**Федеральное государственное образовательное**

**бюджетное учреждение высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**ИНСТИТУТ ЗАОЧНОГО И ОТКРЫТОГО ОБРАЗОВАНИЯ**

**ДЕПАРТАМЕНТ АНАЛИЗА ДАННЫХ И МАШИННОГО ОБУЧЕНИЯ**

**ПРАКТИЧЕСКАЯ РАБОТА № 1**

**по дисциплине «Практикум по программированию»**

**Вариант №5. Игра «Лоскутное одеяло»**

Направление подготовки - 09.03.03 «Прикладная информатика»

Студент(ка) **Белоусов А.А.**

(Ф.И.О.)

Курс2 № группы **ЗБ-ПИ19-1**

Преподаватель **Горохова Р.И.**

(Ф.И.О.)

Москва 2020

Задание

Реализовать программу, при помощи которой 3 игрока могут играть в игру «Лоскутное одеяло». Правила игры следующие. На поле, имеющем размер 4 на 5 клеток за один ход каждый игрок должен заполнить одну клетку своим символом. Игрок старается, чтобы его символы были как можно дальше друг от друга. В ходе игры ведется подсчет очков: за каждое соседство клеток с одинаковыми символами игроку, владельцу символа добавляется одно штрафное очко. Соседними считаются клетки, имеющие общую сторону или расположенные наискосок друг от друга. Выигрывает тот, у кого в конце игры меньше всего штрафных очков.

Взаимодействие с программой производится через консоль. Игровое поле изображается в виде 4 текстовых строк и перерисовывается при каждом изменении состояния поля. При запросе данных от пользователя программа сообщает, что ожидает от пользователя (например, координаты очередного хода) и проверяет корректность ввода. Программа должна уметь автоматически определять количество штрафных очков и окончание партии и ее победителя.

Сама программа НЕ ходит, т.е. не пытается заполнять клетки символами с целью выиграть игру.

Код программы

import shutil  
import copy  
import os  
from colorama import Fore  
from colorama import Backstate = [[]]  
marker = {**'x'**: 0, **'y'**: 0}  
currentPlayer = **'1'**playersCount = {**'1'**: 0, **'2'**: 0, **'3'**: 0}  
  
  
def init():  
 global state  
 global marker  
 global currentPlayer  
 global playersCount  
 state = [[**'s'** for j in range(5)] for i in range(4)]marker = {**'x'**: 0, **'y'**: 0}  
 currentPlayer = **'1'** playersCount = {**'1'**: 0, **'2'**: 0, **'3'**: 0}  
  
  
def getTableContent(drawMarker=True):tableContent = [[[] for j in range(5)] for i in range(4)]  
 cellTypes = {**'s'**: [[**' '**] for j in range(4)],  
 **'1'**: [[Fore.RED + **'+'** \* 6 + **' '** \* 3 + **'+'** + Fore.CYAN],  
 [Fore.RED + **'+'** \* 4 + **' '** \* 4 + **'+'** \* 2 + Fore.CYAN],  
 [Fore.RED + **'+'** \* 2 + **' '** \* 4 + **'+'** \* 4 + Fore.CYAN],  
 [Fore.RED + **'+'** + **' '** \* 3 + **'+'** \* 6 + Fore.CYAN]],  
 **'2'**: [[Fore.BLUE + **'@'** \* 5 + **' '** \* 5 + Fore.CYAN],  
 [Fore.BLUE + **' '** \* 5 + **'@'** \* 5 + Fore.CYAN],  
 [Fore.BLUE + **'@'** \* 5 + **' '** \* 5 + Fore.CYAN],  
 [Fore.BLUE + **' '** \* 5 + **'@'** \* 5 + Fore.CYAN]],  
 **'3'**: [[Fore.GREEN + **' '** \* 4 + **'\*'** \* 2 + **' '** \* 4 + Fore.CYAN],  
 [Fore.GREEN + **' '** \* 3 + **'\*'** \* 4 + **' '** \* 3 + Fore.CYAN],  
 [Fore.GREEN + **' '** + **'\*'** \* 8 + **' '** + Fore.CYAN], [Fore.GREEN + **'\*'** \* 10 + Fore.CYAN]],  
 }  
for i in range(len(state)):  
 for j in range(len(state[0])):  
 tableContent[i][j] = copy.deepcopy(cellTypes[state[i][j]])if drawMarker:  
 def getBackColor():  
 if currentPlayer == **'1'**:  
 return Back.RED  
 elif currentPlayer == **'2'**:  
 return Back.BLUE  
 else:  
 return Back.GREEN  
  
