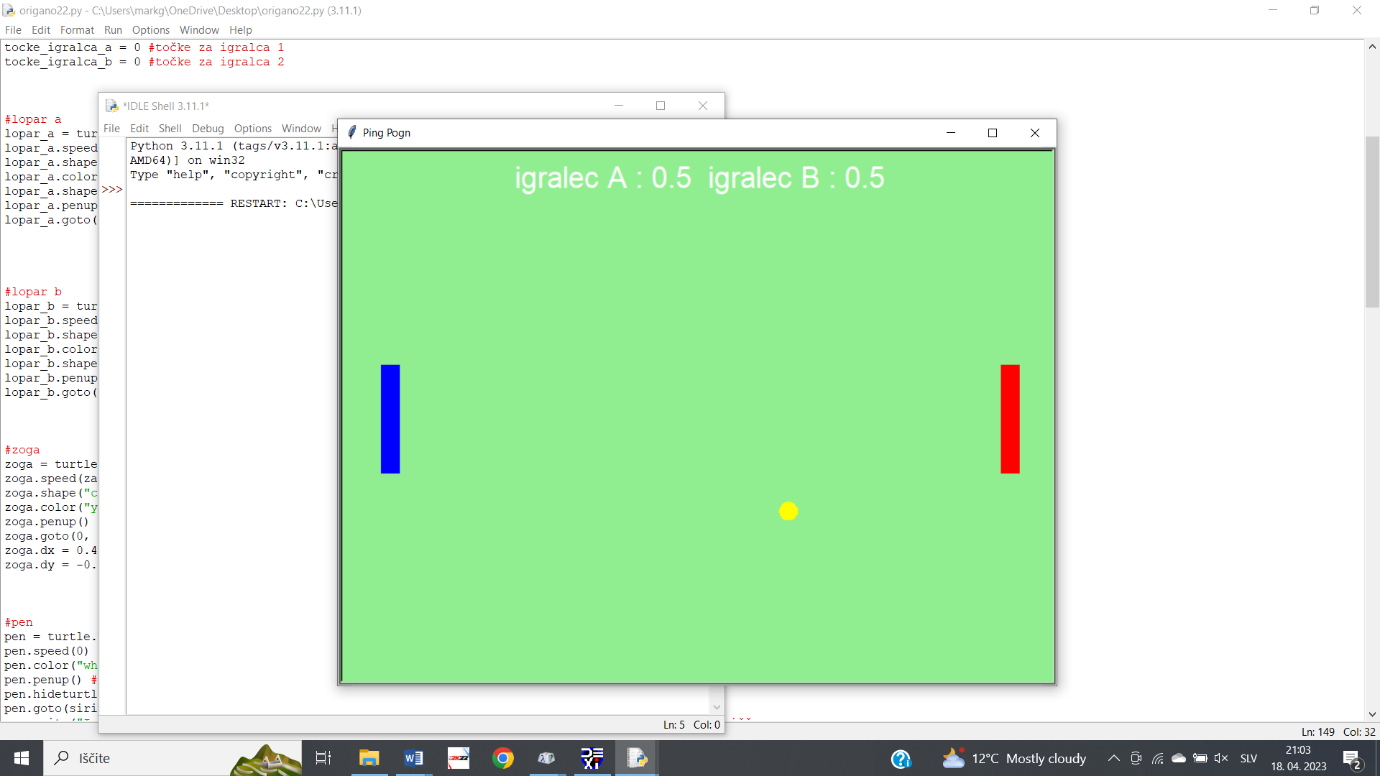


PROGRAMIRANJE IGER V PYTHONU

PING PONG



Slika 1: Naslovna slika

Avtor: Mark Gajšek

Mentor: Vinko Kušar

April 2023

Kazalo vsebine

[Povzetek 5](#_Toc132754158)

[Povzetek izdelave igrice 5](#_Toc132754159)

[Summary 6](#_Toc132754160)

[Summary of game creation 6](#_Toc132754161)

[Ključne besede 7](#_Toc132754162)

[Key words 8](#_Toc132754163)

[Opis informacijskega problema 9](#_Toc132754164)

[Python programski jezik 10](#_Toc132754165)

[Knjižnica Turtle 11](#_Toc132754166)

[Zakaj Turtle in ne Pygame 11](#_Toc132754167)

[Knjižnica Time 12](#_Toc132754168)

[Knjižnica Winsound 12](#_Toc132754169)

[Opis izdelave programa 13](#_Toc132754170)

[Navodila za igro 13](#_Toc132754171)

[Uvoz knjižnic 13](#_Toc132754172)

[Nastavitev spremenljivk 13](#_Toc132754173)

[Nastavitev igralnega okna 14](#_Toc132754174)

[Nastavitev točk 14](#_Toc132754175)

[Nastavitev obeh loparjev 15](#_Toc132754176)

[Nastavitev žoge 16](#_Toc132754177)

[Nastavitev funkcij za premik loparja 17](#_Toc132754178)

[Funkcije za premik loparjev 18](#_Toc132754179)

[Klic funkcij in nastavitev tipk na tipkovnici 19](#_Toc132754180)

[Zanka (Game loop) 19](#_Toc132754181)

[Premikanje žoge in preverjanje koordinat žoge 20](#_Toc132754182)

[Preverjanje trkov žoge z mejami in točkovanje 21](#_Toc132754183)

[Preverjanje trkov žoge z loparji 22](#_Toc132754184)

[Preverjanja konca igre 23](#_Toc132754185)

[Izpis po koncu programa 24](#_Toc132754186)

[Uporaba informacijske tehnologije 25](#_Toc132754187)

[Viri in literatura 27](#_Toc132754188)

Kazalo slik

[Slika 1: Naslovna slika 0](https://d.docs.live.net/fd8f38b9b5a469b0/Desktop/april_2023.docx#_Toc132754190)

[Slika 2: Stopnje reševanja informacijskega problema 9](https://d.docs.live.net/fd8f38b9b5a469b0/Desktop/april_2023.docx#_Toc132754191)

[Slika 3: Python programski jezik 10](https://d.docs.live.net/fd8f38b9b5a469b0/Desktop/april_2023.docx#_Toc132754192)

[Slika 4: Ustavljen program 24](#_Toc132754193)

[Slika 5: Izpis programa 24](#_Toc132754194)

**Stvarno kazalo:**

Informacijska tehnologija, 24

Knjižnica Time, 11

Knjižnica Winsound, 11

Ping Pong, 12

Programiranje, 8

Python, 9

Turtle, 10

# Povzetek

Programiranje iger združuje tako programiranje kot oblikovanje. Python je eden izmed najbolj priljubljenih programskih jezikov, ponuja več knjižnic za programiranje iger, med katerimi je ena izmed najbolj osnovnih in priljubljenih knjižnic za začetnike Turtle.

Knjižnica Turtle je knjižnica za risanje grafik na zaslonu, ki je vgrajena v Python. Omogoča preprosto risanje grafik s pomočjo ukazov, kot so premik, obrat, barva in širina črte. Poleg tega Turtle omogoča upravljanje s koordinatnimi sistemi, tako da lahko uporabniki natančno določijo položaj, smer in hitrost risanja. Programiranje iger z uporabo Turtla je odlična izbira za začetnike, saj zahteva minimalno predznanje v programiranju in se osredotoča na razvoj temeljnih konceptov, kot so logika, pogojni stavki in zanke. Z uporabo Turtla lahko ustvarimo veliko iger, od klasičnih iger, kot so Pac-Man in Tetris, do iger, ki so zahtevnejše za izdelavo.

