**Project Charter**

**Connect Four**

**6/10/2019**

**PROJECT STATEMENT:**

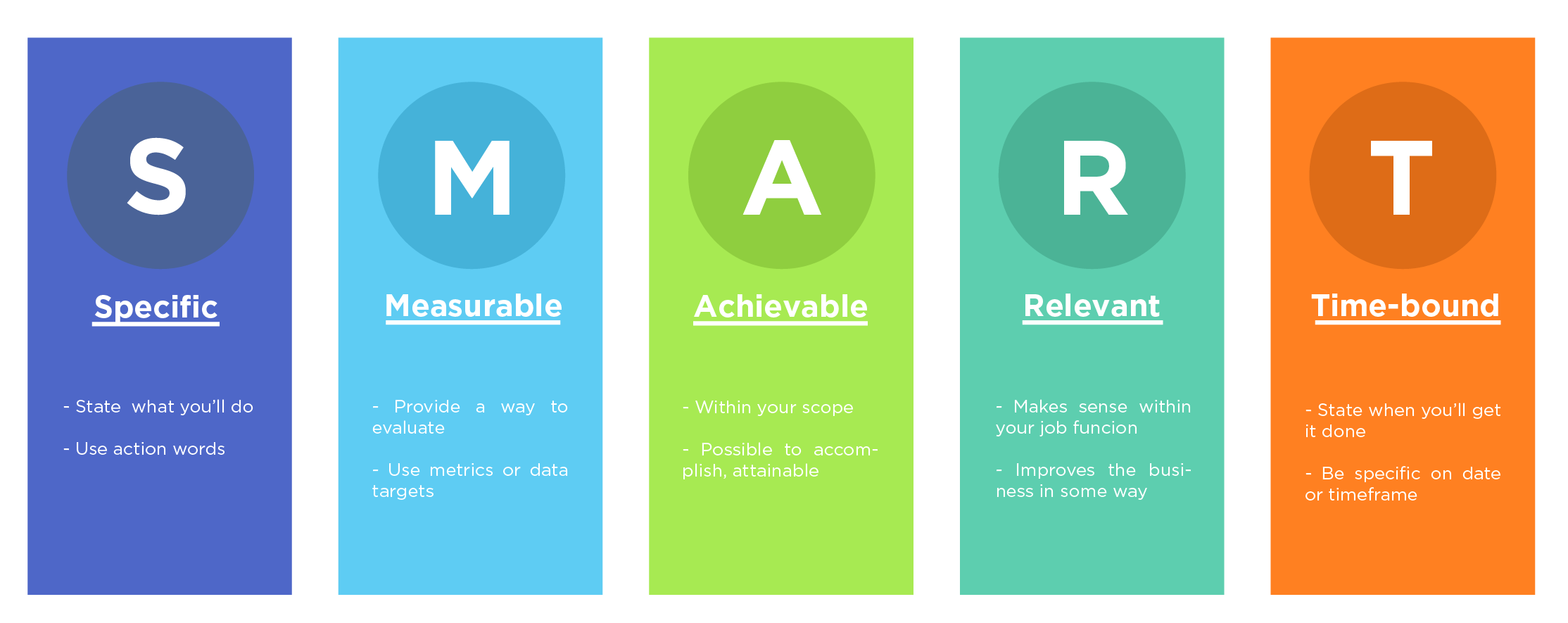
To understand how game algorithms work, our group wanted to create a game that would evaluate our learning of how to create a successful game in computers. To find what game we wanted to make, we went over multiple ideas such as tic tac toe (ultimate) which we discarded because of its simplicity, chess which we discarded because of its vast availability online already and complexity. Although many Connect Four solvers exist online, Connect Four is a “solved game” and these programs are deterministic and will *ALWAYS* win when playing a human opponent. We intend to create a program that is challenging but not frustrating.

**CONCISE PROJECT OVERVIEW:**

We decided to make a Connect 4 board and AI with python. This game would be a single player game against a variable skill-level bot so that the users can play a challenging but not frustrating game whenever they want. To make this game, we used the school computers and personal laptops to code, and we learned how to program in python. We did not need to spend any money on this project so $0 for budget. No approvals needed. Key stakeholders are Neil Hazra, Matthew Seminatore, and Sasank Chennapragada. We are assuming that the user understand how to play the game. The only constraint is time as the project must be finished before the deadline June 10th.

**SCOPE STATEMENT:**

<https://fitsmallbusiness.com/smart-goals-examples/>



**Specific:** Create a working Connect-4 game and bot to play against with variable skill levels.

**Measurable:** A working GUI will make it easy to see that the program effectively achieves the goal. We will test our bot against multiple students and peers to determine if the program is challenging but not frustrating.

**Achievable:** While learning the basics of python is a difficult task, we are up to the challenge. Creating a minimax tree and a game tree is also achievable as we have learned recursion and trees during this years IB CS HL course.

**Relevant:** We all enjoy playing games, so this project makes sense for us to do. Trees and recursion are relevant to our course.

**Time-bound:** The project is due 6/10/19

**STAKEHOLDERS:**

**Neil Hazra:** Will create the game board class to represent that current state of the game. The game board class must contain a representation of the location and colors of the pieces. It should contain helper functions that will be able to “score” the board for red. I.e. quantify who is winning at a particular position. It contains the move needed to make, function that rates who is winning

**Matthew Seminatore:** Will create the visual (UI) aspects of the game, making it more playable and accessible for the end user.

