook.

If you have not done so already, please read all of the relevant “GitHub Guides”, available at https://guides.github.com/, that explain how to use many of the features that GitHub provides. In particular, please make sure that you have read guides such as “Mastering Markdown” and “Documenting Your Projects on GitHub”; each of them will help you to understand how to use both GitHub and GitHub Classroom.

Accessing the Laboratory Assignment on GitHub
Since this is a team-based assignment we will be using a group assignment functionality of GitHub Classroom. For team assignments only one person will be creating the team while the other team members will join that team. Please form a team consisting of two or three members, assign one person to be the designated team manager.

The selected team manager should go into the #labs channel in our Slack team and find the announcement that provides a link for it. Copy this link and paste it into your web browser. Now, you should accept the laboratory assignment and create a new team with a unique and descriptive team name (under “Or Create a new team”).

Now the other members of the team can click on the assignment link in the #labs channel and select their team from the list under “Join an Existing Team”. When other team members join their group in GitHub Classroom, a team is created in our GitHub organization. Teams have pretty cool functionality, including threaded comments and emoji support. Every team member will be able to push and pull to their team’s repository. Your team’s project manager should be the one to resolve any conflicts or merge pull requests.
Unless you provide the instructor with documentation of the extenuating circumstances that you are facing, not working in a team and not accepting the assignment means that you automatically receive a failing grade for it.

**Arduino Project**

In this two-week lab students are invited to implement a rational agent using an open-source electronics platform, Arduino, for some application of your choosing (within the realm of available hardware resources). As you select a specific application, you must identify the agent type, its attributes, and its task environment. The developed applications will be used during outreach activities in Crawford Central school district.

Arduino platform makes it relatively easy to develop interactive applications that utilize both hardware and software by taking inputs from a variety of switches or sensors and controlling a variety of outputs (lights, motors, etc.). You will be working with an Arduino microcontroller board, a small circuit (the board) that essentially contains a small computer on a small chip (the microcontroller). There are different versions of the Arduino board that differ in components and size, you will work with either UNO or DUE versions. Other electronics components that are available are light sensor, infrared distance sensor, LED module, light sensor, push button, sound sensor, temperature sensor, tilt sensor, motion sensor, rotation sensor, buzzer module, servo, among others.

There are many examples for sample Arduino projects, including the following:

- [https://create.arduino.cc/projecthub](https://create.arduino.cc/projecthub)
- [https://www.hackster.io/arduino/projects](https://www.hackster.io/arduino/projects)
- [https://www.elegoo.com/download/](https://www.elegoo.com/download/)

Although you are encouraged to use the sample examples included with Arduino IDE or sample projects from existing online resources, your solution to this lab can not be just an existing project that you have put to practice. You must add at least one small extension or make a small change to the project providing an inspiration for your work. In your report document, you must also include a reference (URL is sufficient) to all resources that you have used when learning about Arduino and possible projects.

**Set Up**

Arduino programs are written in the Arduino Integrated Development Environment (IDE). In ALIC, Arduino IDE is already installed on the machines. If you would like to use your own laptop, you can obtain the necessary software from [https://www.arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software). You also have an option of developing and testing your program on Arduino Wed Editor [https://create.arduino.cc/editor](https://create.arduino.cc/editor).

**Arduino Programming**

Arduino IDE enables you to write sketches (programs in Arduino) for different Arduino boards. The Arduino programming language is based on a very simple language called *processing*, which is similar to the C language. After the sketch is written in the Arduino IDE it can be uploaded on the Arduino board for execution.

Due: 7 February, 2020 at 2:30 pm
The structure of an Arduino program contains a minimum of 2 blocks responsible for preparation and execution, `setup` and `loop` methods respectfully in the example below.

```cpp
void setup( ) {
    statements;
    ...
}
void loop( ) {
    statements;
    ...
}
```

When program is run, the `setup` function is executed first, and it is responsible for initializing the pin modes and starting serial communication. This function has to be included even if there are no statements to execute but it is only executed once. After the `setup` function runs, the execution block runs next. The execution block contains statements that are responsible for specific tasks, such as reading inputs, triggering outputs, checking conditions etc. For example, a `loop` function is a part of execution block and it executes the set of statements repeatedly.

**Testing your Program**

Since this is an open-ended assignment involving hardware you must execute your program(s) and see if they are producing the expected output. When you believe you have completed all of the requirements of the writing for this lab, you can use the GatorGrader tool ([https://github.com/GatorEducator/gatorgrader](https://github.com/GatorEducator/gatorgrader)) to automatically check if your submission meets the minimum submission requirements (by running the “gradle grade” command). If you are using Docker Desktop, you can use the following “docker run” command to start “gradle grade” as a containerized application, using the “DockaGator” Docker image available on DockerHub. You can run the following command to run the “gradle grade” on your project:

```bash
docker run --rm --name dockagator \
    -v "$(pwd)":/project \
    -v "$HOME/.dockagator":/root/.local/share \
    gatoreducator/dockagator
```

This command will use "$(pwd)" (i.e., the current directory) as the project directory and "$HOME/.dockagator" as the cached GatorGrader directory. Please note that both of these directories must exist, although only the project directory must contain some content. To ensure that the previous command will work correctly, you should create the cache directory by running the command “mkdir $HOME/.dockagator”; you will only need to do this once. If the above “docker run” command does not work correctly on the Windows operating system, then you may need to instead run the following command to work around limitations in the terminal window:

```bash
docker run --rm --name dockagator \
    -v "$(pwd)":/project" \
    -v "$HOME/.dockagator:/root/.local/share" \
    gatoreducator/dockagator
```

To enter into an “interactive terminal” in the Docker container, you can instead use the following command.
docker run -it --rm --name dockagator \
-v "$(pwd)":/project \
-v "${HOME}/.dockagator":/root/.local/share \
gatoreducator/dockagator /bin/bash

Now, you can type “gradle grade” to run through the checks that the instructor specified for your assignment.

