For context, this application effectively makes an api call to a website for card images, then shows those images in a home view, and then logs those images in a separate scene to be swapped back and forth.

A diagram of a computer flowchart

AI-generated content may be incorrect.

The DiscordMTG ability app, is rooted in 6 key class; “DiscordMTG.java”, “BasicView.java”, “CardLogView.java”, “ResizeListener.java”, “CardGrabber.java”, “CardDisplayer.java”.

A screenshot of a computer

AI-generated content may be incorrect.

Here, preliminary things that extend the whole project are initialized as it’s the main file.

DiscordMTG extends GLUON’s MobileApplication to be used as a mobile application. The class registers 2 view’s BasicView, which is the main home screen, and the CardLogView, which is the secondary scene to be used later.

DiscordMTG also on startup runs a for loop to clean out any data from past runs in the Cards folder.

A screenshot of a computer

AI-generated content may be incorrect.

BasicView, effectively initializes all the nodes and panes to be used within the main home view. Moreover, it initializes preliminary or static elements that are not going to be changed, all changes made to nodes is done through the ResizeListener.java class

A screenshot of a computer

AI-generated content may be incorrect.

CardLogView, is very similar to BasicView, in that it just initializes all the nodes and alignments for pane to be used for the CardLogScene, however the main difference is that it initializes the cardlog\_button\_array, which is used to store an array of cards relating to passed over cardlog cards.

A screenshot of a computer

AI-generated content may be incorrect.

The ResizeListener class’ main function is to listen for and to update the UI upon changing. Moreover, it sets a scale x and y, based off of the dimensions 270w \* 600h which is scaled down by 4x in relation to be designed for a smart phone. However, the class use the window\_width and height along with the scale in order to update each element whenever the window changes size as there is an x and y listener. This is because the app is designed to be used on any mobile device so everything is relatively scaled through various formulas.

A screenshot of a computer

AI-generated content may be incorrect.

The CardGrabber class functions to make api calls to the website scryfall.com for card image data through the form of a .json file. When the class runs the Gamble() method, it makes an https web request to the scryfall website, then converts the string to a json through the google gson library to be accessed as a dictionary. Then, it access the dictionary for a URI link for the actual png, of which then it is then web requested again. The Gamble method also makes use of the in-class method write\_to\_card\_folder, as after grabbing a card it stores it in the Cards folder in resources. All the image views are sourced from the Cards folder through java input output stream.

A diagram with text and symbols

AI-generated content may be incorrect.

CardDisplayer, runs off the endstep\_trigger method, which is called by a button from the home BasicView. The method uses a constant initialized in the CardGrabber class to always be aware of the current index of cards generated. It’s because of this it then uses it’s own make\_card() method to generate a card and assign the imageview as a graphic icon. The make\_card() method also includes the swipe mechanic on allowing the cards to be dragged and “swiped” to be moved into the cardlog array to be shown later.