## Povzetek izdelave igrice

Začnemo s kreiranjem okenčka igre z določitvijo velikosti zaslona in barve ozadja. Nato narišemo igralna polja, ki sta dva pravokotnika na obeh straneh zaslona. Določimo tudi začetne pozicije polj in nastavimo tako, da se polja premikajo gor in dol z uporabo tipk na tipkovnici. Nato dodamo žogo, ki se premika in odbija od sten in polj. Žogo ustvarimo kot krog in določimo njeno začetno pozicijo in hitrost. Žogo nato premikamo po x in y osi z uporabo spremenljivk, ki predstavljajo pozicijo žoge. Ko žoga zadene steno ali igralno polje, se odbije in spremeni smer gibanja. Če žoga doseže konec igralnega polja, to pomeni, da je nasprotnik osvojil točko. Nasprotniku, pa se pol točke odšteje. Vsakič, ko igralec ali nasprotnik osvojita točko, se na zaslonu prikaže napis, ki označuje rezultat. Če želimo igro Ping Pong prilagoditi po lastnih željah, lahko dodamo različne funkcije, kot so različni zvoki, spremenljive hitrosti, omejitve gibanja ali celo več igralcev. Uporaba Turtle knjižnice omogoča tudi preprosto risanje na zaslon, kar pomeni, da lahko igro prilagajamo vizualno, kot so barve, oblike in slike.

# Summary

Game programming combines both programming and design. Python is one of the most popular programming languages, it offers several libraries for programming games, among which one of the most basic and popular libraries for beginners is Turtle. The Turtle library is a library for drawing screen graphics built into Python. It allows you to easily draw graphics using commands such as shift, rotate, color and line width. In addition, Turtle provides coordinate system management so that users can precisely determine the position, direction and speed of drawing. Game programming using Turtle is a great choice for beginners as it requires minimal prior programming knowledge and focuses on developing fundamental concepts such as logic, conditionals and loops. Using Turtle, we can create many games, from classic games like Pac-Man and Tetris to games that are more challenging to make.

## Summary of game creation

We start by creating the game window by specifying the screen size and background color. Next, we draw the playing fields, which are two rectangles on either side of the screen. We also define the starting positions of the fields and set the fields to move up and down using the keys on the keyboard. Then we add a ball that moves and bounces off walls and fields. We create the ball as a circle and determine its initial position and speed. The ball is then moved along the x and y axes using variables representing the ball's position. When the ball hits a wall or playing field, it bounces and changes direction. If the ball reaches the end of the playing field, it means that the opponent has won a point. Half a point is deducted from the opponent. Each time a player or an opponent wins a point, a text will appear on the screen to indicate the score. If we want to customize the Ping Pong game to our liking, we can add different features such as different sounds, variable speeds, movement restrictions or even multiplayer. Using the Turtle library also makes it easy to draw to the screen, which means we can customize the game visually, such as colors, shapes, and images

# Ključne besede

**Programiranje** - programiranje vključuje pisanje kode v programskih jezikih, ki so zasnovani za komuniciranje z računalnikom, ustvarjanje algoritmov za reševanje problemov, testiranje in odpravljanje napak v kodah za zagotavljanje njegove funkcionalnosti in učinkovitosti.

**Python** - je visokonivojski interpretirani programski jezik, ki je znan po svoji preprostosti in berljivosti kode, široki uporabi, raznolikosti knjižnic

**Informacijski problem** - programiranje iger v Pythonu je informacijski problem, ki zahteva znanje programiranja

**Knjižnica Turtle** – je vgrajena knjižnica v Python za grafiko, je zelo preprosta in primerna za začetnike

**Knjižnica Winsound** – je knjižnica za zvok in zvočne efekte. Omejena je na operacijski sistem Windows

**Knjižnica Time** - je knjižnica za čas, s pomočjo nje lahko določamo in pretvarjamo časovne enote v programu

**Informacijska tehnologija** - Informacijska tehnologija (IT) se nanaša na uporabo računalnikov, strojne in programske opreme, omrežij in drugih tehnologij za obdelavo, shranjevanje in prenos informacij. IT vključuje različne tehnologije in aplikacije, kot so računalniške mreže, internet, računalniški programi, podatkovne baze

# Key words

**Programming** - programming involves writing code in programming languages ​​designed to communicate with a computer, creating algorithms to solve problems, testing and debugging code to ensure its functionality and efficiency.

**Python** - is a high-level interpreted programming language, which is known for its simplicity and readability of code, wide use, variety of libraries

**Information problem** - programming games in Python is an information problem that requires programming knowledge

**Turtle library** - is a built-in Python library for graphics, it is very simple and suitable for beginners

**Winsound library** - is a library for sound and sound effects. It is limited to the Windows operating system

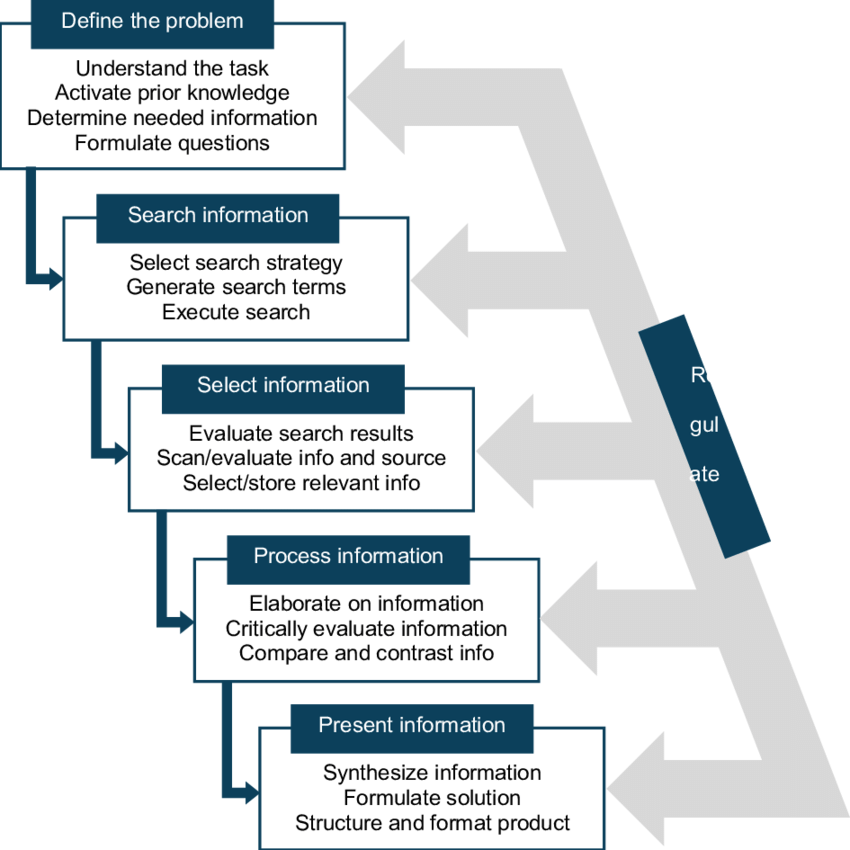
**Time library** - is a library for time, with its help we can define and convert time units in the program

**Information Technology** - (IT) refers to the use of computers, hardware and software, networks and other technologies to process, store and transmit information. IT includes various technologies and applications, such as computer networks, the Internet, computer programs, databases

# Opis informacijskega problema

Programiranje iger v Pythonu je informacijski problem, ki zahteva znanje programiranja, matematike, fizike in oblikovanja. Igralni procesi in vmesniki morajo biti natančno načrtovani in izdelani. Pri ustvarjanju iger v Pythonu je pomembno upoštevati več faktorjev, kot so: algoritmi, animacije, fizika in računalniška grafika. Poleg tega je pomembno razumeti, kako delujejo različni operacijski sistemi in uporabniški vmesniki, da se igra lahko izvaja brez težav na različnih platformah. Pri programiranju iger v Pythonu se lahko soočimo s številnimi izzivi, kot so: načrtovanje igranja, preverjanje omejitev in mehanik igre, ustvarjanje računalniških nasprotnikov, nadzor obdelava vhodov igralca. Poleg tega je pomembno upoštevati tudi uporabniško izkušnjo, kot so grafični vmesniki, zvočni efekti in podobno.

