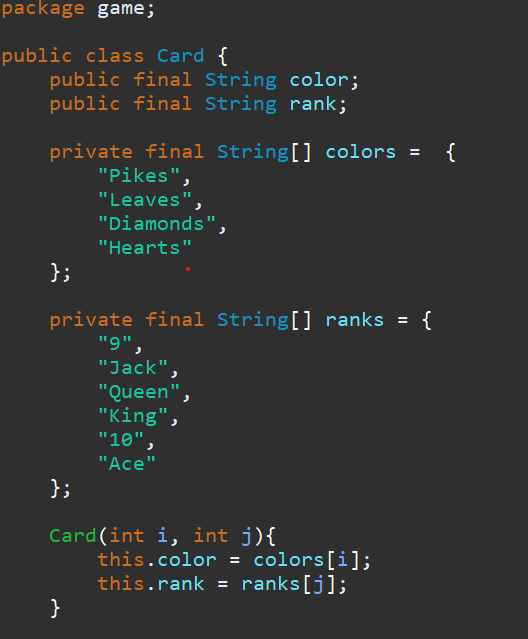
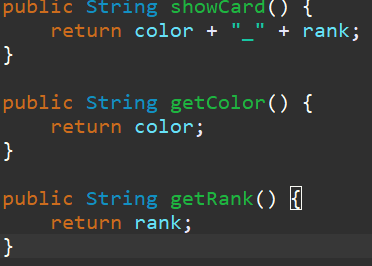
|  |  |  |
| --- | --- | --- |
| Znalezione obrazy dla zapytania pśk logoProjekt Programowanie obiektowe w języku Java  Politechnika Świętokrzyska | | |
| Temat:  Gra „Tysiąc” oparta na architekturze klient-serwer. | | |
| Krzysztof Kiełek  Patryk Bętkowski | Termin oddania projektu:  01.07.2020 | |
| Termin wykonania projektu:  30.06.2020 | |
| Grupa: 2ID11A  Rok Akademicki: 2019/2020 | Prowadzący:  Adam Krechowicz |



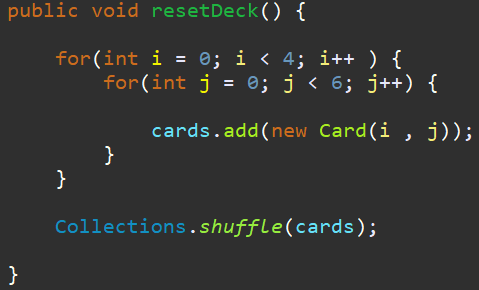
Klasa Card odpowiedzialna za karty w grze, osobno zostały stworzone tablice jednowymiarowe przechowujące odpowiednio kolory oraz figury. W konstruktorze, dwóm zmiennym określającym daną kartę przypisywane są wybrane atrybuty.



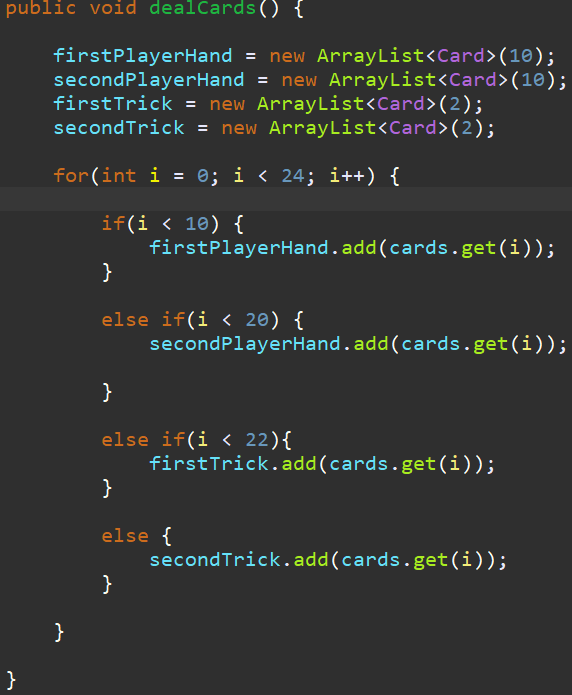
Klasa ta posiada trzy metody, pierwsza zwraca wybrana kartę, kolejna kolor a ostatnia figurę. Zwracane wartości są w postaci tekstu.



