**Šolski center Novo mesto**

**Srednja elektro šola in tehniška gimnazija**

**Šegova ulica 112**

**8000 Novo mesto**

##### SNAKE GAME

(Opis programa)

Avtor: Filip Vrabec, R2A

Mentor: Simon Vovko

Novo mesto, 27.5.2021

Vsebina

[Začetek programa 3](#_Toc74076404)

[Hrana 4](#_Toc74076405)

[Rdeča kocka 4](#_Toc74076406)

[Vijolična kocka 5](#_Toc74076407)

[Bela kocka 5](#_Toc74076408)

[Highscore 6](#_Toc74076409)

[Zaznavanje tipk 6](#_Toc74076410)

[Zvoki in Glasba 7](#_Toc74076411)

[Zvoki 7](#_Toc74076412)

[Glasba 7](#_Toc74076413)

[Trčenje in meje igre 7](#_Toc74076414)

[Sporočila in tekst 8](#_Toc74076415)

[Snake 9](#_Toc74076416)

# Začetek programa

import pygame  
import random  
from pygame import \*  
  
pygame.init*()*white = *(*255, 255, 255*)*yellow = *(*255, 255, 102*)*black = *(*0, 0, 0*)*red = *(*213, 50, 80*)*green = *(*0, 255, 0*)*blue = *(*50, 153, 213*)*purple = *(*184, 61, 186*)*dis\_width = 600  
dis\_height = 400  
  
dis = pygame.display.set\_mode*((*dis\_width, dis\_height*))*pygame.display.set\_caption*(*'Snake Ga... Wait, what am I doing... what is this?'*)*clock = pygame.time.Clock*()*snake\_block = 10  
# size of blocks  
snake\_speed = 15  
# speed of snake  
  
highscore = 0  
#Highscore  
  
font\_style = pygame.font.SysFont*(*"bahnschrift", 25*)*# game lost text  
score\_font = pygame.font.SysFont*(*"Agency FB", 35*)*# score text

Import (knjižnica) – uvozi podatke iz knjižnice v moj program

pygame.init() – inicializira modul, kar omogoči uporabo nekaj prej nedostopnih funkcij programa

(ime barve) = (R, G, B) – določi barve s ime spremenljivke in RGB barve npr. white = (255, 255, 255)

dis – zaslon, ki je omejen na za širino(dis\_width) in višino(dis\_height)

pygame.display.set\_caption – ime zaslona

clock – določi hitrost sličič

snake\_block – spremenljivka za velikost kock na zaslonu

snake\_speed – spremenljivka za hitrost kače

highscore - spremenljivka za rekord

font\_style, score\_font – spremenljivka za temo in velikost pisave

# Hrana

foodx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
foody = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0

pygame.draw.rect*(*dis, green, *[*foodx, foody, snake\_block, snake\_block*])*

if x1 == foodx and y1 == foody:  
 foodx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
 foody = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0  
 Length\_of\_snake += 1  
 Sound\_effect\_Green.play*()* snake\_speed += 1

foodx, foody – naključni koordinati za hrano, ki se določi na začetku programa

pygame.draw.rect – nariše kocko hrane na danih koordinatah

if x1 == foodx and y1 == foody – Če kocko poberemo se kocka ponovno naključno pojavi, zvok za zeleno kocko se predvaja, hitrost kače, score in dolžina kače se poveča za 1

# Rdeča kocka

bgx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
bgy = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0

pygame.draw.rect*(*dis, red, *[*bgx, bgy, snake\_block, snake\_block*])*

if x1 == bgx and y1 == bgy:  
 bgx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
 bgy = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0  
 Length\_of\_snake -= 1  
 snake\_speed -= 1  
 Sound\_effect\_Red.play*()*

bgx, bgy – naključni koordinati za rdečo kocko, ki se določi na začetku programa

pygame.draw.rect – nariše rdečo kocko na danih koordinatah

if x1 == bgx and y1 == bgy – Če kocko poberemo se kocka ponovno pojavi, zvok za rdečo kocko se predvaja, hitrost kače in score se zmanjša za 1, kača se ne zmanjša vendar se tudi povečala ne bo dokler ne poješ dovolj hran

if -5 >= *(*Length\_of\_snake - 1*)*:  
 game\_close = True

if -5 >= (Length\_of\_snake - 1): - če je (Length\_of\_snake - 1) -5 ali manj se igra konča