Vse tehnologije in metode, ki se uporabljajo pri programiranju iger v Pythonu, so zasnovane za izdelavo boljše igralne izkušnje za uporabnika. Skupaj tvorijo informacijski problem, ki ga je treba rešiti z uporabo znanja in spretnosti v različnih področjih, od programiranja do oblikovanja.

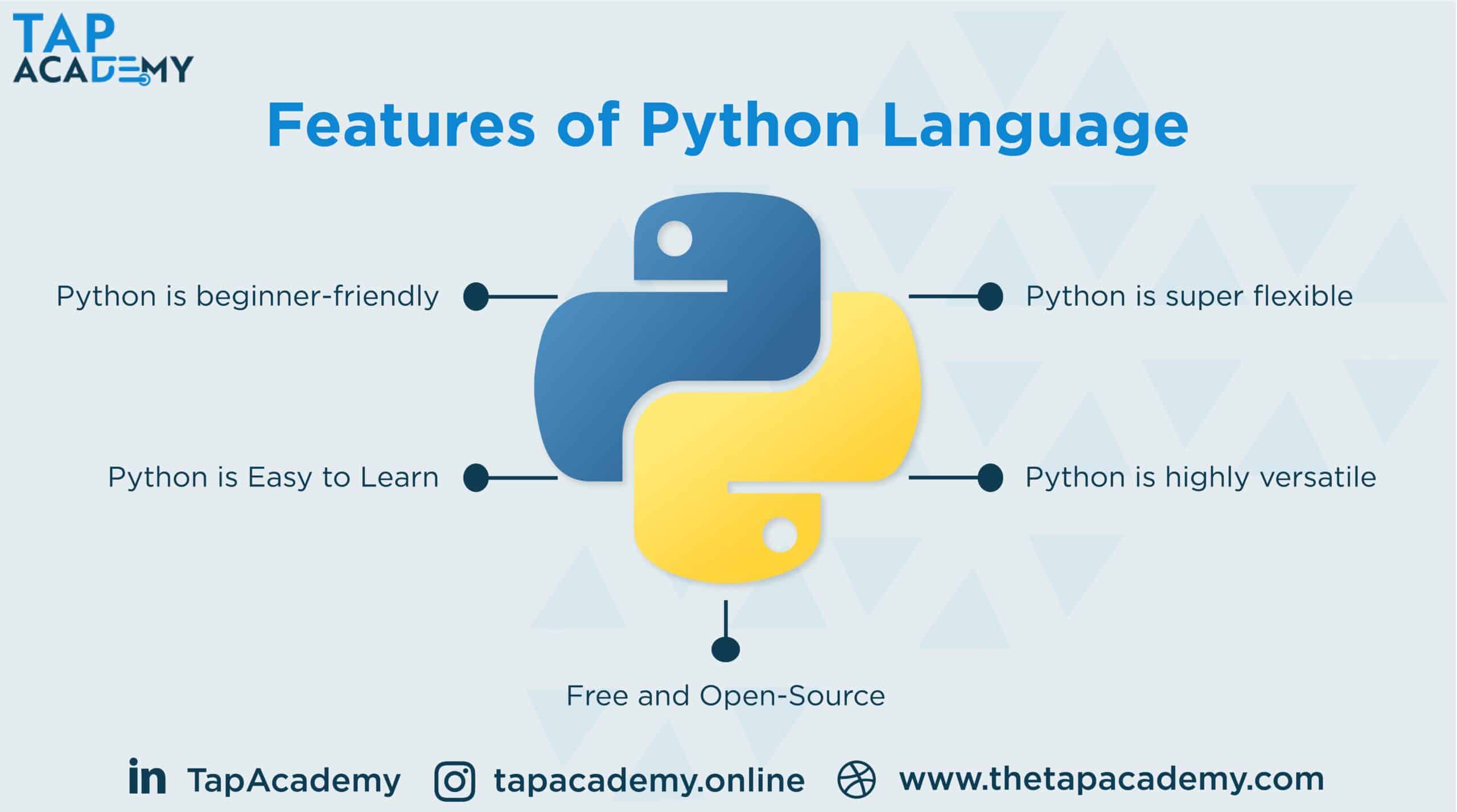
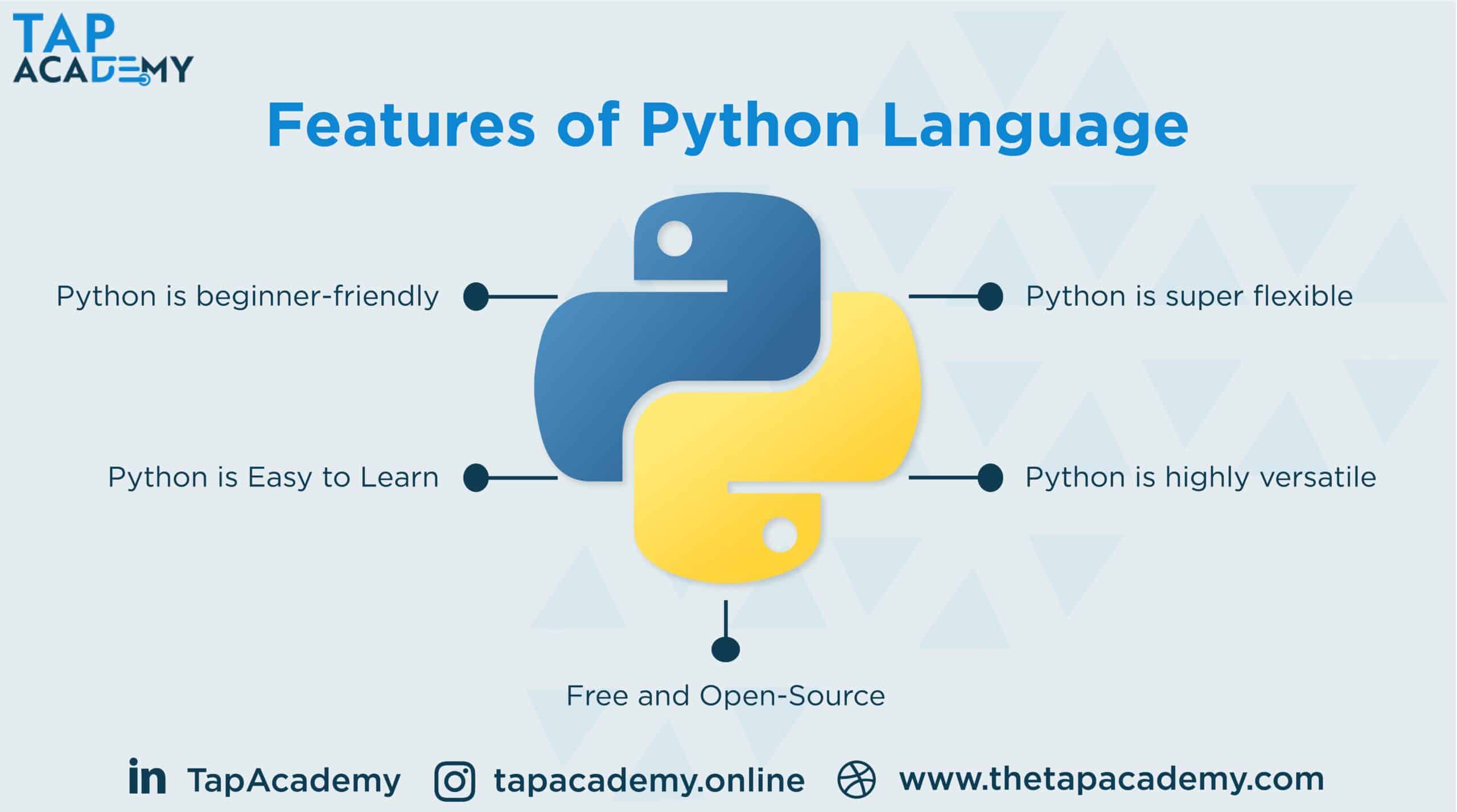


