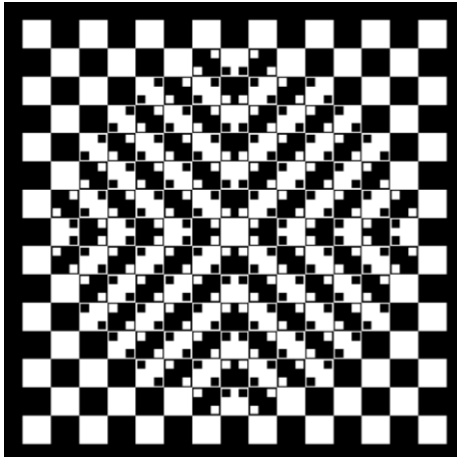


# Optičke iluzije, 3.c

Iluzija Šahovske ploče .....	2
Crtanje fraktala .....	4
Iluzija crvene i bijele boje.....	5
Penroseov trokut.....	6
Optička iluzija .....	8
Nepostojeći likovi .....	9
Iluzija perspektive .....	10
Šetajući krugovi.....	11
Iluzija.....	13
Optička iluzija .....	14
Optička iluzija – Petra Koceić Bilan .....	13
Optička iluzija – Mihaela Burić.....	14

## Iluzija Šahovske ploče

Igor Skopljak  
**Iluzija**



Ova iluzija se sastoji od crno bijele šahovske ploče i malih crno bijelih kvadratića koji su postavljeni u sredinu ploče. Kada se te dvije stvari spoje nastaje prividna slika koja izgleda kao su linije u sredini ploče zakrivljene.

### Kod

#### Crtanje početnog kvadrata:

```
def draw_square(x,y,size,c):  
    turtle.up()  
    turtle.goto(x-size/2,y-size/2)  
    turtle.seth(0)  
    turtle.color(c)  
    turtle.begin_fill()  
    for _ in range(4):  
        turtle.fd(size)  
        turtle.left(90)  
    turtle.end_fill()
```

#### Crtanje kvadratne ploče:

```
def draw_board():  
    for x in range(-7,8,2):  
        for y in range(-7,8,2):  
            draw_square(x,y,1,color1)  
    for x in range(-6,8,2):  
        for y in range(-6,8,2):  
            draw_square(x,y,1,color1)
```

```

def draw_straight(x,y):
    c = color2 if (x+y)%2 == 0 else color1
    if y>0:
        draw_square(x-0.3,y-0.3,0.3,c)
        draw_square(x+0.3,y-0.3,0.3,c)
    elif y<0:
        draw_square(x-0.3,y+0.3,0.3,c)
        draw_square(x+0.3,y+0.3,0.3,c)
    elif x>0:
        draw_square(x-0.3,y-0.3,0.3,c)
        draw_square(x-0.3,y+0.3,0.3,c)
    elif x<0:
        draw_square(x+0.3,y-0.3,0.3,c)
        draw_square(x+0.3,y+0.3,0.3,c)

```

### Crtanje malih kvadratića:

```

def draw_diag(x,y):
    c = color2 if (x+y)%2 == 0 else color1
    if x*y > 0:
        draw_square(x-0.3,y+0.3,0.3,c)
        draw_square(x+0.3,y-0.3,0.3,c)
    elif x*y < 0:
        draw_square(x+0.3,y+0.3,0.3,c)
        draw_square(x-0.3,y-0.3,0.3,c)

```

```

def draw_bulge():
    for x in range(-6,7):
        for y in range(-6,7):
            if abs(x)+abs(y)<=7:
                draw_diag(x,y)
                if x==0 or y==0: draw_straight(x,y)
    x,y = -5,-3

```

```

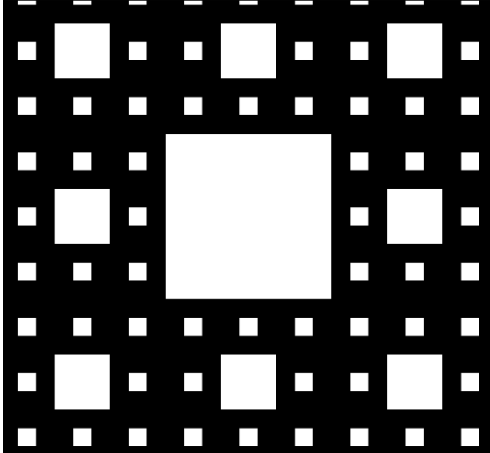
for i in range(3):
    draw_diag(x,y)
    draw_diag(-x,-y)
    draw_diag(x,-y)
    draw_diag(-x,y)
    x += 1
    y -= 1

