**Tenis**

#### **U devetom smo poglavlju simulirali jedan gem tenisa. Sada prilažemo program koji će simulirati kompletnu igru tenisa do 2 ili tri dobivena seta. Set se igra do 6 dobivenih gemova pod uvjetom da je razlika dobivenih gemova veća od 1, odnosno do 7 ako je jedan od igrača poslije rezultata 6:5 dobio sedmi set. Ako je rezultat bio 6:6, igra se sedma igra ili Tie-break.**

Tijekom tie-breaka poeni se broje 0, 1, 2, 3, itd. Prvi igrač koji dobije sedam poena dobiva Gem i Set ukoliko njegov protivnik ima barem dva poena manje. Ukoliko to nije slučaj, tie-break se nastavlja sve dok se ne postigne razlika od dva poena između igrača.

Igrač koji je na redu za servirati servira prvi poen tie-breaka. Sljedeća dva poena servira njegov protivnik. Nakon toga, igrači naizmjenično serviraju po dva poena do kraja tie-breaka.

**🖫Tenis.py**

# Tenis\_GUI.py

# Igraju X (A) i Y (B). Rang igrača je

# vjerojatnost dobivanja poena.

**from** turtle **import** \*

**from** random **import** \*; rnd = random

P = { 0: '0 ', 1: '15', 2: '30', 3: '40',

4: 'A '}

X = Y = A = B = a = b = 0

d = 0.05 # prednost na servisu RX +d

# ili RX -d ako servira igrač Y

S = 0 # odigranih setova

Tie = **False**; Mach = **True**; End = **False**

Ix = 'X'; Iy = 'Y' # Igraju X i Y

title ("TENIS"); setup (600,200, 0,0); ht ()

**def** Semafor (x,a, T1,T2, c0,c1, O = "black"):

tracer (0)

begin\_fill(); width(2); color (O)

pu (); setpos (x, 0); pd()

**for** i **in** range(2):

fd(a); lt(90); fd(60); lt(90)

fillcolor (c0); end\_fill()

**if** x == -120 : x = -135

color (c1); font = ("Consolas", 20,

"bold")

pu(); setpos (x+20, 30); pd()

write (T1, **False**, "left", font)

pu(); setpos (x+20, 0); pd()

write (T2, **False**, "left", font); pu()

Semafor (-250, 130, Ix, Iy, "white",

"black")

Semafor (-120, 20, '\*', ' ', "white",

"orange")

Semafor (-100, 50, '0', '0', "green",

"white")

Semafor ( -50, 50, '0', '0', "dark violet",

"white")

Semafor ( 0, 50, '0', '0', "white",

"black")

**def** Igra ():

**global** X, Y, x, A, B, a, b, d, S, Tie, \

Mach, End

**if not** Mach:

**if not** End : End = **True**

**return**

**def** Set () :

**global** A, B, X, Y, S, Mach

**if** A > B : X += 1

**else** : Y += 1

S += 1

Mach = X != N **and** Y != N

Semafor (-100, 50, str(X), str(Y),

"green", "white")

Semafor (-100 +S\*50, 50, str(A), str(B),

"blue", "white")

**if not** Mach :

Semafor (-120, 20, ' ', ' ',

"white", "white")

Semafor (-48 +S\*50, 100, ' ', ' ',

"white", "white", "white")

**return**

Semafor ( -50 +S\*50, 50, '0', '0',

"dark violet", "white")

Semafor ( 0 +S\*50, 50, '0', '0',

"white", "black")

**def** Servis (d) : # promjena serviranja

d1 = '\*' \*(d > 0); d2 = '\*' \*(d < 0)

Semafor (-120, 20, d1, d2, "white",

"orange")

p = rnd(); q = 'X' \*(d>0) +'Y'\*(d<0)

**if** p < RX +d : a += 1; Q = 'X'

**else** : b += 1; Q = 'Y'

**if** (**not** Tie **and** ((a >= 4 or b >= 4) **and**

abs (a-b) > 1)) \

**or** (Tie **and** ((a >= 7 **or** b >= 7) **and**

abs (a-b) > 1)) :

A += 1\*(a > b); B += 1\*(b > a)

a = b = 0;

**if** Tie **and** abs(A-B) == 1 **or** (**not** Tie

**and** ((A >= 6 **or** B >= 6) **and**

abs (A-B) > 1)) :