Slika 2: Stopnje reševanja informacijskega problema

# Python programski jezik

Python[[1]](#footnote-1) je zasnovan tako, da omogoča enostavno branje in pisanje kode, kar pomeni, da je jezik primeren za začetnike in napredne programerje. Python uporablja sintakso, ki spominja na angleški jezik in ima zato zelo razumljiv zapis kode. Na primer, za izvedbo pogojnega izraza lahko uporabimo ključni besedi "if" in "else", kar je precej bolj razumljivo kot v drugih programskih jezikih.

Python je interpretiran jezik, kar pomeni, da ne potrebuje posebnega prevajanja v strojni jezik, kot je to pri nekaterih drugih programskih jezikih. Namesto tega se koda interpretira v realnem času in se izvaja po vrstnem redu, kot je napisana. Ena izmed glavnih prednosti Pythona je njegova razširljivost. Python ima veliko število vnaprej pripravljenih modulov, ki so na voljo za uporabo, kar lahko prihrani čas pri razvoju programske opreme.

Poleg tega ima Python veliko skupnost, ki redno objavlja različne knjižnice in orodja, ki so na voljo brezplačno. Python je tudi objektno usmerjen programski jezik, kar pomeni, da je zgrajen okoli koncepta objektov, ki vsebujejo lastnosti in metode. Objektno usmerjeno programiranje omogoča modularnost in organizacijo kode, kar olajša razumevanje in spreminjanje kode. Python je znan tudi po svoji sposobnosti za obdelavo podatkov in znanstveno računanje, kar je posledica velikega števila modulov, ki so na voljo za to področje. Python je prav tako zelo prilagodljiv jezik, ki se lahko uporablja za različne vrste projektov.

Slika 3: Python programski jezik

## Knjižnica Turtle

Knjižnica Turtle[[2]](#footnote-2) je modul v programskem jeziku Python, ki se uporablja za izrisovanje grafik v oknu, ki se imenuje grafični okvir. Turtle je osnovan na logiki, ki se imenuje grajenje, kar pomeni, da lahko izrisujemo sestavljene slike z uporabo preprostih ukazov. Vsak ukaz v Turtlu se izvede v trenutku, ko se slika posodobi v oknu, zato lahko programerji vidijo takojšnje spremembe v sliki, ki jih naredijo.

Turtle ponuja širok nabor funkcij, ki se uporabljajo za izrisovanje slik. Med njimi so funkcije za premikanje, risanje linij, izrisovanje krogov in polkrogov, izpolnjevanje oblik in še več. Te funkcije se lahko združijo, da ustvarijo bolj zapletene slike. Turtle je lahko tudi interaktiven, kar pomeni, da lahko uporabnik nadzira smer gibanja želve s pomočjo tipkovnice ali miške. To omogoča ustvarjanje animacij in iger z uporabo Turtla.

Turtle je prav tako priljubljena knjižnica za razvoj spletnih iger in animacij. Uporaba Turtla za izrisovanje animacij lahko pomaga pri razvijanju veščin v programiranju, kot so logika, sintaksa in načrtovanje kode. Ena izmed prednosti Turtla je njegova preprostost. Funkcije so enostavne za uporabo in Turtle ponuja veliko število primerov kode, ki jih lahko uporabite kot izhodišče. Poleg tega lahko programerji prilagodijo željeno obliko želve s spreminjanjem barve, debeline linij in drugih lastnosti.

## Zakaj Turtle in ne Pygame

Prva prednost Turtla pred Pygameom je njegova preprostost. Turtle je namenjen predvsem za začetnike, zato ima enostavno sintakso, ki je enostavna za razumevanje. Poleg tega ima Turtle tudi veliko število vgrajenih funkcij, ki jih lahko hitro in enostavno uporabimo za risanje grafik. Druga prednost Turtla pred Pygameom je njegova interaktivnost. Uporabnik lahko nadzira smer gibanja želve s pomočjo tipkovnice ali miške, kar omogoča interaktivno izkušnjo za uporabnika. To lahko olajša učenje osnov programiranja in matematike, saj lahko uporabnik v realnem času videli spremembe v sliki, ki jih naredi. Poleg tega lahko Turtle pomaga pri vizualizaciji matematičnih konceptov, kot so geometrija in koordinate. Tretja prednost Turtla pred Pygameom je njegova enostavnost risanja oblik. Turtle ima veliko število funkcij za risanje različnih oblik, kot so krogi, kvadrati, trikotniki, elipse itd. Poleg tega lahko uporabnik prilagodi željeno obliko želve s spreminjanjem barve, debeline linij in drugih lastnosti.

## Knjižnica Time

Knjižnica Time[[3]](#footnote-3) je knjižnica za upravljanje s časom v programskem jeziku Python. Uporablja se za delo s časovnimi podatki, kot so datumi, časi, časovni intervali in podobno. Knjižnica Time ima številne uporabne funkcije, ki so zelo koristne za delo s časovnimi podatki. Ena izmed najpogostejših uporab knjižnice Time je upravljanje s časovnimi območji. To vključuje pridobivanje trenutnega časa, pretvorbo časovnih območij med različnimi formati, delo s časovnimi intervali in podobno. Knjižnica Time vsebuje tudi številne funkcije za delo s časovnimi nizi, ki so lahko uporabni pri analizi podatkov. Druga uporabna funkcija knjižnice Time je možnost obračuna časovnih razlik. To lahko vključuje izračun časovne razlike med dvema določenima datumoma, izračun razlike med dvema časi ali izračun časovne razlike.

## Knjižnica Winsound

Knjižnica Winsound[[4]](#footnote-4) je del standardne knjižnice Pythona in se uporablja za predvajanje zvokov na operacijskih sistemih Windows. Z uporabo te knjižnice lahko predvajamo zvoke v obliki valovnih oblik, ki so shranjene v datotekah WAV. Za uporabo knjižnice WinSound moramo najprej uvoziti modul z ukazom "import winsound". Nato lahko s pomočjo funkcije "PlaySound" predvajamo zvok iz datoteke. Funkcija "PlaySound" ima štiri parametre: ime datoteke, način predvajanja, želeni način predvajanja v ozadju in želeno trajanje predvajanja. Način predvajanja je lahko niz, ki predstavlja način predvajanja, kot je SND\_FILENAME, ki se uporablja za predvajanje zvoka iz datoteke WAV, SND\_LOOP, ki predvaja zvok v zanki, ali SND\_ASYNC, ki predvaja zvok asinhrono. Želeni način predvajanja v ozadju lahko uporabimo, če želimo, da se zvok predvaja brez ustavljanja izvajanja programa. Če je ta parameter nastavljen na SND\_ASYNC, bo program nadaljeval z izvajanjem, medtem ko se zvok predvaja.

