

1. uzdevums

Sastādīt **Ziemassvētku** kartiņu.

Kods:

```
# Programmas nosaukums: 1. uzd. MPR16
```

```
# 1. uzdevums MPR16
```

```
# Uzdevuma formulējums: Sastādīt Ziemassvētku kartiņu
```

```
# Versija 11.0
```

```
"""
```

```
globalo mainīgo saraksts šajā programmā:
```

```
slot1, slot2, slot3, slot4, slot5, slot6, slot7, slot8
```

```
slot_on
```

```
metelis_on, metelis_left_side, metelis_right_side
```

```
cilindrs, cilindra_lente
```

```
j, spainis
```

```
indicator_count, i
```

Lai programma strādātu korekti, ir nepieciešams, lai mapītē ar programmu būtu atrodāmi šie attēli:

```
myicon.png
```

```
snowman_smile_to_front.png
```

```
snowman_smile_to_right.png
```

```
dog_house.png
```

```
dog_revealed.png
```

```
red_star.png
```

```
golden_star.png
```

```
"""
```

```
import tkinter
```

```
import random
```

```
# ===== Zīmēšanas rīki =====
```

```
def oval(x0,y0, x1,y1, color): # Viegļāk izveidot ovalus, pēc koordinātam x0,y0, x1,y1 un ar noteiktu krāsu (color)
```

```
    kanva.create_oval(x0, y0, x1, y1, fill = color, outline = color)
```

```
def draw_circle(x0,y0, r, color): # Viegļāk veidot aplus. x0,y0 - apliņa centrs. r - rādiuss. color - krāsa
```

```
    kanva.create_oval(x0-r, y0-r, x0+r, y0+r, fill = color, outline = color)
```

```
def draw_square(x0,y0, r, color): # Viegļāk zīmēt kvadrātus
```

```
    kanva.create_rectangle(x0-r, y0-r, x0+r, y0+r, fill = color, outline = color)
```

```
def create_star_with_n_color(color): # Izveidot zvaigzni
```

```
    kanva.create_polygon(550, 200, 520, 180, 580, 180, fill = color, outline = color)
```

```
    kanva.create_polygon(540, 180, 550, 160, 560, 180, fill = color, outline = color)
```

```
    kanva.create_polygon(550, 200, 560, 190, 565, 210, fill = color, outline = color)
```

```
    kanva.create_polygon(550, 200, 540, 190, 530, 210, fill = color, outline = color)
```

```
# ===== Statiskie objekti =====
```

```
def balta_zeme(): # izveido balto zemi
```

```
    balta_zeme = kanva.create_rectangle(0,1000,1100,550, fill = "white")
```

```
def egle(): # statiska egle bez rotājumiem
```

```
    kanva.create_rectangle(500, 560, 590, 590, fill = "#8B4513", outline = "#8B4513") # egles stumbrs (visszēmāka egles daļa)
```

```
    kanva.create_polygon(448, 380, 550, 200, 640, 380, fill = "#385926", outline = "#385926") # augšējā zaļā egles daļa (augšējais trijstūris)
```

```
    kanva.create_polygon(435, 480, 550, 300, 655, 480, fill = "#4a5f29", outline = "#4a5f29") # vidēja zaļā egles daļa (vidējais trijstūris)
```

```
    kanva.create_polygon(390, 570, 550, 390, 710, 570, fill = "#556B2F", outline = "#556B2F") # zemākā zaļā egles daļa (augšējais trijstūris)
```

```

def dāvanas(): # divas dāvanas un lente

    # kreisā violēta dāvana
    kanva.create_rectangle(260, 500, 340, 590, fill = "#800080", outline = "#800080")

    # kreisās violētas dāvanas lenta
    kanva.create_rectangle(290, 500, 310, 590, fill = "red", outline = "red")
    kanva.create_rectangle(260, 540, 340, 560, fill = "#FF4500", outline = "#FF4500")

    # dāvanas no labās puses
    kanva.create_rectangle(660, 500, 750, 590, fill = "#FF69B4", outline="#FF69B4")

    # dāvanas lenta
    kanva.create_rectangle(690, 500, 710, 590, fill = "#8B0000", outline="#8B0000")

def meness(): # pusmēness
    oval(800,50, 900,150, "white") # Baltais mēness
    oval(830,30, 950,140, "#01112B") # Lai būtu pusmeness, pārklājam to ar citu apli ar dēbess krāsu
    (lai veidotos pusmēness)

def rotājumi(): # egles rotājumi (statiskas bumbas)

    draw_circle(548,300, 7, "red") # 1.egles daļas sarkanā bumba (centrā)
    draw_circle(500,350, 7, "purple") # 1.egles daļas violēta bumba (kreisā puse)
    draw_circle(600,350, 7, "yellow") # 1.egles daļas dzeltenā bumba (labā puse)

    draw_circle(550,400, 7, "yellow") # 2. egles daļas dzeltenā bumba (centrā)
    draw_circle(500,450, 7, "blue") # 2. egles daļas zila bumba (kreisā puse)
    draw_circle(600,450, 7, "red") # 2. egles daļas sarkana bumba (labā puse)

    draw_circle(550,500, 7, "purple") # 3. egles daļas violēta bumba (centrā)
    draw_circle(500,550, 7, "yellow") # 3. egles daļas dzeltena bumba (kreisā puse)
    draw_circle(600,550, 7, "red") # 3. egles daļas sarkana bumba (labā puse)

def sniegavirs(): # statisks sniegavīrs
    draw_circle(160,510, 90, "#CECECE") # zemāka sniegavīra daļa