```

# Crtanje fraktala

Ivan Tomáš

## Iluzija



## Kod

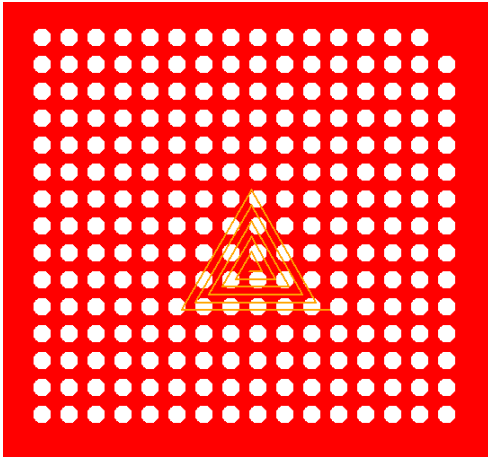
```
def kvadrat(c,a): —————> c-Određuje broj rekurzivnih crtanje kvadrat  
if(c==0): a-Određuje stranicu ukupnog fraktala
```

```
    begin_fill() }  
    fillcolor("black") }  
    for i in range(4): } Crtanje pojedinih kvadrata  
        fd(a) }  
        left(90) }  
    end_fill() }  
else: }  
    for i in range(4): }  
        kvadrat(c-1,a/3) }  
        fd(a/3) }  
        kvadrat(c-1,a/3) } pozivanje funkcije za crtanje pojenih kvadrat  
        fd(a/3) }  
        fd(a/3) }  
        lt(90) }
```

## Iluzija crvene i bijele boje

Ivan Lozić

### Iluzija



Ova optička iluzija ukazuje na to kako receptori u ljudskom oku drugačije percipiraju boje ispred sebe. Na primjer, ako dulje vrijeme gledamo u ovu sliku na bijeloj pozadini nakon 20 sekundi vidimo crveno i obratno. Tamo gdje je bijela boja bila na krugovima sada se nalazi u pozadini.

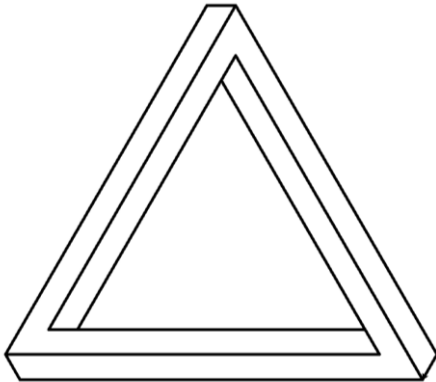
### Kod

```
pocx=0
pocy=0
m=15
n=16
*od trokutaste spirale
u=numinput("Ulaz", "Broj duljina
stranice(5,100):",50,5,100)
t=numinput("Ulaz", "Broj
ponavljanja(5,50):",20,5,100)
pu()
goto(-20,159)
pd()
for i in range (int(n)):
    color("orange")
    pensize(2)
    fd(m)
    lt(120)
    m=m+10
ht()
mainloop()

*pozadine
from turtle import*
speed(1000)
pensize(5)
color('red')
begin_fill()
goto(0,-50)
for i in range(4):
    fd(250)
    lt(90)
    fd(300)
end_fill()
*stupova i redova(dio koda)
goto(-270,0)
dd=30
```

## Penroseov trokut

Mihovil Parčina  
**Iluzija**



Moja optička iluzija vrlo je jednostavna i sastoji se od crtanja Penroseovog trokuta. Penroseov trokut je nemoguć trokut koji nema jasno definiranu plohu te, iako ga se može percipirati, on u praksi ne može postojati. Ime je dobio po matematičaru Rogeru Penroseu koji je dao detaljne opise njegove „nemogućnosti“.

### Kod

```
from turtle import*
speed(10000)
pensize(5)
pu()
goto(0,216)
pd()
for i in range(3): #petlja koja crta početni trokut
    rt(60)
    fd(500)
    rt(60)

lt(60)
fd(50)
rt(120)
fd(650)
rt(120)
fd(700)
rt(60)
fd(50)
```

rt(120)  
fd(650)  
lt(120)  
fd(50)  
pu()  
lt(180)  
fd(50)  
rt(120)  
fd(650)  
pd()  
rt(120)  
fd(700)  
rt(60)  
fd(50)  
rt(120)

