**Tallinna Tööstushariduskeskus  
Logistika IT süsteemide spetsialist**

**Projekti „Wordle“ Pythoni keeles loomine**Praktiline töö

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

**Tallinn 2024**

Sisukord

[Введение 3](#_Toc166486721)

[Sissejuhatus 3](#_Toc166486722)

[Теоретическая часть 4](#_Toc166486723)

[Teoreetiline osa 5](#_Toc166486724)

[Практическая часть 6](#_Toc166486725)

[Praktiline osa 8](#_Toc166486726)

[Внешний вид игры 12](#_Toc166486727)

[Mängu välimus 15](#_Toc166486728)

[Заключение 18](#_Toc166486729)

[Kokkuvõte 18](#_Toc166486730)

[Allikad 19](#_Toc166486731)

[Bibliograafia 19](#_Toc166486732)

# Введение

Цель этого проекта создать хорошую игру для детей и взрослых. Игра сделана по подобию „Wordle“, но разница в том, что в мою игру можно играть много раз, а в „Wordle“ нет. Также эта игра сделана для того, чтобы учить новые слова на эстонском языке и проверить себя на грамотность. В этой игре нужно вводить текст в каждую ячейку, а для удобства в игре добавлена клавиатура. Так же в игре есть функция запустить новую игру.

# Sissejuhatus

Selle projekti eesmärk on luua hea mäng lastele ja täiskasvanutele. Mäng on sarnane "Wordle'", kuid erinevus seisneb selles, et minu mängu saab mängida mitu korda, samas kui "Wordle" ei saa. See mäng on tehtud ka selleks, et õppida uusi sõnu eesti keeles ja testida oma kirjaoskust. Selles mängus tuleb igasse lahtrisse sisestada tekst ja mugavuse huvides on mängul klaviatuur. Samuti on mängus funktsioon uue mängu alustamiseks.

# Теоретическая часть

В программе используется язык Python.

Python появился в 20 февраля 1991. Автор Гвидо ван Россум.

Модули, которые я использовала:

import tkinter as tk

import random

Основные конструкции в программе:

Циклы

for i, täht in enumerate(arvamine):

if täht==sõna[i]:

tulemus.append("#32CD32")

elif täht in sõna:

tulemus.append("#FFFF00")

else:

tulemus.append("#080808")

return tulemus

Открывание файла

def laadi(sõnad\_fail):

with open(sõnad\_fail, 'r', encoding='utf-8') as fail:

sõnad=[rivi.strip().upper() for rivi in fail if len(rivi.strip())==5]

return sõnad

И также используется def, который позволяет создавать новые функции

def liidest(arvamine, tulemus):

for i, täht in enumerate(arvamine):

sisestus=sisestused[üritus][i]

sisestus.delete(0, 'end')

sisestus.insert('end', täht)

sisestus.config(disabledforeground=tulemus[i], state='disabled')

# Teoreetiline osa

Programmis kasutatakse Pythoni keelt.

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

Kasutatud moodulid on järgmised:

import tkinter as tk

import random

Programmi põhikonstruktsioonid:

Loops

for i, täht in enumerate(arvamine):

if täht==sõna[i]:

tulemus.append("#32CD32")

elif täht in sõna:

tulemus.append("#FFFF00")

else:

tulemus.append("#080808")

return tulemus

Faili avamine

def laadi(sõnad\_fail):

with open(sõnad\_fail, 'r', encoding='utf-8') as fail:

sõnad=[rivi.strip().upper() for rivi in fail if len(rivi.strip())==5]

return sõnad

Ja kasutatakse ka def, mis võimaldab luua uusi funktsioone

def liidest(arvamine, tulemus):

for i, täht in enumerate(arvamine):

sisestus=sisestused[üritus][i]

sisestus.delete(0, 'end')

sisestus.insert('end', täht)

sisestus.config(disabledforeground=tulemus[i], state='disabled')

# Практическая часть

Когда я начала делать этот проект, я с начало нарисовала внешний вид, и чтобы я хотела видеть в коде. Затем я начала писать код, много исправляла и добавляла много нового. Не все ошибки я смогла исправить, но всё равно код работает хорошо.

Вот сам код:

import tkinter as tk

import random

def laadi(sõnad\_fail):

with open(sõnad\_fail, 'r', encoding='utf-8') as fail:

sõnad=[rivi.strip().upper() for rivi in fail if len(rivi.strip())==5]

return sõnad

def vali(sõnad):

return random.choice(sõnad)

def kontrolli(arvamine, sõna):

tulemus=[]

for i, täht in enumerate(arvamine):

if täht==sõna[i]:

tulemus.append("#32CD32")

elif täht in sõna:

tulemus.append("#FFFF00")

else:

tulemus.append("#080808")

return tulemus

def liidest(arvamine, tulemus):

for i, täht in enumerate(arvamine):

sisestus=sisestused[üritus][i]

sisestus.delete(0, 'end')

sisestus.insert('end', täht)

sisestus.config(disabledforeground=tulemus[i], state='disabled')

def saatmine():

global üritus, hetke

arvamine=[sisestus.get().upper() for sisestus in sisestused[üritus]]

if len(arvamine)!=5:

silt.config(text="Sisestage täpselt 5 tähte!", fg="#ff0000")

return

tulemus=kontrolli(arvamine, hetke)

liidest(arvamine, tulemus)

if arvamine==list(hetke) or üritus == 5:

if arvamine!=list(hetke):

silt.config(text=f"Õige sõna: {''.join(hetke)}", fg="#ff0000")

else:

silt.config(text="")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='disabled')

else:

üritus += 1

jäänud.config(text=f"Jäänud püüdeid: {6 - üritus}")

for sisestus in sisestused[üritus]:

sisestus.focus\_set()

def algus():

global hetke, üritus

hetke=vali(sõnad)

üritus=0

jäänud.config(text="")

silt.config(text="")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='normal')

sisestus.delete(0, 'end')

def sisesta\_täht(täht):

focused=aken.focus\_get()

if focused:

focused.insert(tk.END, täht)

sõnad=laadi('Sõnad.txt')

hetke=vali(sõnad)

