# Објаснување на апликацијата

За потеклото на Blackjack сè уште се води дискусија, но истражувачите се согласуваат дека Blackjack најверојатно потекнувал од француските казина околу 1700 година. Француските картички биле наречени „ Vingt-et-Un “, што во превод значи „Дваесет и еден“.

Друга теорија е дека Blackjack е пронајдок од Римјаните. Теоријата стои затоа што Римјаните сакале да се коцкаат, но таа не е потврдена. Се верува дека Римјаните ја играле оваа игра со дрвени блокови со различни броеви насликани на нив, наместо со хартиени картички.

Blackjack е најголемиот предизвик во казиното. Целта е да се добијат што повеќе пари од куќата (која е претставена од дилер). Дилерот, за разлика од играчите, не донесува свои одлуки, па сè зависи од трпението, среќата и стратегијата на играчот.

На почетокот на секоја рунда, играчот треба да направи кладба. Откако ќе се поделат карти, облозите не можат да се менуваат до следната рунда. Дилерот ќе ви даде две карти што ја формираат вашата рака и две карти за себе што ја формира неговата рака. Една од картите на дилерот ќе биде скриена на почетокот. За да победите, треба да се осигурате дека комбинираната вредност на вашите карти е поголема од онаа на дилерот.

Можете да додадете повеќе карти, една по една, со избор на “Hit”, вашата цел е да се приближите што е можно поблиску до 21 без да надминувате 21. Ако играчот добие повеќе од 21, губи автоматски. Во случај играчот да е подготвен да играте рунда, ја одбира опцијата “Stand”. Дилерот потоа ќе ја открие својата скриена карта и секогаш мора да влече нови карти (“Hit”) ако има 16 или помалку.Ако збирот на картите му е 17 или повеќе, ќе престане да влече.

Вредноста на картите:

Ace -> 1 или 11 во зависност од другите карти во раката

10, Jack, Queen, King -> 10

Останатите ја имаат својата нормална вредност.

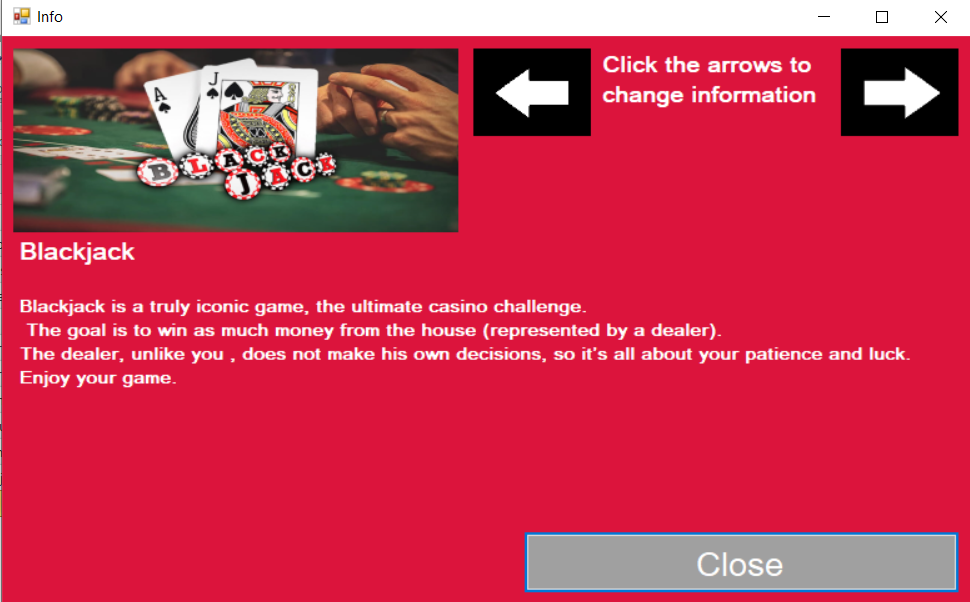
## Да се опише решението на проблемот (кои податоци се чуваат, во какви структури, класи)

