

Laboratorium Architektura Systemów Komputerowych - Sprawozdanie 1	
Mateusz Drzewiecki, 185607 Mikołaj Galant, 188565	Zadanie 1 - uproszczona wersja gry Blackjack

Wprowadzenie

Zadanie ma pokazać przykład zastosowania techniki tworzenia graficznego interfejsu użytkownika, monitora ekranowego, klawiatury i myszki. Zdecydowano się na implementację zmodyfikowanej, nie hazardowej wersji gry karcianej Blackjack.

Zasady gry

Gra polega na uzyskaniu jak najbliżej, ale nie więcej niż 21 punktów. Suma punktów jest przydzielane na podstawie posiadanych przez gracza kart.

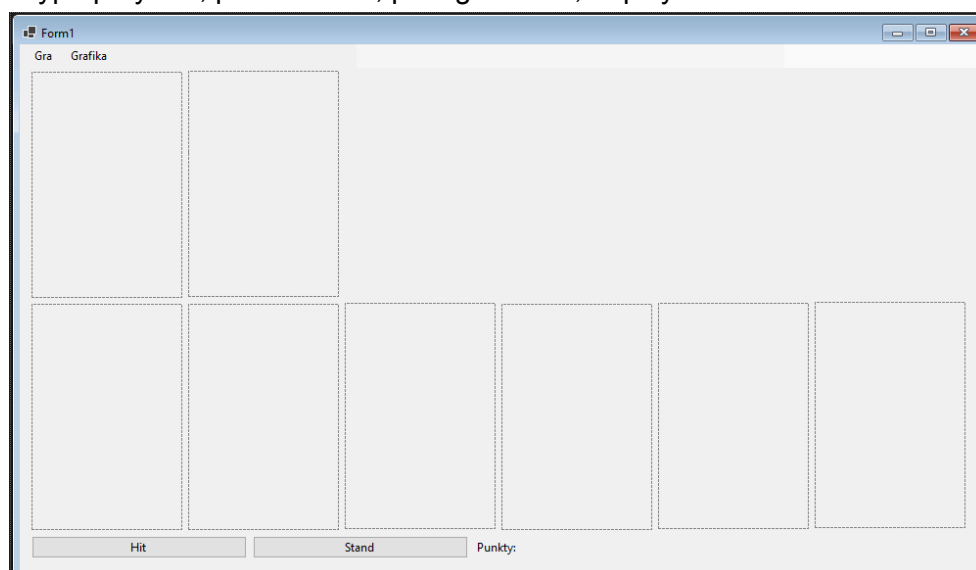
Zasady punktacji:

- karty od 2 do 10 mają wartość równą numerowi karty,
- walet, dama, król mają wartość równą 10 punktom,
- as ma wartość 1 albo 11 - zależy od tego co jest lepsze dla gracza.

Na początku rozgrywki gracz dostaje 2 karty, a przeciwnik - w tym przypadku zawsze w roli dealera - jedną kartę odkrytą i zakrytą. Gracz może dobrać kartę (Hit), bądź uznać, że już nie będzie dobierał i jest usatysfakcjonowany ze zdobytych punktów (Stand). W naszej odmianie przeciwnik nie może dobierać kart. Po zakończeniu dobierania kart następuje zliczanie punktów i określenie kto jest zwycięzcą.

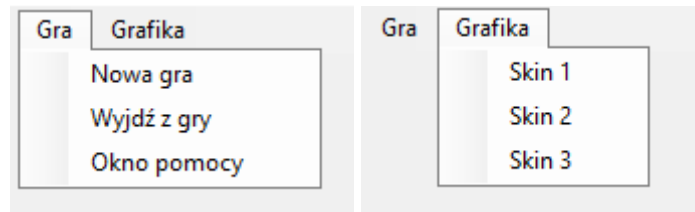
Implementacja gry - interfejs i przebieg rozgrywki.

Grę zrealizowano przy użyciu języka C# i platformy .NET. Posłużono się dostępnymi zasobami typu przycisk, pasek menu, pola graficzne, napisy.

