

# Makao - dokumentacja

## Opis:

### Treść zadania:

Proszę zaimplementować grę karcianą Makao w wariantcie dla od 2 do 4 graczy (użytkownik i od 1 do 3 graczy komputerowych) w wersji klasycznej (bez wersji zagranicznych i wersji Pławińskiej).

### Opis Zadania:

Moim zadaniem było stworzenie klasycznej polskiej gry karcianej Makao. Zasady są znane większości ludzi jednak na tyle obszerne, że umieszczanie ich w tym dokumencie zajęłoby za dużo miejsca, jednak można je zobaczyć [tutaj](#).

### Opis Klas:

**Card:** Bazowa karta reprezentująca kartę do gry, posiada wartość jak i kolor. Z ważniejszych metod mamy metodę "can\_put\_card" która decyduje, czy karta ta może zostać położona na inną, zakładając brak kart specjalnych (Takich jak Asów czy Waletów). Do reprezentacji graficznej użyłem kodów ascii na kolory kart jak i biblioteki colorama, tworząc intuicyjny obraz karty dla gracza.

**CardStack:** Stos kart na których gracze kładą karty ze swojej ręki. Najważniejszymi metodami w niej są "is\_valid\_combo", "add\_cards\_on\_top" oraz "trigger\_top\_effect". Metody te mają dużo interakcji ze sobą, pierwsza sprawdza, czy ciąg podanych kart może być położony w tej kolejności, jeżeli jest to prawda, druga metoda kładzie te karty na stos w tej kolejności, podczas położenia karty wywołwana jest ostatnia metoda, która aktywuje wszystkie potencjalne efekty kart specjalnych.

**Player:** Bazowa klasa gracza, nie bierze ona udziału w grze, jednak inne klasy dziedziczą z niej. Gracz posiada rękę z kartami jak i nazwę. Może on dobierać karty i je usuwać. Posiada także atrybuty pozwalające na zabranianie mu zagrywania pewnych kart (Np po efekcie Asa lub Waleta). Gracz może 'przekazać' swój specjalny efekt innemu graczowi, np po przebiciu zagranej dwójki swoją trójką, w rezultacie czego kolejny gracz będzie musiał dobrać 5 kart.

**MainPlayer:** Klasa gracza głównego, którym sterujemy, nie posiada on wielu różnic, jedyną różnicą jest zapisywanie wiadomości specjalnych do pokazywania graczowi podczas jego tur.

**Opponent:** Klasa gracza, przeciwko któremu gramy, jedyne różnice od klasy z której dziedziczymy to możliwość automatycznego decydowania o strategii na podstawie informacji z gry (AI nie jest bardzo zaawansowane, ale pozwala na rozegranie całej partii ze średnią trudnością)

**Game:** Główna klasa sterująca całą logiką w makao, łączy w sobie ona wszystkie klasy powyższe i odpowiada za interakcje między nimi. Najważniejszą metodą jest metoda "handle\_turn" na podstawie której decydujemy o kolejnych typach tur do wywołania na podstawie specjalnych przypadków (np. zagrany Król Pik). Jest w stanie poprosić gracza o input oraz wykryć, kiedy ci chcą powiedzieć makao, pozwala także graczom na powiedzenie stop makao, w rezultacie czego gracz będzie musiał dobrać 5 kart.

## Obsługa:

Aby rozpocząć rozgrywkę, należy uruchomić plik makao.py, powinno się także upewnić, że istnieją pliki:

- card\_stack.py
- card.py
- deck.py
- game.py
- main\_player.py
- opponent.py
- player.py

Potrzebna jest także biblioteka colorama do poprawnego wyświetlania kolorów kart. Można ją zainstalować poprzez "pip3 install colorama"

## Refleksje:

Pisząc ten projekt bawiłem się świetnie, jest to mój największy projekt w pythonie skalowo jak i pod względem linii. Na początku pisanie go było względnie proste, sama logika w mako nie jest zaawansowana, jednak prawdziwą trudnością okazała się implementacja wielu interakcji jakie mogą zajść pod rząd. Zauważyłem, że portowanie fizycznych gier na komputer jest bardzo zaawansowanym procesem, gdyż rzeczy dla nas oczywiste niekoniecznie mogą być prosto zaimplementowane. Największą przeszkodą była interakcja Króla Pik, gdyż proszenie gracza który już rozegrał swoją kolejkę o input nie jest takie oczywiste.

Nie zdecydowałem się na rozwiązanie graficzne z kilku powodów. Po pierwsze interfejs graficzny nie dodawał aż tak dużo do rozrywki, jak mogłoby się wydawać, po drugie nakład czasowy byłby ogromny i niewymierny do reszty pracy, gdyż biblioteka pygame jest bardzo obszerna.

Niestety nie byłem w stanie wprowadzić zaawansowanego AI, jednak z doświadczenia wiem, że wiele osób grających w makao nie gra o wiele lepiej :P

Projekt ten sprawił, że zdałem sobie sprawę z wad w moim podejściu do programowania, okazuje się, że programowanie "od ogółu do szczegółu" nie zawsze skończy się dobrze, kilka razy musiałem przepisywać funkcję, bo wcześniej założyłem, że nie będzie ona potrzebować nic więcej. Zwiększyło to też mój respekt do osób robiących to na co dzień. "Walka" z potencjalnie dziwnymi interakcjami z użytkownikiem sama w sobie jest męcząca, a im większy projekt, tym więcej takich interakcji.

Uważam, że mój projekt spełnił zadane wymagania i prezentowana wersja makao jest autentyczna, dlatego zasługuje na wysoką ocenę ;)