

Dokumentacja projektu

Języki skryptowe

Tatiana Cieślak, gr III E
Wydział Matematyki Stosowanej kier. Informatyka

30 stycznia 2020

Część I

Opis programu

Program gra z użytkownikiem w klasyczne dziewięciopole kółko i krzyżyk. Wygrywa gracz, który zajmie trzy pola w rzędzie (poziomo, pionowo lub na ukos). W przypadku gdy nikt nie wygrywa i zostają zajęte wszystkie pola na planszy, następuje remis. Nie można także stawiać pól w miejscach na planszy już zajętych. Użytkownik zawsze rozpoczyna grę. Gra zaimplementowana jest w sposób, gdzie użytkownik nigdy nie wygrywa - komputer bowiem grając rozsądnie doprowadza do remisu lub swojej wygranej. Jest to program zrealizowany na podstawie zadania drugiego z konkursu Algorytmion 2015.

Instrukcja obsługi

Aby włączyć program należy otworzyć plik menu.bat znajdujący się w folderze projektu. W nim użytkownik ma do wyboru następujące opcje: - Start programu - Backup - Informacje o projekcie - Wyjście. Po wybraniu opcji Start programu uruchamiana jest gra, a po jej zakończeniu otwierana jest strona html z wynikami gry. Następnie użytkownik wraca do menu i może wybrać jedną z kolejnych opcji.

Menu

```
-----MENU-----
      1. Start programu
      2. Backup
      3. Informacje o projekcie
      4. Wyjście
-----
Wpisz 1/2/3/4:
```

Zakończenie rozgrywki

STATYSTYKA KÓŁKO I KRZYŻYK ~Tatiana Cieślar

Kto wygrał?	Data	Ilość ruchów Gracza
Remis	2020/01/25, 21:36:16	5
Remis	2020/01/25, 21:48:06	5
Komputer	2020/01/25, 21:53:36	3
Komputer	2020/01/25, 21:55:00	3
Remis	2020/01/25, 21:57:20	5
Komputer	2020/01/25, 21:58:52	3
Remis	2020/01/25, 21:59:12	5
Komputer	2020/01/25, 21:59:35	4
Komputer	2020/01/25, 22:03:48	3
Remis	2020/01/25, 22:04:24	5
Remis	2020/01/25, 22:05:10	5
Remis	2020/01/28, 09:25:43	5

Część II

Część techniczna

Projekt składa się z 3 katalogów:

1. dokumentacja: znajduje się tam dokumentacja projektu w formacie .pdf
2. projekt:
 - (a) KolkoKrzyzyk.html (tworzony w trakcie wykonywania programu) - strona html wraz z wynikami rozgrywek
 - (b) KolkoKrzyzyk.py - główny plik programu, w którym toczy się cała rozgrywka
 - (c) menu.bat - menu, z którego użytkownik wybiera opcje
 - (d) Strona.py - plik programu, w którym tworzy się strona html
 - (e) wynik.txt (tworzony w trakcie wykonywania programu) - zamieszczone są tam wyniki poszczególnych rozgrywek
3. backup (tworzony w trakcie działania programu): tworzą się tam backupowe pliki .html

Opis działania

Po uruchomieniu pliku menu.bat i wybraniu opcji Start programu, uruchamiany jest plik KolkoKrzyzyk.py, gdzie użytkownik rozgrywa grę przeciw komputerowi. Po jej zakończeniu otwierany jest plik KolkoKrzyzyk.html, w którym do statystyk dopisana zostaje właśnie rozegrana gra. Użytkownik powraca do menu, gdzie do wyboru ma wyświetlenie informacji o projekcie, wykonanie backupu lub wyjście. Przy wykonywaniu backupu w osobnym folderze o nazwie "backup" pojawiają się pliki KolkoKrzyzyk.html ze zmienioną nazwą na raport-data, gdzie data to dokładna data i czas wykonania backupu przez użytkownika.