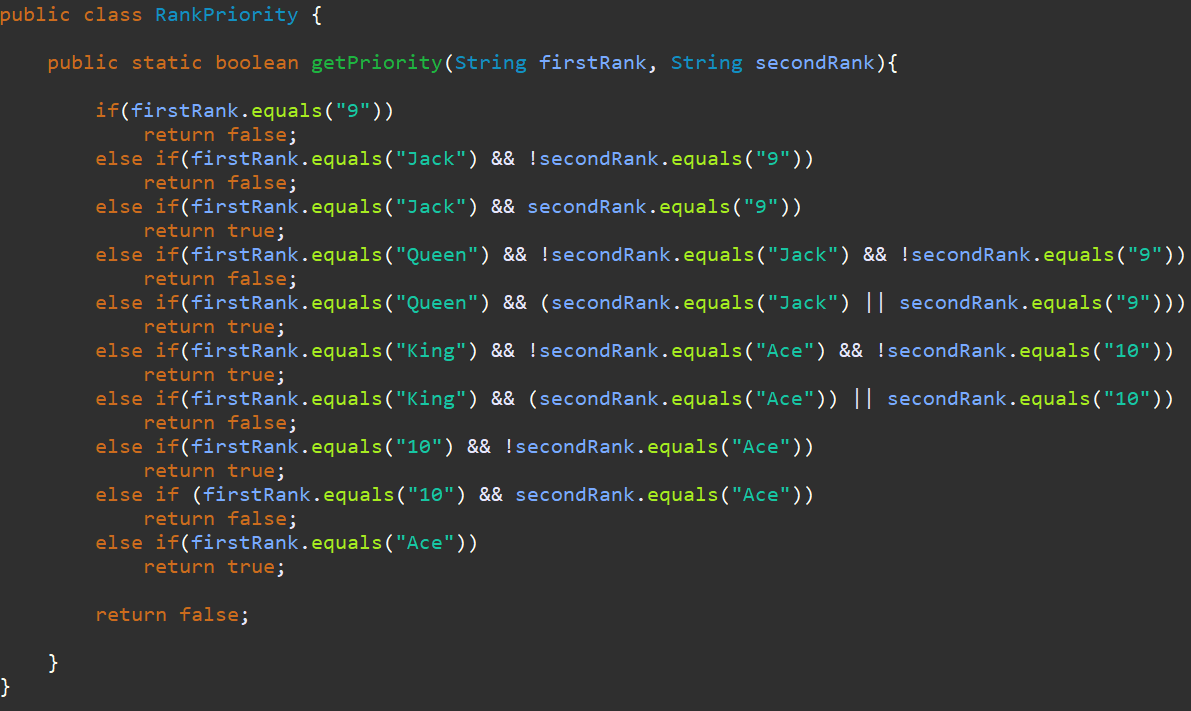
Klasa deck odpowiada za układ kart na stole. W konstruktorze tworzona jest pusta talia, jeszcze bez kolorów czy figur.



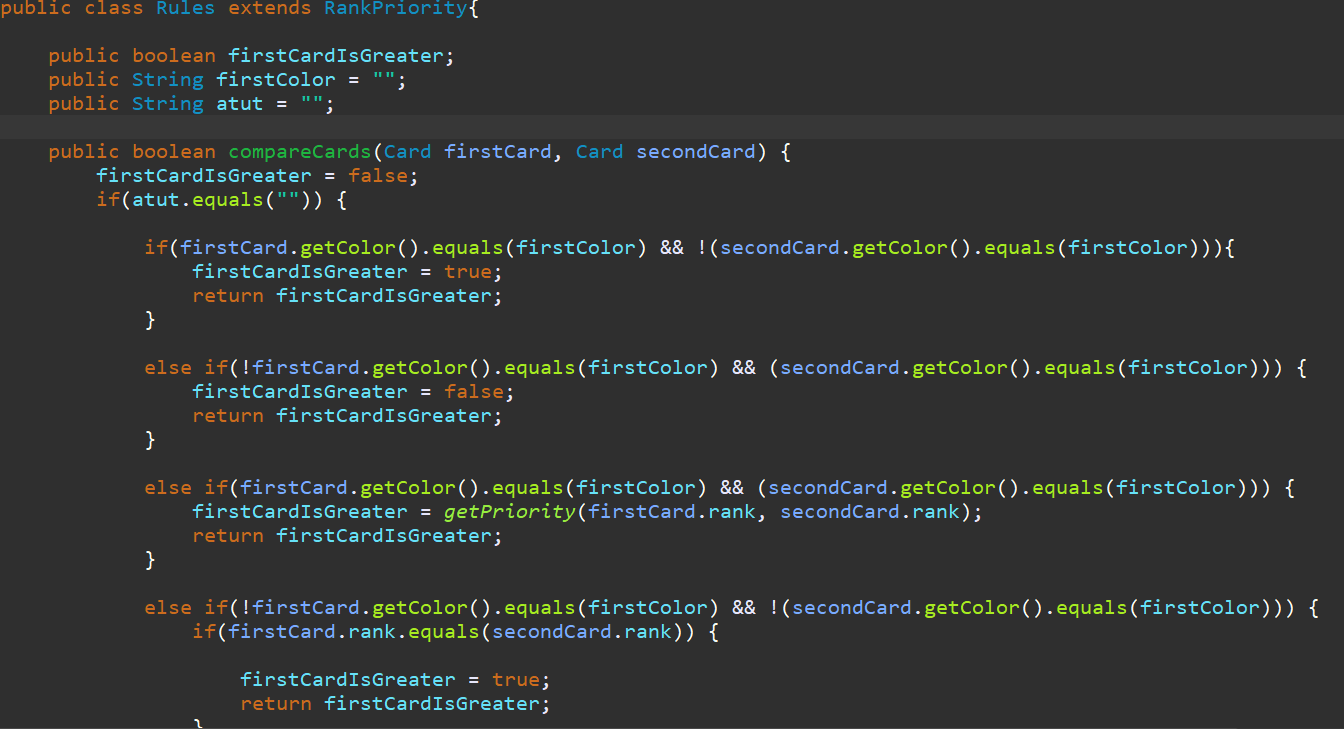
Metoda resetDeck() odpowiada za stworzenie właściwych kart w tali a następnie przetasowanie tali.



