Status Summary:
Project Title: Defend Your Village
Team Members: Anuragini Sinha, Lin Shi
Work done:

- The UI has been completed for everything but tradecart.
    - All buttons included so far have functionality; however, some are incomplete.
    - Adding/subtracting villagers from a job
        - Adding resources based on villagers' job
    - Adding daily tasks that will produce resources
    - Adding buildings
    - Resources, buildings, and villagers are all reflected in the game so far
    - *Note: some of the above functionalities are incomplete, such as needed connection with another class.*
- Basic skeleton for all classes.
    - All decorator: AxeDecorator, SewingMachineDecorator, BaitDecorator, BowDecorator, MatchesDecorator, PickaxeDecorator, StorageDecorator, and MagicitemDecorator. However, the decorator still needs to be connected to the actual gameplay.
    - All People: Gather, Hunter, Trapper, Waterman, Tailor, Miner, Weaponsmith, Villager, Lumberjack, Repairer, Cook.
    - All buildings: Smokehouse, ClothesFactory, Bucket, Mines, Blacksmith, Trap, Hut, Tradecart
    - The two factories: BuildingFactory and JobFactory
    - UserActions class
    - *Note: the classes have all been created. However, the detailed implementation level may vary. For example, UserActions is mostly completed, we may mess around with the associated values, but the decorator classes are not used in the actual game yet.*
- Breakdown of work:
-

Changes:

- Game mechanics (some are just thought processes and have not been implemented):
    - The user gets to choose 3 actions to perform per day.
    - Each villager will cost *x* water, *x* food, and *x* wood per day
        - The villagers will be unhappy if the village does not produce enough resources for the villagers.
        - If the villagers are unhappy for *x* days, they will leave the village.
- Implementation changes:
    - moving the JFrame stuff from the Game class to GameUI class
        - For better readability
    - Name changes throughout the program
    - Small implementation changes throughout the code, but nothing big.

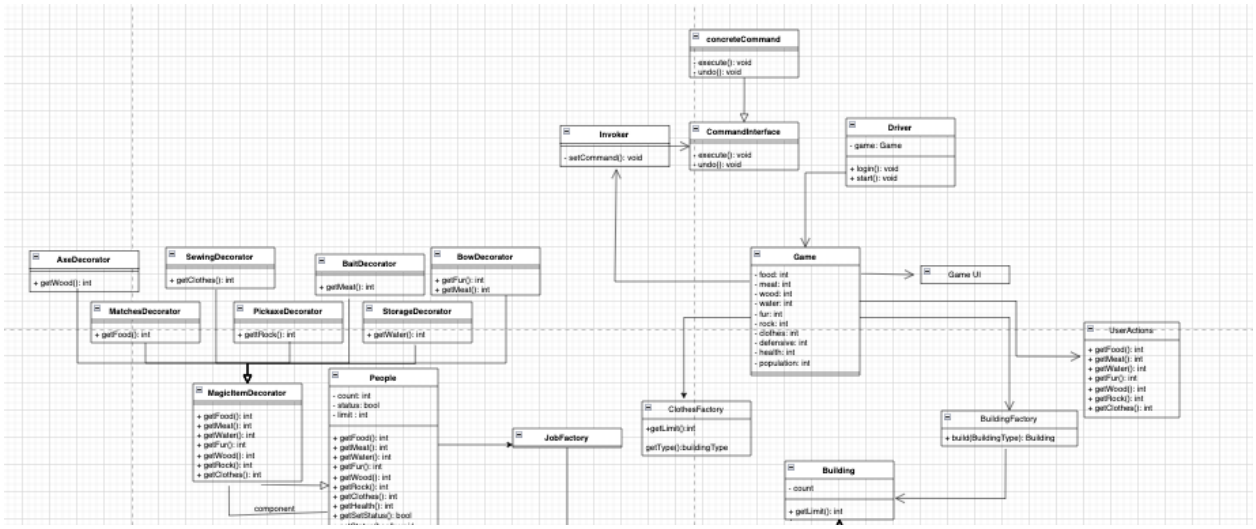
Issues:

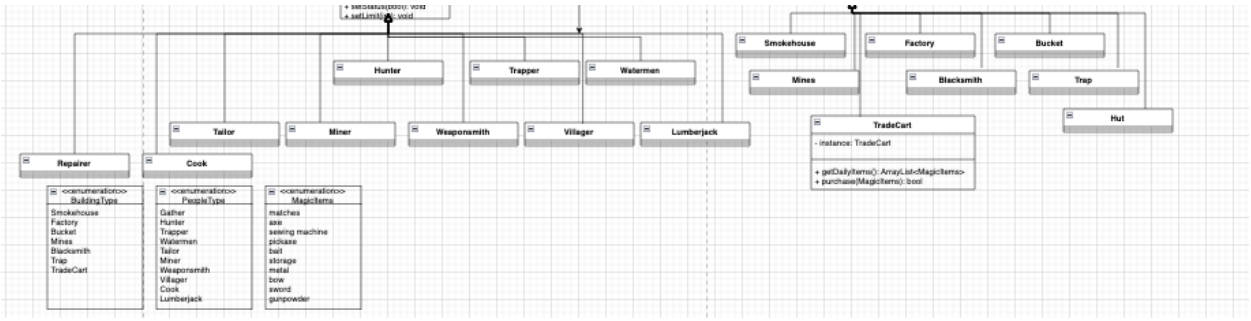
- Formatting the JPanel, (it just needs more time). We did not spend much time formatting the JPanel or assigning default values for our game. We decided these two will be low on our priority list; thus, we will work on that as one of the last tasks.

Patterns:

- Factory Pattern (completed)
    - We have different types of villagers. Each one will produce different resources based on their job. Therefore, creating a factory pattern will be useful.
- Decorator Pattern skeleton (full implementation is not completed).
    - There are magic items that will boost daily resources once purchased. Therefore, a decorator will be useful.
- Singleton Pattern (tradecart completed but not for event announcer).
    - Tradecart is a special building that will unlock stores to purchase magic items. Therefore, it can only be constructed once.
    - Additionally, there will be an event announcer on the side. Therefore, a singleton will be useful.
    - Singleton is a good option, given that we can only have one tradecart and one event announcer.
- Observer (not implemented).
    - The event announcer will announce daily updates when the user tries to complete some action that they are not allowed to perform and possible hints.
    - Given that the event announcer will only update when some action has been performed, an observer is appropriate for this task.
- Command Pattern (not been implemented)
    - There will be random events that are urgent, and the user must act upon them before moving on to the next action. Therefore, a command pattern will be a good fit.

Class Diagram UML:





Plan for the next iteration:

- Login, data storage
    - login page that will store the previous game progress
- event announcer
    - display game information after every action
- Random events (command pattern)
    - random events that could help/hurt the village
- magic items and tradecart
    - Adding buying of resources and magic items
    - Implement the decorator pattern
- Attacks on the village (probably part of random events)
    - Decrease the health
- Connect all the components
    - many pieces are loosely connected. Therefore, connecting everything together is important.

Project 6 was somewhat limited in workflow, given how the user interface was needed to actually implement/test the functionalities. However, we were able to complete the basic skeleton for almost all of the classes that will be used. Therefore, given that all (aside from tradecart) UI has been completed, the workflow will be much faster in the next iteration. We plan to use Discord for communication and claim “tickets,” small tasks, at a time to avoid conflict. Based on what has been completed and what we hope to achieve, we believe that the proposed program in project 5 will be completed by the end of project 7. As a player, you can create buildings and assign villagers to jobs to boost your resources. There will be attacks and random events that could help or hurt your village.