Implementacja

1. Plik KolkoKrzyzyk.py - podział na metody:
 - (a) poczatek()
Jest to metoda wywołana jako pierwsza. Wita użytkownika oraz przedstawia sposób indeksowania na planszy. Informuje użytkownika o tym, że zaczyna grę. Z niej wywołana jest metoda rysujPlansze().
 - (b) rysujPlansze()
Ta metoda w trakcie trwania programu uaktualnia bieżący stan planszy i wyświetla ją na konsoli, a następnie wywołuje funkcję ruchUzytkownika().
 - (c) czyJestZwyciezca()
Metoda ta sprawdza, czy można orzec zwycięzcę lub remis i zakończyć rozgrywkę. Kolejno sprawdza, czy w liniach nie znajdują się trzy symbole 'X' (co oznacza zwycięstwo komputera), trzy symbole 'O' (co oznacza zwycięstwo użytkownika) oraz czy każde miejsce na planszy jest zajęte (co oznacza remis). Zwraca wartość true gdy istnieje zwycięzca, a false gdy gra ma być dalej kontynuowana.

(d) `ruchUzytkownika()`

Metoda `ruchUzytkownika()` prosi użytkownika o wykonanie ruchu, a także sprawdza czy wpisał liczbę całkowitą znajdującą się w przedziale od 1 do 9. Blokowana jest także możliwość postawienia symbolu w miejscu na planszy już zajętym. Gdy wpisze on liczbę rzeczywistą jest ona konwertowana na liczbę całkowitą i zaokrąglana według ogólnych zasad zaokrąglania liczb. Następnie sprawdzane jest, czy gra została zakończona poprzez wywołanie funkcji `czyJestZwyciezca()`. Jeśli zwraca wartość `true`, program przechodzi do funkcji `zakonczenie()`, natomiast jeśli zwraca wartość `false`, wywołana zostaje metoda `ruchKomputera()`.

(e) `zakonczenie()`

W zakończeniu pokazywany jest ostateczny stan planszy, a także wyświetlany na konsoli jest zwycięzca gry (lub informacja o remisie). Zaimplementowany jest tam także zapis rezultatu do pliku „wynik.txt”.

(f) `ruchKomputera()`

Komputer rozgrywa grę w taki sposób, aby doprowadzić do przegranej użytkownika lub do remisu. Zawsze rozpoczyna grę od postawienia symbolu na środku planszy (jeśli jest to możliwe), następnie w różnych przypadkach gry użytkownika stawia symbol w odpowiednich miejscach na planszy. Zaimplementowane jest blokowanie możliwości zwycięstwa użytkownika, a także zawsze wykorzystywanie możliwości zwycięstwa przez komputer. Sprawdzane jest także wiele przypadków, w których użytkownik ma możliwość zwycięstwa. Po tej metodzie sprawdzane jest, czy gra została zakończona poprzez wywołanie funkcji `czyJestZwyciezca()`. Jeśli zwraca `true` to wywoływana jest metoda `zakonczenie()`, a gdy zwraca `false` wywoływana jest funkcja `rysujPlansze()`.

(g) `wygranaKomputera()`

Ta metoda sprawdza, czy komputer ma szansę wygrać w kolejnej kolejce, jeżeli tak, to stawia ‘X’ w miejscu, które to umożliwia (3 symbole w linii). Zwraca wartość `true` lub `false` w zależności od tego, czy ruch został wykonany.

(h) `blokowanie()`

Metoda analogiczna do `wygranaKomputera()` – sprawdza, czy użytkownik ma szansę wygrać w kolejnej kolejce, jeżeli tak, to stawia ‘X’ w miejscu, które to uniemożliwia. Zwraca wartość `true` lub `false` w zależności od tego, czy ruch został wykonany.