üritus=0

aken=tk.Tk()

aken.title("Wordle")

aken.geometry("1000x1000")

aken.config(bg='#87ceeb')

põhiraam=tk.Frame(aken, bg="#87ceeb")

põhiraam.pack(expand=True)

sisestused=[[tk.Entry(põhiraam, font=('Tahoma', 24), width=4, bg="#928aff", fg="#87ceeb", bd=2, relief=tk.GROOVE) for \_ in range(5)] for \_ in range(6)]

for i, rida in enumerate(sisestused):

for j, sisestus in enumerate(rida):

sisestus.grid(row=i, column=j, padx=3, pady=5)

saatmis=tk.Button(põhiraam, text="Saada", command=saatmine, bg="#928aff", fg="#87ceeb")

saatmis.grid(row=6, column=0, columnspan=3, padx=5, pady=10)

jäänud=tk.Label(põhiraam, text="", font=('Tahoma', 24), bg="#87ceeb", fg="#ffffff")

jäänud.grid(row=7, column=0, columnspan=5, padx=10, pady=5)

silt=tk.Label(põhiraam, text="", font=('Tahoma', 24), bg="#87ceeb", fg="#ffffff")

silt.grid(row=8, column=0, columnspan=5, padx=10, pady=5)

nupp=tk.Button(põhiraam, text="Alusta uuesti", command=algus, bg="#928aff", fg="#87ceeb")

nupp.grid(row=6, column=1, columnspan=9, padx=5, pady=5)

tähestik="a b d e f g h i j k l m n o p r s š z ž t u v õ ä ö ü".split()

alphabet\_frame=tk.Frame(põhiraam, bg="#87ceeb")

alphabet\_frame.grid(row=9, column=0, columnspan=5, pady=5)

for i, täht in enumerate(tähestik):

tk.Button(alphabet\_frame, text=täht, font=('Tahoma', 12), width=2, bg="#928aff", fg="#87ceeb", bd=2, relief=tk.GROOVE, command=lambda t=täht: sisesta\_täht(t)).grid(row=0, column=i, padx=3)

aken.mainloop()

# Praktiline osa

Kui ma seda projekti alustasin, siis kõigepealt joonistasin projekti välimuse ja selle, mida ma tahtsin koodis näha. Siis hakkasin koodi kirjutama, parandasin palju asju ja lisasin palju uusi asju. Kõiki vigu ma ei suutnud parandada, kuid kood töötab siiski hästi.

Siin on kood ise:

import tkinter as tk

import random

def laadi(sõnad\_fail):

with open(sõnad\_fail, 'r', encoding='utf-8') as fail:

sõnad=[rivi.strip().upper() for rivi in fail if len(rivi.strip())==5]

return sõnad

def vali(sõnad):

return random.choice(sõnad)

def kontrolli(arvamine, sõna):

tulemus=[]

for i, täht in enumerate(arvamine):

if täht==sõna[i]:

tulemus.append("#32CD32")

elif täht in sõna:

tulemus.append("#FFFF00")

else:

tulemus.append("#080808")

return tulemus

def liidest(arvamine, tulemus):

for i, täht in enumerate(arvamine):

sisestus=sisestused[üritus][i]

sisestus.delete(0, 'end')

sisestus.insert('end', täht)

sisestus.config(disabledforeground=tulemus[i], state='disabled')

def saatmine():

global üritus, hetke

arvamine=[sisestus.get().upper() for sisestus in sisestused[üritus]]

if len(arvamine)!=5:

silt.config(text="Sisestage täpselt 5 tähte!", fg="#ff0000")

return

tulemus=kontrolli(arvamine, hetke)

liidest(arvamine, tulemus)

if arvamine==list(hetke) or üritus == 5:

if arvamine!=list(hetke):

silt.config(text=f"Õige sõna: {''.join(hetke)}", fg="#ff0000")

else:

silt.config(text="")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='disabled')

else:

üritus += 1

jäänud.config(text=f"Jäänud püüdeid: {6 - üritus}")

for sisestus in sisestused[üritus]:

sisestus.focus\_set()

def algus():

global hetke, üritus

hetke=vali(sõnad)

üritus=0

jäänud.config(text="")

silt.config(text="")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='normal')

sisestus.delete(0, 'end')

def sisesta\_täht(täht):

focused=aken.focus\_get()

if focused:

focused.insert(tk.END, täht)

sõnad=laadi('Sõnad.txt')

hetke=vali(sõnad)

üritus=0

aken=tk.Tk()

aken.title("Wordle")

aken.geometry("1000x1000")

aken.config(bg='#87ceeb')

põhiraam=tk.Frame(aken, bg="#87ceeb")

põhiraam.pack(expand=True)

sisestused=[[tk.Entry(põhiraam, font=('Tahoma', 24), width=4, bg="#928aff", fg="#87ceeb", bd=2, relief=tk.GROOVE) for \_ in range(5)] for \_ in range(6)]

for i, rida in enumerate(sisestused):

for j, sisestus in enumerate(rida):

sisestus.grid(row=i, column=j, padx=3, pady=5)

saatmis=tk.Button(põhiraam, text="Saada", command=saatmine, bg="#928aff", fg="#87ceeb")

saatmis.grid(row=6, column=0, columnspan=3, padx=5, pady=10)

jäänud=tk.Label(põhiraam, text="", font=('Tahoma', 24), bg="#87ceeb", fg="#ffffff")

jäänud.grid(row=7, column=0, columnspan=5, padx=10, pady=5)

silt=tk.Label(põhiraam, text="", font=('Tahoma', 24), bg="#87ceeb", fg="#ffffff")

silt.grid(row=8, column=0, columnspan=5, padx=10, pady=5)

nupp=tk.Button(põhiraam, text="Alusta uuesti", command=algus, bg="#928aff", fg="#87ceeb")

nupp.grid(row=6, column=1, columnspan=9, padx=5, pady=5)

tähestik="a b d e f g h i j k l m n o p r s š z ž t u v õ ä ö ü".split()

alphabet\_frame=tk.Frame(põhiraam, bg="#87ceeb")