 backColor = getBackColor()  
  
 for i in range(4):  
 tableContent[marker[**'x'**]][marker[**'y'**]][i][0] = backColor + tableContent[marker[**'x'**]][marker[**'y'**]][i][  
 0] + Back.RESET  
  
 return tableContent  
  
  
def draw(tableContent):def getRow(index):  
 result = **""** for i in range(4):  
 result += **"**\n\t**║"** for j in range(5):result += tableContent[index][j][i][0] + **"║"** return result  
  
 table = **f'''**{Fore.CYAN}  
\t**╔══════════╦══════════╦══════════╦══════════╦══════════╗**{getRow(0)}  
\t**╠══════════╬══════════╬══════════╬══════════╬══════════╣**{getRow(1)}  
\t**╠══════════╬══════════╬══════════╬══════════╬══════════╣**{getRow(2)}  
\t**╠══════════╬══════════╬══════════╬══════════╬══════════╣**{getRow(3)}  
\t**╚══════════╩══════════╩══════════╩══════════╩══════════╝**{Fore.WHITE}**'''** os.system(**"cls"**)  
 print(table)  
  
  
def changePlayerCount():  
 playersCount[currentPlayer] += sum([row[max(0, marker[**'y'**] - 1):marker[**'y'**] + 2].count(currentPlayer) for row in  
 state[max(0, marker[**'x'**] - 1):marker[**'x'**] + 2]]) - 1  
  
  
  
def showPlayerCount():  
 print(**f"**\t{Fore.RED}**Player 1:** {playersCount[**'1'**]}**,** {Fore.BLUE}**Player 2:** {playersCount[**'2'**]}**,** {Fore.GREEN}**Player 3:** {playersCount[**'3'**]}{Fore.RESET}**"**)  
  
  
def move():  
 def nextPlayer():  
 global currentPlayer  
 currentPlayer = str(int(currentPlayer) % 3 + 1)  
  
 def newMarker():  
 for i in range(4):  
 for j in range(5):  
 if state[i][j] == **'s'**:  
 marker[**'x'**] = i  
 marker[**'y'**] = j  
 return  
  
 while True:  
 command = input()  
 if command == **"r"**:for j in range(marker[**'y'**] + 1, 5):  
 if state[marker[**'x'**]][j] == **'s'**:  
 marker[**'y'**] = j  
 break  
 else:  
 for i in range(marker[**'x'**] + 1, 4):  
 for j in range(5):  
 if state[i][j] == **'s'**:  
 marker[**'x'**] = i  
 marker[**'y'**] = j  
 break  
 else:  
 continue  
 break  
 elif command == **"s"**:for i in range(marker[**'x'**] + 1, 4):  
 if state[i][marker[**'y'**]] == **'s'**:  
 marker[**'x'**] = i  
 break  
 else:  
 for j in range(marker[**'y'**] + 1, 5):  
 for i in range(4):  
 if state[i][j] == **'s'**:  
 marker[**'x'**] = i  
 marker[**'y'**] = j  
 break  
 else:  
 continue  
 break  
 elif command == **"l"**:for j in range(marker[**'y'**] - 1, -1, -1):  
 if state[marker[**'x'**]][j] == **'s'**:  
 marker[**'y'**] = j  
 break  
 else:  
 for i in range(marker[**'x'**] - 1, -1, -1):  
 for j in range(4, -1, -1):  
 if state[i][j] == **'s'**:  
 marker[**'x'**] = i  
 marker[**'y'**] = j  
 break  
 else:  
 continue  
 break  
 elif command == **"w"**:for i in range(marker[**'x'**] - 1, -1, -1):  
 if state[i][marker[**'y'**]] == **'s'**:  
 marker[**'x'**] = i  
 break  
 else:  
 for j in range(marker[**'y'**] - 1, -1, -1):  
 for i in range(3, -1, -1):  
 if state[i][j] == **'s'**:  
 marker[**'x'**] = i  
 marker[**'y'**] = j  
 break  
 else:  
 continue  
 break  
 elif command == **''**:  
 state[marker[**'x'**]][marker[**'y'**]] = currentPlayer  
 changePlayerCount()  
 break  
 draw(getTableContent())  
 showPlayerCount()  
 nextPlayer()  
 newMarker()  
  