(i) `wyborRogu()`

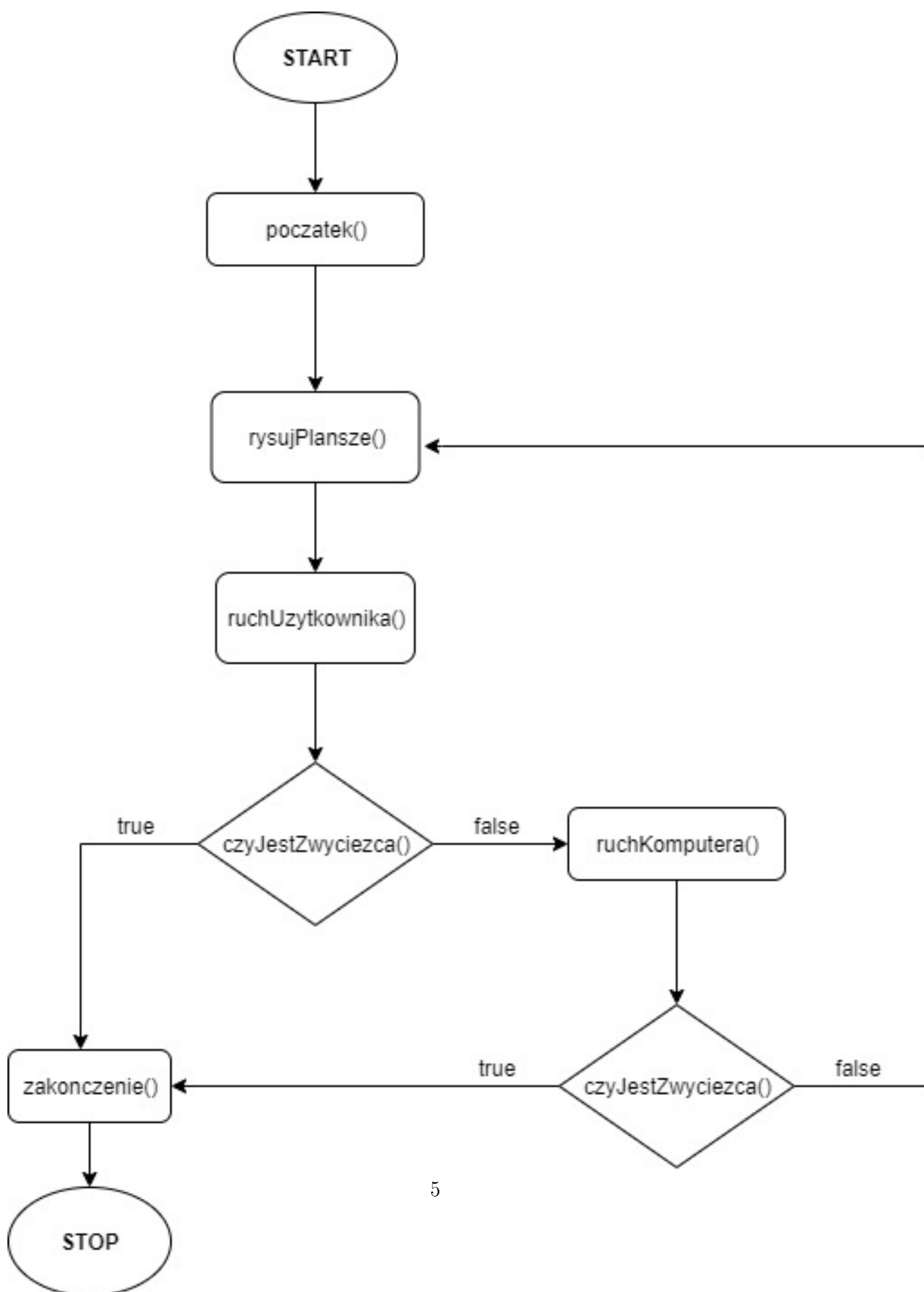
Jest to metoda wykonująca ruch komputera, która stawia symbol ‘X’ na jednym z dostępnych rogów planszy – mają one kolejno indeksy 1,3,7 i 9.

(j) `wyborBoku()`

Jest to metoda wykonująca ruch komputera, która stawia symbol ‘X’ na jednym z dostępnych boków planszy – mają one kolejno indeksy 2,4,6 i 8.

2. Plik `Strona.py` - w tym pliku następuje konwersja wyników z pliku `wynik.txt` do pliku `KolkoKrzyzyk.html`. Używając składni `html` tworzona jest tablica z wartościami pobranymi z pliku.