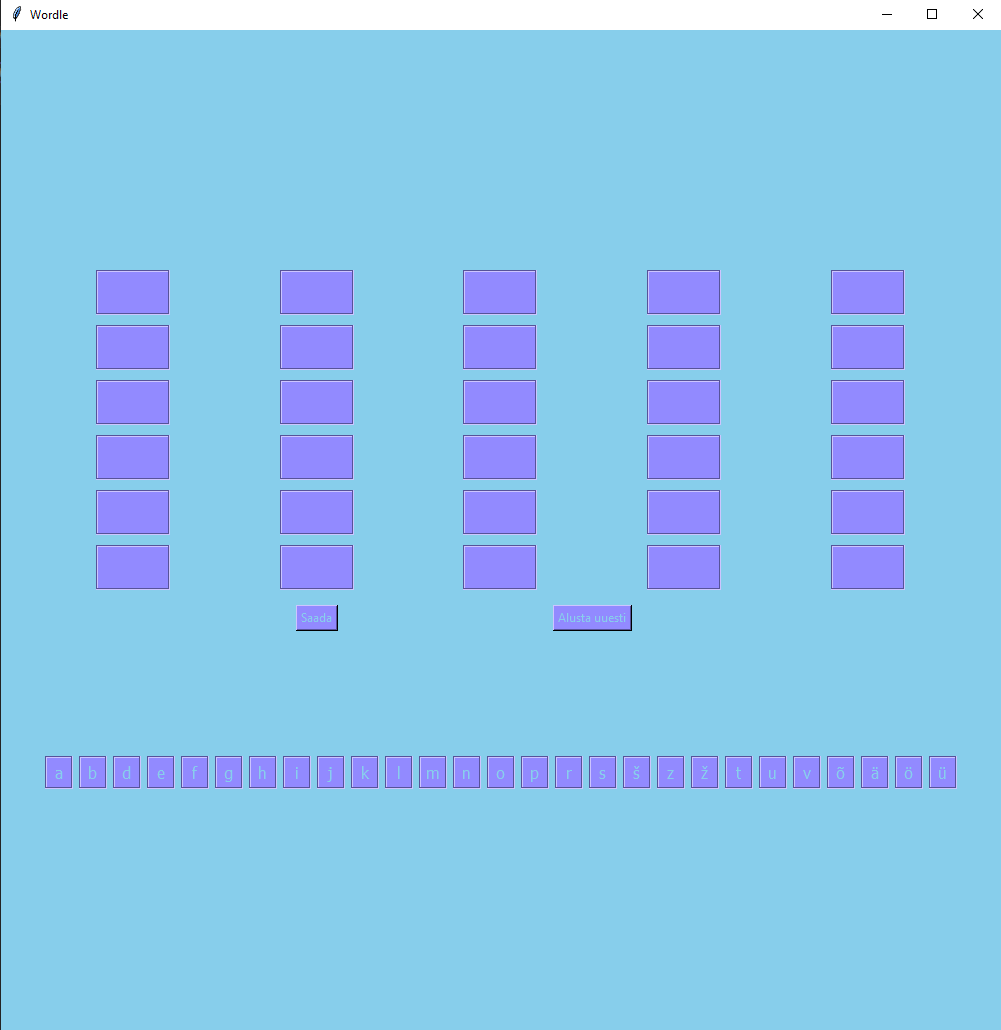
alphabet\_frame.grid(row=9, column=0, columnspan=5, pady=5)

for i, täht in enumerate(tähestik):

tk.Button(alphabet\_frame, text=täht, font=('Tahoma', 12), width=2, bg="#928aff", fg="#87ceeb", bd=2, relief=tk.GROOVE, command=lambda t=täht: sisesta\_täht(t)).grid(row=0, column=i, padx=3)

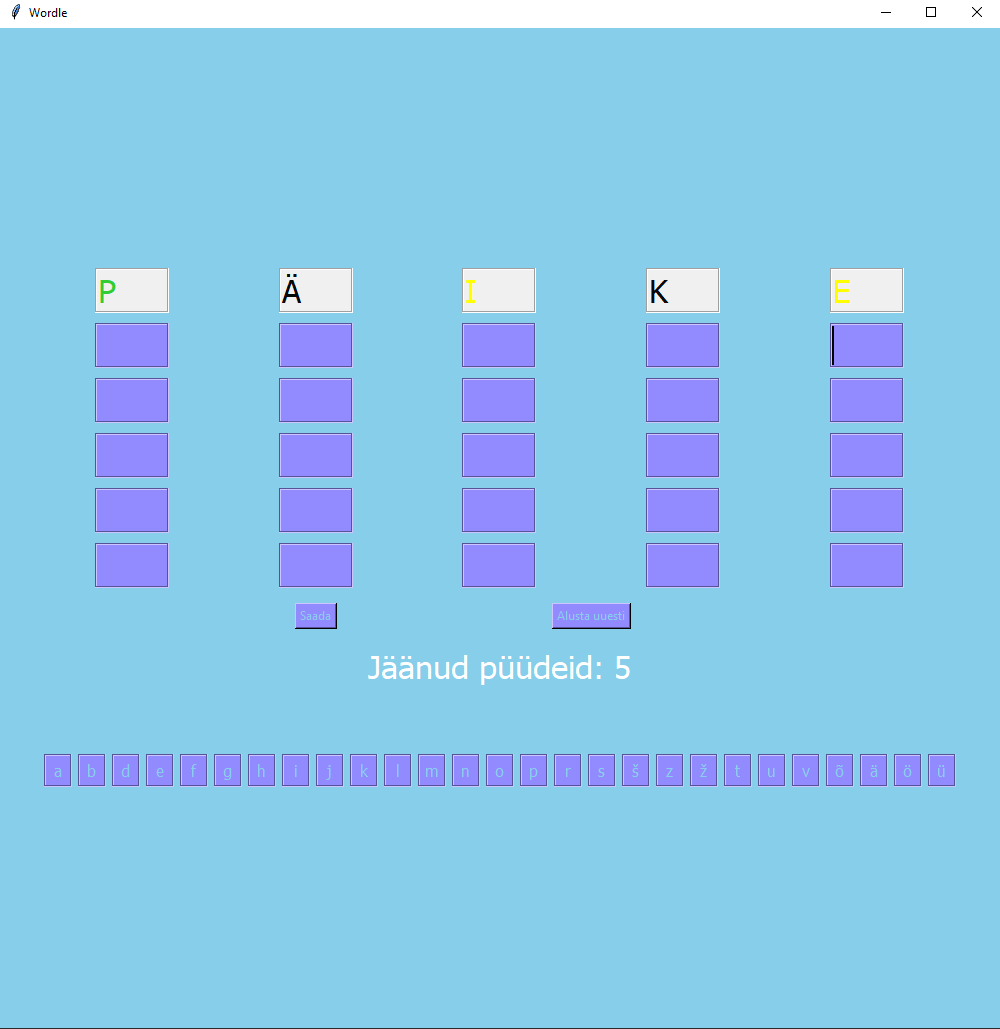
aken.mainloop()