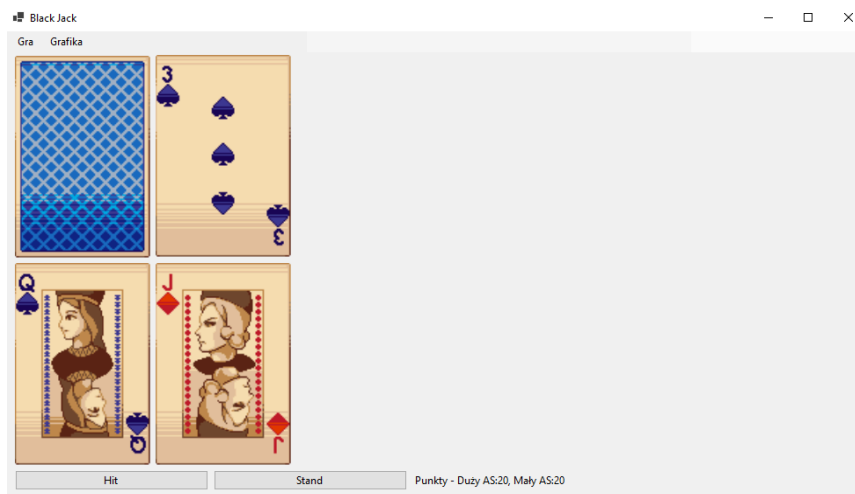


Rys. 1. Elementy interfejsu graficznego widziane z poziomu Visual Studio przed kompilacją programu.

Każdą grę należy rozpocząć od wybrania z paska menu "Gra" opcji "Nowa Gra". Powoduje to przetasowanie kart i rozdanie dwóch dla każdego z graczy. Powoduje to również aktywację przycisków "Hit" i "Stand" na dole okna gry. Dodatkowo zostaje aktywowany licznik punktów gracza obok przycisku "Stand"

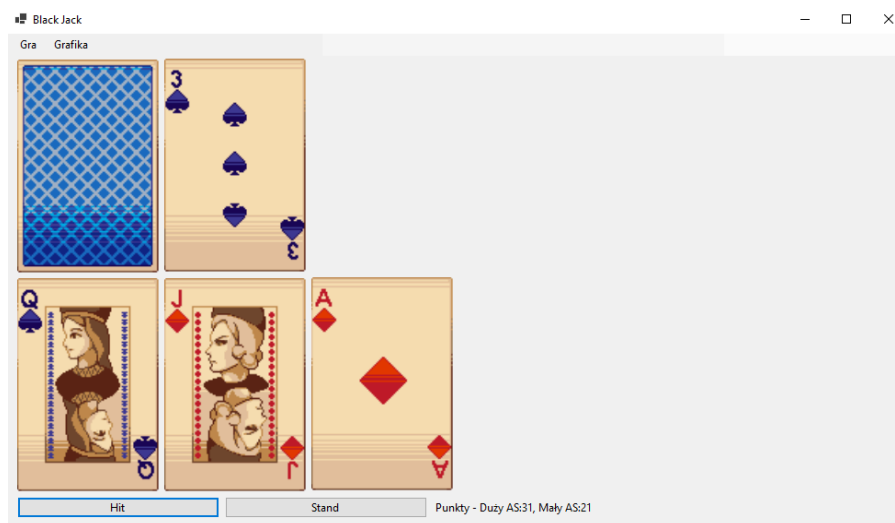


Rys. 2. Menu "Gra" i menu "Grafika".

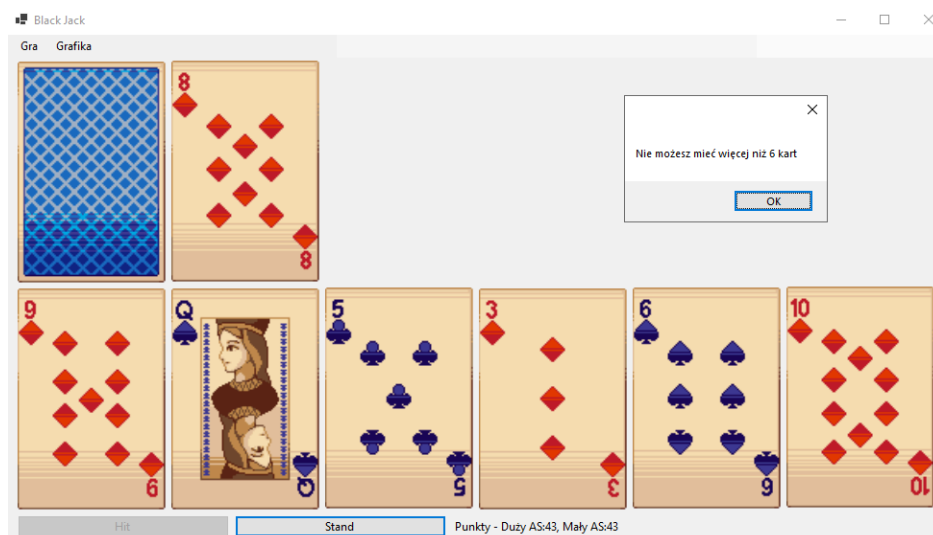


Rys. 3. Pierwsze rozdanie kart.