Poleg funkcije PlaySound ima knjižnica WinSound tudi druge funkcije za predvajanje zvoka, kot so Beep, ki predvaja kratke tone s specifično frekvenco in trajanjem, ter MessageBeep, ki predvaja določen zvok ob obveščanju uporabnika o napaki ali obvestilu. Knjižnica WinSound je uporabna pri različnih aplikacijah, ki zahtevajo predvajanje zvoka. Vendar pa je knjižnica omejena na operacijski sistem Windows.

# Opis izdelave programa

## Navodila za igro

Igra Ping Pong je narejena za dva igralca. Cilj igre je odbiti žogo. Prvi igralec ima na voljo tipki w in s, s katerima premika lopar v obliki pravokotnika in skuša odbiti žogo. Drugi igralec pa ima na voljo tipki puščico za gor in dol. Njegov cilj je enak. Če igralec ne odbije žoge in žoga zadane steno po x koordinati, potem se nasprotnemu igralcu prišteje točka. Prav tako se igralcu, ki ni uspel odbiti žoge odšteje pol točke. Igra se konča v petih primerih. Če igralec A doseže 11 točk, igralec B 11 točk, igralec A -3 točke, igralec B -3 točke, ali pa če imata igralec A in B oba točno 10 točk. V zadnjem primeru je igra neodločena.[[5]](#footnote-5)

## Uvoz knjižnic

import turtle #uvoz knjižnice turtle

import winsound #uvoz knjižnice winsound

import time #uvoz knjižnice time

## Nastavitev spremenljivk

#spremenljivke

zacetni\_cas = time.time()

visina\_okna=600

sirina\_okna=800

zacetna\_hitrost\_loparja=0

zacetna\_hitrost\_zoge=0

## Nastavitev igralnega okna

#nastavi okno

okno = turtle.Screen()#novo okno

okno.title("Ping Pogn") #naslov

okno.bgcolor("light green") #barva ozadja

okno.setup(width=sirina\_okna, height=visina\_okna) #definicija okna s parametri

okno.tracer(0) #animacija se izklopi

Najprej se z ukazom "okno = turtle.Screen()" ustvarimo novo okno, ki se uporablja za izrisovanje. Naslednji ukaz "okno.title("Ping Pong")" nastavi naslov okna na Ping Pong.

Ukaz okno.bgcolor("light green") nastavi barvo ozadja okna na svetlo zeleno.

Ukaz "okno.setup(width=sirina\_okna, height=visina\_okna)" definira okno s parametri, kjer sta širina in višina okna nastavljeni na določene vrednosti. Vrednosti parametrov lahko spreminjamo glede na potrebe in želje uporabnika.

Zadnji ukaz "okno.tracer(0)" izklopi animacijo, kar pomeni, da se risanje slike ne bo prikazovalo korak po korak, temveč bo prikazana celotna slika naenkrat.

## Nastavitev točk

#točke

tocke\_skupaj=0 #točke igralcev skupaj

tocke\_igralca\_a = 0 #točke za igralca 1

tocke\_igralca\_b = 0 #točke za igralca 2

Spremenljivka tocke\_skupaj predstavlja skupno število točk obeh igralcev. Ta spremenljivka se uporablja za beleženje celotnega števila doseženih točk pri igri. Spremenljivki tocke\_igralca\_a in tocke\_igralca\_b predstavljata število točk za vsakega igralca posebej. To se uporablja za beleženje števila točk, ki jih posamezni igralec doseže med igro. Vsakič, ko igralec doseže točko, se vrednost ustrezne spremenljivke poveča za eno. Na koncu igre se lahko uporabi spremenljivko "tocke\_skupaj", da se izračuna skupno število točk, ki so bile dosežene med igro.

## Nastavitev obeh loparjev

#lopar a

lopar\_a = turtle.Turtle() #nov turle lopar a

lopar\_a.speed(zacetna\_hitrost\_loparja) #začetna hitrost

lopar\_a.shape("square") #Oblika

lopar\_a.color("blue") #barva

lopar\_a.shapesize(stretch\_wid=6, stretch\_len=1) #velikost loparja s parametri

lopar\_a.penup() #dvigne turtke

lopar\_a.goto(-345, 0) #levo začne risati

#lopar b

lopar\_b = turtle.Turtle()

lopar\_b.speed(zacetna\_hitrost\_loparja)

lopar\_b.shape("square") #isto kot lopar a

lopar\_b.color("red")

lopar\_b.shapesize(stretch\_wid=6, stretch\_len=1)

lopar\_b.penup()

lopar\_b.goto(345, 0)

Zgornji kodi ustvarjata dva Turtle objekta, ki predstavljata loparje za igro. Vsak lopar se ustvari z ukazom turtle.Turtle(), ki vrne nov objekt turtle. Nato se določijo lastnosti loparjev, kot so hitrost, oblika, barva in velikost.

Loparje se izriše s kvadratno obliko, ki je raztegnjena po višini s pomočjo metode "shapesize(stretch\_wid=6, stretch\_len=1)".

S spremenljivko zacetna\_hitrost\_loparj, se določi začetna hitrost loparjev, ki se lahko kasneje med igro spremeni.

Sledi uporaba metode penup() za dvig turtlla s platna, da se prepreči risanje črte med premikanjem loparja. Z metodo goto(x, y) se določi začetna pozicija loparja na platnu. Lopar\_a se določi na levi strani igralnega polja (-345, 0), lopar\_b pa na desni strani igralnega polja (345, 0).

## Nastavitev žoge

zoga = turtle.Turtle() #nov turtle zoga

zoga.speed(zacetna\_hitrost\_zoge) #zacetna hitrot

zoga.shape("circle") #oblika

zoga.color("yellow")#barva

zoga.penup() #dvigne turtle

zoga.goto(0, 0) #začne risati v sredini

zoga.dx = 0.45 #smer začetnega gibanja žoge po x osi

zoga.dy = -0.45 #smer začetnega gibanja žoge po y soi

Objekt turtle predstavlja žogo. S pomočjo metod se nastavijo lastnosti žoge, kot so hitrost, oblika in barva. Žoga se izriše kot krog z metodo shape("circle") in obarva z metodo color("yellow").

Z metodo penup() se dvigne objekt Turtle s platna, da se prepreči risanje črte med premikanjem žoge. Z metodo goto(x, y) se določi začetna pozicija žoge na sredini platna (0, 0). Nato se določi začetna hitrost žoge z uporabo spremenljivk zacetna\_hitrost\_zoge, dx in dy. Spremenljivki dx in dy predstavljata smer gibanja žoge po x in y osi, ki sta nastavljeni na 0.45. Pozitivna vrednost spremenljivke dx pomeni, da bo žoga premikala desno, negativna vrednost pa pomeni, da bo žoga premikala levo. Podobno, pozitivna vrednost spremenljivke dy pomeni, da bo žoga premikala navzgor, negativna vrednost pa pomeni, da bo žoga premikala navzdol.