```

```

draw_circle(160,420, 65, "#BEBEBE") # vidēja sniegavīra daļa
draw_circle(160,340, 50, "#B4B4B4") # augšēja sniegavīra daļā
kanva.create_polygon(210,400, 260,520, 285,510, 220,400, fill = "brown") # "kreisā roka"
kanva.create_polygon(100,390, 20,500, 30,520, 120,400, fill = "brown") # "labā roka"

def pogas(): # sniegavīra statistikas pogas
    draw_circle(160,405, 5.6, "black") # 1.poga (augšēja)
    draw_circle(160,435, 5.6, "black") # 2.poga (vidēja)
    draw_circle(160,465, 5.6, "black") # 3.poga (zemaka)

# ===== Pēc mouse-1 nospiešanas tiek ģenerēta viena zvaigzne nejaušā vietā uz ekrāna =====

def draw_zvaigzne(x, y, r, color): # tiek izmantota, lai ar mouse-1 ģenerētu zvaigznes
    # zvaigznes figūra
    kanva.create_polygon(x, y-3*r, x+r, y-r, x+3*r, y-r, x+r, y, x+2*r, y+2*r, x, y+r, x-2*r, y+2*r, x-r, y, x-3*r, y-r, x-r, y-r, x, y-3*r, fill = color)

def generet_zvaigzni(event): # pēc mouse-1 nospiešanas, ģenerējam zvaigzni
    zvaigznes_skaits = 1 # cik zvaigznes tiek ģenērētas pēc mouse-1 nospiešanas

    for i in range(zvaigznes_skaits):
        x = int(random.randint(1,1000)) # centrs pēc x
        y = int(random.randint(1,400)) # centrs pēc y
        r = int(random.randint(1,4)) # izmērs
        color = "gold" # krāsa
        draw_zvaigzne(x,y, r, color) # tiek uzzīmēta zvaigzne

# ===== Slotas parādīšana, pēc "s" pogas nospiešanas =====

def slota_paradit(): # slotas parādīšana
    global slota1, slota2, slota3, slota4, slota5, slota6, slota7, slota8

