

Реализация игры «Блэк Джек» на языке программирования Python

Кизянов Антон Олегович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс создания игры «Блэк Джек». Для создания используется язык программирования Python и специальные символы для мастей карт из кодов Unicode. Созданная игра служит наглядным пояснением как устроена игра «Блэк Джек», а также познакомит с созданием визуальных игр с использованием символов из Unicode.

Ключевые слова: Python, блэк джек

Implementation of the BlackJack game in the Python programming language

Kizyanov Anton Olegovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of creating a Blackjack game. For creation, the programming language Python and special characters for card suits from Unicode codes are used. The created game serves as a visual explanation of how the BlackJack game works, and also introduces the creation of visual games using characters from Unicode.

Keywords: Python, blackjack

Блэкджек, также известный как «21», представляет собой карточную игру, в которой игроки стараются набрать как можно ближе к 21 очку, не переходя через край. Эта программа использует изображения, нарисованные текстовыми символами, что называется ASCII art. Американский стандартный код обмена информацией (ASCII) — это отображение текстовых символов на числовые коды, которые компьютеры использовали до того, как его заменил Unicode. Игральные карты в этой программе являются примером ASCII-арта.

Цель исследования — написать игру «Блэк Джек» на языке программирования Python.

Ранее этим вопросом интересовались С.Н. Дворяткина, А.М. Лопухин развивали тему «Эффективность применения деловых игр в обучении математике студентов-юристов» [1] в которой предлагали решение проблемы актуализации эффективных дидактических средств обучения математике студентов юридических направлений подготовки. Целью

исследования явилась разработка технологии организации и проведения деловой игры по математике для студентов-юристов и оценка ее эффективности. Методологическим основанием в контексте структурно-процессуального и содержательно-информационного обеспечения проектирования технологии выступил синтез подходов, среди которых наиболее значимыми были определены деятельностный, контекстный, технологический и синергетический. Представлены отдельные элементы всех этапов разработанной технологии на примере деловой игры «Судебный процесс по правовому регулированию игорного бизнеса». Полученные результаты открывают возможность для дальнейшего выявления и исследования синергетических эффектов у будущих юристов (мотивационный, когнитивный, профессиональный, экономический), эффективной реализации разработанной технологии с целью дальнейшего повышения уровня профессионализма посредством совершенствования профессиональных компетенций в области судебного судопроизводства и правотворчества с применением математических методов.

К.С. Волков с темой «Математика в азартных играх» [2], а подробнее про разные события с разным исходом. Например, подбрасывая вверх монету, игрок не знает, какой стороной она упадет. Производя повторные высокоточные измерения, например, скорости света или очень больших расстояний, обычно получают лишь приблизительно равные, но разные результаты. Все эти эксперименты производятся в одинаковых условиях, а исходы их различны и непредсказуемы. Такие эксперименты и исходы называются случайными. А.С. Храпов опубликовал статью «Кроссплатформенная программа реализации игры "блэк джек"» [3] в статье демонстрируется программа однопользовательской компьютерной карточной игры «Блэк Джек». Одним из игроков является пользователь компьютера, а вторым - компьютер. На экране монитора отображаются кнопки, соответствующие действиям в игре, на которые можно нажимать с помощью мышки. В начале игры генерируется массив индексов, соответствующих картам колоды, случайный порядок размещения индексов в массиве генерируется функцией генерации случайных чисел, предварительно рандомизированной с помощью значения системного времени на компьютере. После завершения каждой игры в массивах суммируется количество выигрышей пользователя и компьютера в качестве игрока, указанные значения отображаются на экране и сохраняются до завершения работы программы.

