

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.

For this project, our team has a management style that is a combination of both waterfall and agile. The main reasoning behind this is we input styles from both within our project. Our timeline is adaptive as we can change/switch things around as time goes on, our client involvement is more on the waterfall side as we have check-ins but not as common as agile, for iteration we follow more closely with waterfall as we operate in phases more than likely, lastly, we have high flexibility which relates to agile and high planning required which relates to waterfall.

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.

Below are the different ways we will track progress throughout this course and the following semester:

Discord - We have used discord throughout the semester already as a form of communication, however, we also have channels that are filled with important dates, information and other goals that we have for this project.

Github - When we start working on the backend logic of the CySim games it will make a lot of sense to have a centralized location where all of our code can be stored and accessed, Github is one of the easiest to use and we all have previous experience with it.

Google Drive - Our group has a shared google drive where we input all of the important documents that we may need and anything else that is important to the 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.

For the task decomposition we are taking more of a waterfall design approach:

- **Requirements**
 - Developing the CySim Field, this is the main interface for where the games will be played.
 - Backend Logic, we will code the different games and scenarios being played so that CySim can actually function properly at the current time.
- **Design**
 - We were provided with a layout of what the client wants the “arena” to look like.
 - SOC/Scoreboard, we are tasked with creating an efficient and good looking design for the scoreboard so that it displays all of the information that the users, spectators and anyone else in the facility may need.
 - Designing the main interface for the field, this will be in connection with the scoreboard.
- **Implementation**
 - Work on the backend logic of the game so that it can be uploaded to the server once functional.
 - Work on the CySim field that will hopefully display the results and other aspects of the backend logic.
 - Implement a way to combine the user interface field with the backend logic so that it all functions properly.
 - Pull our designs for the scoreboard and then implement them into the scoreboard layout to see if it fits properly.
- **Testing**
 - Testing/debugging the backend logic for the CySim functionality to make sure that all of the scenarios are working correctly.
 - Making sure that the user interface as well as the scoreboard layout looks correct and also functions in the way that it should.
- **Maintenance**
 - As time goes on, look for potential problems or errors or even things that can be adjusted for efficiency sake.
 - Along with this we then change/edit code or design features to make sure that the issue no longer occurs or that the design is more efficient.

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).

Below is some key milestones that has been assigned to us within the project:

Backend Logic will be fully functional for all scenarios that the client wants implemented for CySim.

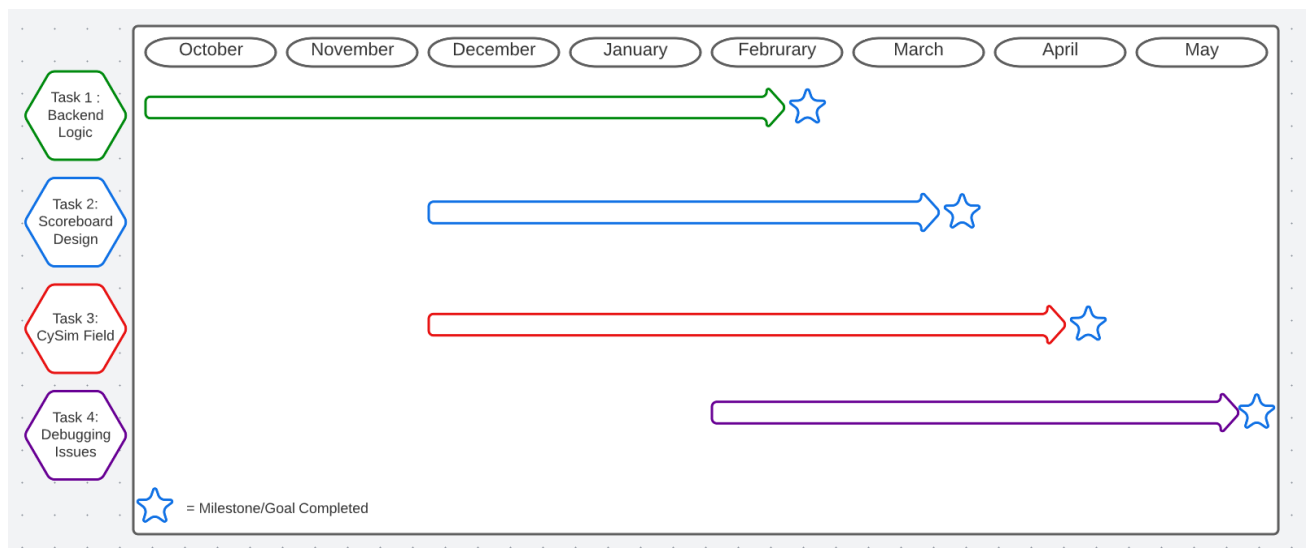
The CySim field will have an approved design and will be implemented in the frontend.