W tej metodzie karty rozdawanie są pomiędzy dwóch graczy oraz dwa musy, odpowiednio po 10 i 2 karty.



Ta klasa odpowiada za priorytet figur w przypadku gdy obaj gracze maja te same kolory, posiada tylko jedna metodę która to sprawdza.



Klasa Rules dziedziczy po RankPriority, posiada metodę compareCards która zwraca prawdę jeśli pierwsza z dwóch podanych jej kart jest mocniejsza. Dwie zmienne firstColor oraz atut przechowują kolor który obecnie jest najmocniejszy przy czym atut ma tutaj większe znaczenie.

Gra oparta jest na architekturze klinet-server, jeden z graczy jest klientem a drugi serwerem.

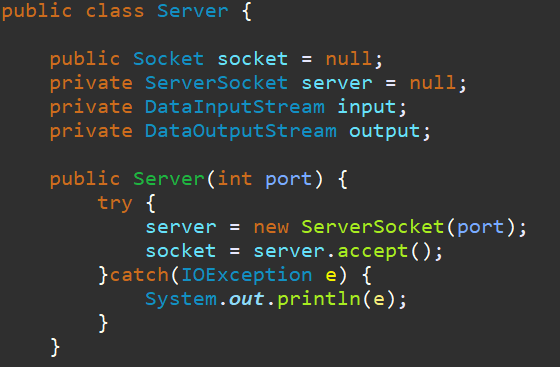


Klasa odpowiedzialna za klienta.





Metody klasy Client odpowiedzialne za zakończenie połącznie, wysłanie liczb do serwera, odebranie liczb z serwera, oraz wysłanie i odebrania ciągu znaków.



Klasa serwera jest bardzo podobna, różni się głownie w konstruktorze. Metody są analogiczne jak w kliencie aby możliwa była dwustronna komunikacja.