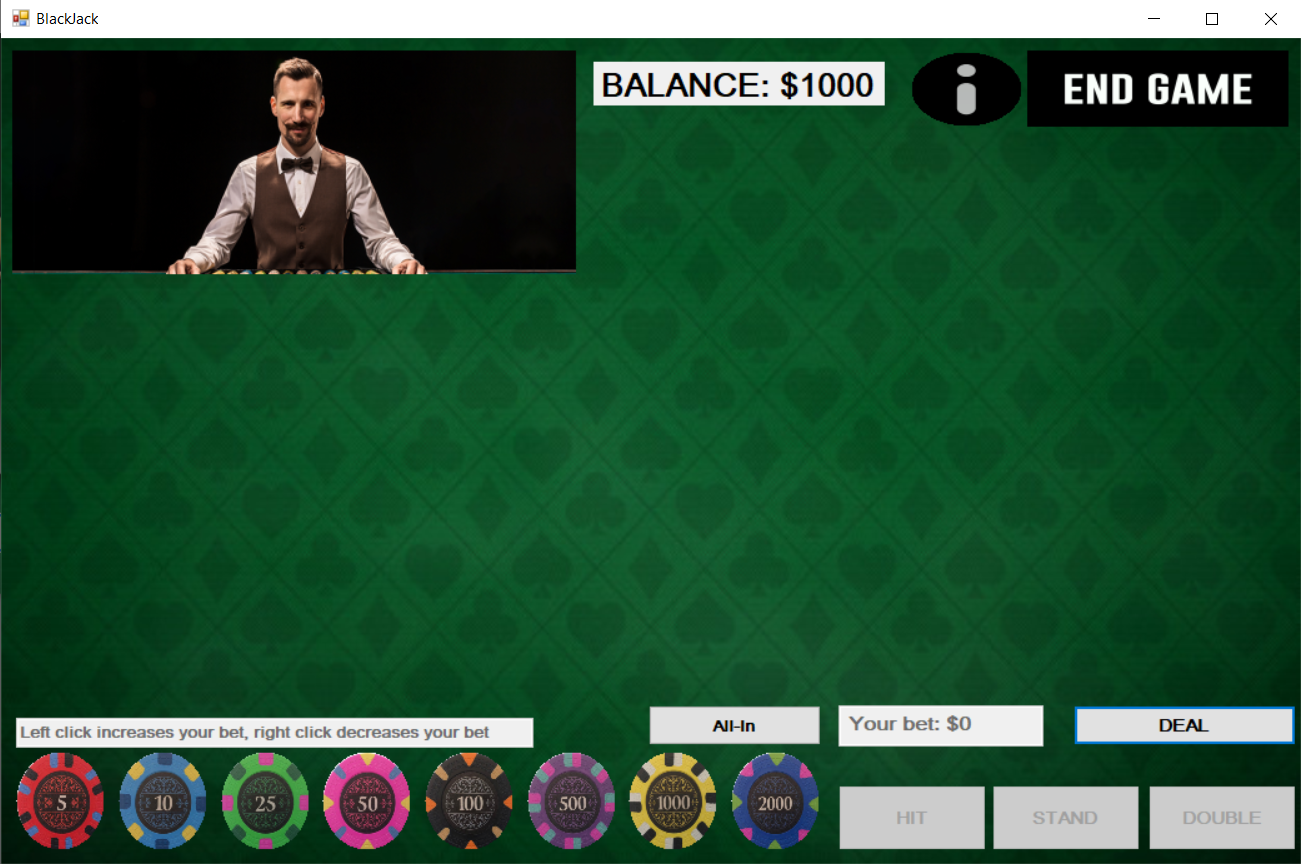
За имплементацијата на логиката употребив 4 форми,3 главни класи: Card,Deck и Hand и две класи кои наследуваат од класата Hand.

