**Tallinna Tööstushariduskeskus  
Logistika IT süsteemide spetsialist**

**Projekti „Labürint“**Praktiline töö

**Juhendaja**: Marina Oleinik **Koostaja**: Julia Postnikova **Rühm**: LOGITpv23

**Tallinn 2024**

Sisukord

[Sissejuhatus 3](#_Toc168299032)

[Teoreetiline osa 4](#_Toc168299033)

# Sissejuhatus

Selle projekti eesmärk on luua hea ja huvitav tähelepanelikkuse mäng lastele ja täiskasvanutele. Tehakse labürint mõningate modifikatsioonidega. Mängus on kolm taset ja pärast mängu läbimist küsitakse, kas soovite uuesti mängida või väljuda. On ka lõbus muusika, et sul ei oleks igav.

# Teoreetiline osa

Programm kasutab Pythoni keelt.

See kasutab ka Pygame'i mooduleid.

Python ilmus 20. veebruaril 1991. Autor Guido van Rossum.

Pygame ilmus 28. oktoobril 2000. Autor Pete Shinners.

Minu kasutatud moodulid on järgmised:

import pygame

import random

Programmi põhikonstruktsioonid:

Loops

while True:

screen.fill(lBlue)

end\_text = font.render("Sa oled kõik tasemed läbinud!", True, black)

continue\_text = font.render("Kas soovite jätkata (J) või väljuda (V)?", True, black)

screen.blit(end\_text, (scr\_w // 2 - end\_text.get\_width() // 2, scr\_h // 2 - 40))

screen.blit(continue\_text, (scr\_w // 2 - continue\_text.get\_width() // 2, scr\_h // 2 + 10))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

quit()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_j:

restart\_game()

return

elif event.key == pygame.K\_v:

pygame.quit()

quit()

def, mis võimaldab luua uusi funktsioone

def create\_cell(x, y):

return {"x": x, "y": y, "walls": [True, True, True, True], "visited": False}

Klahvide juhtimine

if keys[pygame.K\_LEFT] and plr\_x > plr\_r:

new\_x -= plr\_speed

if keys[pygame.K\_RIGHT] and plr\_x < scr\_w - plr\_r:

new\_x += plr\_speed

if keys[pygame.K\_UP] and plr\_y > plr\_r:

new\_y -= plr\_speed

if keys[pygame.K\_DOWN] and plr\_y < scr\_h - plr\_r:

new\_y += plr\_speed

Audio

pygame.mixer.music.load('audio.mp3')

pygame.mixer.music.play()

# Praktiline osa

Kui ma hakkasin seda projekti tegema, joonistasin kõigepealt koodi välimuse ja selle, mida ma

tahtsin koodis näha. Mõtlesin, et mängul oleks mitu taset ja et inimene valiks, kas ta tahab uuesti mängida või lõpetada. Siis hakkasin koodi kirjutama, parandasin palju ja lisasin palju uusi asju.

Siin on kood ise:

import pygame

import random

pygame.init()

scr\_w, scr\_h = 640, 480

cell\_size = 40

cols, rows = scr\_w // cell\_size, scr\_h // cell\_size

screen = pygame.display.set\_mode((scr\_w, scr\_h))

pygame.display.set\_caption('Labürint')

pygame.mixer.music.load('audio.mp3')

pygame.mixer.music.play()

lBlue = (153, 255, 255)

lPurple = (204, 153, 255)

blue = (51, 153, 255)

red = (255, 0, 0)

black = (0, 0, 0)

purple = (76, 0, 153)

plr\_r = 10

plr\_speed = 5

start\_x, start\_y = 0, rows - 1

end\_x, end\_y = cols - 1, 0

plr\_x = start\_x \* cell\_size + plr\_r

plr\_y = start\_y \* cell\_size + plr\_r

clock = pygame.time.Clock()

level = 0

max\_levels = 3

font = pygame.font.SysFont(None, 36)

def create\_cell(x, y):

return {"x": x, "y": y, "walls": [True, True, True, True], "visited": False}

def draw\_cell(cell, cell\_size, screen, blue):

x = cell["x"] \* cell\_size

y = cell["y"] \* cell\_size

if cell["walls"][0]:

pygame.draw.line(screen, blue, (x, y), (x + cell\_size, y), 2)

if cell["walls"][1]:

pygame.draw.line(screen, blue, (x + cell\_size, y), (x + cell\_size, y + cell\_size), 2)

if cell["walls"][2]:

pygame.draw.line(screen, blue, (x + cell\_size, y + cell\_size), (x, y + cell\_size), 2)

if cell["walls"][3]:

pygame.draw.line(screen, blue, (x, y + cell\_size), (x, y), 2)

def create\_grid():

return [[create\_cell(x, y) for x in range(cols)] for y in range(rows)]

def generate\_maze():

grid = create\_grid()

stack = []

current\_cell = grid[start\_y][start\_x]

current\_cell["visited"] = True

while True:

neighbors = []

x, y = current\_cell["x"], current\_cell["y"]