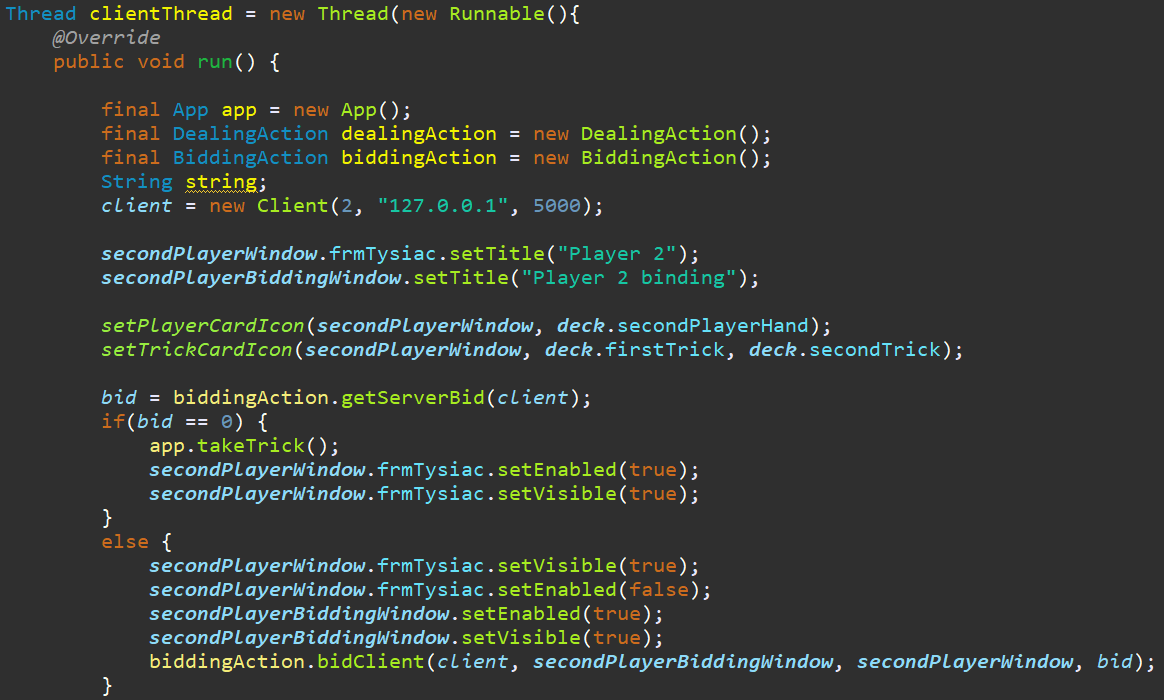


Klasa App w której zebrane są, opisane wcześniej klasy odpowiedzialne za karty, zasady, komunikacje oraz wyświetlanie wszystkich komponentów na ekran. Mamy również dwie listy w których przechowywane są karty jakie dany gracz wygrał podczas rundy w celu wyliczenia punktów jakie zdobył.

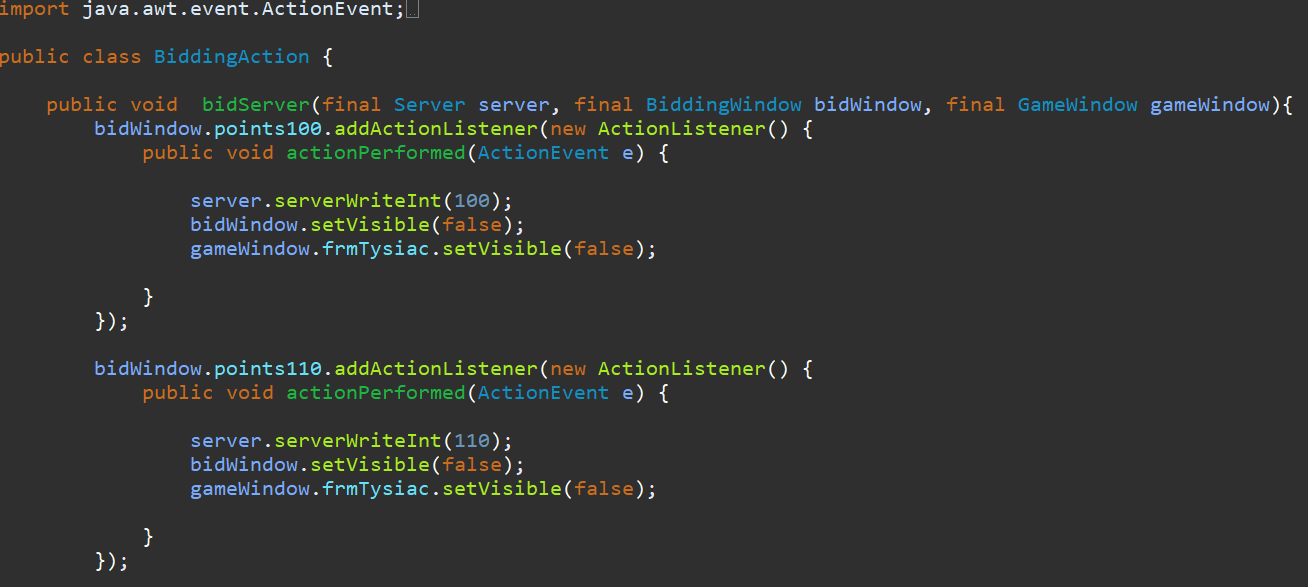
W klasie App tworzone są dwa wątki dla graczy.



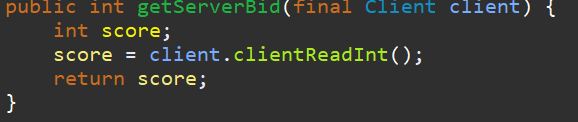
Wątek serwera którym jest jeden z graczy.



