**General Stuff**

**App email:** [DMTransitionZone@gmail.com](mailto:DMTransitionZone@gmail.com) **Password:** dicemastersrulez

**Keystore password:** dicemastersrulez

**App Name**

Technically the app’s name is Dice Masters Companion, and I believe behind the scenes that’s how it’s named. That was the original name before I had to change it on Google Play because once when I tried to update it complained about copyright issues. I removed all mention of “dice masters” on the store and it seemed to fix the issue.

**Naming**

I generally named each die image as the name of the character all lowercase no punctuation. When needed, set initials were added. Same logic goes for the card image, they’re based on the card titles. If it’s an alternate art or blank card, I added “alt” or “blank”. I can’t guarantee they’re 100% named this way however, and for any future cards added, any naming convention that android studio will allow will work just fine.

**Spreadsheet Stuff**

**Foils**

If the card has a foil version, then the “Foils Owned” column should be 0. Otherwise -1. The app will only add a foil button if the count is 0 or higher. I didn’t intentionally add in foil imaged due to space concerns. Some images might be the foil version because it was all I could find.

**Rarity**

The rarity codes are weird because I wanted to be able to sort by them. The first letter is just for sorting purposes, the other letter(s) are the actual rarity. There’s a rarely used “fz” for zombie cards. “gp” is supposed to be all promo cards except alternate art versions of existing cards, those should be “halt”. I can’t remember if the app does anything different between those two, so I think those could possibly be combined into one if desired.

**Names**

If there’s a “ or other weird character at the beginning of the name, then putting a ‘ before the text in the spreadsheet will usually make it show up correctly.

**Things Not Used**

Max Dice is not used anywhere, that was something I didn’t get around to using in the app. I think most of the values aren’t filled correctly. Char/Die/Card ID, I don’t think these are used at all. I was attempting to come up with a way to uniquely identify characters, dice, and cards individually, but I don’t think I went anywhere with it. With each new card I kept adding those ID’s in case I ever used them in the future. I can’t say 100%, but I think that only the main ID value is ever actually used in the code.

**Versions Tab**

I kept all of the version information here. What I added with each version and what I wanted to add in future versions, which included user suggestions.

**Removed Cards Tab**

I originally had foils as separate card entries until they became a normal thing. These are just backups of entries that were removed.

**Important Code**

**MySQLiteHelper.java**

This is where the card database is populated. I would have set it up differently if I had known at the beginning how many sets there would be and how annoying this would get. The string data is all copied out of the spreadsheet, so that part is just tedious. There’s a limit to how long each string can be (I think because the SQL query can only be so long), so I have each set as its own string so it’s a little more neat. Every time the app is updated, the full table is dropped and recreated. A backup table is created with the cards/dice that are owned, and then those values are populated back into the table after it’s remade. I made a mess at one point with an update, the if statements in onUpgrade are a result of that.

**SQLDataSource.java**

This holds all of the SQL queries, each one with its own function. Some of these probably aren’t used anymore because CardCollector.java isn’t used, but I didn’t clean it up. There’s probably a better way of getting all of the card information than this, but it works.

**CardCollectorNEW.java**

This is where most of the work is done when the app pulls up a list of cards. It calls to the SQL table to get lists of all of the card info then passes all that to the list adapter.

**ExpandableListAdapter.java**

This sets the display for list of cards. I pass in viewType so that I didn’t have a different adapter for each way you can view the cards. As a result, there are several if statements looking at the type and changing display elements accordingly.

**DataManagement.java**

This controls the import/export feature. The actual work is done in SQLDataSource though. Not sure why I did it that way, seems confusing.

**ListBuilder.java**

I’m pretty sure this was made obsolete with CardCollectorNEW.java and is not used any more.

**Search.java**

Pretty self explanatory, it’s the code for the search feature. Android Studio yells at me for having too many views in the activity, but I didn’t know a better way of doing it. The code has gotten very long and cluttered due to the nature of what it is.

**Tools.java**

Not much to say about this other than it’s where the bulk set adding code is kept.

**Credits.java**

I have a donate link set up here. You’ll want to either change it to something of your own or remove it from the code.

**Updates**

1. Get all of the card images and save them to the drawables folder.
   1. Cards are all sized down to 500 px if larger
   2. Die images are pulled from the card image and sized to 240 x 40 px
      1. Resize full card images to about 259x367 to get dice right size
   3. These numbers aren’t set in stone, it’s just what looked best on my phone
2. Enter data for the cards into the spreadsheet, columns J through Y
3. Copy down the formulas in column Z and columns A through E
   1. Z is for copy/paste into the app
   2. Others are to help fill out ID values. So only column F is really needed if you scrap the half done ID system.
4. Columns C and D tell you what the CharID of that character is, and the total number of cards that have that ID. These numbers can be typed into columns G and I.
   1. If these are #N/A, then the character name needs to be added to the “Char Counts” tab
   2. If the rows color, then there are duplicate ID values somewhere. Double check all the colored rows.
5. Column H is really just filled in manually. I either remember how many different dice there are for that character, or I look it up. It’s a pain.
6. Update MySQLiteHelper.java with the new set.
   1. Add a new string
   2. Add an execSQL statement to onCreate
7. Copy the information from column Z into the proper string in MySQLiteHelper.java for each set that is new or had changes.
8. Update the search function for the new set and any new affiliations it has
   1. Add new checkbox(es) to search.xml layout
   2. Setup checkbox(es) in Search.java
   3. Add the search logic into the appropriate section in the on click listener below
   4. Add the clearing logic to the clear button on click listener
9. Update bulk add function
   1. Add a new entry to mass\_add.xml layout for the new set with the appropriate checkbox
   2. Setup the checkbox in Tools.java
   3. Add criteria logic into the add button on click listener below
10. Set the update message in MainActivity.java
11. Test to make sure nothing broke and you’re finally done!