Set ()

**if** **not** Mach : **return**

A = B = 0

**if not** Tie :

Semafor (-50 +S\*50, 50, str(A),

str(B), "dark violet", "white")

Semafor (0 +S\*50, 50, '0', '0',

"white", "black")

d = -d; Servis (d)

Tie = **False**

**if not** Tie **and** (A == B == 6) :

Tie = **True**;

a = b = 0 # tie break

**if not** Tie :

**if** a == b == 4 : a = b = 3

Semafor (S\*50, 50, P[a], P[b], "white",

"black")

**else** :

Semafor (S\*50, 50, a, b, "white",

"red")

**if** (a+b) % 2 : d = -d; Servis(d)

# Poen se dobiva poslije "serviranja",

# s "Enter".

onkey (Igra, "Return")

**while True** :

N = int (textinput ("TENIS",

'Igra se do 2 ili 3'))

**if** N in {2, 3} : **break**

# RX rang prvog igrača od 0.2 do 0.8. Drugi igrač ima ranga 1 -RX

**while True** :

RX = float (textinput ("TENIS",

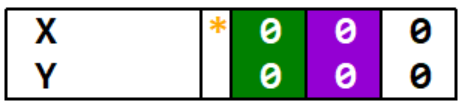
'Rang prvog igrača (0.2 - 0.8) '))

**if** 0.2 <= RX <= 0.8 : **break**

listen(); mainloop()

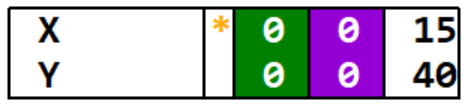
Poslije zadavanja do koliko se dobivenih setova igra ( 2 ili 3) i zadavanja „ranga“ Rx igrača X (rang igrača Y je 1 -Rx) evo nekoliko poena u prvom gemu prvog

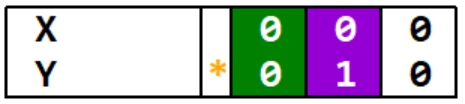
seta:



Graphical user interface

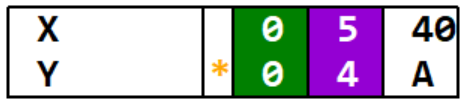
Description automatically generated with low confidence

****

****

****

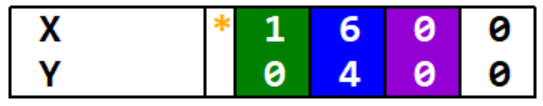
Stanje u 10-tom gemu, igrač Y ima prednost:

****

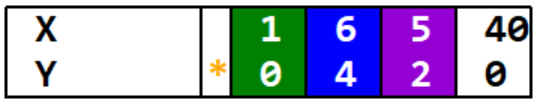
Igrač X je izjednačio i ima prednost:

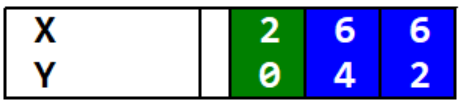
****

Igrač X je iskoristio gem-set loptu i poveo s 1:0:

****

Pogledajmo stanje u osmom gemu drugog seta. Igrač X ima tri set lopte (na servis igrača Y)!:

****

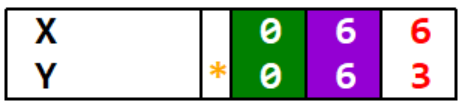
****

Pobjeda igrača X u dva dobivena seta sa 6:4 i 6:2!

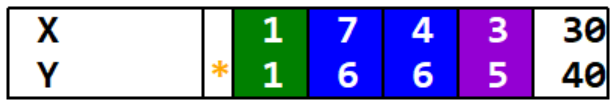
U drugom „polufinalnom“ meču igrači su izjedna-čeni i poslije reziltata 6:6 u prvom setu, igraju tie-break:



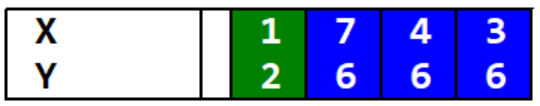
Igrač X ima tri set lopte:



i dobiva prvi set poslije treće lopte. Igrač Y je dobio drugi set i igra se deveti gem trećeg seta:



Igrač Y je pobijedio sa rezultatom 7:6, 4:6 i 3:6!