The scoreboard will also have an approved layout that makes it an excellent visual to fit all of the different players/spectators/members needs.

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.

Below is our Gantt chart showing our Project Timeline and schedule:



We will continue to update this and add more as time goes on within the project.

https://lucid.app/lucidspark/6f8a980b-7b45-4b7b-8c35-d79a9547095d/edit?viewport_loc=-25%2C-9%2C1920%2C1075%2Co_o&invitationId=inv_dcee85d2-ed18-40ca-bd6a-0efa6e7eef02#

That's the link to the chart I made if anyone wants to edit it

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.

Task 1: Creating the Backend Logic to the scenarios within CySim (Make it playable/functional)

- **Risks**
 - **Bugs within the logic we develop that we don't find/see**
 - **Risk probability: 0.6**
 - **This could lead to the scenarios failing for no reason due to various bugs**
- **Mitigation Plan**
 - **In depth debugging by each group member to ensure that everything has been tested to the fullest and with that we can confirm confidently that we don't have any errors.**

Task 2: Developing the scoreboard design so that it can show everything the client wants it to show as well as be visible to everyone.

- **Risks**
 - **The ratio of our design does not accurately represent the scoreboard and because of that everything we designed doesn't work.**
 - **Risk Probability: 0.2**
 - **Easy fix, clarify dimensions with client ahead of time**

Task 3: Creating and implementing the CySim field which will display the user interface for playing the different scenarios

- **Risks**
 - **The user interface has trouble relating to the backend logic and doesn't work properly because of that.**
 - **Risk Probability: 0.3**
 - **Other issues such as items not displaying or loading in properly as well could be an issue.**
 - **Risk Probability: 0.2**

Task 4: Debugging and testing all of the previous tasks to make sure that no existing errors occur and everything works properly.

- Risks
 - While going through and testing we find something major that could cause a major change in design and if we take too long we could risk running out of time.
 - Risk Probability: 0.5
- Mitigation Plan
 - I would say the best way to prevent this would be to implement debugging and testing even more than normal and do it as we move forward in the project, this helps limit big issues getting pushed to the end as well as allowing us to finish with plenty of time.

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.

Below is our personnel effort requirements:

Team Members	Tasks to be worked on	Projected Person Hours
Brady Schlotfeldt	Backend Logic, UI and Scoreboard Design, further Testing and Debugging	55 Hours
Matthew Daoud	Team Management, Backend Logic, Testing and Debugging	60 Hours
Bailey Heinen	Backend Logic, UI and Scoreboard Design, further Testing and Debugging	55 Hours
Ethan Swan	Network Architecture, Scenario Design, Testing and Debugging	60 Hours
Jacob Boicken	Network Architecture, Scenario Design, Testing and Debugging	60 Hours

2.7 OTHER RESOURCE REQUIREMENTS

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

Below is a list of resources that are provided to us:

- **Server**
 - We are provided from the client with a server for which we will use to implement our backend logic
- **A floor layout/map of projected facility**
 - A design provided by the client that displays how they want the facility to be laid out.
 - This helps us know where people will be so that we can take that and input it into our scoreboard design.