if y > 0 and not grid[y - 1][x]["visited"]:

neighbors.append(grid[y - 1][x])

if y < rows - 1 and not grid[y + 1][x]["visited"]:

neighbors.append(grid[y + 1][x])

if x > 0 and not grid[y][x - 1]["visited"]:

neighbors.append(grid[y][x - 1])

if x < cols - 1 and not grid[y][x + 1]["visited"]:

neighbors.append(grid[y][x + 1])

if neighbors:

next\_cell = random.choice(neighbors)

stack.append(current\_cell)

if next\_cell["x"] == current\_cell["x"] + 1:

current\_cell["walls"][1] = False

next\_cell["walls"][3] = False

elif next\_cell["x"] == current\_cell["x"] - 1:

current\_cell["walls"][3] = False

next\_cell["walls"][1] = False

elif next\_cell["y"] == current\_cell["y"] + 1:

current\_cell["walls"][2] = False

next\_cell["walls"][0] = False

elif next\_cell["y"] == current\_cell["y"] - 1:

current\_cell["walls"][0] = False

next\_cell["walls"][2] = False

current\_cell = next\_cell

current\_cell["visited"] = True

elif stack:

current\_cell = stack.pop()

else:

break

return grid

def reset\_player():

global plr\_x, plr\_y

plr\_x = start\_x \* cell\_size + plr\_r

plr\_y = start\_y \* cell\_size + plr\_r

def next\_level():

global level, grid

level += 1

if level < max\_levels:

grid = generate\_maze()

reset\_player()

else:

end\_game()

def end\_game():

pygame.mixer.music.stop()

while True:

screen.fill(lBlue)

end\_text = font.render("Sa oled kõik tasemed läbinud!", True, black)

continue\_text = font.render("Kas soovite jätkata (J) või väljuda (V)?", True, black)

screen.blit(end\_text, (scr\_w // 2 - end\_text.get\_width() // 2, scr\_h // 2 - 40))

screen.blit(continue\_text, (scr\_w // 2 - continue\_text.get\_width() // 2, scr\_h // 2 + 10))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

quit()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_j:

restart\_game()

return

elif event.key == pygame.K\_v:

pygame.quit()

quit()

def restart\_game():

global level, grid

level = 0

grid = generate\_maze()

reset\_player()

pygame.mixer.music.play()

grid = generate\_maze()

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

quit()

keys = pygame.key.get\_pressed()

new\_x, new\_y = plr\_x, plr\_y

if keys[pygame.K\_LEFT] and plr\_x > plr\_r:

new\_x -= plr\_speed

if keys[pygame.K\_RIGHT] and plr\_x < scr\_w - plr\_r:

new\_x += plr\_speed

if keys[pygame.K\_UP] and plr\_y > plr\_r:

new\_y -= plr\_speed

if keys[pygame.K\_DOWN] and plr\_y < scr\_h - plr\_r:

new\_y += plr\_speed

plr\_rect = pygame.Rect(new\_x - plr\_r, new\_y - plr\_r, plr\_r \* 2, plr\_r \* 2)

collision = False

for row in grid:

for cell in row:

cx, cy = cell["x"] \* cell\_size, cell["y"] \* cell\_size

if cell["walls"][0] and plr\_rect.colliderect(pygame.Rect(cx, cy, cell\_size, 2)):

collision = True

if cell["walls"][1] and plr\_rect.colliderect(pygame.Rect(cx + cell\_size - 2, cy, 2, cell\_size)):

collision = True

if cell["walls"][2] and plr\_rect.colliderect(pygame.Rect(cx, cy + cell\_size - 2, cell\_size, 2)):

collision = True

if cell["walls"][3] and plr\_rect.colliderect(pygame.Rect(cx, cy, 2, cell\_size)):

collision = True

if not collision:

plr\_x, plr\_y = new\_x, new\_y

if (plr\_x // cell\_size == end\_x and plr\_y // cell\_size == end\_y):

next\_level()

screen.fill(lBlue)

for row in grid:

for cell in row:

draw\_cell(cell, cell\_size, screen, blue)

pygame.draw.rect(screen, black, (start\_x \* cell\_size, start\_y \* cell\_size, cell\_size, cell\_size))

pygame.draw.rect(screen, red, (end\_x \* cell\_size, end\_y \* cell\_size, cell\_size, cell\_size))

pygame.draw.circle(screen, lPurple, (plr\_x, plr\_y), plr\_r)