  
def showPlayer(index):  
 if index == **'1'**:playerStr = Fore.RED + **f'''**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++++++++ ++ +++ ++ ++ ++++++++ ++++++++ ++ ++ ++ ++++ ++ ++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++++ ++ ++ ++ ++ +++ ++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++ ++ ++ ++ ++ ++++ ++ ++ ++ ++ ++ ++ ++ ++ ++++ ++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++++++++ ++ ++ ++ ++ ++++++ ++++++++ ++ ++ ++ ++ ++ ++ ++ ++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++ ++ +++++++++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++ ++ +++**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}**++ ++++++++ ++ ++ ++ ++++++++ ++ ++ ++++++ +++ +++ ++++ ++ ++'''** \  
 + Fore.RESET  
 print(playerStr)  
 elif index == **'2'**:playerStr = Fore.BLUE + **f'''**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@@@@@@@ @@ @@@ @@ @@ @@@@@@@@ @@@@@@@@ @@@@@@@ @@ @@ @@@@ @@ @@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@@ @@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@ @@ @@ @@ @@ @@@@ @@ @@ @@ @@ @@ @@ @@ @@ @@@@ @@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@@@@@@@ @@ @@ @@ @@ @@@@@@ @@@@@@@@ @@@@@@@ @@ @@ @@ @@ @@ @@ @@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@ @@ @@@@@@@@@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@@@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@ @@@**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}**@@ @@@@@@@@ @@ @@ @@ @@@@@@@@ @@ @@ @@@@@@@@@ @@@ @@@ @@@@ @@ @@'''** \  
 + Fore.RESET  
 print(playerStr)  
 else:playerStr = Fore.GREEN + **f'''**{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\*\*\*\*\*\*\* \*\* \*\*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\* \*\* \*\*\*\* \*\* \*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*\* \*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\* \*\* \*\* \*\* \*\* \*\*\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*\*\* \*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\*\*\*\*\*\*\* \*\* \*\* \*\* \*\* \*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\* \*\* \*\*\*\*\*\*\*\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*\*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*\***{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}**\*\* \*\*\*\*\*\*\*\* \*\* \*\* \*\* \*\*\*\*\*\*\*\* \*\* \*\* \*\*\*\*\*\*\* \*\*\* \*\*\* \*\*\*\* \*\* \*\*'''** \  
 + Fore.RESET  
 print(playerStr)  
  
  
def showWinner():  
 count = len(set(playersCount.values()))  
 if count == 1:  
 tempStr = Fore.CYAN + **f'''**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## ## ## ## ## ## ########**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## ## ### ## ## ## ## ##**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## ## #### ## ## ## ## ##**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**######### ## ### ## ######### ######## ########**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## #### ## ## ## ## ## ##**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## ### ## ## ## ## ## ##**{Back.CYAN + **' '** \* 10 + Back.RESET + **' '**}**## ## ## ## ## ######## ## ##'''** + Fore.RESET  
 print(tempStr)  
 elif count == 2:  
 winners = [key for key, value in playersCount.items() if value == min(playersCount.values())]for winner in winners:  
 showPlayer(winner)  
 else:showPlayer([key for key, value in playersCount.items() if value == min(playersCount.values())][0])  
  