Poprzez wciśnięcie przycisku "Hit" gracz dobiera kolejną kartę z talii. Efektem tego jest pojawienie się kolejnej karty i aktualizacja licznika punktów. Gracz może wciskać przycisk do czasu posiadania 6 kart w ręce. Kolejne wciśnięcie przycisku powoduje pojawienie się komunikatu o posiadaniu maksymalnej liczby kart i dezaktywację przycisku "Hit".

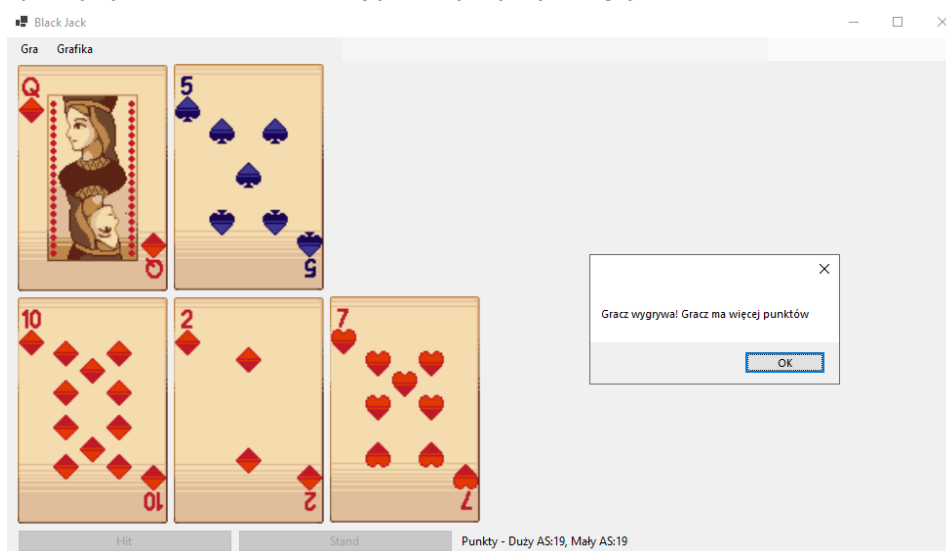


Rys. 4. Efekt wciśnięcia przycisku "Hit".



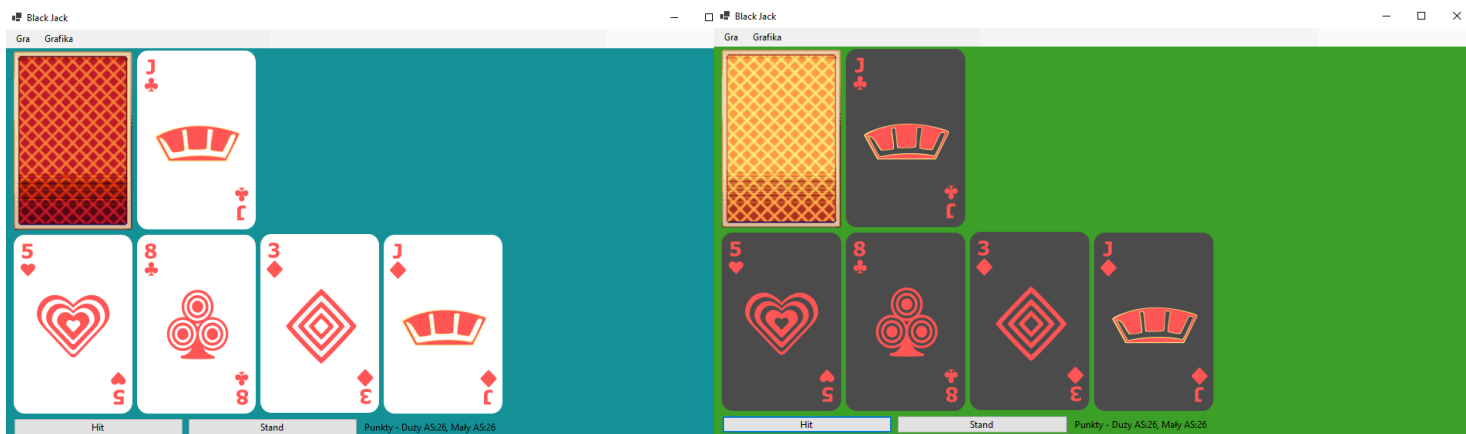
Rys. 5. Wciśnięcie przycisku "Hit" przy posiadaniu 6 kart w ręce.