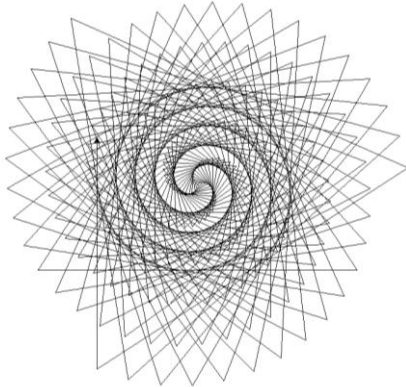
fd(650)  
rt(60)  
lt(180)  
fd(50)  
lt(180)  
fd(50)  
rt(120)  
fd(650)  
rt(120)  
fd(700)  
rt(60)

fd(50) #sve navedene naredbe dodatno  
opisuju početni trokut

#početni trokut zajedno s opisanim dijelovima  
daje finalan oblika

## Optička iluzija

Bruna Panžić  
**Iluzija**



>> Program koristi kornjačinu grafiku kako bi nacrtao 3 prividne spirale od izlomljenih linija.

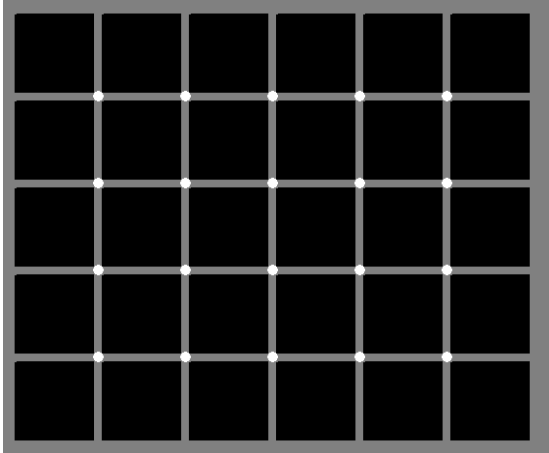
### Kod

```
from turtle import* #uvodimo kornjacinu grafiku
def iluzija(n): #definiranje funkcije koju cemo pozvati kasnije
    if n>0:
        iluzija(n-3) #svaki put je sve manja
        fd(n) #uputa za crtanje naprijed
        rt(123) #uputa za skretanje u desno
    print('unesite broj')
print('N.B. preporučene vrijednosti su višekratnici broja 100, ali ne veći od 800')
a=int(input()) #korisniku dajemo mogućnost unosa vrijednosti
iluzija(a) #pozivanje funkcije iluzija
```



## Nepostojeći likovi

Tonći Kaselj  
**Iluzija**



Gledajući u nacrtani geometrijsku strukturu, uočit ćemo pojavu crnih točaka koje zapravo ne postoje.

### Kod

```
from turtle import*    #Uvodimo kornjačinu grafiku
```

```
speed(1000)
```

```
def prvi_red():
```

```
    dot(11.35,'white')
```

```
    pu()
```

```
    goto(pocz+z*dd,pocw+w*dd)
```

```
    pd()
```

```
    clear()
```

```
    goto()
```

```
def kvadrat():
```

```
    pensize(2)
```

```
    fillcolor('black')
```

```
    begin_fill()
```

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

```
        fd(80)
```

```
        rt(90)
```

```
    end_fill()
```

```
bgcolor('grey')
```

```
title('Prvi')
```

```
dd=90
```

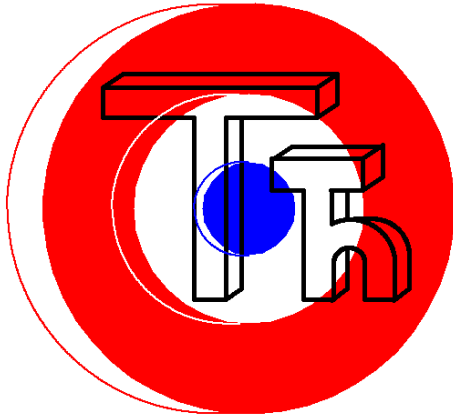
#Funkcija koja crta stupac bijelih  
krugova

#Funkcija koja crta kvadrat i boja ga

# Iluzija perspektive

Mario Rubić

## Iluzija



Iluzija se sastoji od dva objekta. Prvi dio su slova T i h koja su, iako nacrtana na 2D podlozi, prikazana u 3D prostoru na način da je promatrač gleda s desne strane. Drugi dio je crveno-bijelo-plavi krug koji je također prikazan u 3D prostoru, ali na način da ga promatramo s lijeve strane. S obzirom da se ta dva dijela preklapaju i da imaju isto središte odakle su nacrtani, nemoguće je odrediti s koje strane (lijeve ili desne) promatrač stoji ispred objekata.