```

```
slota1 = kanva.create_rectangle(34,350, 45,590, fill = "brown", outline = "brown")
slota2 = kanva.create_rectangle(34,350, 45,590, fill = "brown", outline = "black")
slota3 = kanva.create_rectangle(28,350, 51,380, fill = "brown", outline = "black")
slota4 = kanva.create_line(20,260, 30,350, width = 3, fill = 'brown')
slota5 = kanva.create_line(30,260, 35,350, width = 3, fill = 'brown')
slota6 = kanva.create_line(40,260, 40,350, width = 3, fill = 'brown')
slota7 = kanva.create_line(50,260, 45,350, width = 3, fill = 'brown')
slota8 = kanva.create_line(60,260, 50,350, width = 3, fill = 'brown')
```

```
def slota_delete(): # slotas nodzēšana
```

```
    kanva.delete(slota1)
    kanva.delete(slota2)
    kanva.delete(slota3)
    kanva.delete(slota4)
    kanva.delete(slota5)
    kanva.delete(slota6)
    kanva.delete(slota7)
    kanva.delete(slota8)
```

```
def paradi_vai_nonemt_slota(e): # slotas parādīšana vai noņemšana
```

```
    global slota_on
```

```
    if slota_on == False: # Ja nav slotas un nospiežām uz "s", tad parādam to
```

```
        slota_paradi()
        slota_on = True
```

```
    else:
```

```
        slota_delete() # Ja ir parādīta slota un nospiežām uz "s", tad noņemam to
        slota_on = False
```

```

# ===== Sniegavīra mētelis =====

def paradiť_vai_nonemt_metelis(e): # sniegavīra metelis, lai to izsauķu ir jāklikšķinā uz mouse-2

    global metelis_on, metelis_left_side, metelis_right_side # tiek izmantots global, lai paņēmtu
    metelis_on un metēja "vērtības" no galvēnas programmas (lai nebūtu kļūdas local variable
    'metelis_on' referenced before assignment)

    if metelis_on == False: # metelis_on == False nozīme, ka uz sniegavīra nav mēteļa

        metelis_left_side = kanva.create_polygon(100,390, 15,510, 160,390, fill = "orange")

        metelis_right_side = kanva.create_polygon(160,390, 280,510, 220,390, fill = "orange")

        metelis_on = True # tas nozīme, ka uz sniegavīra ir mētelis

    else:

        kanva.delete(metelis_left_side)

        kanva.delete(metelis_right_side)

        metelis_on = False # nav mētelis

# ===== Sniegavīra cepures animācija pēc Mouse-3 nospiešanas =====

def create_cilindrs(): # izveidot sniegavīra cilindru

    global cilindrs, cilindra_lente

    cilindrs = kanva.create_polygon(115,180, 205,180, 205,270, 235,270, 235,315, 82,315, 82,270,
    115,270, fill = "black", outline = "black")

    cilindra_lente = kanva.create_rectangle(115,270, 205,260, fill = "red", outline = "red")

def delete_cilindrs(): # Nodzēst cilindru

    kanva.delete(cilindrs)

    kanva.delete(cilindra_lente)

def izmainit_sniegavira_cepuri(e): # sniegavīrs: 1. galva bez cepures 2. galva ar spaini 3. galva ar
cilindru

    global j, spainis # j - skaitītājs. j ir nepieciešams, lai cepures mainītos secība 1. 2. 3.

    if j == 0:

```

```

delete_cilindrs()

kanva.delete(spainis)

j = j + 1 # palielinām skaitītāju par vienu


elif j == 1: # noņēmam cilindru un uzlikam (izveidojam) spaini
    delete_cilindrs()
    spainis = kanva.create_polygon(130,230, 95,315, 225,315, 185,230, fill = "#797983", outline=
"black")
    j = j + 1


elif j == 2: # noņēmam spaini un uzlikam (izveidojam) cilindru
    kanva.delete(spainis)
    create_cilindrs()
    j = j + 1


elif j == 3: # noņēmam cilindru, sniegavīrs palik ar pliku galvu
    delete_cilindrs()
    j = 1 # Lai skaitītājs atkāļ būtu viens


# -----
# ===== GALVENĀ PROGRAMMA =====
# -----


# Loga definēšana

logs = tkinter.Tk()
logs.geometry("1000x600")
logs.title("Ziemassvētku kartīte")
photo = tkinter.PhotoImage(file = "myicon.png") # zvaigznes ikona
logs.iconphoto(False, photo)