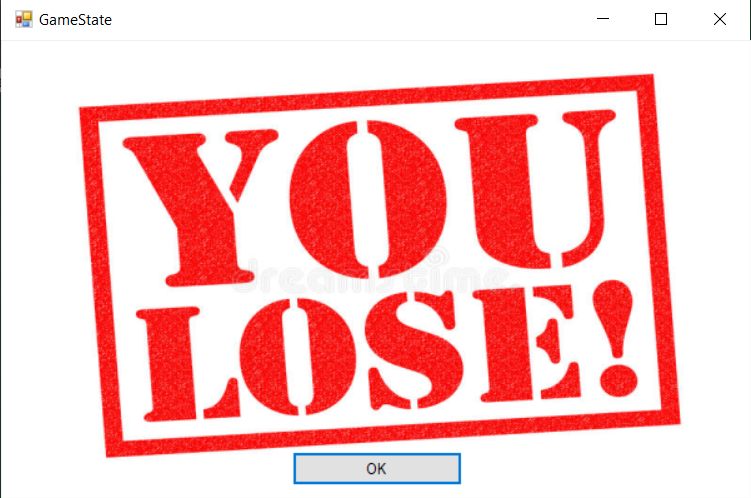
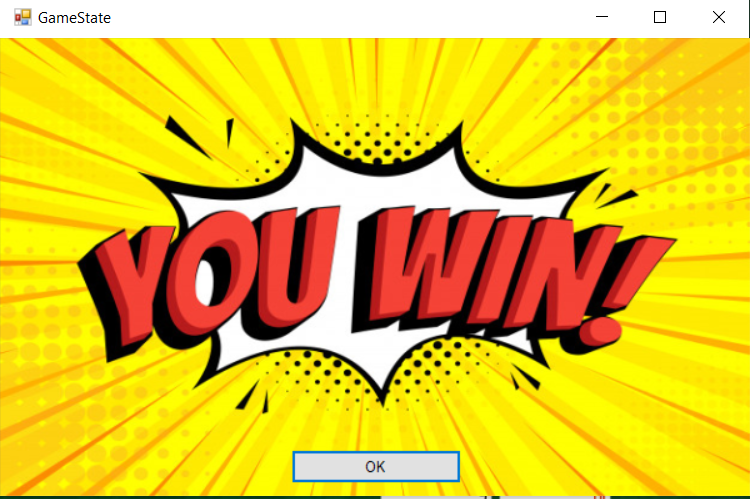
## Форми

Една форма за почетното мени:

Овде има 3 опции: Start која ја започнува играта,Info која дава информации за начинот на играње и Quit Game со која се исклучува играта.



Со помош на стрелките се менува информацијата која е прикажана.

Blackjack формата е главната игра, а четвртата форма е форма која не известува за состојбата на играта. Односно дали сме победиле,изгубле, дали сме имале блекџек или дилерот имал блекџек.

## Класи

Во кодот се започнува од класата Card за неа чувам:

* Enum SUIT (вид)
* Вредност
* Слика
* bool isAce (булеан вредност која ни кажува дали картата е ас или не)

Кодот на најосновната класа Card:

public enum SUIT

{

SPADE,

HEART,

CLUB,

DIAMOND

}

public class Card

{

public SUIT suit{ get; set; }

public int value { get; set; }

public Image image { get; set; }

public bool isAce { get; set; }

public Card(SUIT suit, int value, Image image)

{

this.suit = suit;

this.image = image;

if(value == 11)

{

value = 11;

isAce = true;

}else if (value > 10)

{

this.value = 10;

isAce = false;

}

else

{

this.value = value;

isAce = false;

}

}

}

Наредна класа е класата Deck која се состои од:

* Низа од карти Card[] deck
* Објект од класата Random

Во конструкторот ги креираме картите (Card) и ги додаваме во шпилот (Card[] deck). Има уште две функции, едната е shuffle() функција која го меша шпилот, а другата е generateCard() која ни враќја random карта од шпилот.