### Kod (dio koda kojim se crtaju slova T i h)

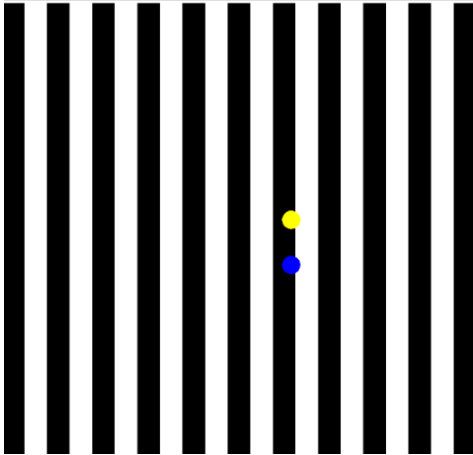
```
pendown()          fd(100)          lt(90)          pu()
pensize(5)         rt(90)          fd(33)         rt(180)
lt(90)             fd(100)        lt(90)         fd(100)
fd(100)           lt(90)         fd(40)         rt(90)
rt(90)            pd()           for i in range(180):
fd(100)           fd(33)         rt(1)         fd(35)
lt(35)            rt(55)         fd(0.3)       rt(90)
fd(27)            fd(27)         fd(40)        fd(17)
lt(55)            pu()           lt(90)        lt(90)
fd(33)            lt(180)        fd(90)        pd()
lt(90)            fd(27)         fd(33)        fd(31)
fd(235)           rt(35)         lt(90)        pu()
lt(35)            pd()           fd(40)        fd(32)
fd(27)            fd(235)        for i in range(115):
lt(55)            pu()           lt(1)         pd()
fd(33)            home()         fd(0.85)      fd(45)
lt(90)            lt(90)         rt(115)       rt(53)
fd(100)           fd(20)         fd(37)        fd(21)
rt(90)            rt(90)         penup()       rt(37)
fd(200)           fd(115)        home()        pu()
lt(90)            pendown()      speed(500)    rt(180)
fd(35)            fd(35)         lt(90)        fd(68)
lt(35)            lt(35)         fd(20)        pendown()
fd(21)            fd(27)         rt(90)        lt(37)
lt(55)            lt(55)         fd(115)       fd(21)
fd(188)           fd(33)         fd(35)        lt(53)
penup()           lt(90)         lt(90)        fd(37)
lt(180)           fd(100)        pendown()     for i in range(90):
fd(188)           lt(35)         fd(35)        lt(1)
rt(55)            fd(27)         rt(55)        fd(0.70)
fd(21)            lt(55)         fd(27)        fd(20)
rt(35)            fd(33)         penup()       pu()
rt(90)            lt(90)         rt(180)       home()
pendown()         fd(35)         fd(27)        pensize(2)
fd(100)           rt(90)         rt(35)        speed(1000)
pu()              fd(120)        pendown()

```

# Šetajući krugovi

Mislav Zelić

## Iluzija



Kod crta stupce (crne i bijele boje) i 2 kruga (plave i žute boje) koje se kreću prema desno.

Iluzija nastaje pri prijelazu iz jednog u drugi stupac gdje nam priviđa da krugovi usporavaju kad prelaze preko crnog i ubrzavaju kada prelaze preko bijelog stupca.

## Kod

### CRTANJE STUPACA

```
def stupci():  
    boje=["black", "white"]  
    for i in boje:  
        color(i)  
        begin_fill()  
        for j in range(2):  
            fd(50)  
            lt(90)  
            fd(1000)  
            lt(90)  
        end_fill()  
        pu()  
        fd(50)  
        pd()  
    return()  
mainloop()  
for k in range(10):  
    stupci()
```

- Stvara 2 pravokutnika, crni i bijeli
- 2 puta se ponavlja stvaranje pravokutnika, svaki put se mijenja boja
- Stvara pravokutnik i završava punjenje bojom
- Poziva crtanje stupaca

### DEFINIRANJE KORNJAČA

```
krug1 = turtle.Turtle()  
krug1.color('yellow')  
krug1.hideturtle()  
krug1.penup()  
krug1.goto(-500, 0)  
krug1.pendown()
```