Wciskając przycisk "Stand" następuje odkrycie kart przeciwnika, zliczenie jego punktów. Pojawia się przy tym okno z informację o zwycięzcy rozgrywki.



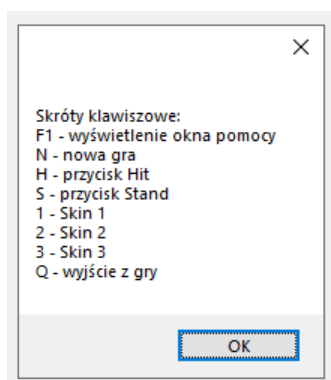
Rys. 6. Efekt wciśnięcia przycisku "Stand".

Zaimplementowano obsługę 3 skórek, które można podmienić w każdym momencie trwania rozgrywki - po rozpoczęciu gry - i do momentu jej zakończenia - pojawienia się komunikatu o zwycięzcy. Można tego dokonać poprzez wybór skórki z menu "Grafika" - patrz rys. 2.



Rys. 7. Zastosowane alternatywne skórki gry.

Każdą akcję można wykonać również z poziomu klawiatury. Zastosowane sterowanie zostało omówione w oknie pomocy, które można wywołać wybraniem opcji pomoc z menu “Gra” - patrz rys. 2.



Rys. 8. Okno pomocy.

Opis kodu

Rozgrywka zaczyna się od wciśnięcia opcji “Nowa Gra”, co wywołuje następującą funkcję, która przygotowuje interfejs użytkownika oraz tasuje i rozdaje karty graczom, a następnie przeszukuje folder z assetami w celu wyświetlenia odpowiedniej grafiki karty.

```
private void nowaGraToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Stand_Button.Enabled = true;
    this.Hit_Button.Enabled = true;

    this.karty_dealera.Clear();
    this.karty_playera.Clear();
    this.talia.Clear();
    //rest grafiki
    dealer_1.Image = null;
    dealer_2.Image = null;

    player_1.Image = null;
    player_2.Image = null;
    player_3.Image = null;
    player_4.Image = null;
    player_5.Image = null;
    player_6.Image = null;

    Shuffle(); //tasowanie kart
    AddCardToPlayer(2, karty_playera); //dodanie kart do reki gracza
    AddCardToPlayer(2, karty_dealera); //dodane kart do reki dealera
    LoadCardsOnScreen(this.curretSkin, 2, 2); //ładowanie kart
    WriteScoreToLabel(this.karty_playera); //aktualizacja wyświetlania punktów
}
```

Rys. 9. Funkcja rozpoczęcia rozgrywki.

Wciśnięcie przycisku Hit wywołuje następującą funkcję, która sprawdza posiadaną liczbę kart przez gracza. Jeżeli nie przekracza narzuconego limitu, dodaje kartę do ręki oraz szuka odpowiedniej grafiki z folderu z assetami. Następnie odnosi się do pierwszego wolnego pola graficznego gracza i dodaje do niego znalezioną grafikę i aktualizuje licznik punktów..

```
private void Hit_Button_Click(object sender, EventArgs e)
{
    int ilosc_kart = this.karty_playera.Count;
    //liczone od zera
    if (ilosc_kart > 5)
    {
        this.Hit_Button.Enabled = false;
        MessageBox.Show("Nie możesz mieć więcej niż 6 kart");
        return;
    }
    else
    {
        AddCardToPlayer(1, karty_playera);

        int kolor_karty = karty_playera[ilosc_kart].kolor;
        string kolor = "";
        switch (kolor_karty)
        {
            case 0:
                kolor = "karo";
                break;
            case 1:
                kolor = "kier";
                break;
            case 2:
                kolor = "pik";
                break;
            case 3:
                kolor = "trefl";
                break;
        }

        int wartosc_karty = karty_playera[ilosc_kart].wartosc;
        string wartosc = "";
    }
}
```

```
string wartosc = "";
switch (wartosc_karty)
{
    case 1:
        wartosc = "A";
        break;
    case 11:
        wartosc = "J";
        break;
    case 12:
        wartosc = "Q";
        break;
    case 13:
        wartosc = "K";
        break;
    default:
        wartosc = wartosc_karty.ToString();
        break;
}

PictureBox pb = (PictureBox)this.Controls.Find("player_"
+ (ilosc_kart+1), true)[0];
//dostosowanie rozmiaru grafiki do rozmiaru pictureboxa
pb.SizeMode = PictureBoxSizeMode.StretchImage;
pb.Image = Image.FromFile("assets/" + (this.curretSkin + 1)
+ "/" + kolor + "/" + wartosc + ".png");

//aktualizacja score
WriteScoreToLabel(karty_playera);
}
```

