# 2 Project Plan

## 2.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

Which of agile, waterfall, or waterfall+agile project management style are you adopting? Justify it with respect to the project goals.

- Waterfall+agile is the system style we plan to implement. For parts that require our expertise (Electrical and Software) we want to split up into groups and divide the work. This follows an agile style. Once these parts are finished and to comes to building and testing, we need a structured flow to make sure we can both meet deadlines and test our system correctly.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

- We plan on using Github for the software side of the project to organize and collaborate with code from all the members of the software team. We can also use the documentation tools of GitHub to show the intricacies of our project.

## 2.2 TASK DECOMPOSITION

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

-To complete the building of the cabinet, we first have to get the required resources, such as the wood, wiring, computer, and glass to put it together.

-To complete the software side, we first have to make a UI, code games into the program, and make sure the backend and front end are communicating with each other. We could also set up automatic deployment to the cabinet for easy updates to the software.

## 2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

- key milestones
  - Collect 100% of the necessary parts
  - creation of all control circuits
  - Create a basic UI that can be controlled with keyboard controls
  - run a basic game
  - creation of the physical box
  - final project test
  - design and aesthetics
  - Final presentation

## 2.4 PROJECT TIMELINE/SCHEDULE

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar (including both 491 and 492 semesters). The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Project schedule/Gantt chart can be adapted to Agile or Waterfall development model. For agile, a sprint schedule with specific technical milestones/requirements/targets will work.
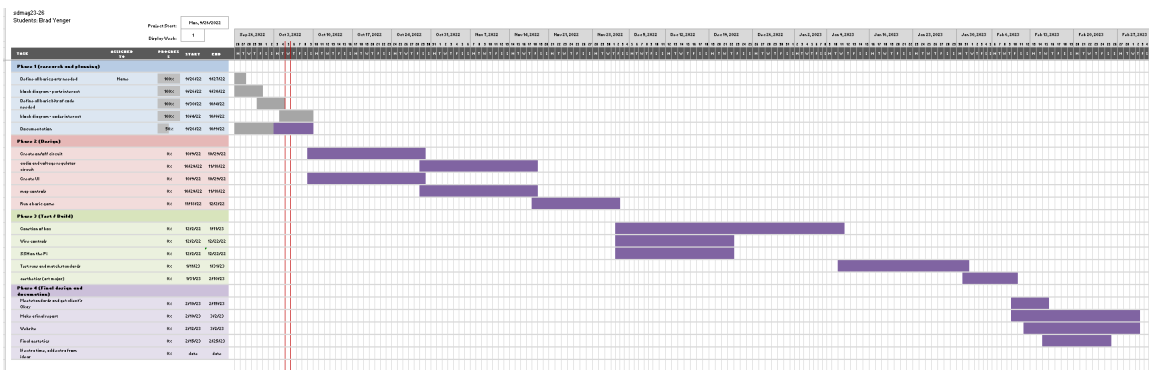
| Phase 1 (research and planning) | | | | |
|---|---|---|---|---|
| Define all basic parts needed | Name | 100% | 9/26/22 | 9/27/22 |
| block diagram - parts interact | | 100% | 9/26/22 | 9/30/22 |
| Define all basic bits of code needed | | 100% | 9/30/22 | 10/4/22 |
| block diagram - codes interact | | 100% | 10/4/22 | 10/9/22 |
| Documentation | | 50% | 9/26/22 | 10/9/22 |

| Phase 2 (Design) | | | |
|---|---|---|---|
| Create on/off circuit | 0% | 10/9/22 | 10/29/22 |
| audio and voltage regulator circuit | 0% | 10/29/22 | 11/18/22 |
| Create UI | 0% | 10/9/22 | 10/29/22 |
| map controls | 0% | 10/29/22 | 11/18/22 |
| Run a basic game | 0% | 11/18/22 | 12/2/22 |

| Phase 3 (Test / Build) | | | |
|---|---|---|---|
| Constion of box | 0% | 12/2/22 | 1/11/23 |
| Wire controls | 0% | 12/2/22 | 12/22/22 |
| SSH on the Pi | 0% | 12/2/22 | 12/22/22 |
| Test runs and match standards | 0% | 1/11/23 | 1/31/23 |
| aesthetics (art major) | 0% | 1/31/23 | 2/10/23 |

| Phase 4 (Final design and documation) | | | |
|---|---|---|---|
| Meet standards and get client's Okay | 0% | 2/10/23 | 2/15/23 |
| Make a final report | 0% | 2/10/23 | 3/2/23 |
| Website | 0% | 2/12/23 | 3/2/23 |
| Final aestetics | 0% | 2/15/23 | 2/25/23 |
| If extra time, add extra from ideas | 0% | date | date |

The full excel file can be emailed on request (bradleyy@iastate.edu)

## 2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Agile project can associate risks and risk mitigation with each sprint.

|  | LikelyHood | Severity | Mitigation | severity after mitigation |
|---|---|---|---|---|
| Overheat/fire | moderate/low | major | Apply external cooling | low |
| Raspberry pi not able to run software | moderate/low | moderate | have other options to run software | low |
| Power being disconnected | low | low | ensure wire and cords are properly stowed neatly | low |
| software crashing | high/moderate | low | extensive testing and allow a reboot option for user | low |

## 2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in the total number of person-hours required to perform the task.

| Jobs | Expected time commitment (hours) |
|---|---|
| power circuit | 3 - 5 |
| on / off control circuit | 2 - 5 |
| audio control circuit | 1- 5 |
| UI | 10 - 15 |
| Memory and adding new game protocall | 7 - 15 |
| mapping controls | 10-20 |
| carperty for frame | 2-5 |
| constructing frame | 5 |
| testing | 20 |
| final documation | 10 |

## 2.7 OTHER RESOURCE REQUIREMENTS

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

Requirements for completion:

- Lumber
- Workshop tools (saw. planer, hammer, nails, etc.)
- Monitor
- Raspberry Pi 3
- Keyboard
- Mouse
- Retro game controllers
- Arcade joystick and buttons
- Speaker and other audio equipment
- Various wires (power adapter, hdmi, etc.)
- Laptop with coding software

ca