

Autor: Maciej Cieřlik

Cel i opis projektu

Celem projektu jest napisanie gry makao w języku Python. Program umożliwia użytkownikowi grę w makao z przeciwnikami komputerowym. Kod napisany jest zgodnie z zasadami programowania obiektowego. Problem został podzielony na mniejsze problemy, które są rozwiązywane przy pomocy stworzonych klas i funkcji.

Założenia przyjęte przy tworzeniu gry:

Użytkownik ma możliwość wyboru od jednego do trzech przeciwników. Wszyscy przeciwnicy grają na poziomie łatwym lub trudnym w zależności od wyboru użytkownika. Program wykorzystuje interfejs graficzny, reagujący na zachowania myszy.

W grze nie występują karty typu joker. Dodatkowo w jednej turze na króle kier i pik nie można położyć następnego króla. Przeciwnik na łatwym poziomie trudności zagrywa pierwszą możliwą do zagrania kartę od lewej. Na poziomie trudnym przeciwnik spośród kart możliwych do zagrania stara się zagrywać karty w kolejności:

Karty od 5 do 10 → as → 4 → walet → 2 → 3 → król → dama

W sytuacji gdy zawodnik następny po bocie ma 2 karty lub mniej to bot stara się go zaatakować i zmienia kolejność zagrywania kart na:

3 → 2 → walet → 4 → król → as → karty od 5 do 10 → dama

Dodatkowo gdy bot zagra waleta lub asa, to żąda takiego numeru lub koloru, którego pojawia się najczęściej w jego kartach na ręce. Boty na obu poziomach trudności zawsze mówią makao gdy zostanie im na ręce jedna karta.

Użyte biblioteki:

pygame, random, csv, enum, time, abc

Struktura i dekompozycja programu. Opis najważniejszych klas i funkcji

Program składa się z klas i funkcji, które są rozdzielone na dziewięć modułów o nazwach odpowiadających powierzonym rolom.

Moduł buttons.py :

Moduł zawiera wiele funkcji odpowiedzialnych za rysowanie przycisków i napisów na ekranie: przyciski komend, przyciski kart gracza, przyciski kart przeciwników, przycisk karty na środku stołu, napis określający liczbę kart jaka ma przeciwnik, dowolny napis.

- Klasa CardButton – Definiuje obiekty będące przyciskami w postaci kart do gry. Przyciski te są wyświetlane w oknie gry i umożliwiają użytkownikowi wybór karty.
- Klasa Button – Definiuje obiekty będące przyciskami z wyświetlanymi na ich środku napisami. Przyciski te umożliwiają sterowanie grą oraz nawigację między oknami gry.

Moduł cards_piles.py :

- Klasa Pile – Definiuje obiekty będące stosami kart, które reprezentowane są jako listy kart. Klasa ta jest klasą nadrzędną dla bardziej wyspecjalizowanych klas.
- Klasa CardsPile – Definiuje obiekt będący stosem kart, z którego gracz może pobierać karty. Jest podklasą klasy Pile. Klasa umożliwia tasowanie kart na stosie oraz dobieranie kart przez graczy.
- Klasa BinCardsPile – Definiuje obiekt będący stosem kart odrzuconych. Jest podklasą klasy Pile. Klasa pozwala dodawać użyte karty na stos kart odrzuconych.

Moduł game.py :

- Klasa Game – Główna klasa programu, sterująca działaniem pozostałych klas i funkcji. Klasa kontroluje bloki wykonujące określone zadania i ustawia je w odpowiedniej kolejności. Odpowiada między innymi za sterowanie turą gracza i obsługuje podejmowanie decyzji.

Moduł main.py :

- Funkcja main – Wyświetla po kolei ekrany gry i wywołuje główną jej pętlę. Łączy ze sobą wszystkie elementy programu.

Moduł play_service.py :

- Klasa PlayService – Klasa techniczna, sterująca parametrami gry. Odpowiada za zmianę wartości parametrów, bazując na tym jaką akcję w swojej turze wykonał zawodnik.

Moduł players.py :

- Klasa Participant – Klasa abstrakcyjna definiująca obiekt, którym jest zawodnik. Obsługuje parametry sterujące zawodnikiem. Klasa ta jest klasą nadrzędną dla bardziej wyspecjalizowanych klas.
- Klasa Player – Definiuje obiekt będący fizycznym graczem. Jest podklasą klasy Pile. Zawiera w sobie logikę, która umożliwia pobranie karty bądź komendy wybranej przez użytkownika i przekazanie jej dalej w zmienionej postaci.
- Klasa Opponent – Definiuje obiekt będący przeciwnikiem komputerowym. Jest podklasą klasy Pile. Zawiera w sobie bloki, które odpowiadają za wybór karty oraz podejmowanie finalnej decyzji przez bota na łatwym i trudnym poziomie trudności.

