Design Proposal [50 pts]
Your Design Proposal should be submitted as a directory containing several files:

- **Project Proposal** [30 pts]
  Write up a proposal file (in the file proposal.txt, or .docx, or .pdf) which should include the following components:

    - **Project Description** [5 pts]: The name of the term project and a short description of what it will be.
        - Growing up, my family and I were often immersed in games of Scrabble on our Friday nights more than shopping trips and dinners.
        - Therefore, this project will be a computerized version of the board game Scrabble with two versions. It will be called Ultimate Scrabble. The first version will be a two-player multiplayer game in which one can play with a friend using the same device. The players will have a three minute time frame to play each round. The second version is one-player player vs. computer version in which the player can choose the computer's level of experience (from novice, intermediate, and advanced) and play against AI. The goal will be to attain the highest points by forming words in rounds of seven tiles (until there are no more tiles left) in both versions.
    - **Competitive Analysis** [5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.
        - Pogo Scrabble
            - My project will be similar to this online version in that there will be a player vs. computer option which will run with AI. The player will be able to choose the AI difficulty (from novice, intermediate, and advanced) to play against. There will also be an option to play against another player (but from the same device).
            - My project will be different to this online version in that there will not be advertisements (which are important for this particular website to pay for it running). Also, there will be no registration requirement to play a multiplayer version. The multiplayer version will also run from the same device (not over the web).
        - Funky Potato Scrabble
            - My project will be similar to this online version in that the player will be able to play against the computer. The design of the UI will also be similar and the board will have rewards such as double/triple letter/scores, etc.
            - My project will be different than this one since the player will be able to choose the experience level of the AI. Also, there will be a multiplayer version and a time factor involved. If the player(s) do(es) not make a move within a three minute time frame, the player's turn will end and it will be the other player(s)' turn.

        - In essence, my game will combine features from multiple versions of online Scrabble without the unnecessary registration/payment requirements to enhance user experience and reduce inconvenience.
    - **Structural Plan** [5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or objects.
        - Mode startScreen
        - Mode leaderBoard
        - Mode gameMode
            - Calls and runs scrabbleGame class
            - Function timer
            - Function drawScreen
                - Function drawBoard
                - Function drawScore
                - Function drawTiles
        - Class object scrabbleGame
            - Boolean one/two player
            - Function calculateWordScore
            - Function possibleWords
            - Function createBoard
            - Calls class geneticAlgorithm to determine best moves
        - Class letterBag
        - Class geneticAlgorithm
            - Function findBestLocation
            - Function testPossibleMove
            - Function runAI
    - **Algorithmic Plan** [5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project.
        - AI to create a computerized player with different experience levels
            - Calculating the move that results in the highest score for each computer round will be the most difficult part of this project.
            - The reason for implementing a genetic algorithm in this case is because I am not sure of the specific weight the individual tile scores must have compared to their placement on the board (for the external rewards like double/triple letter/word scores).
            - My geneticAlgorithm class will run testPossibleMove for each word in possibleWords and store the data for which word placement results in the best score. I will test this for several different combinations of tiles to optimize the weights of the coefficients in the genetic algorithm equation. I will also add a threshold for a certain score for the computer player. If the score is reached by the AI, I will store the weight values (to use for optimization). I will increment the potential score to reach it using AI every time, and in the end, I will average the weights of all of these stored weights to result in the best overall AI

(especially for the advanced version). I will repeat these steps for a lower score threshold for the novice and intermediate computer experience levels.

- Player gaming experience
  - Drag and drop tiles on the board using the mouse
  - Word validity will be checked through a dictionary of potential scrabble words (with scores attached to each individual tile)
- Creating and displaying a leaderboard
  - I will store the best scores after the game has been played and store these scores along with a timestamp in a .txt file. I will later display this .txt file as part of my mode leaderBoard

- **Timeline Plan** [5 pts]: A timeline for when you intend to complete the major features of the project.
  - 11/20 Board generated
  - 11/23 Tile placement and game strategy implemented
  - 11/26 AI genetic algorithm implemented
  - 11/29 Leaderboard implemented
  - 12/1 Multiplayer version implemented

- **Version Control Plan** [3 pts]: A short description and image demonstrating how you are using version control to back up your code. Notes:
  - **You must back up your code somehow!!!**
  - **Your backups must not be on your computer** (ideally, store them in the cloud)
  - I will be utilizing Google Docs to backup my code using version control. I will be storing my files of codes in Google Docs, and using the Manage Versions feature of Google Docs, I will be able to access older versions of my code by checking the revision history of the doc. I will also store my all my files in Github.

- **Module List** [2 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!
  - Pygame
  - Numpy (potentially, but not for MVP)

TP2 Update:
Structure has slightly changed with player now a class (to allow multiplayer and the ability to initialize multiple players as objects)
Brute force AI currently being used (genetic algorithm may not be the best fit since there are no “weights” to

Currently looking into the DAWG algorithm as well as the implementation of certain heuristics such to re-evaluate scores of certain words

TP3 UPDATE:

- Could not get multiplayer feature to work
- Implemented computer AI so user can observe two computer players play each other
    - Calculated possible scores for all possible words and locations on the entire board while ensuring that the words are attached
- Help page implemented
- Implemented frequency heuristic as more complex AI to punish the computer score if letters that appear the most commonly in the Scrabble dictionary (which are also the lowest scoring letters) are used too frequently
    - See Scrabble.frequencyheuristic()