**MasterMind**

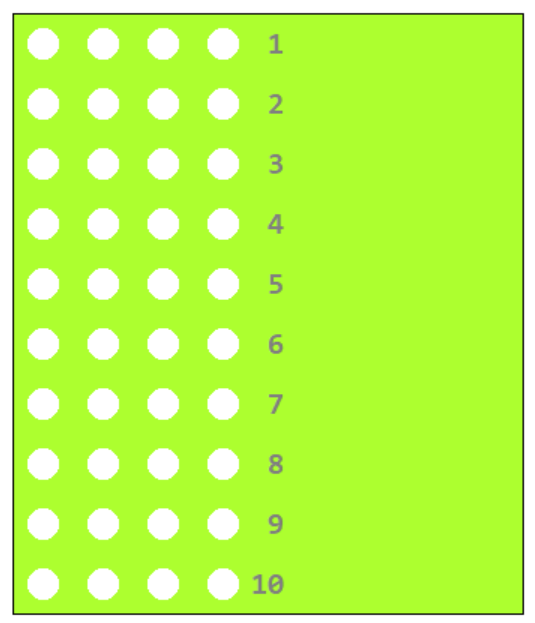


Mastermind („glavni um“) igra je pogađanja kombinacija za dva igrača. Modernu igru s čiodama izmislio je, 1970. godine, Morcedal Meirowitz, upravnik pošte i ekspert u oblasti telekomunikacija. Treba pogoditi boju i mjesto čioda koje postavlja prvi igrač, a nevidljivo je za drugog igrača. Ima ukupno šest boja. Označili smo ih brojkama 0 do 5:

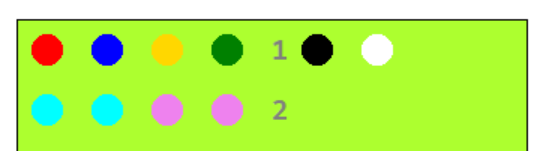


Boje se mogu ponavljati (npr. sve četiri jednake ili dvije jednake, a dvije različite itd.). Drugi igrač postavlja u rupice redom po četiri čiode u izabranim bojama. Poslije postavke prvi igrač mu daje obavijest koliko ima pogođenih i na pravom mjestu, a koliko pogođenih boja, ali na pogreš-nom mjestu. Za to se koriste crna čioda za pravo mjesto i bijela za pogrešno.

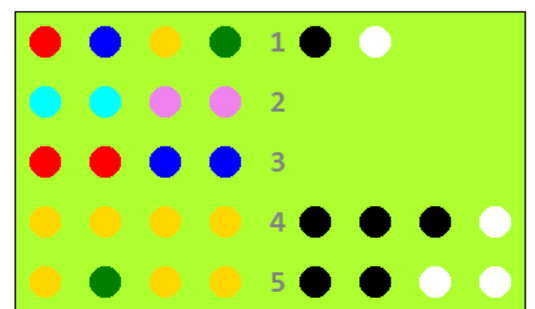
Na primjer, prvi igrač je postavio jednu zelenu i tri žute čiode, nevidljive za drugog igrača.



Drugi igrač je postavio redom boje 0 1 2 3 i dobio odgovor jedna crna (boja na pravom mjestu) i jedna bijela, a poslije 4 4 5 5 nema pogođene boje.



Slijedi zaključak da su postavljene dvije boje, po dvije od svake ili tri od jedne i jedna od druge. Poslije provjere boja 0 i 1 zaključujemo da su u igri boje 2 i 3. Pitamo se koliko je boja 2 pa u redu 4 postavljamo četiri takve boje. Iz odgovora da su 3 na pravom mjestu zaključujemo da tražimo tri boje 2 (žute) i jednu 3 (zelena). Postavljamo četiri žute i poslije odgovora da su 3 na pravom mjestu i iz reda 1 zaključujemo da su to treće i četvrto mjesto, a treća je na prvom ili drugom mjestu. Iz petog reda zaključujemo da je na drugom mjestu, a zelena na prvom i pogađamo!