**Sasank Chennapragada:** Will do the documentation, make the code more readable, and proof-read it to make sure there are no errors or time-consuming parts of code. He will re-write any code that is not time-efficient.

**TIMELINE:**

May 30th: Discussed ideas for the end of the year project

May 31st: Was not present due to physics field trip

June 3rd: Started the project by downloading python and creating the project. Installed numpy and tkinter for matrix manipulation and graphics. Created git repository for the project. Started the documentation and the algorithms at home.

June 4th: Familiarize ourselves with python and continue work on the solver and gameboard class.

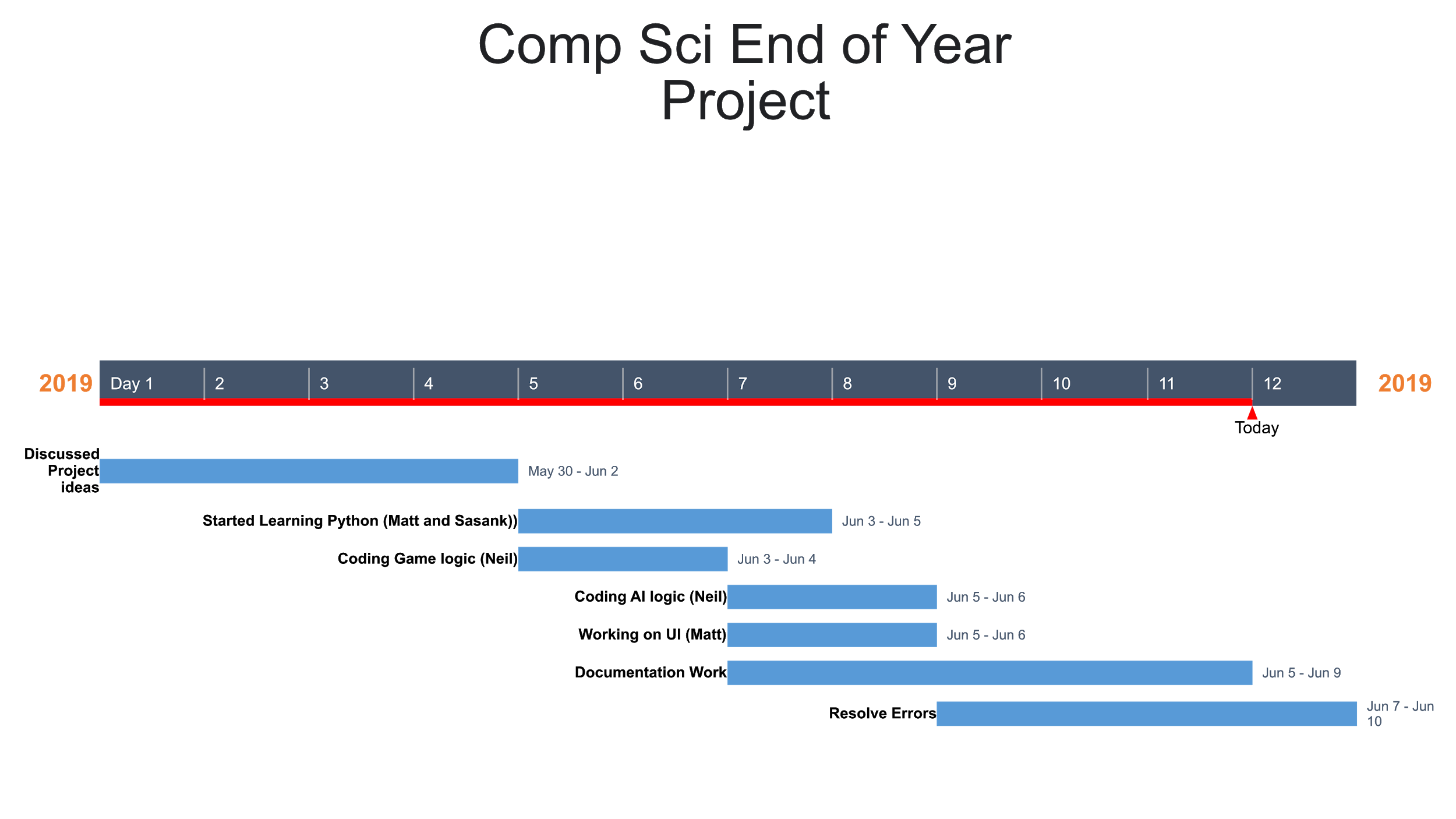
June 5th: Finished Project Statement and Concise Project Overview. Debug the algorithms and logic of the solver class and start the UI part of the project.

June 6th: Finish UI and pair program to catch bugs in development

June 7th: Resolve errors and tidy up the code

June 8th & 9th: Complete documentation

**Gantt Diagram/ Work Breakdown**



**BUDGET ESTIMATE(S):**

Money-wise, the project did not cost anything . We have under 4.5 hours of class time to finish up the project due to the small class days and the physics field trip. This forced the group to do 18 hours of coding and documenting outside of school. We used school computers and personal laptops to complete this project

**RISKS AND CONTINGENCY PLANS:**

**Difficulties Learning Python** (We could switch back to Java, a language we are more comfortable using)

**Difficulties Learning Alpha-Beta pruning** (this is optional, a backup to this would be to create a less powerful/efficient AI that scanned fewer subtrees)

**Not enough time needed to finish all we want to include**

If we need to cut some of the project out, we can reduce the complexity of the AI.