Rys. 10. Funkcja przypisana do przycisku "Hit".

Funkcja wywoływana przez przycisk "Stand" jest bardzo podobna do tego opisanej wyżej z tą różnicą, że dodatkowo orzeka kto wygrał daną rozgrywkę.

```
//obliczenie punktów dealera
int suma_A11_dealer = 0; //Asa liczymy jako 11
int suma_A1_dealer = 0; //Asa liczymy jako 1
foreach (Karta karta in karty_dealera)
{
    if (karta.wartosc == 1)
    {
        suma_A11_dealer += 11;
        suma_A1_dealer += 1;
    }
    else if (karta.wartosc > 10)
    {
        suma_A11_dealer += 10;
        suma_A1_dealer += 10;
    }
    else if (karta.wartosc <= 10)
    {
        suma_A11_dealer += karta.wartosc;
        suma_A1_dealer += karta.wartosc;
    }
}
```

```
//obliczenie punktów gracza
int suma_A11_gracza = 0; //Asa liczymy jako 11
int suma_A1_gracza = 0; //Asa liczymy jako 1
foreach (Karta karta in karty_playera)
{
    if (karta.wartosc == 1)
    {
        suma_A11_gracza += 11;
        suma_A1_gracza += 1;
    }
    else if (karta.wartosc > 10)
    {
        suma_A11_gracza += 10;
        suma_A1_gracza += 10;
    }
    else if (karta.wartosc <= 10)
    {
        suma_A11_gracza += karta.wartosc;
        suma_A1_gracza += karta.wartosc;
    }
}
```

```

//sprawdzenie kto wygrał
if (suma_A11_dealer > 21 && suma_A1_dealer > 21)
{
    MessageBox.Show("Gracz wygrywa! Dealer przekroczył 21 punktów");
}
else if (suma_A11_gracza > 21 && suma_A1_gracza > 21)
{
    MessageBox.Show("Dealer wygrywa! Gracz przekroczył 21 punktów");
}
else if (suma_A11_dealer > suma_A11_gracza && suma_A11_dealer <= 21)
{
    MessageBox.Show("Dealer wygrywa! Dealer ma więcej punktów");
}
else if (suma_A1_dealer > suma_A1_gracza && suma_A1_dealer <= 21)
{
    MessageBox.Show("Dealer wygrywa! Dealer ma więcej punktów");
}
else if (suma_A11_dealer == suma_A11_gracza)
{
    MessageBox.Show("Remis! Obaj mają tyle samo punktów");
}
else if (suma_A1_dealer == suma_A1_gracza)
{
    MessageBox.Show("Remis! Obaj mają tyle samo punktów");
}
else
{
    MessageBox.Show("Gracz wygrywa! Gracz ma więcej punktów");
}

```

Rys. 11. Fragmentu funkcji wywoływanej przez wciśnięcie przycisku "Stand" orzekające zwycięzcę rozgrywki.

Komentarz do kodu

Środowisko .NET pozwala przy małym wysiłku mieć kontrolę nad wieloma aspektami i funkcjami prostych aplikacji graficznych. W połączeniu z możliwościami języka C# można w krótkim czasie osiągnąć efekty, którym trzeba by poświęcić więcej czasu w innych środowiskach.

Napisany przez nas kod można by usprawnić o dodanie kilku funkcji, które poprawiłyby jego czytelność - tutaj w szczególności mowa o kilkukrotnym przeszukiwaniu zasobów graficznych i tłumaczeniu elementów klasy Karta na nazwę grafiki jak i obliczaniu aktualnych punktów gracza.

Po sprawdzeniu kodu tak, by zajmował mniej miejsca, ale nie stracił znacząco na czytelności zajmuje on prawie 500 liniiek w pliku Form1.cs. Jest to dość spory narzut kodu w porównaniu do prostoty zaimplementowanej gry.