- definiranje prve kornjače kao žuti krug
- postavlja je na početni položaj

### KRETANJE KORNJAČA

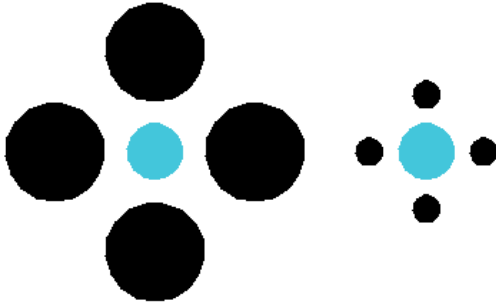
```
def moving_object1(krug1): - definiranje kretanja kornjače
    krug1.fillcolor("yellow")
    krug1.begin_fill()
    krug1.circle(20) - kada se pozove, kornjača ponovno postaje žuti krug
    krug1.end_fill()
```

### POKRETANJE KRETANJA KORNJAČA

```
while True : - izvršava kretnju kornjače
    krug1.clear() - kornjača se briše i ponovno stvara
    moving_object1(krug1)
    screen.update()
    krug1.forward(0.2) - kreće se unaprijed i ponavlja radnju
    - stvara privid horizontalnog kretanja kornjače
```

# Iluzija

Petra Koceić-Bilan  
Iluzija



## Kod

```
import turtle
def krug(x, y, radius, color): #Kada pozovemo funkciju, odredit ćemo varijable koje se u njoj koriste. To
su koordinate središta(x i y), radijus kruga kojicrtamo i boja kruga.
    global myPen
    myPen.setheading(0)
    myPen.penup() #olovka se podiže
    myPen.color(color)
    myPen.fillcolor(color)
    myPen.goto(x,y-radius) #šaljemo olovku na zadane koordinate središta
    myPen.begin_fill() #olovka započinje ispunjavanje bojom
    myPen.circle(radius) #olovka crta krug zadanog radijusa
    myPen.end_fill() #olovka završava ispunjavanje bojom

myPen = turtle.Turtle()
myPen.shape("arrow")
myPen.speed(500)

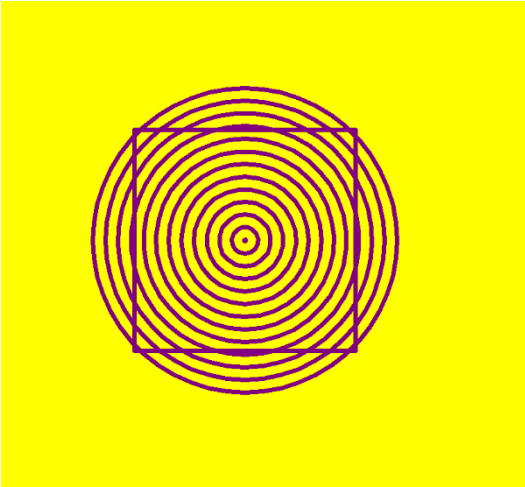
krug(-90, 0, 20 , "#43C6DB") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(-90, 70, 35, "black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(-90, -70, 35, "black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(-160, 0, 35, "black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(-20, 0, 35, "black") #pozivamo funkciju određujući koordinate, radijus i boju krugova

krug(100,0,20,"#43C6DB") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(60,0,10, "black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(140,0,10,"black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(100,40,10,"black") #pozivamo funkciju određujući koordinate, radijus i boju krugova
krug(100,-40,10,"black") #pozivamo funkciju određujući koordinate, radijus i boju krugova

myPen.hideturtle() #olovka prestaje biti vidljiva
```

## Optička iluzija

Mihaela Burić  
Iluzija



**Kod**

```
#optička iluzija
from turtle import*
import random
from math import*
speed(10000000)
pensize(7)
c=0
d=1

def program():
    b=20
    a=350
    pu()
    rt(135)
    fd(a/2*sqrt(2))
    pd()
    rt(225)

for i in range(3):
    if c==1:
        pencolor("purple")
        bgcolor("yellow")
    elif c==2:
        pencolor("green")
        bgcolor("red")
    elif c==3:
        pencolor("orange")
        bgcolor("blue")
for i in range (4):
    fd(a)
```

```
    rt(270)
    pu()
    goto(0,0)
    for i in range(1,250,20):
        rt(90)
        fd(i)
        rt(270)
        pd()
        circle(i)
        pu()
        #color[boje_krug]
        home()

#boje_krug=["purple","green","orange"]
#boje_bg=["yellow","red","blue"]
#for i in range(boje_krug):
    #i=rand(input(boje_krug))

#pencolor(random(boje_krug))
#bgcolor(random(boje_bg))

while c!=d:
    c=random.randint(1,3)
    d=random.randint(1,3)
    #print(c)
    #print(d)
if c==d:
    program()

ht()
mainloop()
```