https://github.com/sjgilligan/EC327-Final-Project

Team Roles

The project has five team roles:

1. ***Project Lead*** **- This person coordinates all efforts of the project.**
   - **The project lead is responsible for organizing and maintaining the schedule of all work being done for the project. This includes:**
   - **o Setting up the timeline, milestones, and objectives.**
   - **o Interfacing with all team members to maintain progress during development, including calling and organizing meetings.**
   - **o Completing all required work that other members fail to do (i.e. a catch-all for ambiguous tasks).**
   - **Submitted documentation for timelines, objectives, and meetings. –** ***ProjectTimeline.doc***
2. ***Specification Lead (Matthew Guacho)*** **– This person is responsible for the specification of the project in terms of performance, functionality, and software architecture. This includes:**
   - **Checking all design specifications for system-level problems and performing system-level integrated testing.**
   - **Commenting and readability of the code.**
   - **Testing regime - thoroughly test your code and provide comments/code for easy testing.**
   - **Overall software architecture description and images –** ***ProjectArchitecture.ppt***
3. ***Interface Lead*** **– This person is responsible for how users interact with the software. They will be responsible for the following areas that will be graded as part of the project:**

- **Crisp and intuitive - how easy is it for a user to understand what to do; is the feedback (e.g. error messages) clear and complete?** ***Scale of 1-10 evaluated by entire course staff.***
- **Robust - is it possible to crash the GUI or have it do inappropriate things (e.g., leave windows or icons on the screen, perform incorrect actions)?** ***Scale of 1-10 evaluated by entire course staff.***

1

- Visually pleasant - does it properly utilize icons, backdrops, and other artifacts to improve the user's experience? Keep in mind that sometimes less is more (and sometimes it isn't). ***Scale of 1-10 evaluated by entire course staff.***

4. ***Technical Lead*** **– (Sebastian Gilligan) This person is responsible for the algorithms, data structures, and core technical aspects of the project. They will be responsible for the following areas that will be graded as part of the project:**

- **Figuring out what data should be passed between the GUI and the backend code (if applicable). and implementing the routines that will do it.**
- **Testing routines to demonstrate the proper passage of data.**
- **Efficiency - the implementation should be efficient enough to provide a good user experience.** ***Scale of 1-10 evaluated by select course staff.***
- **Robustness - the back-end should never crash or enter an infinite loop; it should also always provide reasonable data to the other components.** ***Scale of 1-10 evaluated by select course staff.***
- **Basic security - there should not be reasonably obvious ways to get the project to do something wrong, -** ***Scale of 1-10 evaluated by select course staff.***

5. *Documentation Lead (Daniel Juweid)*- This person manages all documentation and marketing for the project. This includes:

- Setting up a version control system for maintaining all code and documentation used by the project. *Provide link to course staff showing your repo (GitHub).*

- Packaging all code into one application, with appropriate resources, and distributing it.

- Putting together (and writing, if necessary) all documentation for the project, both front-end and back-end, in an organized, concise, and appealing fashion. *ProjectDocumentation.doc*

- Getting all team members to agree to a statement of work at the end of the project, describing what each member contributed. *StatementOfWork.doc*

- Putting together a short video presentation of the project, including motivation for why the project is useful, who will use it, how they will use, and why they will use it. *Video on course YouTube page.*

Timelines:

- First Team Meet Up (Setting up leads/roles) **November 28th, 2022**
    - Project Lead: **Steven Phung**
    - Specification Lead: **Matthew Guacho**
    - Interface Lead: **Nick Ramondo**
    - Technical Lead: **Sebastian Gilligan**
    - Documentation Lead: **Daniel Juweid**
    - **Extra help: Zhilang Gui**

Objectives:

The objective of this project is to create a Connect-4 game with only using C++ programming language. For starters, Connect-4 is a 2-player board game with a size of 6 rows and 7 columns. Each player takes turns dropping their piece in one of the 7 columns. The winner is decided if one of the sides makes a 4 in a row. It can be horizontal (----), vertical (|), positive diagonal (/) or negative diagonal (\ ). However, if both sides are unable to make a move once the board has 42 pieces and there is no 4 in a row, it is considered a draw. With our game we would like to add in the option to make the opponent a computer as well with varying levels of difficulty.

Outside of programming:

- **The whole team meets up in one place prepping for the project**
- **Once we finish the objectives below and have everything we need, let's meet up and work together to put all in one place**

Anything in red text is optional if we want to make connect-4 interesting.

I'll set up each part we need to complete the project (Everyone is allowed to program).

- **One of us creates a GitHub link for all of us to post our code**
- **Create a game board of Connect-4/A Display( ) function**
    - It's up to you to decide what the board should look like
- **Create a function that checks the endgame conditions:**
    - If one of the sides got a 4 in a row
        - horizontal (----)
        - vertical (|)
        - positive diagonal (/)
        - negative diagonal (\ )
    - If the game is a draw
        - If the board == 42 pieces
        - Should have a counter for how many pieces, in total, dropped
    - Display who is the winner, otherwise display draw
- **Create a display on the console**
    - Title: "Welcome to Connect 4!"

- **Prompting players**
    - Each player will provide their names
    - Each player will have their own piece
        - Player 1 is “X”
        - Player 2 is “O”
        - We can randomize who can go first. If not, then Player 1 will always go first.
        - If we want to get more interesting, we can add picture of the pieces
- **Create a function that keeps track of the score (amount of wins) for each player**
- **Create a function that reset the board if prompted to play again, otherwise terminate the program.**
- **Create computer player**
    - **Will be a class with member functions**
    - **Need a member function to “read” the board and determine wether any moves will win the game outright**
        - **No matter if they are reachable or not**
        - **If reachable for computer player they will do that move and either win or block you from winning**
        - **If not reachable the computer will avoid**
    - **Another function to analyze moves that will not win but advance play**
        - **Attach values to each move**
        - **Ex +0 for moves that give 1 in a row, +200 for 2 in a row, +400 for 3 in a row**
        - **Program will always select the highest value move**
            - **If 2 are the same value it will select at random**
- **Will culminate in an engaging youtube video demonstrating our game with the “World Cup of Connect 4” where we will compete against each other to demonstrate the functionality of our output and how we used class concepts to develop our program. This video will verify that all the objectives outlined above were achieved.**
- **Create restart function**

- **Two commands Y for Yes and N for NO.**
- **If YES, restart the game.**
- **If NO, printout goodbye.**

In-Person Meetings:

- Preliminary Meeting: November 28, 20222
- Second meeting: November 29, 2022
- Third Meeting: December 1, 2022
- Fourth Meeting: December 2, 2022
- Fifth Meeting: December 7, 2022
- Final Meeting (Video Filming): December 9, 2022