## Nastavitev funkcij za premik loparja

#pen

pen = turtle.Turtle() #risanje nov turtle

pen.speed(0) #začetna hitrost risala

pen.color("white") #barva risala

pen.penup() #dvigni turtle

pen.hideturtle() #skrij turtle risalo

pen.goto(sirina\_okna/sirina\_okna, 250) #začetek risanja

pen.write("Igralec A : 0 Igralec B :0", align="center", font=("Arial", 25, "normal")) #napis nad igriščem

Turtle objekt z imenom pen, ki se uporablja za risanje besedila na igralnem zaslonu. Hitrost turtla je nastavljena na 0, da se giblje takoj brez zamika. Barva pisala je nastavljena na belo. Pisalo je dvignjeno z uporabo metode penup() in nato skrito z metodo hideturtle(), da ostane nevidno na igralnem zaslonu. Pisalo nato zapiše besedilo "Igralec A: 0 Igralec B: 0" na igralnem zaslonu, s središčno poravnano in z velikostjo pisave Arial 25.

## Funkcije za premik loparjev

#funkcije za premik loparja

def lopar\_a\_gor(): #definiraj premikanje loparja gor

y = lopar\_a.ycor() #y koorinata loparja

y += 40 #hitrost za gor

lopar\_a.sety(y) #nove koordinate loparja

def lopar\_a\_dol(): #definiraj premikanje loparja dol

y = lopar\_a.ycor() #y koorinata loparja

y -= 40 #hitrost za dol

lopar\_a.sety(y) #nove koordinate loparja

def lopar\_b\_gor():

y = lopar\_b.ycor()

y += 40

lopar\_b.sety(y) #enako kot lopar a

def lopar\_b\_dol():

y = lopar\_b.ycor()

y -= 40

lopar\_b.sety(y)

Funkcije so namenjene premikanju loparjev v igri. Vsaka funkcija predstavlja premik loparja v določeno smer. Funkcija lopar\_a\_gor() premika lopar A navzgor za 40 enot, kar predstavlja hitrost premikanja. Najprej se pridobi y koordinata loparja A, nato se tej vrednosti doda 40 enot, kar predstavlja novo koordinato loparja. Nato se ta nova koordinata nastavi kot y koordinata loparja A z uporabo metode sety(). Funkcija lopar\_a\_dol() premika lopar A navzdol za 40 enot. Delovanje funkcije je podobno funkciji lopar\_a\_gor(), le da se tu vrednost y koordinate zmanjša za 40 enot. Enako velja za lopar b.

## Klic funkcij in nastavitev tipk na tipkovnici

okno.listen() #bere tipkovnico medtem ko program runna

okno.onkeypress(lopar\_a\_gor, "w") #w kliče funkcijo gor

okno.onkeypress(lopar\_a\_dol, "s") #s kliče funkcijio dol

okno.onkeypress(lopar\_b\_gor, "Up") #puščica gor kliče funkcijo gor

okno.onkeypress(lopar\_b\_dol, "Down") #puščica dol kliče funkcijo dol

Prva vrstica okno.listen() omogoča programu, da bere vhod tipkovnice, medtem ko se izvaja. To pomeni, da lahko program ujame dogodke, ki jih uporabnik sproži s tipkovnico, medtem ko igra.

Nato je dodana več vrstic kode, ki določajo, kateri dogodki tipkovnice kličejo katere funkcije. Vrstice okno.onkeypress povežejo funkcijo za premikanje loparja( lopar\_a\_gor) z določenim tipom tipke. V primeru okno.onkeypress(lopar\_a\_gor, w), ko uporabnik pritisne tipko "w" na tipkovnici, bo klicana funkcija lopar\_a\_gor(), ki premakne lopar a gor. S tipko s pa bo klicana funkcija lopar\_dol. Enako velja za lopar b le, da sta tam pušica gor in puščica dol.

## Zanka (Game loop)

#zanka

while True:

okno.update() #posodobitev okna

skupni\_cas = time.time() - zacetni\_cas #izračun skupnega časa

Zanka while predstavlja neskončno zanko ali game loop[[6]](#footnote-6), ki se bo izvajala, dokler ne pride do prekinitve. V tem primeru se izvaja igra, zato je notranjost zanke zgrajena tako, da se izvaja celoten proces igre. Vsakič, ko se vrti zanka, se posodobi okno s klicem metode update(), ki ponovno izriše okno z vsemi spremembami v elementih igre. Nato se izračuna skupni čas, ki je enak trenutnemu času minus času, ki je bil shranjen pri zagonu igre. To omogoča izračun časa, ki je pretekel od začetka igre.

Zanka se konča v petih primerih. Če igralec a doseže 11 ali -3 točke, igralec b 11 ali -3 točke, ali, če imata igralec a in istočasno 10 točk.

## Premikanje žoge in preverjanje koordinat žoge

#premik žoge

zoga.setx(zoga.xcor() + zoga.dx) #prišteva koordinate od žoge v x smeri

zoga.sety(zoga.ycor() + zoga.dy) #prišteva koordinate od žoge v y smeri

#preverjanje kordinat žoge

if zoga.ycor() > 300: #če je več kot 300 v y smeri

zoga.sety(300) #se odbije

zoga.dy \*= -1 #spremeni smer gibanja

winsound.PlaySound("lala.wav", winsound.SND\_ASYNC) #in zaigra se sound

if zoga.ycor() < -300: #enako na levi strani kot na desni

zoga.sety(-300)

zoga.dy \*= -1

winsound.PlaySound("lala.wav", winsound.SND\_ASYNC) #zaigraj zvok

V tem delu kode se premika žoga po koordinatni mreži z dodajanjem vrednosti dx (spremenljivka, ki določa premik v x smeri) k trenutni koordinati x in dy (spremenljivka, ki določa premik v y smeri) k trenutni koordinati y. Če je y koordinata žoge večja od 300, se žoga odbije s stropa (nastavi se y koordinata na 300) in spremeni smer gibanja v y smeri tako, da se spremeni predznak spremenljivke dy (množi se z -1). Nato se predvaja zvok. Če je y koordinata žoge manjša od -300, se žoga odbije s tal (nastavi se y koordinata na -300) in spremeni smer gibanja v y smeri tako, da se spremeni predznak spremenljivke dy (množi se z -1). Nato se predvaja zvok.

## Preverjanje trkov žoge z mejami in točkovanje

#lopar x koordinata preverjanje

if zoga.xcor() > 400: #če je večje kot 400

zoga.setx(400) #gre na 400

zoga.dx \*= -1 #spremeni smer žoge

tocke\_igralca\_a += 1 #in doda točko

tocke\_igralca\_b -= 0.5 #odvzame točko

pen.clear() #izbris niza

pen.write("igralec A :{} igralec B : {}".format(tocke\_igralca\_a, tocke\_igralca\_b), align="center", font=("Arial", 25, "normal")) #napis in posodobitev točk

if zoga.xcor() < -400:

zoga.setx(-400)

zoga.dx \*= -1 #enako na levi kot na desni starni

tocke\_igralca\_b += 1

tocke\_igralca\_a -= 0.5

pen.clear()

pen.write("igralec A : {} igralec B : {}".format(tocke\_igralca\_a, tocke\_igralca\_b), align="center", font=("Arial", 25, "normal"))

Ta del kode preverja, ali je žoga prečkala mejo, ki jo tvorita dve navpični črti na vsaki strani okna, in določa, kateri igralec je dobil oziroma zgubil. Če žoga prečka mejo na desni strani okna (njen x koordinata postane večja od 400), se postavi na to mejo in spremeni smer gibanja v nasprotni smeri. Hkrati se točka dodeli igralcu A in se pol točke odvzame igralcu B. S tem se tudi posodobi prikaz števila točk na zaslonu. Če žoga prečka mejo na levi strani okna (njen x koordinata postane manjša od -400), se postavi na to mejo in spremeni smer gibanja v nasprotni smeri. Hkrati se točka dodeli igralcu B in se pol točke odvzame igralcu A. S tem se tudi posodobi prikaz števila točk na zaslonu.