Schemat blokowy



Pełen kod programu

- KolkoKrzyzyk.py

```
import datetime
import random
import sys
import os
from time import sleep

plansza = ["_","_","_","_","_","_"," "," "," "]
ktoryRuch = 1
rogi = [0,2,6,8]
boki = [1,3,5,7]
mozliweRzedyWygrania = [[0,1,2],[3,4,5],[6,7,8],[0,3,6],
[1,4,7],[2,5,8],[0,4,8],[2,4,6]]

print("Zagrajmy w kółko i krzyżyk!")
print("0 to indeksy na planszy:")
print("1|2|3")
print("4|5|6")
print("7|8|9\n")
print("Zaczynasz grę.")
sleep(2)

def drukujPlansze(plansza, ktoryRuch):
    os.system("cls")
    print (plansza[0]+ "|" +plansza[1] + "|" + plansza[2])
    print (plansza[3]+ "|" +plansza[4] + "|" + plansza[5])
    print (plansza[6]+ "|" +plansza[7] + "|" + plansza[8])

    ruchGracza(plansza, ktoryRuch)

def ruchGracza(plansza, ktoryRuch):

    index = input("Wybierz indeks gdzie chcesz postawić '0': ")

    if index.isdigit():
        index = int(index) - 1
        if int(index)>8 or int(index)<0:
            print("Musisz wybrać numer z zakresu 1-9")
            ruchGracza(plansza, ktoryRuch)
```

```

        if(plansza[int(index)]=="X" or plansza[int(index)]=="O"):
            print("To miejsce jest już zajęte")
            ruchGracza(plansza, ktoryRuch)

        else:
            plansza[int(index)] = "O"

        if(czyKtosWygral(plansza, mozliweRzedyWygrania)==True):
            zakonczenie(plansza)
        else:
            ruchKomputera(plansza, rogi, boki, ktoryRuch, mozliweRzedyWygrania)
    else:
        print("Musisz wpisać liczbę całkowitą")
        ruchGracza(plansza, ktoryRuch)

def ruchKomputera(plansza, rogi, boki, ktoryRuch, mozliweRzedyWygrania):
    wykonanyRuch = False

    def wyborRogu(rogi, plansza):
        mozliweWybory = []
        for i in rogi:
            if (plansza[i]==" " or plansza[i]=="_"):
                mozliweWybory.append(i)
        k = random.choice(mozliweWybory)
        plansza[k]="X"

    def wyborBoku(boki, plansza):
        mozliweWybory = []
        for i in boki:
            if (plansza[i]==" " or plansza[i]=="_"):
                mozliweWybory.append(i)

        if mozliweWybory==[]:
            wyborRogu(rogi, plansza)

        else:
            plansza[random.choice(mozliweWybory)]="X"

    #sprawdzenie czy środek jest wolny w 1 kolejce
    if ktoryRuch==1:
        if (plansza[4]==" " or plansza[4]=="_"):
            #wybór środka

```

```

        plansza[4]="X"
        ktoryRuch += 1
        wykonanyRuch = True
    else:
        #randomowy róg
        wyborRogu(rog, plansza)
        ktoryRuch += 1
        wykonanyRuch = True

else:
    if wykonanyRuch==False:
        #wygranie komputera
        for i in mozliweRzedyWygrania:
            if(plansza[i[0]]=="X" and plansza[i[1]]=="X"
            and (plansza[i[2]]==" " or plansza[i[2]]=="_")):
                plansza[i[2]]="X"
                ktoryRuch += 1
                wykonanyRuch = True
                break
            if wykonanyRuch==False:
                if(plansza[i[0]]=="X" and (plansza[i[1]]==" "
                or plansza[i[1]]=="_") and plansza[i[2]]=="X"):
                    plansza[i[1]]="X"
                    ktoryRuch += 1
                    wykonanyRuch = True
                    break
            if wykonanyRuch==False:
                if((plansza[i[0]]==" " or plansza[i[0]]=="_")
                and plansza[i[1]]=="X" and plansza[i[2]]=="X"):
                    plansza[i[0]]="X"
                    ktoryRuch += 1
                    wykonanyRuch = True
                    break

        #blokowanie
        if wykonanyRuch==False:
            for i in mozliweRzedyWygrania:
                if(plansza[i[0]]=="0" and plansza[i[1]]=="0"
                and (plansza[i[2]]==" " or plansza[i[2]]=="_")):
                    plansza[i[2]]="X"
                    ktoryRuch += 1
                    wykonanyRuch = True
                    break

