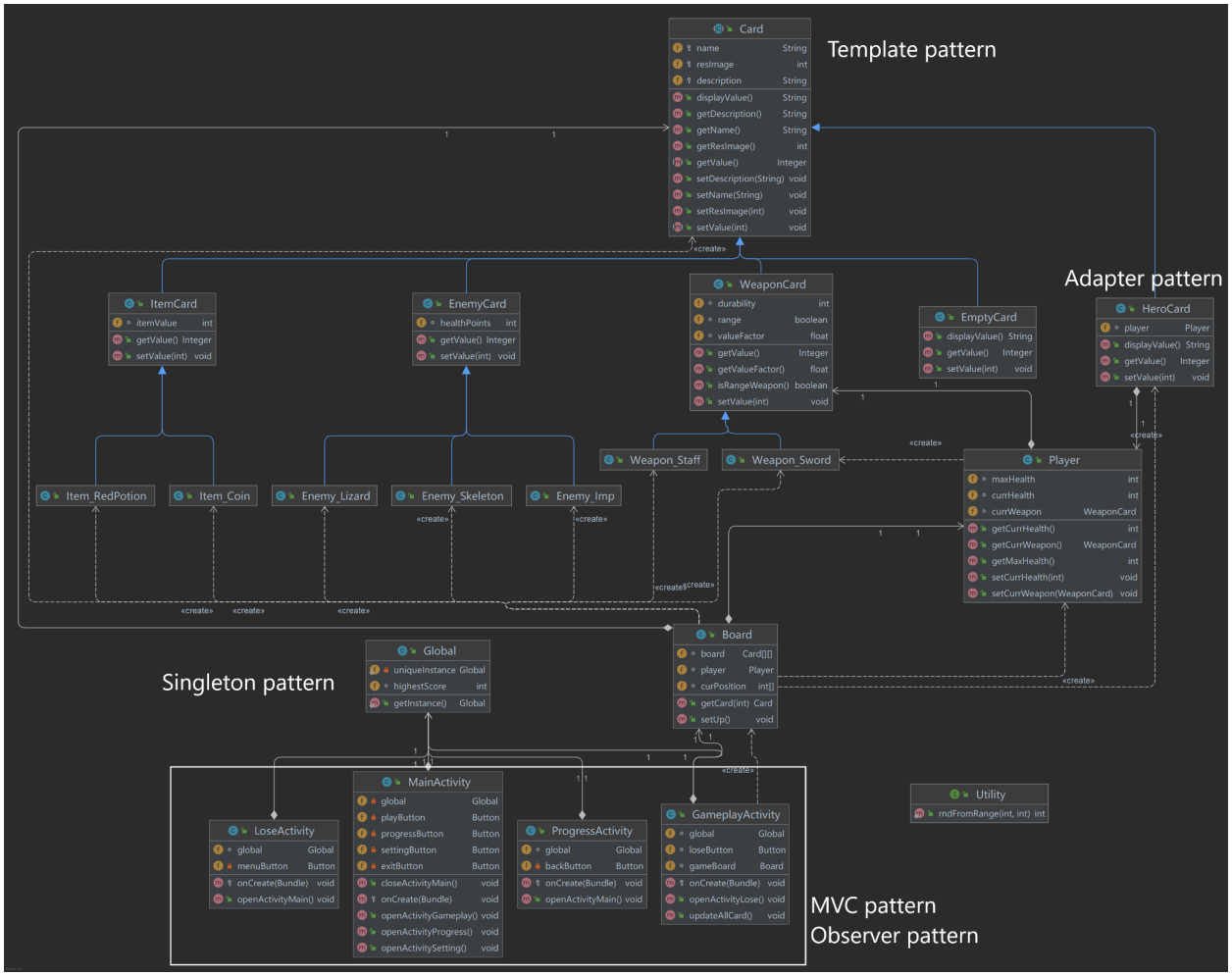
Status Summary

- **Project: Dungeon Crawler**
    - Elijah Aldinger, Daniel Helfrich, Tiancheng Shao
- **Work Done**
    - Elijah
        - Write the XML Activity files for the screens in the game (menu, gameplay, progress, lose screen) and the navigation logic/buttons to move between them on Android screen
        - Setup Java classes and structure for the Card system classes and Player class
        - Create custom OnClickerEvent class with a parameter so each button object connects to a method and passing in its index as parameter
        - Extend board class with checkCard to see if moving to a given card index is valid under the current player state
        - Detect button clicks and move the player card to the new clicked button, regenerating and rendering the board.
    - Tiancheng
        - Re-design the UI layout for gameplay screen
        - Complete Card abstract superclass, create its concrete subclasses, and assign corresponding data to them
        - Create Board class to handle actual gameplay and interactions during the game
        - Implement several design patterns during the process above
    - Daniel
        - Settings menu
        - Worked to get android studio and IntelliJ working
        - Draw home page
- **Changes/Issues Encountered**
    - Android development
        - None of us had done it before so it was a learning curve to get a build working for all of us and up on VCS
        - Once we did it was just Java though which we are familiar with
- **Patterns**
    - Singleton
        - Implemented as a sort of “global class”
    - Model/View/Controller
        - Implemented with Android Activity class defining the UI layout during the game, our actual logic/Controller in the Java classes being informed of actions by user input, and the Model for now is classes holding our image resources and card information.

- Observer
  - Implemented by setting onClickListener for buttons in activity classes, the corresponding function gets called once the button is pressed
- Template
  - Implemented with abstract superclass Card as a Template for all the subclasses. The superclass has concrete methods, abstract methods, and hook methods
- Adapter
  - Implemented by adapt Player object to a Card object, encapsulate the actual detail inside

Class Diagram



Google drive:

https://drive.google.com/file/d/1GNowehannpmGutIceEtdSTuPDoccRESn/view?usp=sharing

Plan For Next Iteration

- Work on the game logic/card interactions
- Work on different card types/cards within the types
- Work on saving/loading player data to Android file
- Work on Android XML/better way to implement the graphics
- Work on progress screen and ideas for game progress

Demonstration

4:30 11/19/21 with Bruce