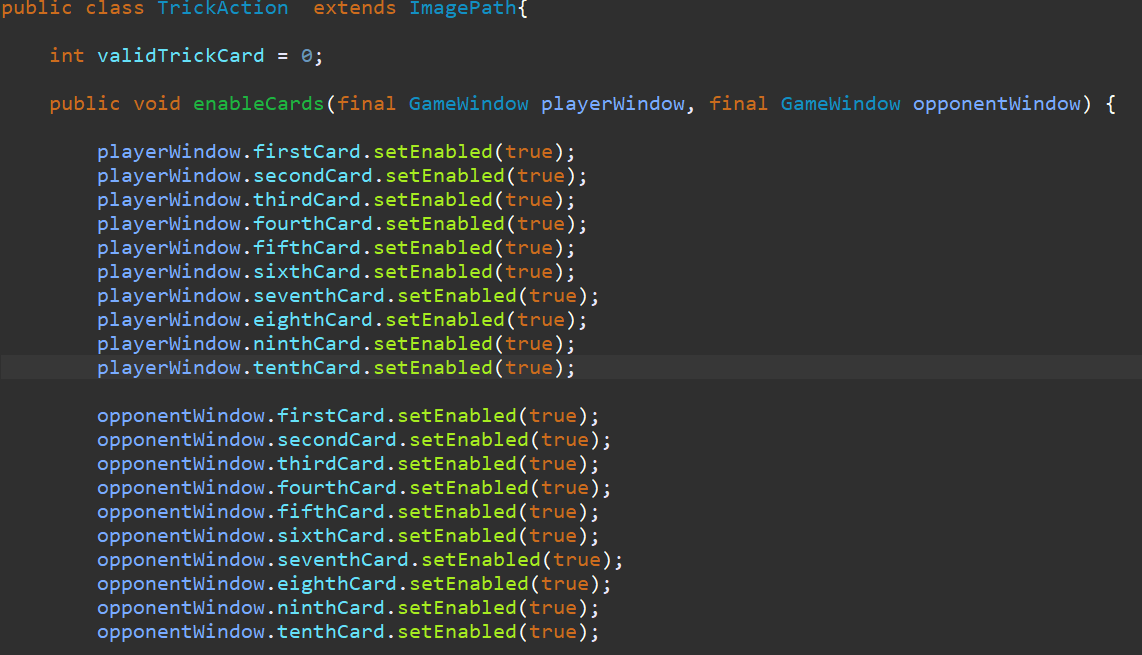
Wątek klienta którym jest drugi z graczy.



Klasa ta jest odpowiedzialna za licytacje na początku gry. W metodzie bidServer gracz będący serwerem deklaruje np. 100 punktow i wysyła tą informacje drugiemu graczowi.



Tą metody wywołuje klient aby odebrać od serwera informacje o liczbie punktów.

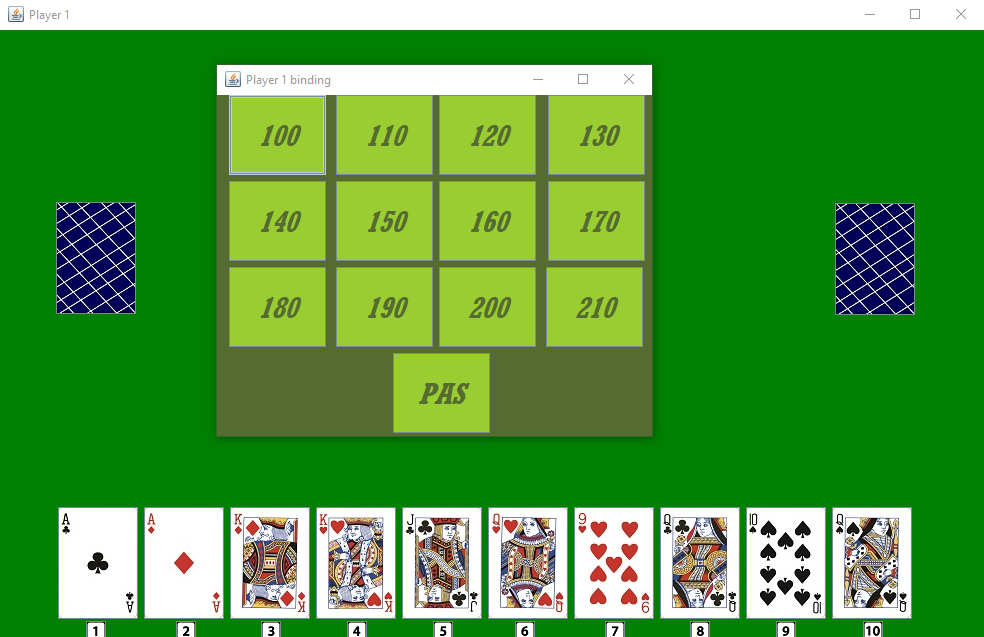


Klasa TrickAction odpowiada za zebranie musiku przez gracz który wygrał licytacje.