# Внешний вид игры



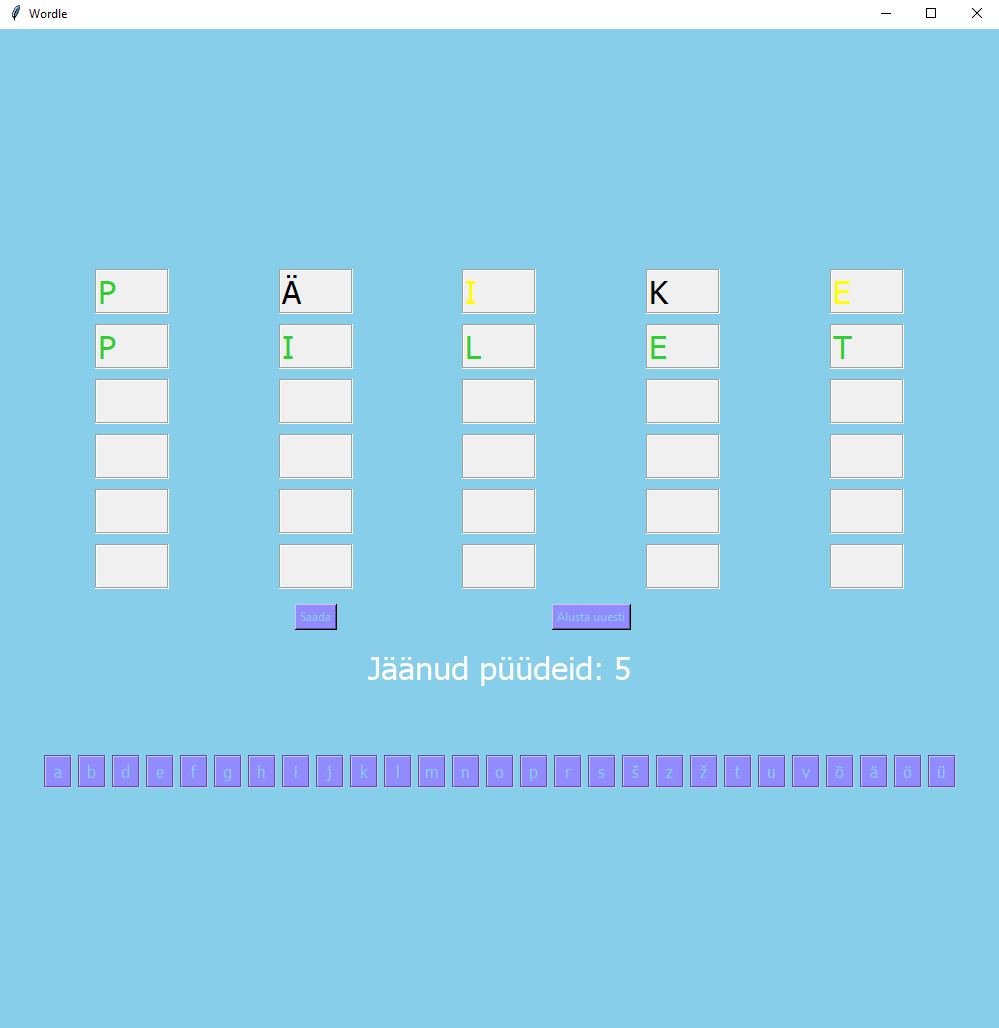
Pilt 1 "Kujundis loomine"

Чтобы ввести букву нужно нажимать на каждую ячейку отдельно, под ячейками находится клавиатура с помощью, которой удобнее вводить слова.

На первой картинке показан внешний вид игры.