```

```
# Kanvas definēšana
```

```
kanva = tkinter.Canvas(logs, width = 1000, height = 600, bg = "#01112B")
```

```
kanva.place(x = -2, y = 0)
```

```
# BIND, kas tiek izdarīts kad nospiežām uz kreiso, vidējo un labo peles pogu
```

```
logs.bind("<Button-1>", generet_zvaigzni ) # Button-1 - kreisā peles poga. Tiek ģenerēta viena  
zvaigzne nejauša vietā
```

```
logs.bind("<Button-2>", paradit_vai_nonemt_metelis) # Button-2 - peles ritenis. Sniegavīram  
parādas metelis
```

```
logs.bind("<Button-3>", izmainit_sniegavira_cepuri) # Button-3 - labā peles poga. Mainās sniegavīra  
cepure
```

```
logs.bind("<s>", paradit_vai_nonemt_slota) # Kad nospiedam uz "s" burtu uz klaviatūras, parādas  
slota
```

```
# Sniegavīras sējas animācijas un attēli
```

```
def enter_snowman(e): # kad uzlikam kursoru virs sniegavīra sējas
```

```
    snowman_smile_to_front = tkinter.PhotoImage(file = "snowman_smile_to_front.png") # =) to  
front
```

```
    snowman["image"] = snowman_smile_to_front
```

```
    snowman.image = snowman_smile_to_front
```

```
def leave_snowman(e): # kad nolikam kursoru nost no sniegavīra sējas
```

```
    snowman_smile_to_right= tkinter.PhotoImage(file = "snowman_smile_to_right.png") # =) to right
```

```
    snowman["image"] = snowman_smile_to_right
```

```
    snowman.image = snowman_smile_to_right
```

```
snowman_smile_to_front = tkinter.PhotoImage(file = "snowman_smile_to_front.png")
```

```
snowman = tkinter.Label(logs, borderwidth = -2)
```

```
snowman.image = snowman_smile_to_front
```

```
snowman.place(x = 130, y = 320)
```



```
snowman.bind("<Enter>", enter_snowman)
snowman.bind("<Leave>", leave_snowman)
snowman["image"] = snowman_smile_to_front
```

```
# Suņa parādīšanas animācija
```

```
def enter_dog_house(e): # kad uzlikam kursoru virs suņu būdas
    dog_house = tkinter.PhotoImage(file = "dog_house.png") # suns iekšā budiņā
    house["image"] = dog_house
    house.image = dog_house

def leave_dog_house(e): # kad nolikam kursoru nost no suņu būdas
    dog_revealed = tkinter.PhotoImage(file = "dog_revealed.png") # suns parādas no būdas
    house["image"] = dog_revealed
    house.image = dog_revealed
```

```
dog_house = tkinter.PhotoImage(file = "dog_house.png") # suns iekšā budiņā
house = tkinter.Label(logs, borderwidth=-2)
house.image = dog_house
house["image"] = dog_house
house.place(x=780, y=395)
house.bind("<Enter>", enter_dog_house)
house.bind("<Leave>", leave_dog_house)
```

```
# ===== Egles dedzināšana =====
```

```
def enter_start_golden_star(e): # notikums, kad nolikam kursoru virs egles zvaigznes
    kanva.create_text(330,65, text = "Priecīgus Ziemassvētkus!", fill = "red", font = "Manrope 35 bold") # parādam "Priecīgus Ziemassvētkus!"
    red_star = tkinter.PhotoImage(file = "red_star.png") # Mainām zvaigzni uz sarkano
    star.image = red_star
    star["image"] = red_star
```

```

def leave_red_star(e): # notikums, kad nolikam kursoru nost no egles zvaigznes

    kanva.create_text(355,85, text = "\n" "un Laimīgu Jauno gadu!", fill = "#EEEEF0", font = "Manrope
35 bold") # parādam "un Laimīgu Jauno gadu!"

    star.destroy() # Noņemam sarkano zvaigzni

    snowman_smile_to_right = tkinter.PhotoImage(file = "snowman_smile_to_right.png") # Sniegavīrs
sāk skatīties uz iededzināto eglīti

    snowman["image"] = snowman_smile_to_right

    snowman.image = snowman_smile_to_right

    snieg() # Egles dedzināšana un sniegs


golden_star = tkinter.PhotoImage(file = "golden_star.png") # sākotnēji parādas dzeltenā zvaigzne
star = tkinter.Label(logs, borderwidth = -2)

star.image = golden_star


star.place(x = 518, y = 160)
star.bind("<Enter>", enter_start_golden_star)
star.bind("<Leave>", leave_red_star)
star["image"] = golden_star


# ===== Statisko objektu parādīšana =====


balta_zeme()
egle()
sniegavirs()
davasas()
rotajumi()
meness()
pogas()

