Description:
Collect & Conquer: 112. A strategy game where players compete locally against each other to become the dominant civilization. Players collect resources to build up their cities, fight other players on the map, and learn skills to aid in both of these efforts.

Similar Projects:
The main inspiration for this project is the game, The Battle of Polytopia. There is a map that is a square of tiles that may/may not have resources on them. There are cities around the map that can be acquired to aid in collecting "currency". The actual game contains extra features like water terrain, choices of civilization that affect spawn rates and skills, map exploration (but will likely be cut out due to time/complexity constraints) . I would like my project to most resemble this game in terms of base functionality. Some similar things would be farming of resources, turn-based stages, creating units, combat, learning new skills, city levels and expansion, etc...
The Battle of Polytopia trailer for reference: https://youtu.be/p-_HR7Uiens

Structural Plan:
The main structure of the game will rely on classes. There will be a large player class that stores various information about the player & their game status (such as current units, skill tree progress, city locations, currency amount, etc...). This class will manage/store information for the player, and it will use methods that interact with different classes. These different classes will account for actual game implementation. There will be a unit class that contains various subclasses for special units [Contains health, movement, defense, attack]. Methods include movement, checking for health There will be a tile class that contains subclasses for special tiles (like mountains, cities, grass, etc...). This class will be used in complement with the graphics to create the basic board of the game. There will also be a Skill Tree class that keeps track of game skills. Specific actions will locked unless the corresponding tree node is "True"

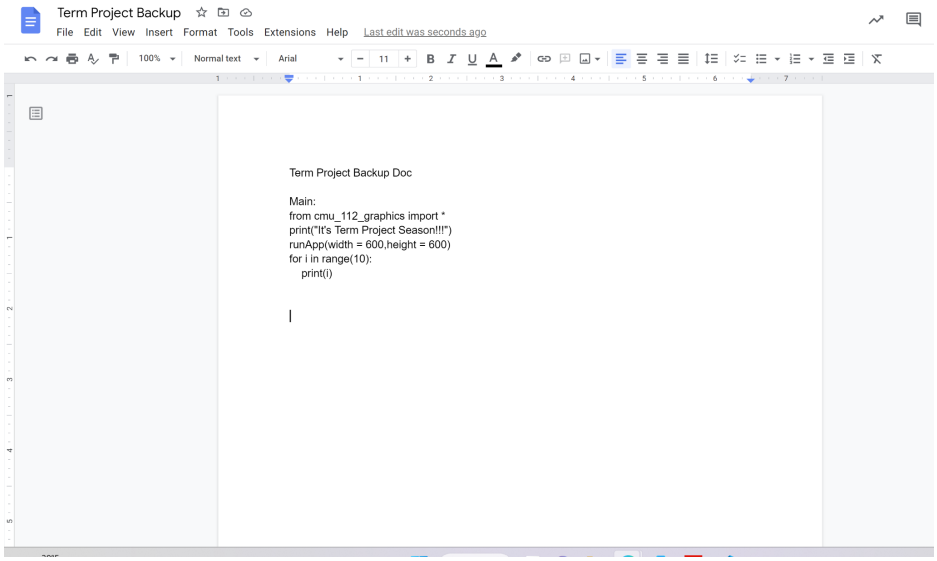
Algorithmic Plan:
Difficult implementation will occur in (possible) terrain and resource generation and accurate combat interactions between players, etc... Terrain and resource generation would rely on random number generation to make tiles and determine what resources are on them. Once a map is generated, a series of checks can be made to ensure that it is feasible. From there, mouse events be used for accurate interaction between the player and components of the game (resources, units, etc…) For combat, a turn-based system would be put in place. From there, movement checks and attack checks would be needed, and if an attack is carried, it should correspond with specific animations and update information for both players.

Timeline Plan:
11/17 - 11/18: Game Map
By this date range, I would like to have a viable map, consisting of an array of tile classes. The board creation process would have a minimum number of resources and cities available for both players. Information related to the map and tile would be store within the class, methods for collecting resources would be implemented (although some bugs would likely be present). Additionally, the player class will begin to have some structure. Relevant information will relate to the map, like the cities they have, how many resources they have/gain, levels and expansion, etc…

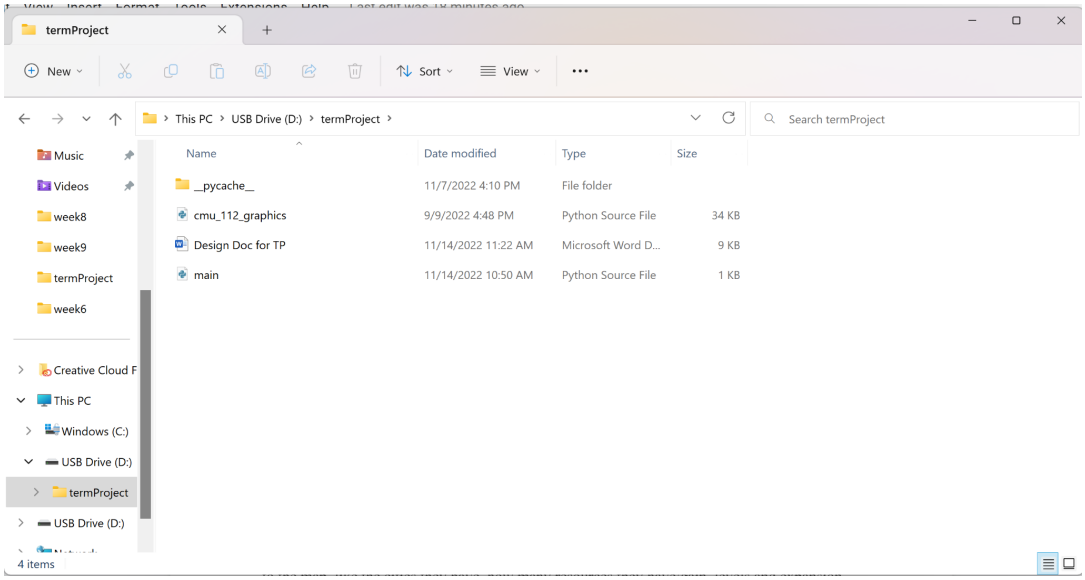
11/25: Full Units and Map Allocation
By this date, I would like to have implementation of units and civilizations without the restriction of a skill tree. Players will able to create any units they want and move them once per turn. Combat will be implemented with precise health, attack, and defense. There will also be some endgame functionality so that MVP can be achieved by TP2 deadline.

11/30 & Onwards: Skill Tree + Final Implementations
By this date and beyond, I would like to have implementation of the skill tree that locks aspects of the game, such as performing certain farming actions & making certain units. If the skill tree becomes a hassle, a working game should still be viable without it. Progression through the tree must be done with in game currency. From, here additional features can be added, such as more units, more resources available to players, increased map size, and more simultaneous players.

Version Control Plan: 2 Methods
*Copy and pasting code into Google Docs for cloud uploading



*Using an external USB flash drive to save files on a physical device.



Module List: No outside modules being used

TP1 Update:
Structurally, major changes include cut implementations of ruins and skill tree. Also considering using dictionaries to store Tiles and Units on the board in opposition to lists. Otherwise, program relies on classes, methods, and attributes as it should. Algorithmically, no major updates. Map generation using backtracking has been mostly successfully. Further time will spent resolving user mouse interaction and combat algorithm/complexity. Version control has been changed to using GitHub.