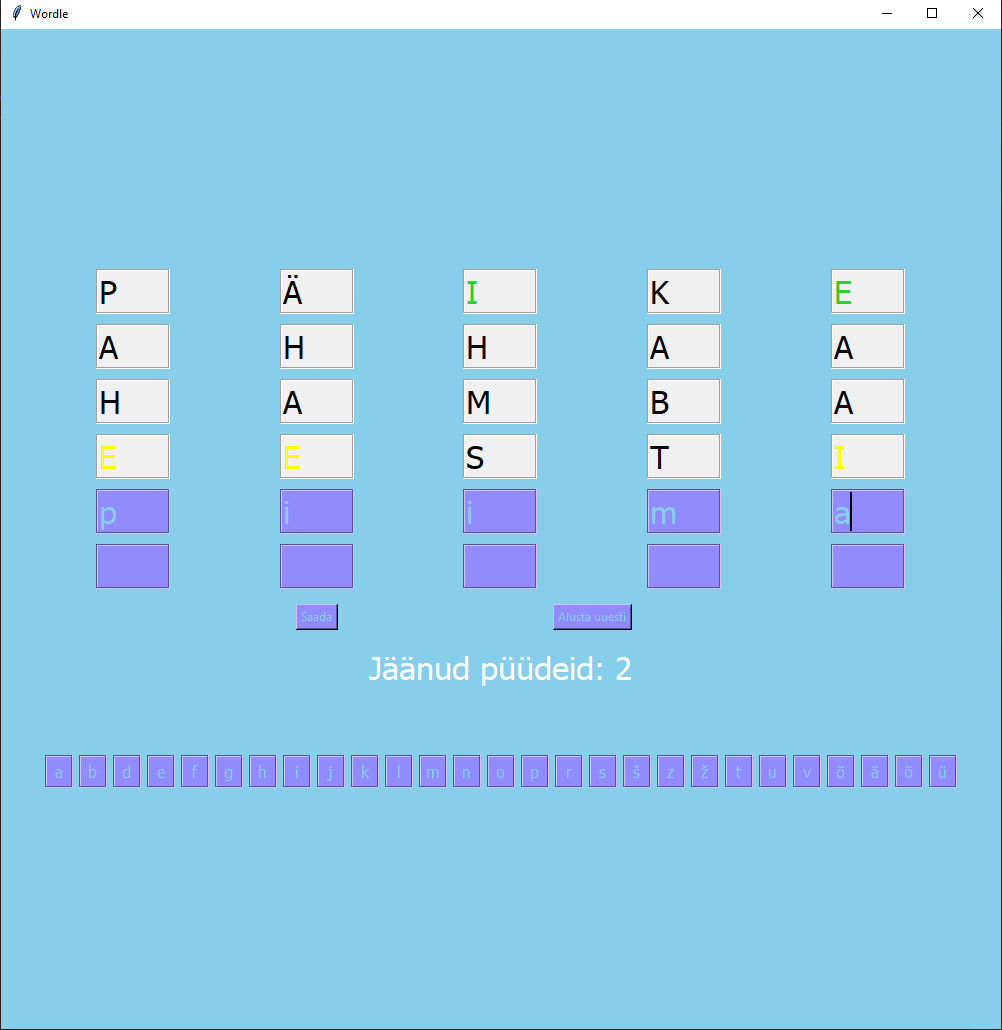
Pilt 2 "Kujundis loomine"

На второй картинке изображено что будет если ввести слово, зелёным цветом подсвечивается правильная бука и на правильном месте, жёлтый означает что буква правильная, но не на том месте, а чёрный что буква не правильная. Внизу написано сколько осталось попыток.



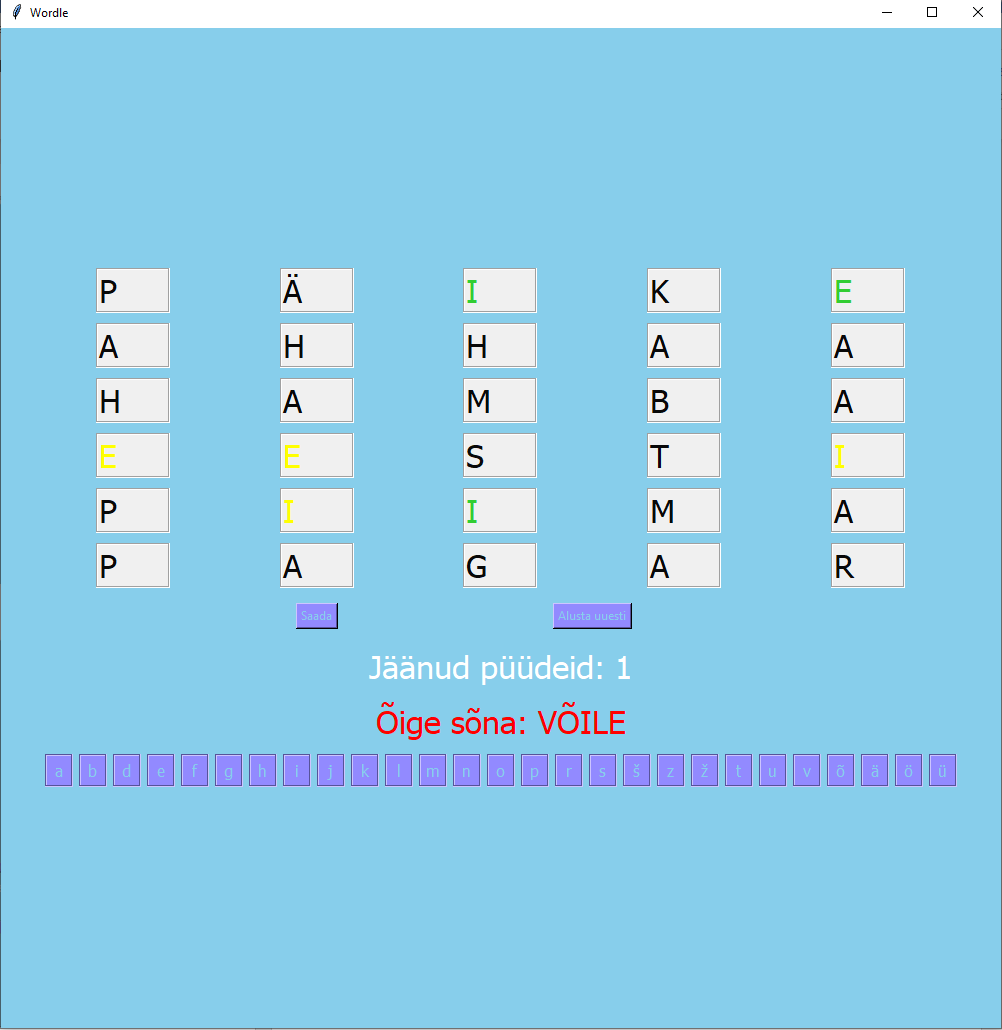
Pilt 3 "Kujundis loomine"

На картинке показано если слово угадали.



На картинке показано как выглядит ввод букв, а также сколько попыток осталось.