# Vijolična kocka

cgx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
cgy = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0

pygame.draw.rect*(*dis, purple, *[*cgx, cgy, snake\_block, snake\_block*])*

if x1 == cgx and y1 == cgy:  
 game\_close = True

if game\_close == True:  
 background\_music.stop*()* if x1 == cgx and y1 == cgy:  
 Sound\_effect\_Purple.play*()* else:  
 Sound\_effect\_lost.play*()*

cgx, cgy – naključni koordinati za vijolično kocko, ki se določi na začetku programa

pygame.draw.rect – nariše vijolično kocko na danih koordinatah

if x1 == cgx and y1 == cgy – Če kocko poberemo se kocka ponovno pojavi, igra konča in poseben zvok za izgubo se igra

# Bela kocka

srx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
sry = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0

pygame.draw.rect*(*dis, white, *[*srx, sry, snake\_block, snake\_block*])*

if x1 == srx and y1 == sry:  
 srx = round*(*random.randrange*(*0, dis\_width - snake\_block*)* / 10.0*)* \* 10.0  
 sry = round*(*random.randrange*(*0, dis\_height - snake\_block*)* / 10.0*)* \* 10.0  
 Sound\_effect\_White.play*()* if snake\_speed > 10:  
 snake\_speed = snake\_speed / 2  
# Reduce speed blocks

srx, sry – naključni koordinati za belo kocko, ki se določi na začetku programa

pygame.draw.rect – nariše belo kocko na danih koordinatah

if x1 == srx and y1 == sry – Če kocko poberemo se kocka ponovno pojavi, igra se zvok za belo kocko, program preveri če je hitrost kače čez 10 in če je se hitrost deli z 2, če hitrost ni čez 10 se nič ne zgodi

# Highscore

global highscore  
if Length\_of\_snake - 1 > highscore:  
 highscore = Length\_of\_snake - 1  
 #highscore

global – poskrbi, da se spremenljivka ne resetira vsak loop

if Length\_of\_snake - 1 > highscore: - če je trenutni score večji od highscore, naj bo highscore enak trenutnem score

# Zaznavanje tipk

for event in pygame.event.get*()*:  
 if event.type == pygame.QUIT:  
 game\_over = True  
 if event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_LEFT:  
 x1\_change = -snake\_block  
 y1\_change = 0  
 elif event.key == pygame.K\_RIGHT:  
 x1\_change = snake\_block  
 y1\_change = 0  
 elif event.key == pygame.K\_UP:  
 y1\_change = -snake\_block  
 x1\_change = 0  
 elif event.key == pygame.K\_DOWN:  
 y1\_change = snake\_block  
 x1\_change = 0  
 elif event.key == pygame.K\_ESCAPE:  
 game\_over = True  
# Game input

Med igro zaznava vnose in sicer:

* Za tipko levo se premika v levo
* Za tipko desno se premika v desno
* Za tipko dol se premika dol
* Za tipko gor se premika gor
* Za tipko ESC se igra zapre

for event in pygame.event.get*()*:  
 if event.type == pygame.KEYDOWN:  
 if event.key == pygame.K\_q:  
 game\_over = True  
 game\_close = False  
 if event.key == pygame.K\_c:  
 gameLoop*()*# End screen input

Na končnem zaslonu zaznava:

* Za tipko q zapre igro
* Za tipko c ponovno igra igro

# Zvoki in Glasba

## Zvoki

Music = True  
Sound\_effect\_Red = mixer.Sound*(*"Red.wav"*)*Sound\_effect\_Green = mixer.Sound*(*"Green.wav"*)*Sound\_effect\_White = mixer.Sound*(*"White.wav"*)*Sound\_effect\_Purple = mixer.Sound*(*"Purple.wav"*)*Sound\_effect\_lost = mixer.Sound*(*"Lost.wav"*)*# music & sounds

Spremenljivko Music nastavi na True

Vnese vse zvoke iz mape, da jih uporabi v programu npr. pri koncu igre

(ime zvoka) = mixer.sound(ime datoteke)

if game\_close == True:  
 background\_music.stop*()* if x1 == cgx and y1 == cgy:  
 Sound\_effect\_Purple.play*()* else:  
 Sound\_effect\_lost.play*()*

Glasba za ozadje se neha predvajati

Zvok za izgubo igro se predvaja

Predvaja se zvok drugačen če si izgubil zaradi vijolične kocke

## Glasba

if Music== True:  
 background\_music = mixer.Sound*(*"Background.wav"*)* background\_music.play*()* background\_music.set\_volume*(*0.3*)* Music = False  
# Background music

Preveri če je spremenljivka Music == True in, če je predvaja glasbo za ozadje

Vnese glasbo iz mape, da jih uporabi v programu

background\_music.play() – glasbo predvaja

background\_music.set\_volume(0.3) – nastavi zvok glasbe

Spremeni Music na False

# Trčenje in meje igre

if x1 >= dis\_width or x1 < 0 or y1 >= dis\_height or y1 < 0:  
 game\_close = True

for x in snake\_List*[*:-1*]*:  
 if x == snake\_Head:  
 game\_close = True  
 # if snake crash in snake lose game

Če se kača zabije vase izgubiš igro

Če gre kača iz zaslona izgubiš igro

# Sporočila in tekst

def Your\_score*(*score*)*:  
 value = score\_font.render*(*"Your Score: " + str*(*score*)*, True, yellow*)* dis.blit*(*value, *[*0, 0*])* # score tracker  
  
def message*(*msg, color*)*:  
 mesg = font\_style.render*(*msg, True, color*)* dis.blit*(*mesg, *[*dis\_width / 6, dis\_height / 3*])*def message2*(*score*)*:  
 value = score\_font.render*(*"Your Highscore was: " + str*(*score*)*, True, white*)* dis.blit*(*value, *[*dis\_width / 5, dis\_height / 2*])*

message*(*"You Lost! Press C-Play Again or Q-Quit", red*)*message2*(*highscore*)*Your\_score*(*Length\_of\_snake - 1*)*pygame.display.update*()*# Displays & Messages

Sporočila in tekst na zaslonu po igri:

* Tvoj score
* Tvoj highscore
* Sporočilo da si igro izgubil

Your\_score*(*Length\_of\_snake - 1*)*# score  
  
pygame.display.update*()*

Sporočila med igro:

* Tvoj score

# Snake

def our\_snake*(*snake\_block, snake\_list*)*:  
 for x in snake\_list:  
 pygame.draw.rect*(*dis, black, *[*x*[*0*]*, x*[*1*]*, snake\_block, snake\_block*])*

snake\_List = *[]*Length\_of\_snake = 1  
# number of snake blocks

snake\_Head = *[]*snake\_Head.append*(*x1*)*snake\_Head.append*(*y1*)*snake\_List.append*(*snake\_Head*)*# The players character on the screen

if len*(*snake\_List*)* > Length\_of\_snake:  
 del snake\_List*[*0*]*

our\_snake*(*snake\_block, snake\_List*)*# game character

pygame.display.update*()*

Kača se riše in premika s tem, da nariše kocko kjer se premakne in prejšnjo pobriše, ko se Lenght\_of\_snake poveča se poveča tudi koliko kock obdrži preden se pobrišejo.