**Previous Submission**

The previous submission explained almost everything in code comments. I am sorry I didn't bother to rewrite them as some weird essay-like mess. I really don't feel comfortable writing in any style that is not highly program-like. I even wrote this report in VS-Code, to avoid stress.

Anyways, I guess it helps to define what I did make in the last report. The last report was literally just Uno, with one new card added: All ++. The game also had added rules:

* non-number cards can be stacked (before their effect triggers; delaying their effect);
* the stacked cards will yield a new effect as a result; the new effect is described in a MASSIVE doc-string in new.py.

I don't know if you read the doc-string. Maybe you didn't understand it? I would feedback on whether you understand the doc-string now. I am referring to the doc-string right below compatability\_matrix.

**Todo**

I guess I should just write out the rules for the little prototype I made.

**Changes**

**This is novelty, right?**

This report will only describe changes from the previous report.

The main change with this report was that I redefined the goal for this project.

**New Goal**

Make a prototype for my card game. I will also remake existing games, because I want to be able to use them as a basis for my game. I have always wanted to make a card game, but the actual card game I am wanting to make is not complete at a conceptual level. If I am making a prototype, then I will need to make a library. Why? Well, I will end up changing the prototype many times in the future, after completing this project. I want to save future me lots of time and energy, so I will design this project as a simple library.

**New Title**

"Card Game Library (and Web App Demo for *King Taco*)" is the new complete title for this project. Obviously, the parenthesis can be omitted, leaving us with just **"Card Game Library"**.

**Achieving the Goal**

There are only a few problems with the current design of the main script. The problems are:

1. The main script is not flexible enough.
2. The main script does not have enough helper functions.
3. The main script lacks documentation in many areas.
4. The main script is not very modular.

I don't currently know how to do 4., and 3. is a low priority problem, since I try to make my code readable. I do include lots of comments and some documentation here and there, so I don't really need to worry about 3.. I will work on it eventually, once I have a better idea of how the code actually needs to be structured. 4. is going to require a lot more understanding of the game's design before I can accomplish it.

1. is extremely general, but I do have very specific sub goals. The run\_effect function takes in effect\_stack as an implied parameter and yields a tuple or **response data**. All of the return values are initialized and then modified. The modifications are done in a straight-forward and linear manner. So, I can turn the response data into a **class**, and make methods for adding cards to the effect stack.

The function already uses organized for-loops with lots of if-statements, so I can just extract the if-statements into methods. I will need a reset method, for when a new effect stack is formed, and I will have to move the run the for-loop iterations in the turn function.

**Global Vars**

I already have a bunch of global vars, such as effect\_stack. I may as well move them into a Game class and put the main functions into the Game class as well.

* this unfortunately means I will be including tons of nasty self.s, but that's okay.

**Deliverables**

The deliverables are to achieve the goal be solving the subproblems above.

**New Dependencies**

I am considering adding a new Python dependency, which I would only use for performance. This is the vector module. vector also require numpy. So, I guess that's 2 dependencies. I will probably use numpy more in the future, so I can make a maximum performance game. I would love to be able to run 1\_000\_000 game simulations per second.

**Next Up**

I already made an UNO clone. I may as well take this all the way and make a SUPER UNO clone. SUPER UNO is a variant of UNO, where I replaced all of the number cards with effect cards. This means that every play results in absurd effect combos.

The cards in SUPER UNO are:

* += 2
* --
* All ++
* x 2
* Skip
* Inverse
* NOPE
* Shield
* Rage
* Calm
* Lock
* Go Fish
* Curse
* Harvest

This set of cards is pretty wacky. I am not sure how balanced it is though. Good thing I can test it in a program now, I guess. Oh, and the wilds also have effects on them. The wilds are simply amplified versions of the colored effect cards. This would probably be more fun with more colors, since its way too hard to get rid of cards otherwise. It would probably also be good if there was some kind of currency system to keep players under control, such as mana that regenerates when you draw cards. Ok, actually, that sounds really fun. I think I am gonna add in mana and HP.

Alright, I have just written a lot of the rules for the new game, which is basically SUPER UNO with strategic resource management.

**Rename new.py**

I am renaming new.py to *"0th prototype"*, since that is what it is.

* new.py => prototype\_0.py

**King Taco, 1st Prototype**

Add the super uno cards, but just call it "King Taco".

**Next Up After That**

I guess I would like to program more existing card games. I may as well design the code to make them work. It would be super fun to combine the different cards from different kinds of games. I will start off with Monopoly Deal. It is a fun game. Another fun one is Canasta. I may as well implement their various rules, so I can consider different library designs.

Games to implement:

* Monopoly Deal
* Canasta
* Exploding Kittens
* Blackjack

**What is Happening**

**This is the only important section!~**

So, I spent a few minutes coming up with the UI for the game. Well, I actually spent hours imagining it in my head. Ah, I just love sitting back and imagining stuff in my head.

12:20 PM of 11/29/2023:

* I have already implemented quite a bit of my planned UI as HTML.

Some things on my todo list, for the UI:

* add side-menus, with cute little folder tabs, on the left side of the page:
  + add popups, with a little exclamation side-menu
  + add a chat menu, so you can message other players in real time
  + add a settings menu, including a button to "exit" to the "Game Entrance" menu
  + add side menu for the rules
* add tooltips (next WIP RN)
* connect PyScript to HTML
* animate the cards

Perhaps, for the UI:

* add a tutorial of some kind? Nah, I would rather just link to a YouTube video of me playing it, and even that is just going too far!

Next up after that, for the script:

* make everything modular
  + add in a rules system, to some degree
* record game actions, oh and add a way to save games
* add an undo function
* maybe list ((valid moves) / (current possible (decisions / actions)))

Projects for another day, maybe?:

* add in smart card-classifying; example: if player has to draw because they can't play, then their hand must have N cards of a certain type
  + yeah, that is kinda complex; maybe a simple check with some simple blacklists (and a report on how long its been since then and like how many cards have been drawn or something) would be good though.
* add macros to autoplay the game?

Oh, this fits nicely in 4 page 😊.

Oh, and I have to upload my code, I guess? I’ll upload the most recent file, since you said it doesn’t need to be functional, and you got the functional game prototye\_0 code in the previous report 😊.