Red is critical, blue is floating, green is bonus

* Planning phase / Design documents (Consider the board games we want to cover and what features we need for each one)
  + Framework for UI
  + Object diagram
  + Documenting the scripting rules (if this then that) we will include.
* Implement the backend code for the scripting rules, piece movement, etc.
  + Code for the scripting rules
  + Code for the board structure
  + Code for piece movement and triggering events
  + Code for structure of a turn
* UI for scripting
* Design UI for game creator (longest part)
  + Board selection
  + Piece creation (sprite selection, movement rules, etc.)
  + Tile conditions
  + Win conditions
* Design UI for gameplay
  + Displaying board, pieces, inventory, score and other non-interactable UI, etc.
  + Moving pieces
  + Triggering effects
  + Passing between turns
  + Endgame screen once somebody’s won
* Create options for sprites for pieces (close to beginning)
* Code to save completed created game as text file (middle to end)
* Testing by creating games and making sure they work (close to end)
* Documentation for the scripting language

Possible Schedule

* Phase 1 (Design Phase)
  + Framework for the UI
  + Object diagrams
  + List of rules/features to implement
* Phase 2 (more independent work)
  + Make classes with dummy functions (board, piece types, etc.)
  + Backend code for scripting rules
  + Backend code for board, pieces, etc.
  + Board selection and general game creation screens, with some dummy functions
  + Piece sprites
* Phase 3 (merging the two sides)
  + Make the scripting UI functional (connect UI with functions)
  + Attach scripting to pieces and tiles
  + Finish adding functionality to the UI
  + Win conditions
  + Save codes
* Phase 4 (playing the game)
  + Finish any backend gameplay-specific piece/board interactions not done previously (all creation elements will be done before this point)
  + Displaying board, pieces, etc. in the gameplay screen
  + Adding interactivity to the pieces and other UI elements
  + Turn progression
* Phase 5
  + Testing
  + LAN hosting if there’s time
  + Adding features we didn’t get to yet if there’s time
  + More testing

4) Propose a plan for updating the GANTT chart each week. Will you meet to update it collectively? Will you identify one person who will be in charge of actually updating the GANTT chart and exporting a PDF version along with some high level notes about how it has changed (more subtasks added, some subtasks completed, critical path slipping, new estimates)

* We will discuss GANTT chart updates during our regular meetings, which we should have at least once a week.

5) Describe your anticipated/target MVP as well as at least one "Plan B" MVP for if the schedule is slipping and a "Plan A+" if things are going really well.

* Our target MVP will include enough features in the game creator for users to be able to create games in the following styles: checkers/chess, tic-tac-toe/Connect Four, Battleship, Qwirkle.
* Our Plan B MVP is to ditch the idea of this being a general-purpose board game creator and just focus on building support for adding special rules, pieces, etc. to checkers/chess variations.
* If we have extra time, we would add LAN hosting support.

6) What is on the critical path in your GANTT chart towards your MVP?

* See the five phases above. The items in red are critical.

7) Identify a list of key risks to the success of your project. Make a plan for resolving the risks as early as is possible.

* Implementing the scripting might be harder than we’re expecting. It may end up being too hard for us to make a scripting system that is easy enough for users to use while also being versatile enough to make games in. The best way we can resolve this is just by implementing the scripting sooner rather than later. If the scripting proves to be too much of a challenge to implement, that will force us to downgrade to the checkers/chess variants only version of the project.