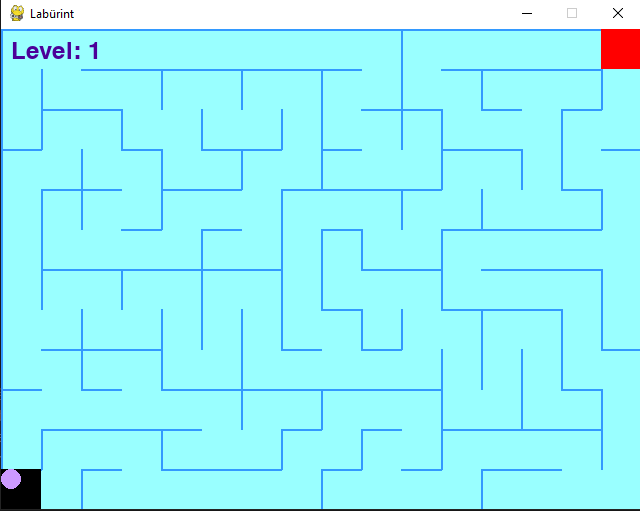
level\_text = font.render(f"Level: {level + 1}", True, purple)

screen.blit(level\_text, (10, 10))

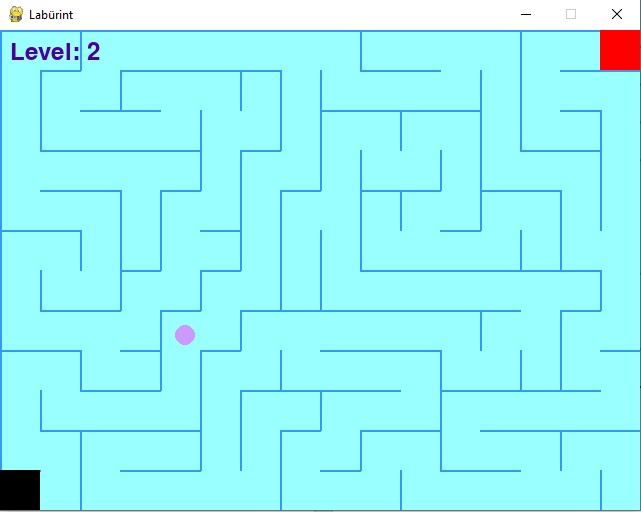
pygame.display.update()

clock.tick(60)

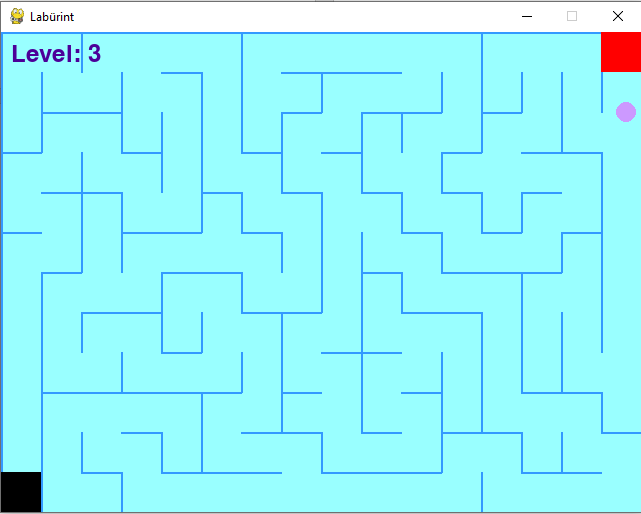
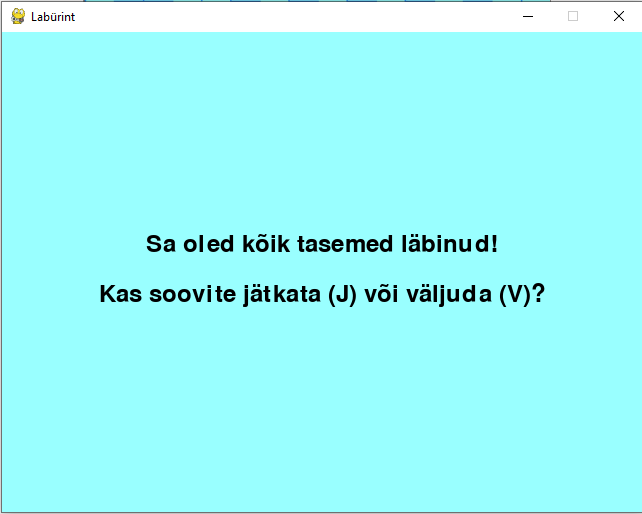
# Mängu välimus



Pilt 1 "Kujundus loomine"



Pilt 2 "Kujundus loomine"



Pilt 3 "Kujundus loomine"

Pilt 4 "Kujundus loomine"

Pildid näitavad, kuidas mäng on kujutatud. Must ruut on see, kust me alustame, punane ruut on see, kuhu me peame minema. Samuti on kirjas, millisel tasemel me oleme. Kui oleme mängu läbinud, siis on küsimus, mida me tahame edasi teha.

# Kokkuvõte

Mäng töötab hästi ja ma olen sellega rahul. Tulevikus lisaksin veel funktsioone, aga ma ei ole veel ühtegi välja mõelnud.

# Allikad

# Bibliograafia

Rossum, G. v. (20. veebruar 1991. a.). Allikas: https://ru.wikipedia.org/wiki/Python

Shinners, P. (28. oktoober 2000. a.). Allikas: https://ru.wikipedia.org/wiki/Pygame