  
def showRules():  
 os.system(**"cls"**)  
 print(**"Игра лоскутное одеяло"**.center(shutil.get\_terminal\_size().columns))  
 print()  
 print()  
 print(**"Правила игры"**.center(shutil.get\_terminal\_size().columns))  
 print(  
 **'На поле, имеющем размер 4 на 5 клеток за один ход каждый игрок должен заполнить одну клетку своим символом. '  
 'Игрок старается, чтобы его символы были как можно дальше друг от друга. В ходе игры ведется подсчет очков: '  
 'за каждое соседство клеток с одинаковыми символами игроку, владельцу символа добавляется одно штрафное очко. '  
 'Соседними считаются клетки, имеющие общую сторону или расположенные наискосок друг от друга. '  
 'Выигрывает тот, у кого в конце игры меньше всего штрафных очков.'**)  
 print()  
 print(**"Управление"**.center(shutil.get\_terminal\_size().columns))  
 print(**"Для выбора клетки вводить буквы: w,s,d,a и жмите Enter"**)  
 print(**"Затем чтобы зафиксировать выбор нажмите Enter"**)  
 print(**"Для выхода из игры нажмите Ctrl+C"**)  
 print()  
 print(**"Маркеры игроков"**.center(shutil.get\_terminal\_size().columns))  
 tempStr = **f'''**{Back.RED + **' '** \* 10 + Back.RESET + **' '**}{Fore.RED + **'+'** \* 6 + **' '** \* 3 + **'+'** + Fore.CYAN}  
{Back.RED + **' '** \* 10 + Back.RESET + **' '**}{Fore.RED + **'+'** \* 4 + **' '** \* 4 + **'+'** \* 2 + Fore.CYAN}  
{Back.RED + **' '** \* 10 + Back.RESET + **' '**}{Fore.RED + **'+'** \* 2 + **' '** \* 4 + **'+'** \* 4 + Fore.CYAN}  
{Back.RED + **' '** \* 10 + Back.RESET + **' '**}{Fore.RED + **'+'** + **' '** \* 3 + **'+'** \* 6 + Fore.CYAN}**'''** print(tempStr)  
 tempStr = **f'''**{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}{Fore.BLUE + **'@'** \* 5 + **' '** \* 5 + Fore.CYAN}  
{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}{Fore.BLUE + **' '** \* 5 + **'@'** \* 5 + Fore.CYAN}  
{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}{Fore.BLUE + **'@'** \* 5 + **' '** \* 5 + Fore.CYAN}  
{Back.BLUE + **' '** \* 10 + Back.RESET + **' '**}{Fore.BLUE + **' '** \* 5 + **'@'** \* 5 + Fore.CYAN}**'''** print(tempStr)  
 tempStr = **f'''**{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}{Fore.GREEN + **' '** \* 4 + **'\*'** \* 2 + **' '** \* 4 + Fore.CYAN}  
{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}{Fore.GREEN + **' '** \* 3 + **'\*'** \* 4 + **' '** \* 3 + Fore.CYAN}  
{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}{Fore.GREEN + **' '** + **'\*'** \* 8 + **' '** + Fore.CYAN}  
{Back.GREEN + **' '** \* 10 + Back.RESET + **' '**}{Fore.GREEN + **'\*'** \* 10 + Fore.CYAN}**'''** print(tempStr)  
 print()  
 print(**"Что-бы запустить игру -нажмите Enter"**.center(shutil.get\_terminal\_size().columns))  
 input()  
  
  
def game():  
 try:  
 showRules()  
 while True:  
 init()  
 for i in range(19):  
 draw(getTableContent())  
 showPlayerCount()  
 move()  
 state[marker[**'x'**]][marker[**'y'**]] = currentPlayer  
 changePlayerCount()  
 draw(getTableContent(False))  
 showPlayerCount()  
 showWinner()  
 input()  
 except KeyboardInterrupt:  
 exit(0)  
  
  
def main():  
 game()  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

Результат выполнения программы

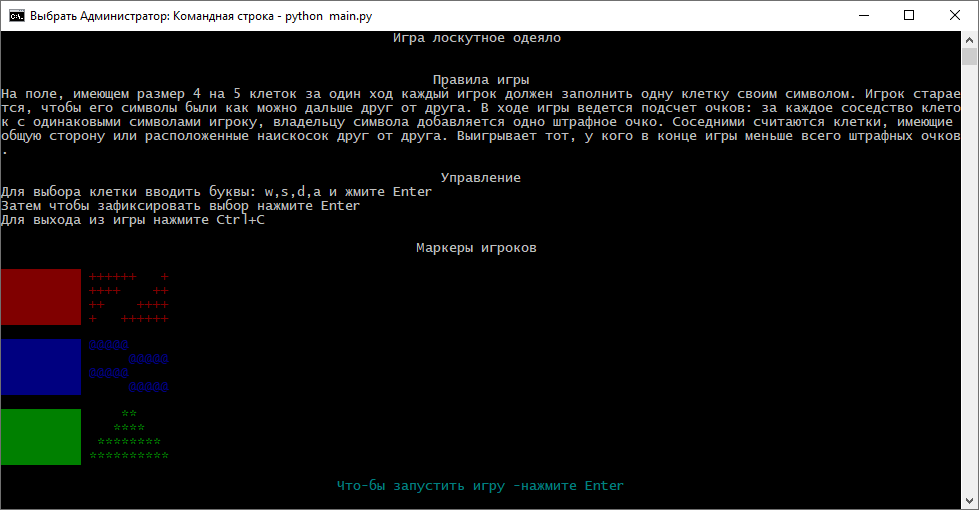


Рис.1 – Стартовый экран игры



Рис.2 – Игровой интерфейс



Рис.3 – Итоговый экран