## Preverjanje trkov žoge z loparji

if (zoga.xcor() > 340 and zoga.xcor() < 350) and (zoga.ycor() < lopar\_b.ycor() + 40 and zoga.ycor() > lopar\_b.ycor() - 40): #preverjanje trka med žogo in loparjem lopar je med 340 in 350

zoga.setx(340) #nova koordinata žoge

zoga.dx \*= -1 #spreminjanje smeri žoge

winsound.PlaySound("lala.wav", winsound.SND\_ASYNC) #igraj zvok

if (zoga.xcor() < -340 and zoga.xcor() > -350) and (zoga.ycor() < lopar\_a.ycor() + 40 and zoga.ycor() > lopar\_a.ycor() - 40): #lopar b enako kot lopar a

zoga.setx(-340)

zoga.dx \*= -1

winsound.PlaySound("lala.wav", winsound.SND\_ASYNC)

V tem delu koda preverjamo, ali se žoga trči z loparjem igralca A ali B. Če žoga doseže določeno x koordinato (med 340 in 350 ali -340 in 350) in hkrati se nahaja v okviru y koordinat loparja (+/-40 od loparjeve y koordinate), potem se žoga odbije nazaj in spremeni smer gibanja, ter se predvaja zvok z uporabo funkcije winsound.PlaySound".

## Preverjanja konca igre

if (tocke\_igralca\_a == 11): #če igralec a doseže 11 končaj in izpiši

print("Zmagovalec je igralec a")

break; #prekine while True

if (tocke\_igralca\_b == 11):

print("Zmagovalec je igralec b") #če igralec b doseže 11 končaj in izpiši

break; #prekine while True

if (tocke\_igralca\_b == 10 and tocke\_igralca\_a==10):

print("Neodločeno je, ponovi!") #če igralec a in b imata 10 točk je neodločeno in se program zaključi

break; #prekine while True

if (tocke\_igralca\_a == - 3): #če igralec a doseže -3 točke končaj in izpiši

print("Igralec a je avtomatsko zgubil")

break; #igra se konča, če igralec doseže 0 točk

if (tocke\_igralca\_b == -3):

print("Igralec b je avtomatsko zgubil") #če igralec b doesže -3 točke končaj izpiši

break; #prekine while True

print("Čas igre: ", round(skupni\_cas), "sekund.") #izpiši čas igre

Ta del kode predstavlja preverjanje konca igre, tj. ali je kateri od igralcev dosegel 11 točk oziroma -3 točke ali pa imata oba igralca po 10 točk, kar pomeni neodločen rezultat. V tem primeru se izpiše sporočilo o zmagovalcu ali poražencu, nato pa se igra prekine s pomočjo break. Na koncu se izpiše tudi skupni čas igre.

## Izpis po koncu programa

1. Možnost: Igralec A dosegel 11 točk.

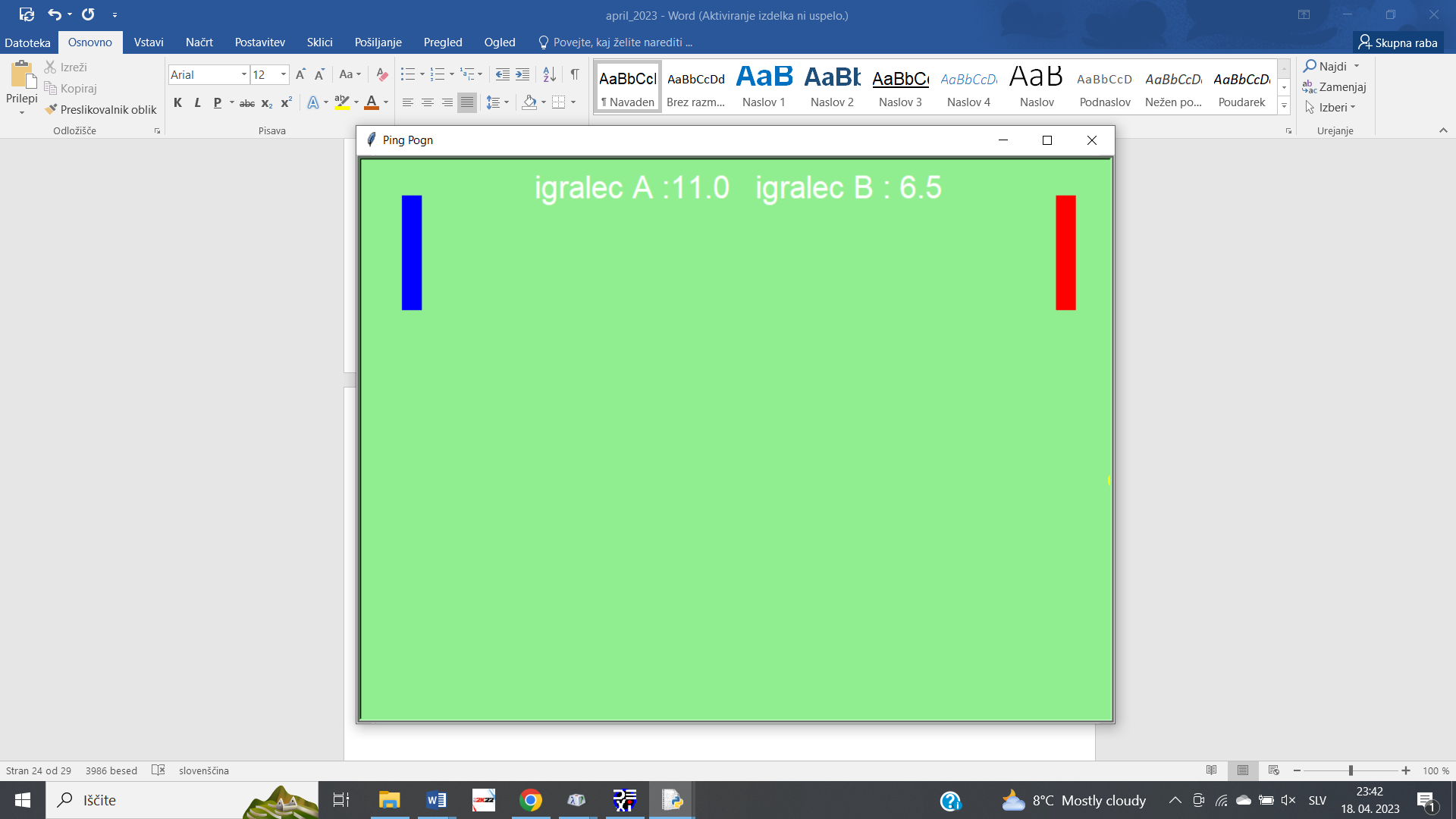
2. Možnost: Igralec A dosegel -3 točke.