```

```
# ===== Globālie skaitītāji/mainīgie =====
```

```
# Regulē žilēti (mouse-2)
```

```
metelis_on = False
```

```
# Regulē slotu (s on keyboard)
```

```
slot_a_on = False
```

```
# globalais i = 0, regulē lampiņas uz eglēs un zvaigzni uz egles
```

```
i = 0
```

```
# regulē sniegavīra cepuri
```

```
j = 1
```

```
# Lai cilindrs un cilindra_lente būtu definēti
```

```
cilindrs = kanva.create_rectangle(115, 180, 205, 315, fill = "black", outline = "black")
```

```
kanva.delete(cilindrs)
```

```
cilindra_lente = kanva.create_rectangle(115, 270, 205, 260, fill = "red", outline = "red")
```

```
kanva.delete(cilindra_lente)
```

```
# ===== Sniega/egles lampiņas animācija =
```

```
def sniegs(s, n):
```

```
    for i in range(130):
```

```
        x = random.randint(1, 1000)
```

```
        y = random.randint(-600*n - 8, 600*(1-n))
```

```
        w = random.randint(0, 10)
```

```
        kanva.create_oval(x, y, x+w, y+w, fill = 'white', tag = s)
```

```
def kustiba_un_lampinas():
```

```
    global indicator_count, i
```

```
i = i + 1
```

```
kanva.move("group1", 0, 1)
```

```
kanva.move("group2", 0, 1)
```

```
if i == 1:
```

```
    draw_circle(548,300, 7, "red")
```

```
    draw_circle(600,350, 7, "yellow")
```

```
    draw_circle(500,350, 7, "purple")
```

```
    draw_circle(550,400, 7, "yellow")
```

```
    draw_circle(600,450, 7, "red")
```

```
    draw_circle(500,450, 7, "blue")
```

```
    draw_circle(600,550, 7, "red")
```

```
    draw_circle(500,550, 7, "yellow")
```

```
    draw_circle(550,500, 7, "purple")
```

```
    draw_circle(548,300, 7, "red")
```

```
    create_star_with_n_color("gold")
```

```
elif i == 11:
```

```
    draw_circle(548,300, 7, "yellow")
```

```
    draw_circle(600,350, 7, "purple")
```

```
    draw_circle(500,350, 7, "blue")
```

```
    draw_circle(550,400, 7, "purple")
```

```
    draw_circle(600,450, 7, "yellow")
```

```
    draw_circle(500,450, 7, "red")
```

```
    draw_circle(600,550, 7, "yellow")
```

```
    draw_circle(500,550, 7, "purple")
```

```
    draw_circle(550,500, 7, "blue")
```

```
    create_star_with_n_color("red")
```

```
elif i == 21:
```

```
draw_circle(548,300, 7, "purple")
draw_circle(600,350, 7, "blue")
draw_circle(500,350, 7, "red")
draw_circle(550,400, 7, "blue")
draw_circle(600,450, 7, "purple")
draw_circle(500,450, 7, "yellow")
draw_circle(600,550, 7, "purple")
draw_circle(500,550, 7, "blue")
draw_circle(550,500, 7, "red")
create_star_with_n_color("blue")
```

elif i == 31:

```
draw_circle(548,300, 7, "blue")
draw_circle(600,350, 7, "red")
draw_circle(500,350, 7, "yellow")
draw_circle(550,400, 7, "red")
draw_circle(600,450, 7, "blue")
draw_circle(500,450, 7, "purple")
draw_circle(600,550, 7, "blue")
draw_circle(500,550, 7, "red")
draw_circle(550,500, 7, "yellow")
create_star_with_n_color("pink")
```

elif i == 32:

```
i = 0
```

```
kanva.move(indicator, 0, 1)
```

```
if kanva.coords(indicator)[1] < 600+1:
```

```
    logs.after(1, kustiba_un_lampinas)
```

else:

```

kanva.move(indicator, 0, -600-5)

logs.after(1, kustiba_un_lampinas)

if indicator_count == 0:
    kanva.delete("group1")
    sniegs("group1", 1)
    indicator_count = 1
else:
    kanva.delete("group2")
    sniegs("group2", 1)
    indicator_count = 0

def snieg():
    global indicator, indicator_count

    indicator = kanva.create_oval(23, -5, 28, 0, fill = 'white')
    indicator_count = 0
    sniegs("group1", 0)
    sniegs("group2", 1)
    kustiba_un_lampinas()


    logs.update()

# =====

logs.mainloop() # Lai logs būtu redzāms visu laiku

```

Testa piemēri:

Programmas starts (kursora ikona,  parāda, kur ir novietots peles kursors šajā attēlā):



Peles kursors novietots virs zvaigznes:

Dzeltenā zvaigzne kļūst sarkanā un parādās “Priecīgus Ziemassvētkus!”



Peles kursora novietots ārpus zvaigznes:

Zvaigzne sāk mirgot, sāk snieg, sāk mirgot lampiņas uz egles (bumbas), parādās “un Laimīgu Jauno gadu!”. Sniegavīrs sāk skatīties uz eglīti.