Символов масти карт не существуют на клавиатуре, поэтому в коде вызывается функцию `chr()` для их создания. Целое число, переданное в `chr()`, называется кодовой точкой Unicode, уникальным числом, которое идентифицирует символ в соответствии со стандартом Unicode.

```
import random
import sys
```

```
HEARTS = chr(9829)
```

```
DIAMONDS = chr(9830)
SPADES = chr(9824)
CLUBS = chr(9827)
BACKSIDE = 'backside'
```

```
def main():
    print('''Blackjack,
Правила: постарайтесь приблизиться к 21,
не переходя через край. Короли, дамы и валеты приносят 10 очков.
Тузы приносят 1 или 11 очков.
Карты со 2 по 10 соответствуют своей номинальной стоимости.
(H) это взять другую карту.
(S) прекратить принимать карты.
В вашей первой игре вы можете (D) удвоить ставку, чтобы увеличить ставку,
но должны сделать еще один ход, прежде чем остановиться.
В случае ничьей ставка возвращается игроку''')

    money = 5000
    while True:
        if money <= 0:
            print("Ты проиграл!")
            print("Хорошо, что ты не играл на настоящие деньги.")
            print('Спасибо за игру!')
            sys.exit()

        print('Деньги:', money)
        bet = getBet(money)
        deck = getDeck()
        dealerHand = [deck.pop(), deck.pop()]
        playerHand = [deck.pop(), deck.pop()]
        print('Ставка:', bet)
        while True:
            displayHands(playerHand, dealerHand, False)
            print()
            if getHandValue(playerHand) > 21:
                break

            move = getMove(playerHand, money - bet)
            if move == 'D':
                additionalBet = getBet(min(bet, (money - bet)))
                bet += additionalBet
                print('Ставка увеличена до {}'.format(bet))
                print('Ставка:', bet)

            if move in ('H', 'D'):
                newCard = deck.pop()
                rank, suit = newCard
                print('Вы нарисовали {} из {}'.format(rank, suit))
                playerHand.append(newCard)

            if getHandValue(playerHand) > 21:
                continue
            if move in ('S', 'D'):
                break

        if getHandValue(playerHand) <= 21:
            while getHandValue(dealerHand) < 17:
                print('Дилера ход...')
                dealerHand.append(deck.pop())
                displayHands(playerHand, dealerHand, False)

            if getHandValue(dealerHand) > 21:
                break
```

```
        input('Нажмите Enter, чтобы продолжить...')
        print('\n\n')

    displayHands(playerHand, dealerHand, True)
    playerValue = getHandValue(playerHand)
    dealerValue = getHandValue(dealerHand)
    if dealerValue > 21:
        print('Перебор дилера! Вы выигрываете {} $!'.format(bet))
        money += bet
    elif (playerValue > 21) or (playerValue < dealerValue):
        print('Ты проиграл!')
        money -= bet
    elif playerValue > dealerValue:
        print('Вы выиграли {} $!'.format(bet))
        money += bet
    elif playerValue == dealerValue:
        print('Ничья, ставка вам возвращается.')

    input('Нажмите Enter, чтобы продолжить...')
    print('\n\n')

def getBet(maxBet):
    while True:
        print('Сколько вы ставите? (1- {} или ВЫЙТИ)'.format(maxBet))
        bet = input('> ').upper().strip()
        if bet == 'ВЫЙТИ':
            print('Спасибо за игру!')
            sys.exit()

        if not bet.isdecimal():
            continue

        bet = int(bet)
        if 1 <= bet <= maxBet:
            return bet

def getDeck():
    deck = []
    for suit in (HEARTS, DIAMONDS, SPADES, CLUBS):
        for rank in range(2, 11):
            deck.append((str(rank), suit))
        for rank in ('J', 'Q', 'K', 'A'):
            deck.append((rank, suit))
    random.shuffle(deck)
    return deck

def displayHands(playerHand, dealerHand, showDealerHand):
    print()
    if showDealerHand:
        print('ДИЛЕР:', getHandValue(dealerHand))
        displayCards(dealerHand)
    else:
        print('ДИЛЕР: ???')
        displayCards([BACKSIDE] + dealerHand[1:])
    print('ИГРОК:', getHandValue(playerHand))
    displayCards(playerHand)

def getHandValue(cards):
    value = 0
    numberOfAces = 0
```

```

for card in cards:
    rank = card[0]
    if rank == 'A':
        numberOfAces += 1
    elif rank in ('K', 'Q', 'J'):
        value += 10
    else:
        value += int(rank)

value += numberOfAces
for i in range(numberOfAces):
    if value + 10 <= 21:
        value += 10

return value

def displayCards(cards):
    rows = ['', '', '', '', '']
    for i, card in enumerate(cards):
        rows[0] += '_____'
        if card == BACKSIDE:
            rows[1] += '|## |'
            rows[2] += '|###|'
            rows[3] += '|_##|'
        else:
            rank, suit = card
            rows[1] += '|{} |'.format(rank.ljust(2))
            rows[2] += '| {} |'.format(suit)
            rows[3] += '|_{}|'.format(rank.rjust(2, '_'))
    for row in rows:
        print(row)

def getMove(playerHand, money):
    while True:
        moves = ['(H) ВЗЯТЬ', '(S) ОСТАНОВИТЬСЯ']
        if len(playerHand) == 2 and money > 0:
            moves.append('(D) УДВОИТЬ')

        movePrompt = ', '.join(moves) + '> '
        move = input(movePrompt).upper()

        if move in ('H', 'S'):
            return move

        if move == 'D' and '(D) УДВОИТЬ' in moves:
            return move

if __name__ == '__main__':
    main()

```

В блэкджеке есть несколько специальных правил, которые можно применить. Например, если первые две карты имеют одинаковое значение, игрок может разделить их на две руки и поставить на них по отдельности. Кроме того, если игрок получает «блэкджек» (туз пик и черный валет) за свои первые две карты, игрок выигрывает выплату десять к одному.

Результат работы кода выше можно наблюдать на рисунке 1.

- проблемы физико-математических наук. Материалы III Международной научно-практической конференции. Под общ. ред. Т.Н. Можаровой. 2017. С. 461-468. URL: <https://elibrary.ru/item.asp?id=35261108> (Дата обращения: 05.01.2022)
2. Волков К.С. Математика в азартных играх // Экономика и социум. 2019. № 2 (57). С. 82-84. URL: <https://elibrary.ru/item.asp?id=38186716> (Дата обращения: 05.01.2022)
 3. Храпов А.С. КРОССПЛАТФОРМЕННАЯ ПРОГРАММА РЕАЛИЗАЦИИ ИГРЫ "БЛЭК ДЖЕК" // Свидетельство о регистрации программы для ЭВМ RU 2016612438, 26.02.2016. Заявка № 2015660320 от 28.10.2015. URL: <https://elibrary.ru/item.asp?id=39344439> (Дата обращения: 05.01.2022)