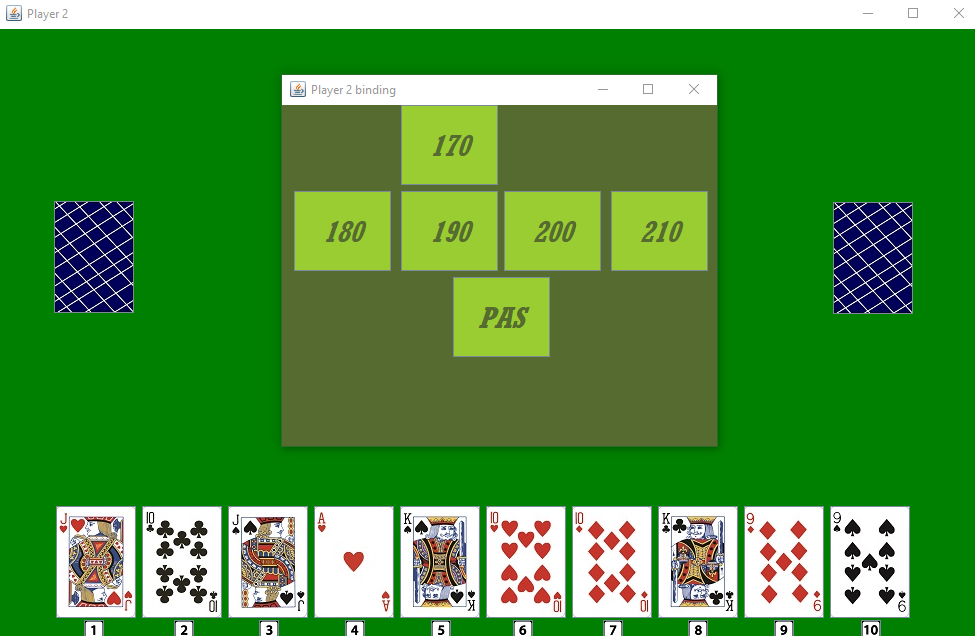


Jest to metoda klasy TrickAction odpowiedzialna za zebranie jednego z dwóch z musików do swojej tali.