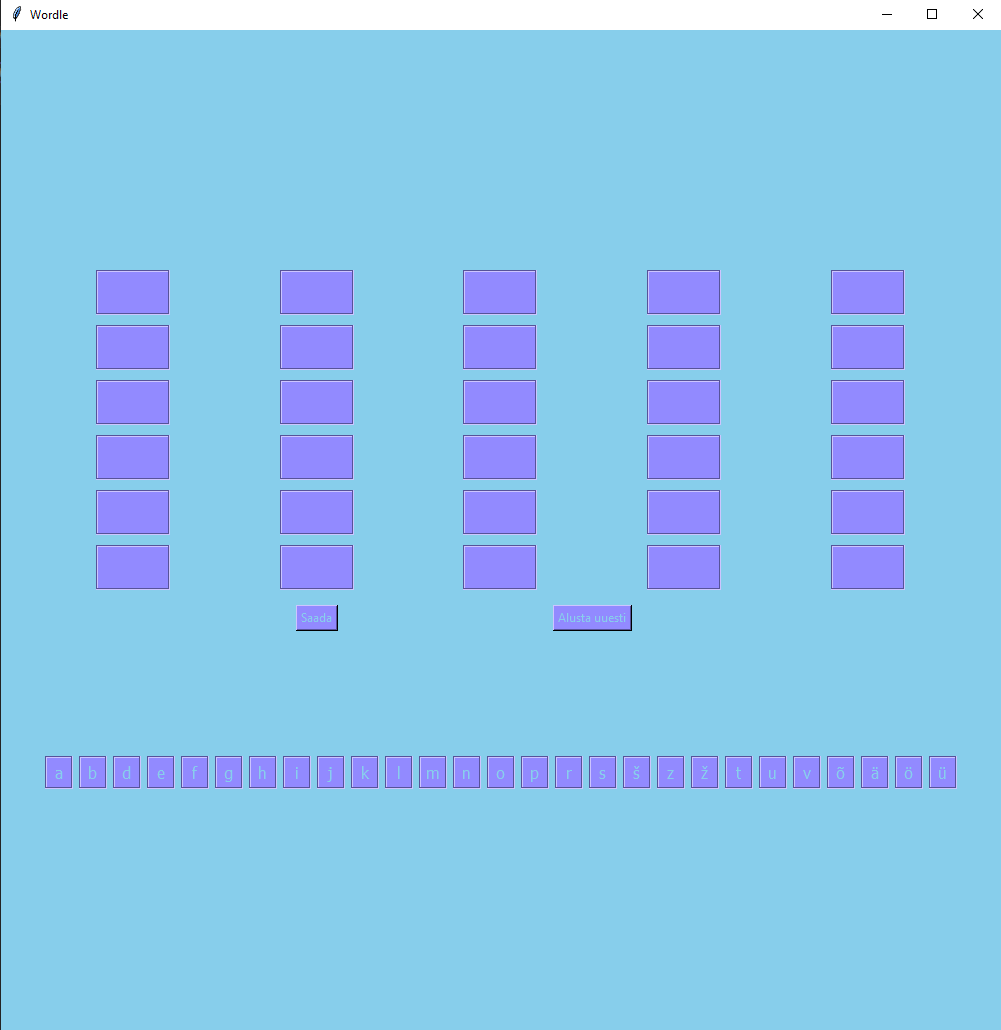
Pilt 4 "Kujundis loomine"

****

Pilt 5 "Kujundis loomine"

На картинке показан правильный ответ если вы неправильно угадали шесть раз.

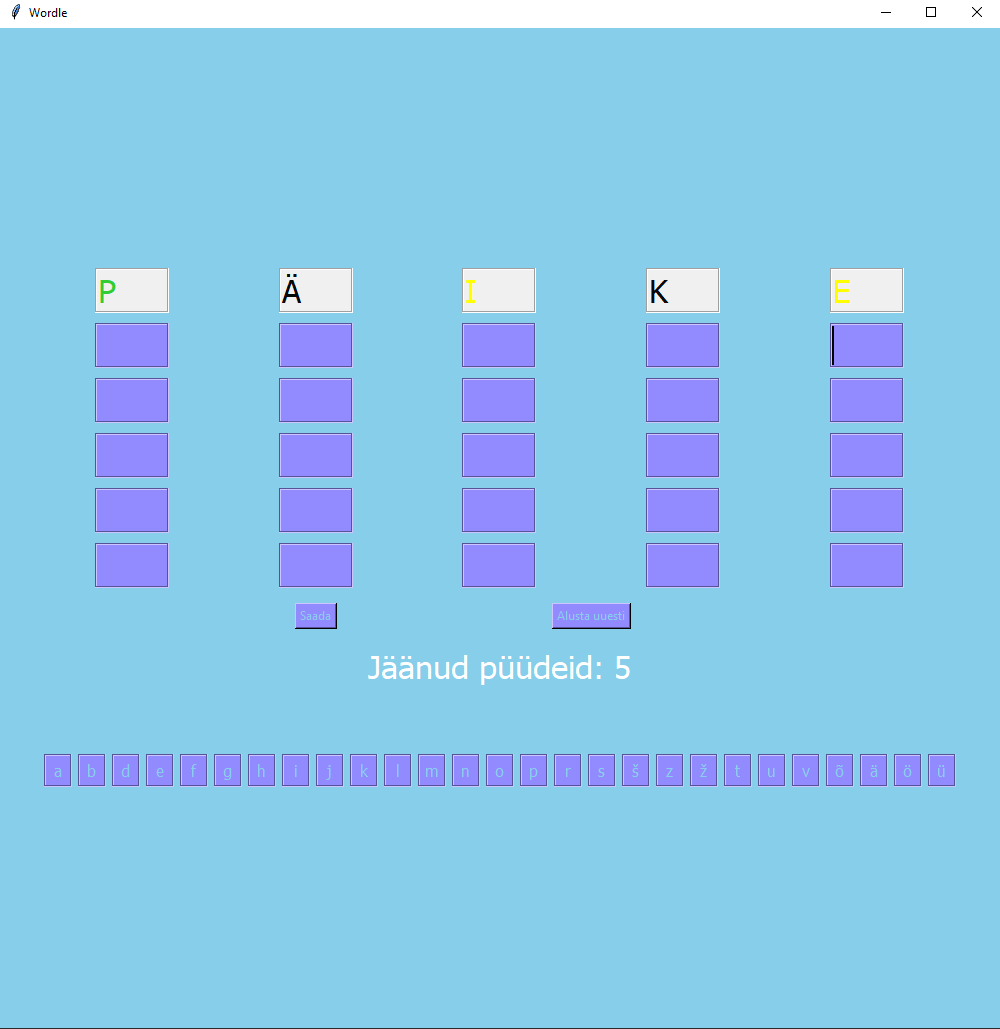
Mängu välimus



Tähe sisestamiseks tuleb klõpsata igal lahtril eraldi, lahtrite all on klaviatuur, mille abil on mugavam sõnu sisestada.