**🖫Mastermind.py**

# Mastermind.py

**from**  turtle **import** \*

**from**  random **import** \*

**global** I, J, Z, B, END, b, T, O, Q

# parametri

setup (500, 1000, 0, 0)

d = 40

X0, Y0 = -4.5\*d, 6\*d

x0, y0 = X0 -d/2, 4\*d +d/2

M = 11; N = 4

x, y = x0 +d , Y0 -d/2

Y = [y -d\*i **for** i **in** range (M)]

X = [x +d\*i **for** i **in** range (N)]

b = ('red', 'blue', 'gold', 'green',

'cyan', 'violet')

**def** Prikaz (I, J, k) :

color (b[k])

goto (X[I], Y[J]); stamp()

**def** \_Y () :

**for** i **in** range(4): Prikaz (i, 0, Z[i])

**def** \_X () :

**for** i **in** range(4): color ('white'); \

goto (X[i], Y[0]); stamp()

**def** Write (x, y, s, c = 'black', f = 20) :

pu(); goto (x, y); color (c); pd()

write(s, **False**, "left", ("Consolas",

f, "bold"))

pu()

**def** Igra (k) :

**global** I, J, B, Z, END, b, T, O, Q

**if** END : **return**

**if** I < 4 :

B.append (k)

shape ('circle'); shapesize (1, 1)

Prikaz (I, J, k)

I += 1

**if** I > 3 :

**for** i **in** range (4) :

y = B[i]

if y == Z[i] : T += 1

**elif** y in Z : O += 1

P = [] +['black']\*T +['white']\*O

**for** i **in** range (len (P)) :

color (P[i])

goto (X[3] +d/2 +d\*(i+1), Y[J]);

stamp()

**if** T == 4 :

Write (X[3] +d, Y[0]-d/2,

'BRAVO!', 'red', 30)

**for** i **in** range(4) : Prikaz(i, 0, Z[i])

END = **True**

Q += 1

**if** Q == 10 :

\_Y ()

Write (X[0] -2\*d, Y[0] +d,

'Nisi pogodi(o/la) iz 10 '

'pokušaja!', 'red', 18)

END = **True**

I = 0; J += 1; B = []; T = O = 0

**def** Ploča () :

pu(); goto (4\*d, Y0-d); pd()

tracer(0); fillcolor ("green yellow")

begin\_fill()

**for** \_ **in** range(2) :

rt(90); fd ( 10 \*d); rt(90); fd (8.5 \*d)

end\_fill()

pu (); goto (d, Y0)

color ('white'); pu()

shape ('circle'); shapesize (1, 1)

**for** i **in** range (N) :

**for** j **in** range (M) :

goto (X[i], Y[j]); stamp()

up(); color ('black')

x = X0 +4\*d; y = Y0-1.8\*d

**for** j **in** range (1, 11):

Write(x, y, "%2s" % str(j), 'gray', 15)

y -= d

pu (); x = X0 +2\*d

shape ('circle'); shapesize (1, 1)

**for** i **in** range (6) :

color (b[i])

goto (x +i\*d, -5.5\*d); stamp()

Write (x -5 +i\*d, -6.5\*d, str(i))

up()

Proc = """

**def** \_C () :

**global** I, J, B, Z, END, b, T, O

Igra (C)

onkey (\_C, "C")

"""

**for** C **in** '012345' :

proc = Proc.replace ('C', C)

exec (proc)

onkey (\_Y, 'Y'); onkey (\_X, 'X')

END = **False**

I = 0; J = 1; B = []; T = O = Q = 0

pu (); ht(); Ploča ()

Ok = textinput ("MASTERMIND", 'Zadaješ boje'

' (d/n)')

**if** Ok[0].upper() == 'D' :

Z = eval (textinput

("MASTERMIND",

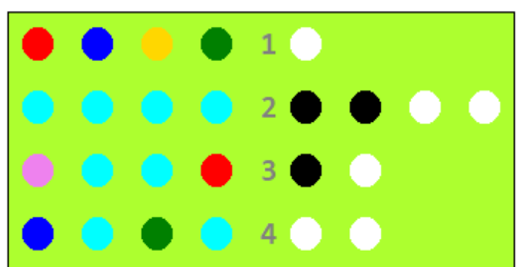
'Zadaj boje (b1, b2, b3, b4)'))

**else** : Z = [randint (0, 5)

**for** i **in** range (4)]

listen(); mainloop()

Na kraju, analizirajmo sljedeći primjer:



Da li se iz prva četiri reda moglo jednoznačno zaključiti koje je rješenje (peti red)?

Background pattern

Description automatically generated