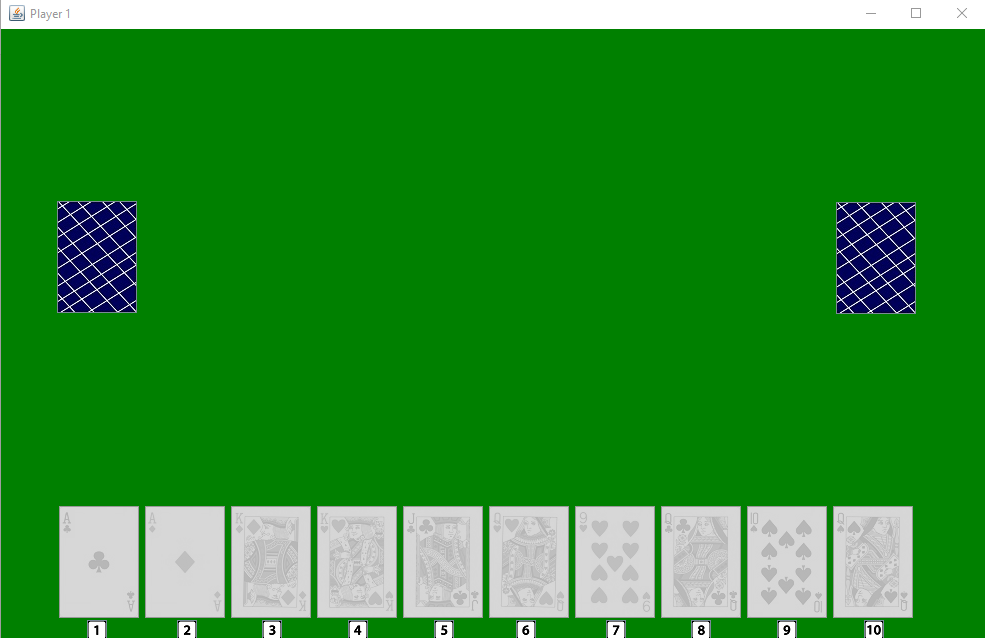
**Prezentacja działania:**

****

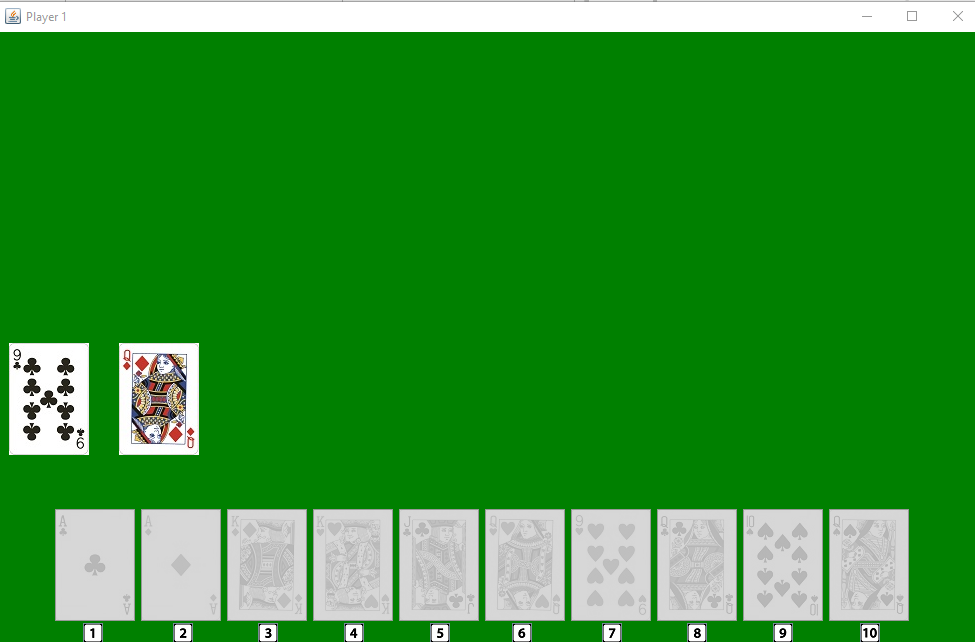
Okno licytacji wyświetla się przed oknem stołu i tylko ono jest obecnie zdolne do interakcji. Gdy klikniemy w wybrana liczbę, okna pierwszego gracza znikną, a pojawią się okna drugiego gracza i również możliwa będzie tylko interakcja z oknem licytacji.

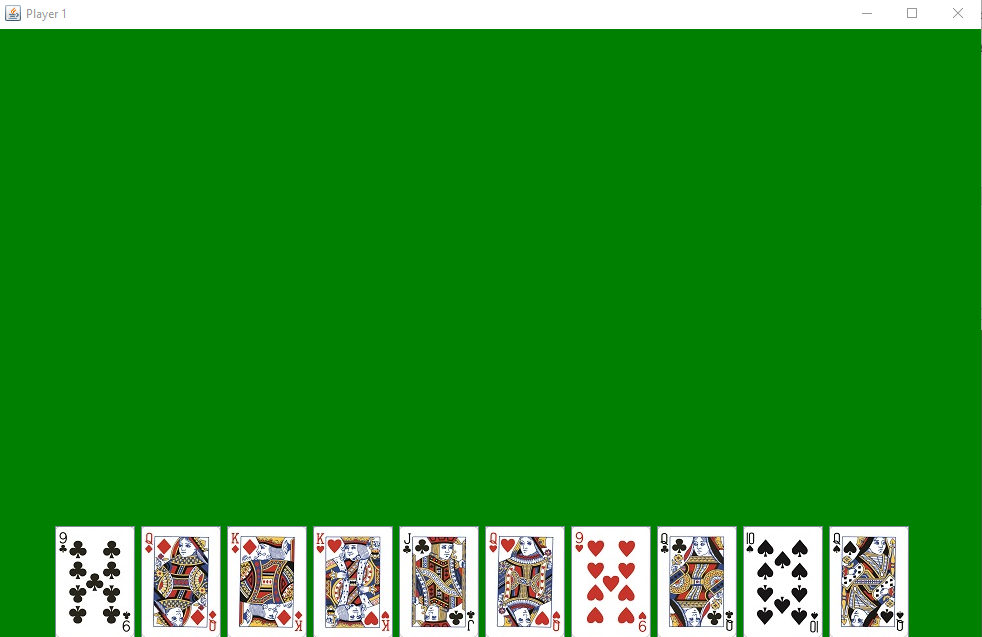


Drugi gracz nie może licytować mniej niż gracz pierwszy.