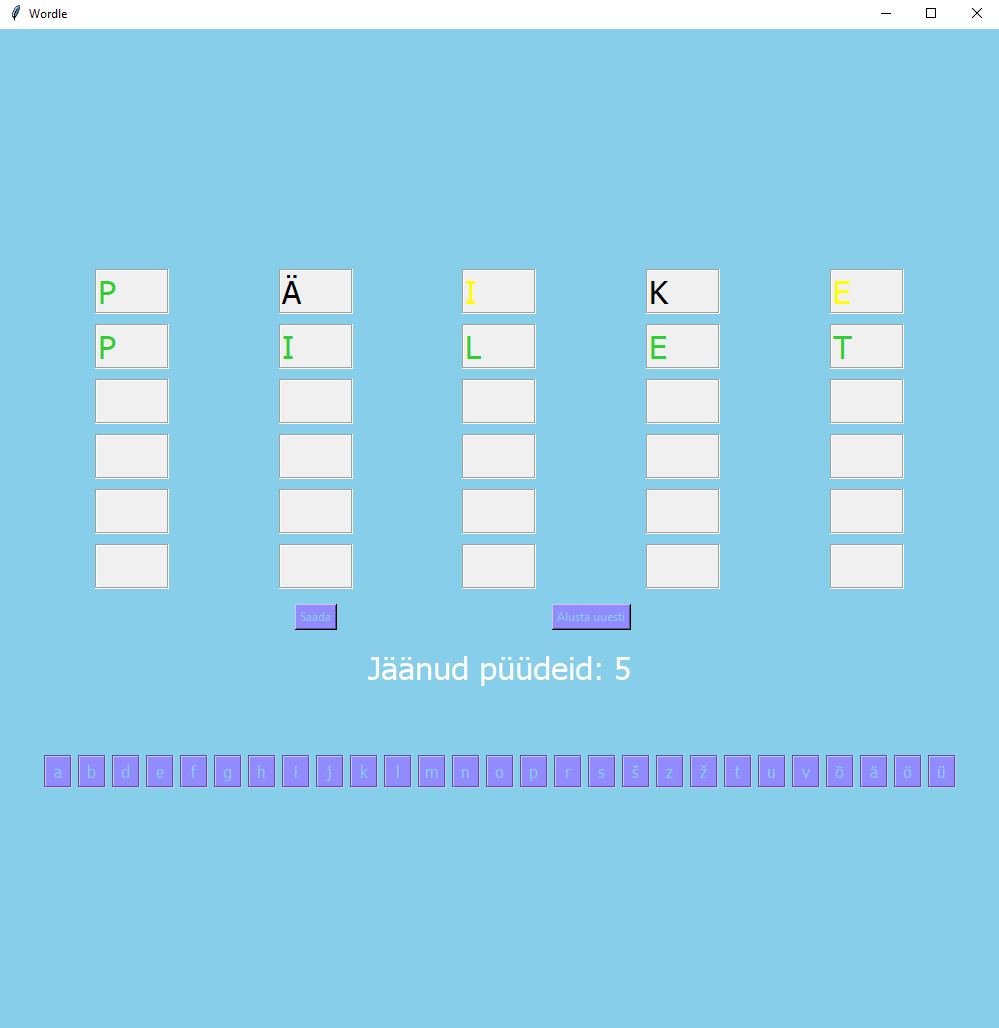
Esimene pilt näitab mängu väljanägemist.

Pilt 1 "Kujundis loomine"



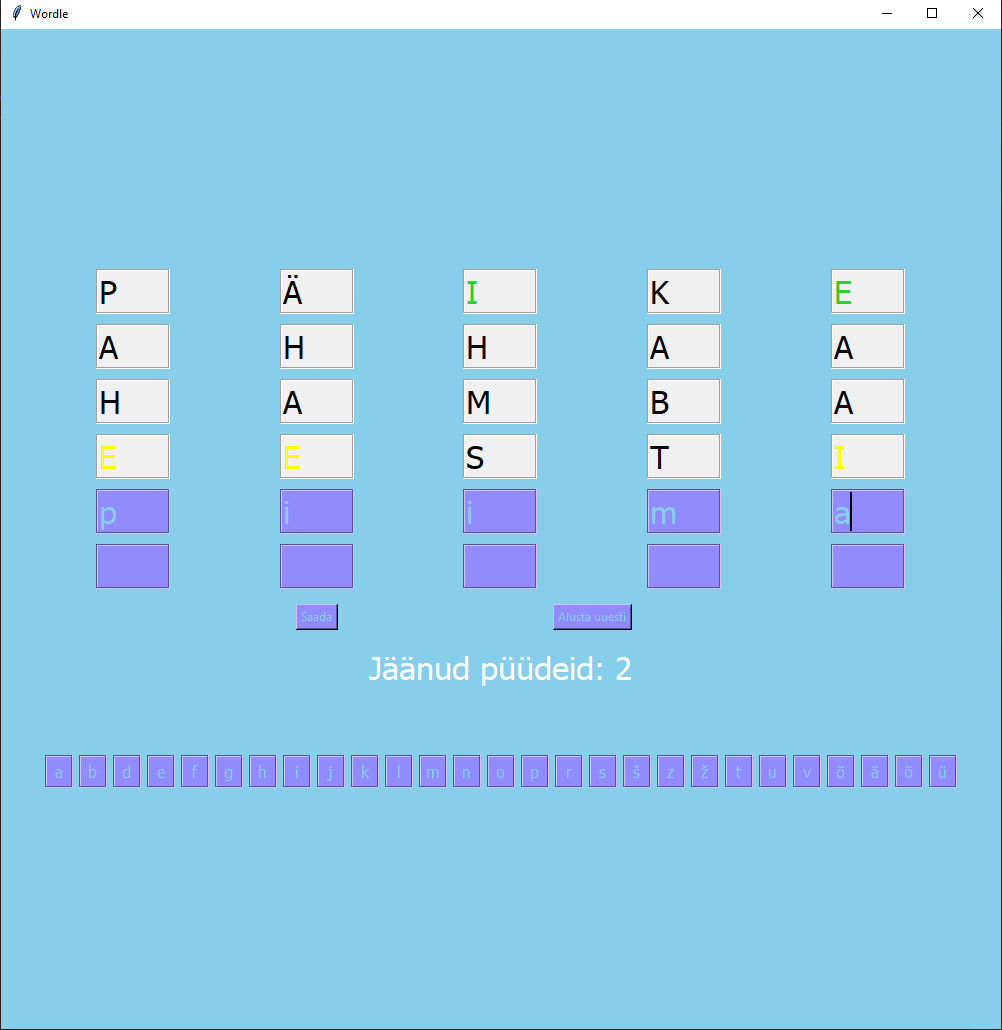
Teine pilt näitab, mis juhtub, kui sisestate sõna, roheline värv tõstab esile õige tähe ja õiges kohas, kollane tähendab, et täht on õige, kuid vales kohas, ja must tähendab, et täht ei ole õige. Allosas on kirjas, mitu katset on jäänud.