Ja novietojam peles kursoru virs sniegavīra sējas, tad viņš sāk skatīties uz mums:



Ja novietojam peles kursoru "ārpus" sniegavīra sājas, tad viņš sāk skatīties uz eglīti:



Ja pirmoreiz novietojam kursoru virs suņa būdas. Tad nekas nenotiek:



Bet tad, ja pārvietosim kursoru ārpus suņa būdas. Tad parādās suns:



Un tagad, ja pārvietosim kursoru uz suņu būdas, suns pazūd:



Ja nospiedām labo peles pogu, tad sniegavīram parādās spainis:



Ja atkal nospiedām uz labo peles pogu, tad sniegavīram parādās cilindrs un spainis pazūd:



Ja atkal nospiedām uz labo peles pogu, tad sniegavīram pazūd cilindrs (un tā pēc secības):



Ja nospiedām uz peles centrālo pogu (peles ritenis), tad sniegavīram parādās "mētelis" :



Ja atkal nospiedām uz peles centrālo pogu (peles ritenis), tad sniegavīram pazūd "mētelis" :



Ja nospiedām pogu "s" uz tastatūras, tad parādās slota:



Ja atkal nospiedām pogu “s” uz tastatūras, tad pazūd slota:



Tāda veidā var kombinēt kostīmus:





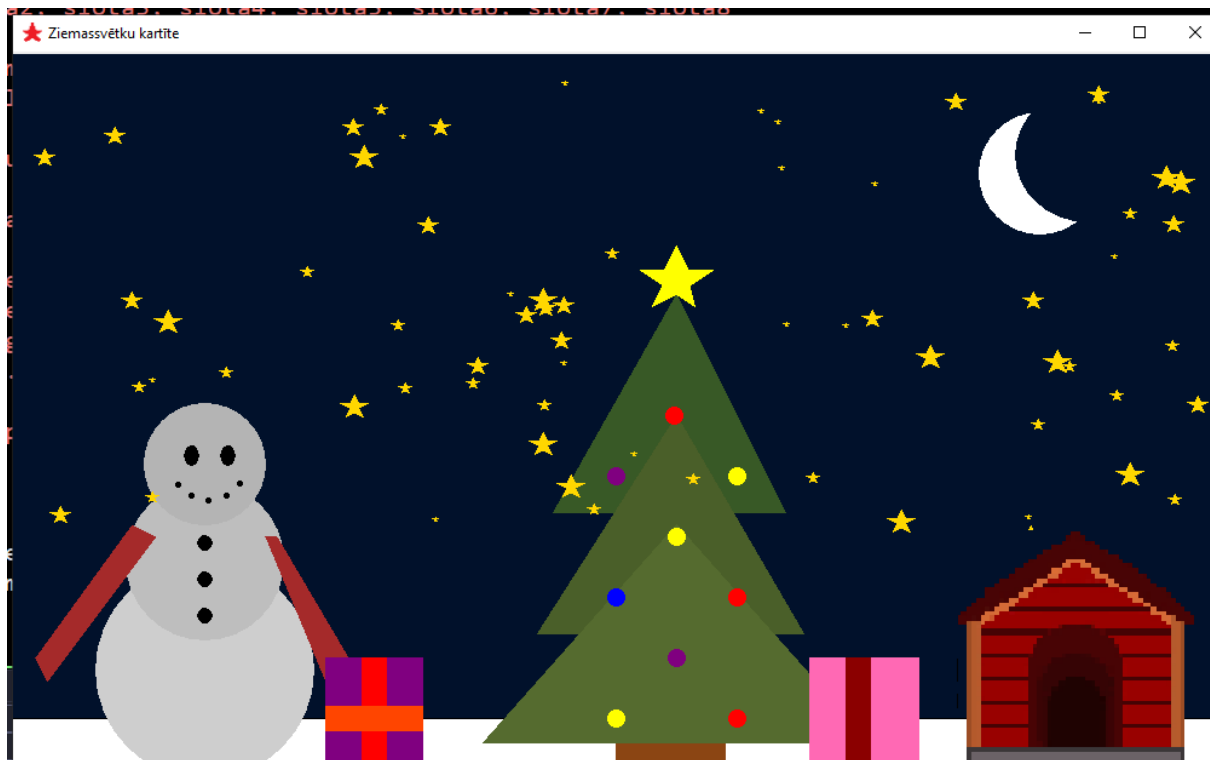




Ja nospiedām peles kreiso pogu, tad veidojas vienā zvaigznē nejauša vietā:



Tā var izveidot lielo daudzumu zvaigznes:





Sīkāk testa piemērus var apskatīt video:

“Video_Tests.mp4”

Vai izmantojot saiti:

<https://youtu.be/1roV4ByJr4>