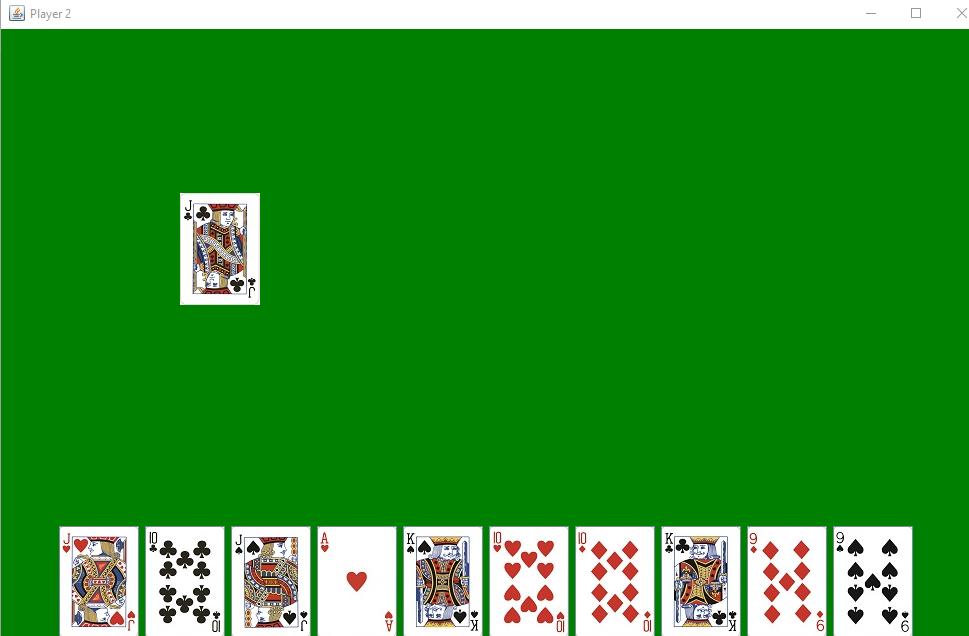


Gracz który wygrał licytacje zbiera musik i rozpoczyna rozgrywkę.

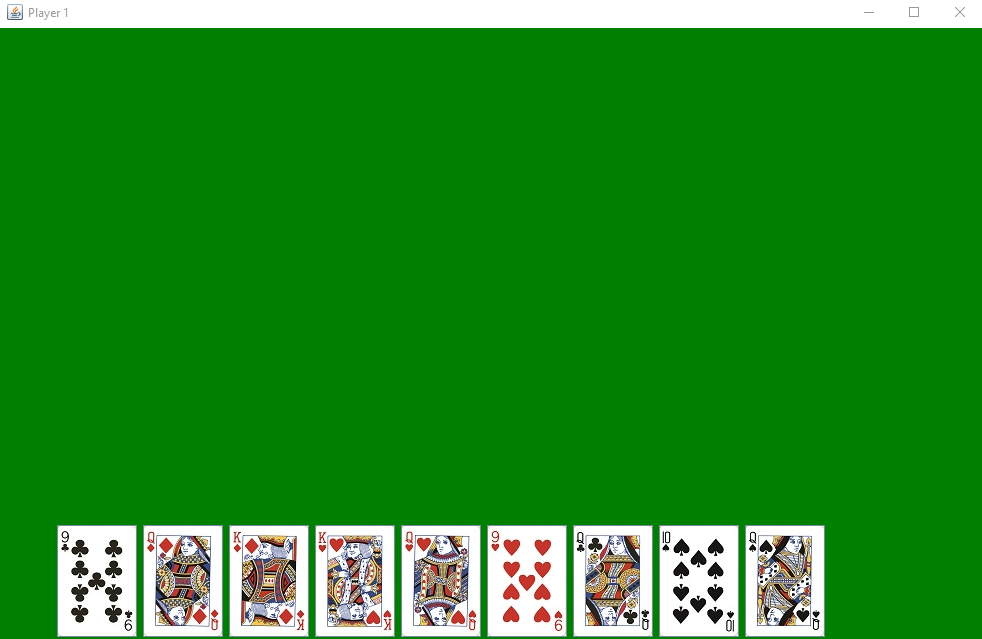




Cyfry pod kartami są po to aby wskazać które karty chcemy oddać.



Pierwszy gracz daje kartę i pojawia się okno gracza drugiego na którym pokazana jest karta jąka zagrał pierwszy gracz.



Wybrana karta znika z tali gracza pierwszego. Rozgrywka jest kontynuowana do ostatniej karty, a następnie podliczane są punkty graczy z kart które wygrali.

**Wnioski**

Jednym z większych problemów okazała się komunikacja klient-serwer, ponieważ zdarzały się momenty że ta komunikacja zawodziła. Staraliśmy się dzielić wszystko na mniejsze klasy co pozwoliło wyeliminować błędy oraz uprościć kilka rzeczy.