```



```

        if wykonanyRuch==False:
            if(plansza[i[0]]=="0" and (plansza[i[1]]==" "
            or plansza[i[1]]=="_") and plansza[i[2]]=="0"):
                plansza[i[1]]="X"
                ktoryRuch += 1
                wykonanyRuch = True
                break
        if wykonanyRuch==False:
            if((plansza[i[0]]==" " or plansza[i[0]]=="_")
            and plansza[i[1]]=="0" and plansza[i[2]]=="0"):
                plansza[i[0]]="X"
                ktoryRuch += 1
                wykonanyRuch = True
                break

if wykonanyRuch==False:
    if ktoryRuch==2 and plansza[4]=="X":

        mozliweRzedyBokowWygranej = [[1,4,7],[3,4,5]]
        bokiGracza = 0

        for i in boki:
            if(plansza[i]=="0"):
                bokiGracza += 1

        #Sytuacje tego typu:
        # 0|_|_ <- tutaj musi być postawiony X
        # _|X|0
        #  |  |

        # _|_|_
        # 0|X|
        #  | |0 <- na pierwszym miejscu musi być postawiony X
        if bokiGracza == 1:
            mozliweRzedyInne = [[0,7],[2,7],[1,8],[1,6],[2,3],
            [3,8],[0,5],[5,6]]

            if wykonanyRuch==False:
                for i in mozliweRzedyInne:
                    if(plansza[i[0]]=="0" and plansza[i[1]]=="0"):
                        for j in mozliweRzedyBokowWygranej:
                            if ((plansza[j[0]]==" "or plansza[j[0]]=="_")
                            and (plansza[j[2]]==" "or plansza[j[2]]=="_")):
                                #tutaj mogloby byc tez j[2]
                                plansza[j[0]]="X"

```

```

                                ktoryRuch += 1
                                wykonanyRuch = True

# 0| | _      _| |0
# _|X|_      _|X|_
# _| |0      0| | _
elif bokiGracza == 0:
    #wybiera bok
    wyborBoku(boki, plansza)
    ktoryRuch += 1
    wykonanyRuch = True

else:
    # _|0|_      _| | _      _| | _
    # _|X|0      _|X|0      0|X|0
    # _| | _      _|0|_      _| | _ <- wybiera ten róg
    if plansza[5]=="0":
        plansza[8]="X"
        ktoryRuch += 1
        wykonanyRuch = True

    # _|0|_      _| | _
    # 0|X|_      0|X|_
    # _| | _ -> _|0|_ wybiera ten róg
    elif plansza[3]=="0":
        plansza[6]="X"
        ktoryRuch += 1
        wykonanyRuch = True

    # _|0|_
    # _|X|_
    # _|0|_
    else:
        wyborRogu(rog, plansza)
        ktoryRuch += 1
        wykonanyRuch = True

elif ktoryRuch==2 and plansza[4]=="0":
    #randomowy corner
    wyborRogu(rog, plansza)
    ktoryRuch += 1
    wykonanyRuch = True

# 0|_|_ <- tu trzeba dać X
# X|X|0

```

```

# |0|

elif ktoryRuch==3 and plansza[4]=="X":
    noweRzedy = [[0,5,7,3,4],[0,5,6,3,4],[1,5,6,3,4],
    [2,3,8,4,5],[1,3,8,4,5],[2,3,7,4,5]]

    for i in range(0, len(noweRzedy)):
        if (plansza[noweRzedy[i][0]]=="0"
        and plansza[noweRzedy[i][1]]=="0"
        and plansza[noweRzedy[i][2]]=="0"
        and plansza[noweRzedy[i][3]]=="X"
        and plansza[noweRzedy[i][4]]=="X"):
            if (i==0):
                plansza[2]="X"
                ktoryRuch += 1
                wykonanyRuch = True

            elif (i==1 or i==2):
                plansza[8]="X"
                ktoryRuch += 1
                wykonanyRuch = True

            elif (i==3 or i==5):
                plansza[0]="X"
                ktoryRuch += 1
                wykonanyRuch = True

            elif (i==4):
                plansza[6]="X"
                ktoryRuch += 1
                wykonanyRuch = True

    if wykonanyRuch==False:
        bokiGracza = 0
        for i in boki:
            if(plansza[i]=="0"):
                bokiGracza += 1

        if bokiGracza >= 1:
            #wybiera corner
            wyborRogu(rogi, plansza)
            ktoryRuch += 1
            wykonanyRuch = True
        else:
            #wybiera bok

```

```

        wyborBoku(boki, plansza)
        ktoryRuch+=1
        wykonanyRuch = True

    else:
        bokiGracza = 0
        for i in boki:
            if(plansza[i]=="0"):
                bokiGracza += 1

        if bokiGracza >= 1:
            #wybiera corner
            wyborRogu(rogi, plansza)
            ktoryRuch += 1
            wykonanyRuch = True
        else:
            #wybiera bok
            wyborBoku(boki, plansza)
            ktoryRuch+=1
            wykonanyRuch = True

    if (czyKtosWygral(plansza, mozliweRzedyWygrania)==True):
        zakonczenie(plansza)
    else:
        drukujPlansze(plansza, ktoryRuch)

def czyKtosWygral(plansza, mozliweRzedyWygrania):

    global winner

    for i in mozliweRzedyWygrania:
        if(plansza[i[0]]=="X" and plansza[i[1]]=="X" and plansza[i[2]]=="X"):
            winner = "Komputer"
            return True
        if(plansza[i[0]]=="0" and plansza[i[1]]=="0" and plansza[i[2]]=="0"):
            winner = "Gracz"
            return True

    filled = 0

    for j in range(0, len(plansza)):
        if (plansza[j]!=" " and plansza[j]!="_"):
            filled = filled + 1