Throughout the process of working on this assignment you should remember to transfer your files to GitHub using the “git commit” and “git push” commands. Remember, to correctly complete this assignment you need to commit all modified files to GitHub. Also, please utilize appropriate GitHub practices while working with your team members to avoid common merge conflicts.

When you use the “git push” command to transfer your source code to your GitHub repository, Travis CI will initialize a “build” of your assignment, checking to see if it meets all of the minimum requirements. If your commit meets all of the minimum submission established requirements, then you will see a green ✓ in the listing of commits in GitHub after awhile. If your submission does not meet the requirements, a red ✗ will appear instead. Please note GatorGrader and Travis are used to just ensure that all materials have been submitted. A Travis pass does not indicate a correct completion of the technical details of the lab due to the open ended nature of the lab.

Planning

Once you have chosen the Arduino project you want to implement and the type of agent you want to develop, identify the tasks that need to be completed for this lab. Then, by midnight on Monday, January 27th, complete and submit the planning portion in your report that includes the tasks your team has to complete and a timetable for their completion. In this section of your report you must also indicate all of the hardware components that your team will need for the lab completion.

Project Walkthrough

During the lab session on Friday, January 31st, each team will participate in the project walkthrough process. Project walkthrough is an informal process where the instructor leads the process of reviewing the progress of the project and the written code is reviewed for technical accuracy with the objective of finding errors and improving the quality of the code. The purpose of this walkthrough is to motivate continuous progression on the project, identification of any conceptual issues, and detection of any technical errors. When the walkthrough is finished, the authors of the project are responsible for taking the necessary actions to correct the identified issues.

By this project walkthrough, each team should have identified the agent they plan to develop, have gotten to know Arduino IDE and hardware component interactions, and have started writing some code. During the walkthrough, the team members will collaboratively lead the walkthrough process, which should last 5-10 minutes for each team. Each team should:

- Describe the chosen application.
- Discuss what makes the agent being developed in this application “an agent”, what its attributes, its type, and its environment are.
- Explain the written code.
- Identify the steps left to complete for this project.
**Project Demonstration**

At the beginning of the lab session on Friday, February 7th, each team will be given an opportunity to demonstrate their project. When the lab session starts, teams will be given a few minutes to set up their demonstrations and get them running. Then, class members will participate in an interactive demonstration session, where everyone will be able to view each demonstration.

**Required Deliverables**

This assignment invites you to submit the following deliverables through your team repository.

1. Planning portion of the report due on January 27 by midnight. The rest of the written requirements are due at 2:30 pm on February 7.
2. A properly completed and commented source program(s). Please make sure your source code is titled “Application” and located inside “src/lab01/Application” directory in your lab01 repository.
3. The report, stored in `/writing/report.md` and written in Markdown, that contains the planning portion as described above, and provides answers in all remaining sections (follow the prompts inside the report document).
4. Lab session on January 31 will be used to conduct project walkthroughs and for lab work.
5. The beginning of the lab session on February 7 will be used for demonstrations.

**Evaluation of Your Laboratory Assignment**

The grade that a student receives through Sakai on this assignment will have the following components:

- **Percentage of Correct GatorGrader Checks and Travis CI Build Status [up to 15%]**: Students are encouraged to repeatedly revise their submission to ensure that it passes all of GatorGrader’s checks about, for instance, the length of the report, its appropriate use of Markdown, and inclusion of minimal programming constraints.

- **Mastery of Verbal Explanation during Walkthrough and Demonstration [up to 20%]**: Since the continuous and timely project development and the ability to communicate technical details of a project is crucial to building successful software and hardware applications, a portion of students’ lab grade will be determined based on the quality of the project walkthrough and the project demonstration.

- **Mastery of Technical Writing [up to 20%]**: Students will also receive a portion of the lab grade when the responses to the technical writing questions presented in the ‘writing/report.md’ reveal a mastery of both writing skills and conceptual and technical knowledge. To receive this portion of the grade, the submitted writing should have correct spelling, grammar, and punctuation in addition to following the rules of Markdown and providing conceptually and technically accurate answers.

- **Mastery of Technical Knowledge and Skills [up to 45%]**: Students will receive a portion of their assignment grade when their project implementation reveals that they have mastered all of the technical knowledge and skills developed during the completion of this project. As a part of this grade, the instructor will assess aspects of the project including, but not limited to, the correctness of the program, the completeness and correctness of the software and hardware integration, the use of effective source code comments and Git commit messages.