3. Možnost: Igralec B dosegel 11 točk.

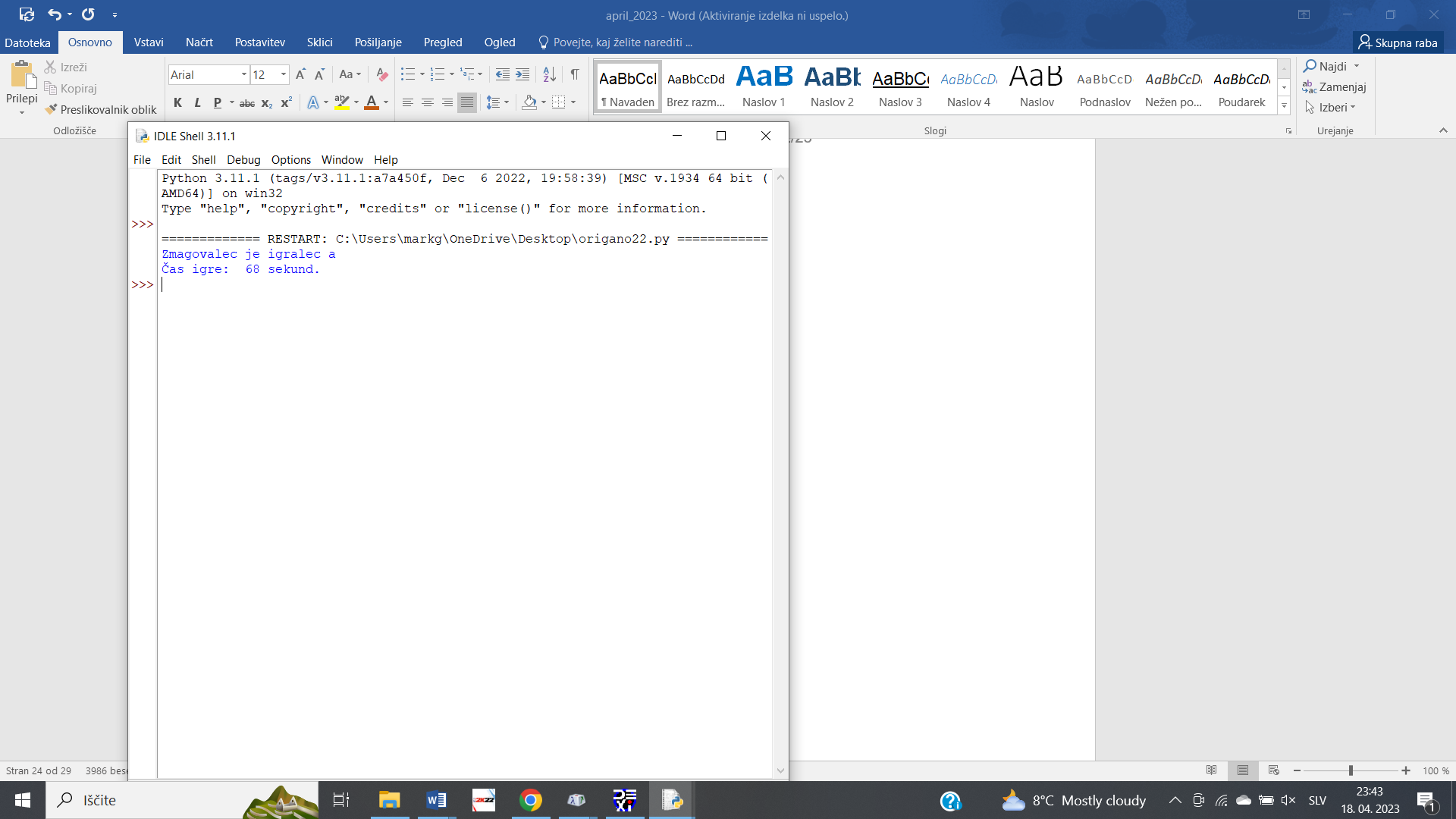
4. Možnost: Igralec B dosegel -3 točke.

5. Možnost: Igralca A in B imata istočasno 10 točk

Program se ustavi:



Slika 4: Ustavljen program



Slika 5: Izpis programa

# Uporaba informacijske tehnologije

Informacijska tehnologija[[7]](#footnote-7) pri izdelavi igre ping pong v Pythonu s knjižnico Turtle je uporaba informacijske tehnologije ključna. Python je visoko nivojski programski jezik, ki se pogosto uporablja za izdelavo igric in drugih aplikacij. Knjižnica Turtle pa nam omogoča preprosto risanje in animiranje grafik na zaslonu. Pri izdelavi igre ping pong z uporabo Pythona in Turtle knjižnice se lahko srečamo z različnimi težavami, ki jih lahko rešujemo s pomočjo informacijske tehnologije. Ena izmed teh težav je na primer zaznavanje trkov med žogico in loparjem. Pri tem nam lahko pomaga algoritem za zaznavanje trkov, ki ga lahko implementiramo s pomočjo matematičnih izračunov. Poleg tega lahko s pomočjo informacijske tehnologije v igro dodamo tudi različne funkcionalnosti, kot so zvočni učinki, glasba in različni efekti pri udarcu žogice.

Pri izdelavi igre ping pong lahko uporabimo tudi različne tehnologije za izboljšanje uporabniške izkušnje, kot so na primer umetna inteligenca in strojno učenje. S pomočjo teh tehnologij lahko izboljšamo igralnost igre in naredimo igro bolj dinamično ter zanimivo za igralce. Poleg tega lahko informacijska tehnologija pomaga pri optimizaciji igre, saj lahko s pomočjo algoritmov za optimizacijo povečamo hitrost igre, izboljšamo grafično zmogljivost in zmanjšamo porabo sistemskih virov. Tako lahko zagotovimo boljšo uporabniško izkušnjo, ne glede na to, katero napravo ali računalnik igralec uporablja. Skupaj lahko povzamemo, da je uporaba informacijske tehnologije ključna pri izdelavi igre ping pong v Pythonu s knjižnico Turtle. Z njo lahko izboljšamo kakovost igre.

# Viri in literatura

Spletne strani:

<https://www.w3schools.com/python/>

<https://www.programiz.com/python-programming>

<https://www.geeksforgeeks.org/history-of-python/>

<https://www.kaggle.com/learn/python>

<https://realpython.com/beginners-guide-python-turtle/>

Videoposnetki:

<https://realpython.com/beginners-guide-python-turtle/>

<https://www.youtube.com/watch?v=0EWN23EOq0w>

<https://www.youtube.com/watch?v=0EWN23EOq0w>

<https://www.youtube.com/watch?v=C6jJg9Zan7w>

Knjige:

1. Python – višji interpretiran programski jezik [↑](#footnote-ref-1)
2. Turtle – je vgrajena knjižnica, ki se uporablja za grafiko [↑](#footnote-ref-2)
3. Knjižnica Time – se uporablja za delo s časovnimi podatki [↑](#footnote-ref-3)
4. Knjižnica Winsound – se uporablja za predvajanje zvoka in je omejena na Windows [↑](#footnote-ref-4)
5. Ping Pong – računalniška igra podobna namiznemu tenisu [↑](#footnote-ref-5)
6. Game loop – glavna zanka igre, ki se izvaja, dokler se program ne prekine [↑](#footnote-ref-6)
7. Informacijska tehnologija - uporabo računalnikov, računalniških omrežij, programske opreme, strojne opreme in drugih tehnologij za zbiranje, obdelavo, shranjevanje in prenos informacij [↑](#footnote-ref-7)