```

```

        if filled==9:
            winner = "Remis"
            return True

    return False

def zakonczenie(plansza):
    print("Ostateczna plansza:")
    print (plansza[0]+ "|" +plansza[1] + "|" + plansza[2])
    print (plansza[3]+ "|" +plansza[4] + "|" + plansza[5])
    print (plansza[6]+ "|" +plansza[7] + "|" + plansza[8])

    if winner=="Remis":
        print("Remis")
    else:
        print("Zwycięzca to: "+ winner)

    iloscRuchowGracza = 0
    for i in range(0,len(plansza)):
        if (plansza[i]=="0"):
            iloscRuchowGracza+=1

    print(f"Gra zakończyła się po wykonaniu {iloscRuchowGracza} ruchów Gracza.")

    plik_wyjsciowy = open("wynik.txt","a")
    plik_wyjsciowy.write(f"\n{winner}\n")
    plik_wyjsciowy.write(datetime.datetime.now().strftime("%Y/%m/%d,
%H:%M:%S")+ "\n")
    plik_wyjsciowy.write(f"{iloscRuchowGracza}")
    sys.exit(0)

drukujPlansze(plansza, ktoryRuch)

```

- Strona.py

```
table = " "
k = 0

plik = open("wynik.txt","r")
przeczytany = plik.read().split("\n")

html = open("KolkoKrzyzyk.html","w")

for line in przeczytany:
    if (k%3==0):
        table += "<tr>"
        table += "<td>" + line
        k+=1

top = "<html>
<center>
<style>
table {width: 40%}
</style>
<head> <h1> STATYSTYKA KÓŁKO I KRZYŻYK ~Tatiana Cieślár </h1> </head>
<body>
<table border='1' align='center'>
    <thead align='center'>
        <tr> <th> Kto wygrał? <th> Data <th> Ilość ruchów Gracza
    <tbody align='center'>
"

bottom = "
</table>
</body>
</center>
</html>
"

html.write(top + table + bottom)
plik.close()
html.close()
```

- menu.bat

```
@echo off

goto MENU
:MENU
cls
echo -----MENU-----
echo                1. Start programu
echo                2. Backup
echo                3. Informacje o projekcie
echo                4. Wyjscie
echo -----

set /p choice="Wpisz 1/2/3/4: "
If %choice%==1 goto START
If %choice%==2 goto BACKUP
If %choice%==3 goto INFO
If %choice%==4 exit

:START
cls
KolkoKrzyzyk.py
Strona.py
KolkoKrzyzyk.html
pause
goto MENU

:BACKUP
set date = date/t
set hour=%TIME:~0,2%
if "%hour:~0,1%" == " " set hour=0%hour:~1,1%
set minute=%TIME:~3,2%
set czas=%hour%.%minute%
set name="%date%-%czas%"

cd../
mkdir "backup"
copy projekt\KolkoKrzyzyk.html backup\raport_%name%.html

cls
echo -----BACKUP-----
echo.
echo        Wykonany backup z data %name%
echo.
echo -----
```

```

cd projekt
set /p powrot="Wcisnij ENTER aby powrocic do MENU..."
goto MENU

:INFO
cls
echo -----INFORMACJE O PROJEKCIE-----
echo.
echo      Autorka projektu: Tatiana Cieslar
echo      Tytul projektu: Kolko i krzyzyk
echo.
echo      Program gra z uzytkownikiem w klasyczne
echo      dziewieciopolowe kolko i krzyzyk. Uzytkownik
echo      zawsze rozpoczyna gre, jednak nigdy nie
echo      wygrywa - komputer bowiem grajac "rozsadnie"
echo      doprowadza do remisu lub swojej wygranej.
echo      Jest to program zrealizowany na podstawie
echo      zadania drugiego z konkursu Algorytmion 2015.
echo.
echo -----

set /p powrot="Wcisnij ENTER aby powrocic do MENU..."
goto MENU

```