Pilt 2 "Kujundis loomine"



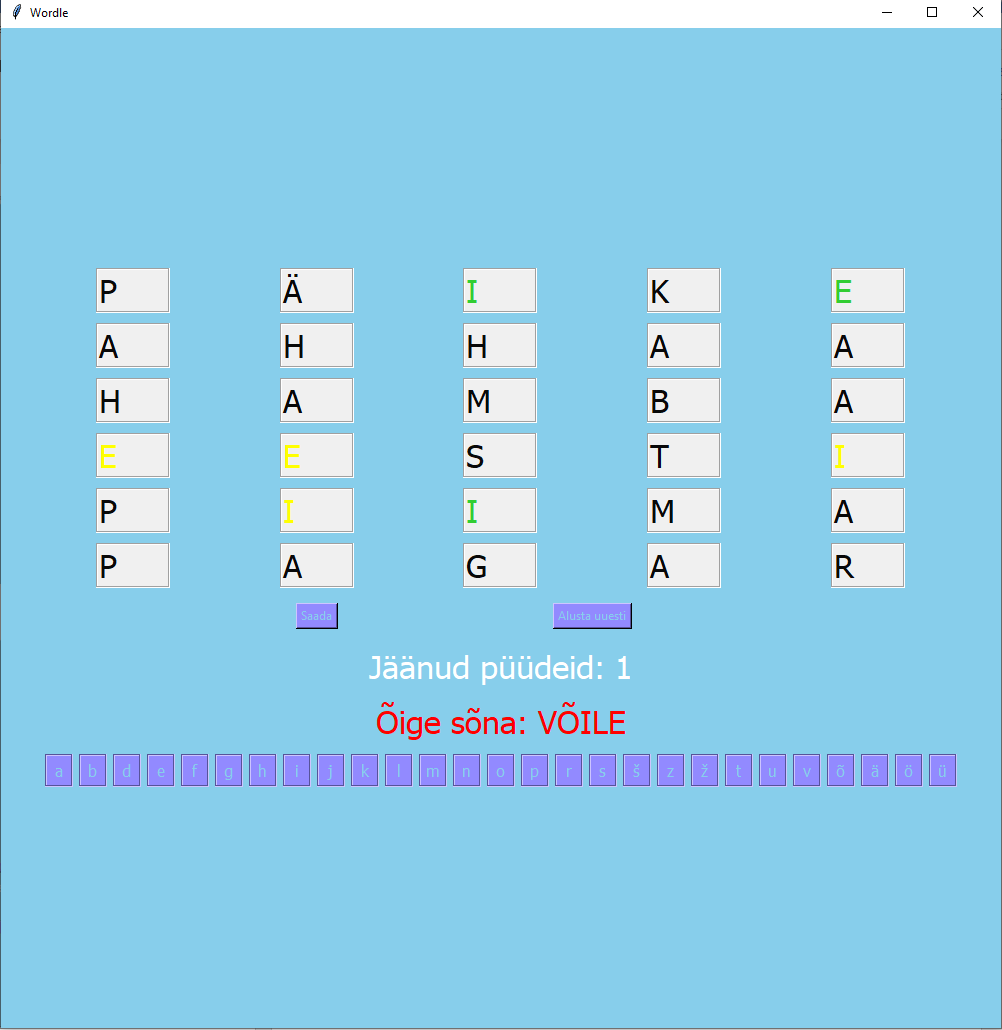
Pilt näitab, kas sõna on ära arvata.

Pilt 3 "Kujundis loomine"



Pildil on näha, kuidas näeb välja tähtede sisestamine ja mitu katset on jäänud.

Pilt 4 "Kujundis loomine"

****

Pilt 5 "Kujundis loomine"

Pildil on õige vastus, kui arvasite kuus korda valesti.

# Заключение

Игра работает хорошо, но в дальнейшем я бы хотела добавить функции, одна из которых будет изменять слова, другая будет удалять слова. Ещё хочу добавить функцию, которая будет передвигать ввод букв, то есть, когда я введу букву то ввод автоматически перейдёт в следующую ячейку. И хочу сделать чтобы букв которых нет на клавиатуре подсвечивались серым, если есть но не нету, то жёлтым, и если правильно стоит, то зелёным.

# Kokkuvõte

Mäng töötab hästi, kuid tulevikus tahaksin lisada funktsioone, millest üks muudab sõnu, teine kustutab sõnu. Samuti tahan lisada funktsiooni, mis liigutab tähtede sisestamist, st kui ma sisestan tähe, liigub sisestus automaatselt järgmisesse lahtrisse. Ja ma tahan teha, et tähed, mis ei ole klaviatuuril esile halliks, kui on, aga ei ole, siis kollaseks ja kui õige seisab, siis roheliseks.

# Allikad

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