public class Deck

{

public Card[] deck { get; set; }

Random random = new Random();

public Deck()

{

deck = new Card[52];

Image image;

int count = 0;

foreach (SUIT s in Enum.GetValues(typeof(SUIT)))

{

for (int value = 2; value < 15; value++)

{

if(s == SUIT.CLUB)

{

image = Image.FromFile(@"Pictures\Cards\" + value + "C.png");

}

else if(s == SUIT.DIAMOND)

{

image = Image.FromFile(@"Pictures\Cards\" + value + "D.png");

}

else if(s == SUIT.HEART)

{

image = Image.FromFile(@"Pictures\Cards\" + value + "H.png");

}

else

{

image = Image.FromFile(@"Pictures\Cards\" + value + "S.png");

}

deck[count] = new Card(s, value, image);

count++;

}

}

}

public void shuffle()

{

Random rnd = new Random();

for (int i = 51; i > 0; i--)

{

int random = rnd.Next(0, 52);

Card tmp = deck[i];

deck[i] = deck[random];

deck[random] = tmp;

}

}

public Card generateCard()

{

return deck[random.Next(0, 52)];

}

}

Последната класа на оваа имплементација е класата Hand која е апстрактна класа исодржи само листа од карти и неколку функции.

public abstract class Hand

{

public List<Card> hand { get; set; }

public Hand()

{

hand = new List<Card>();

}

public Card getCard(int cardPos)

{

if (cardPos >= 0 && cardPos < hand.Count)

{

return (Card)hand.ElementAt(cardPos);

}

else

{

return null;

}

}

public void addCard(Card card)

{

if (card != null)

{

hand.Add(card);

}

}

public int getCardCount()

{

return hand.Count;

}

public abstract int getTotalValue();

}

Од класата Hand наследуваат PlayerHand и DealerHand. Единствена разлика меѓу овие класи е функцијата getTotalValue() која ни го дава вкупниот резултат на раката на играчот или дилерот.

Кодот во PlayerHand:

public class PlayerHand : Hand

{

public override int getTotalValue()

{

int total = 0, count = 0;

Card []aces = new Card[5];

for(int i = 0; i < hand.Count; i++)

{

if (hand[i].isAce)

{

aces[count] = hand[i];

count++;

}

else

{

total += hand[i].value;

}

}

for(int i=0; i < count; i++)

{

if (total >= 11)

{

total += 1;

}

else if (total < 11)

{

total += 11;

}

}

return total;

}

}

Овде getTotalValue() ги поминува картите во раката и ако картата е ас(зависи од вредноста на isAce) ја додава во посебна низа а ако не е ја додава на total односно збирот. Потоа во зависно од збирот ги додава со вредност 11 ако збирот е помал од 11 или ги додава со вредност 1 ако збирот е поголем од 11. На крај го враќа резултатот.

Кодот во DealerHand:

public class DealerHand : Hand

{

public bool stand { get; set; }

public DealerHand(bool stand)

{

this.stand = stand;

}

public bool Blackjack()

{

if(hand[0].isAce && hand[1].value == 10)

{

return true;

}else if (hand[0].value == 10 && hand[1].isAce)

{

return true;

}

else

{

return false;

}

}

public override int getTotalValue()

{

int total = 0,count = 0;

if (stand)

{

Card[] aces = new Card[5];

for (int i = 0; i < hand.Count; i++)

{

if (hand[i].isAce)

{

aces[count] = hand[i];

count++;

}

else

{

total += hand[i].value;

}

}

for (int i = 0; i < count; i++)

{

if (total >= 11)

{

total += 1;

}

else if (total < 11)

{

total += 11;

}

}

}

else

{

total = hand[0].value;

}

return total;

}

}

Во класата DealerHand дополнително се чува bool променлива stand која ни означува дали играчот ја искористил опцијата Stand. Оваа bool променлива е потребна за пресметување на збирот кога дилерот има една една превртена и кога сите негови карти се видливи.