Moduł screens.py :

Moduł zawiera wiele funkcji odpowiedzialnych za wyświetlenie dodatkowych ekranów: ekran początkowy, ekran wyboru liczby przeciwników, ekran wyboru trudności przeciwników, ekrany żądań numeru i koloru, ekran końcowy oraz ekran końca kart

- Funkcja game_loop_screen – Funkcja wyświetla pole gry zgodne z aktualnym stanem i zwraca wybraną przez gracza kartę lub komendę

Instrukcja użytkownika:

Sterowanie grą odbywa się wyłącznie przy pomocy myszy. Po uruchomieniu programu wyświetlany jest ekran startowy. Kliknięcie PLAY wyświetla ekran wyboru ilości przeciwników, natomiast EXIT zamyka program. Po kliknięciu przycisku z wybraną liczbą przeciwników, wyświetlany jest ekran umożliwiający wybór poziomu trudności gry. Wciśnięcie przycisku EASY ustawi łatwy poziom przeciwników, a HARD trudny. Po wybraniu poziomu trudności gra rozpoczyna się i wyświetlony zostaje jej początkowy stan. Aby zagrać kartę należy kliknąć na jej grafikę. Kliknięcie w kartę leżącą na środku stołu lub w kartę przeciwnika nie wywołuje żadnej akcji. Jeśli karta jest możliwa do zagrania to zostanie usunięta z ręki i wyświetli się na środku stołu. Jeśli wybrana karta lub komenda jest nieprawidłowa to nic się nie dzieje. Aby zakończyć turę należy kliknąć w przycisk END TURN. W sytuacji gdy gracz może zagrać kilka kart o tym samym numerze, nie można zakończyć tury przed zagranie przynajmniej jednej z dozwolonych kart. Gdy po wykonaniu akcji gracz ma jedną kartę to przed kliknięciem przycisku END TURN powinien kliknąć przycisk MAKAO, w przeciwnym razie dostanie on 5 karnych kart. Dobór kart realizowany jest automatycznie, gdy gra wykryje, że gracz nie ma możliwości zagrania żadnej karty. W

momencie gdy któryś z zawodnik zagrywa waleta lub asa, na ekranie wyświetlany jest komunikat jaki numer bądź kolor jest żądany. Dodatkowo w przypadku waleta wyświetlana jest liczba tur, przez które będzie trwać żądanie. Gdy zagrana zostanie karta atakująca, wyświetlany jest komunikat wskazujący ilość kart, którą dobierze gracz, lub ile tur będzie stał jeśli nie będzie w stanie obronić się przed atakiem. Gracz ma również możliwość kliknięcia przycisku stop makao, jednak przycisk ten nic nie robi. Spowodowane jest to tym, że przeciwnicy komputerowi są zaprogramowani tak, aby za każdym razem gdy mają 1 kartę, mówić makao, więc sytuacja w której gracz mógłby powiedzieć stop makao nigdy nie wystąpi. Gra kończy się gdy gracz pozbędzie się wszystkich kart lub gracz zostanie jedynym zawodnikiem mającym jeszcze karty, lub gdy skończą się karty. Po skończeniu gry wyświetlana lista zawodników, wraz z zajętymi przez nich miejscami. Klikając przycisk MENU wyświetlany jest ekran początkowy, z poziomu którego gracz może rozpocząć nową grę lub wyjść z programu

Pliki programowe

Pliki: buttons.py, cards_piles.py, cards.py, commands.py, game.py, main.py, play_service.py, players.py, screens.py zawierają kod programu i powinny zostać umieszczone w jednym katalogu o nazwie makao. W tym samym katalogu powinien zostać umieszczony plik configuration.txt. Jest to plik konfiguracyjny, którego pierwsza linia zawiera ścieżkę względną do pliku cards.txt, a linie następne opisują biblioteki wraz z wersjami, które należy pobrać, aby program działał prawidłowo. W katalogu makao należy utworzyć katalog opisy i umieścić w nim plik cards.txt, w którym znajdują się dane potrzebne do stworzenia kart. W katalogu makao należy również stworzyć katalog images, w którym należy umieścić pliki: background.png - który zawiera grafikę tła gry, red_back.png – który zawiera grafikę rewersu karty oraz red_back_rotated.png – który zawiera grafikę odwróconego rewersu karty. W katalogu images należy utworzyć katalog cards, w którym powinny znaleźć się pliki zawierające grafikę każdej z kart. Pliki te powinny być nazwane zgodnie z regułą: Pierwszy znak to numer karty – cyfra bądź duża litera, drugi znak to duża litera zgodna z kolorem karty: H – heart, D – diamond, S – spade, C – club. Po dwóch znakach powinno znaleźć się rozszerzenie .png.

Część refleksyjna

Udało mi się napisać całkiem dobrze działającą grę makao. W grę można grać w sposób komfortowy, jednak momentami(choć bardzo rzadko) widoczny jest problem ze stabilnością. Niestety pomimo prób nie udało mi się zaimplementować działającej opcji zgłaszania stop makao oraz możliwości wstrzymania się od ruchu(obecnie trzeba zagrać wszystkie dostępne karty). Powodem tego jest to, że przy próbie zaimplementowania powyższych rozwiązań gra przestawała działać prawidłowo i pojawiało się dużo błędów w działaniu pozostałych opcji, na których poprawę nie wystarczyło mi czasu. Szczęśliwie udało się wprowadzić dobrze działając interfejs graficzny, którego z początku bardzo się obawiałem oraz działającą logikę do inteligentnego bota. Podsumowując jestem bardzo zadowolony z efektów mojej pracy, przewyższyły one moje oczekiwania, ponieważ z początku obawiałem się, że to zadanie mnie przerośnie i sobie z nim nie poradzę. Spędziłem nad projektem dużo czasu, ale uważam, że było warto, ponieważ podczas jego pisania nauczyłem się wielu umiejętności np.: tworzenie interface'u graficznego przy użyciu biblioteki pygame, korzystania z metod abstrakcyjnych oraz innych. Najważniejszą rzeczą jest jednak to, że zacząłem wierzyć w swoje możliwości przy pisaniu kodu co pozwoliło mi pokonać wyzwanie, którego się obawiałem.