Tallinna Tööstushariduskeskus **Logistika IT süsteemide spetsialist**

**Projekti „Oma Mäng“**

Praktiline töö

Juhendaja: Marina Oleinik  
Koostaja: Karina Linter  
Rühm: LOGITpv23

Tallinn 2024

# Содержание

[Содержание 2](#_Toc168393260)

[Картинки 3](#_Toc168393261)

[Введение 4](#_Toc168393262)

[Теория 5](#_Toc168393263)

[Основные модули, используемые в проекте: 5](#_Toc168393264)

[Основные конструкции, используемые в коде: 5](#_Toc168393265)

[Практическая часть 6](#_Toc168393266)

[Заключение 13](#_Toc168393267)

[Источники 14](#_Toc168393268)

[Библиография 14](#_Toc168393269)

# Картинки

[Pilt 1 Внешний вид 9](#_Toc168392683)

[Pilt 2 Изображение виселицы 9](#_Toc168392684)

[Pilt 3 Проигрыш 10](#_Toc168392685)

[Pilt 4 Победа 10](#_Toc168392686)

[Pilt 5 Диалоговое окно при проигрыше 11](#_Toc168392687)

[Pilt 6 Диалоговое окно при победе 11](#_Toc168392688)

# Введение

Цель этого проекта - создать свою игру.

Я создала игру, которая называется "виселица"

Виселица – игра в слова. Игроку загадывается слово, которое они должны угадать, используя буквы алфавита и возможность совершить ограниченное количество ошибок.

Слова загаданы на тему Python. Игрок должен отгадать название функции или цикла.

При запуске игры Вы видите перед собой английский алфавит. Вы должны нажать на выбранную Вами букву. Если Вы угадали, буква загорается зеленым цветом и ставится на свое место в нижнем поле. Если не угадали - красным и рисуется часть виселицы. Когда рисунок будет закончен - Вы проигрываете, но перед Вами появляется диалоговое окно, в котором можно выбрать - закончить игру или продолжить. При продолжении загадывается новое слово. При угадывании такой же выбор.

# Теория

Игра разработана на языке программирования Python. Игра на английском языке, которая проверяет Ваши знания и грамматику.

Задумка по реализации языка появилась в конце 1980-х годов, а разработка его реализации началась в 1989 году сотрудником голландского института CWI Гвидо ван Россумом. Для распределённой операционной системы Amoeba требовался расширяемый скриптовый язык, и Гвидо начал разрабатывать Python на досуге, позаимствовав некоторые наработки для языка ABC (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале 1991 года Гвидо опубликовал исходный текст в группе новостей alt.sources. С самого начала Python проектировался как объектно-ориентированный язык.

## Основные модули, используемые в проекте:

* **Pygame:** для создания игрового окна и обработки событий в игре.
* **random**: для выбора случайного слова из списка
* **sys:** позволяет работать с системными параметрами и функциями, такими как завершение программы с помощью sys.exit()

## Основные конструкции, используемые в коде:

* **Функции:** используются для организации кода и выполнения определенных задач, таких как загрузка слов, проверка введенного слова, обновление интерфейса и т. д.
* **Циклы:** используются для выполнения повторяющихся операций
* **Условные операторы:** используются для принятия решений на основе определенных условий, например, проверка правильности введенного слова.

# Практическая часть

import pygame

import random

import sys

#инициализация Pygame и его музыкального модуля

pygame.init()

pygame.mixer.init()

#размеры окна игры и его заголовка

WIDTH, HEIGHT = 800, 800

win = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("Игра Виселица")

#изображения виселицы

hangman\_images = [pygame.transform.scale(pygame.image.load(f"{i}.png"), (300, 300)) for i in range(1, 7)]

defeat\_image = pygame.transform.scale(pygame.image.load("youlose.jpg"), (300, 300))

win\_image = pygame.transform.scale(pygame.image.load("youwin.jpg"), (300, 300))

#звук

win\_sound = pygame.mixer.Sound("win\_sound.wav")

lose\_sound = pygame.mixer.Sound("lose\_sound.wav")

#слова и выбор

words = ["elif", "if", "print", "for", "whiletrue", "init", "int", "import", "else", "delete", "pop", "remove"]

word = random.choice(words).upper()

guessed = []

#шрифты

LETTER\_FONT = pygame.font.SysFont('comicsans', 40)

WORD\_FONT = pygame.font.SysFont('comicsans', 60)

DIALOG\_FONT = pygame.font.SysFont('comicsans', 30)

#функция для отрисовки игрового экрана

def draw(word, guessed):

#заполнение экрана белым цветом

win.fill((255, 255, 255))

#отрисовка букв алфавита и кружков вокруг них

for i in range(26):

x = 50 + (i % 13) \* 50

y = 300 + (i // 13) \* 50

ltr = chr(65 + i)

if ltr in guessed:

color = (0, 255, 0) if ltr in word else (255, 0, 0)

else:

color = (0, 0, 0)

text=LETTER\_FONT.render(ltr, 1, color)

pygame.draw.circle(win, color, (x, y), 20, 3)

win.blit(text, (x - text.get\_width() // 2, y - text.get\_height() // 2))

#подсчет количества неправильных попыток

wrong\_guesses = len([g for g in guessed if g not in word])

#отображение изображения виселицы в зависимости от количества ошибок

if wrong\_guesses < len(hangman\_images):

win.blit(hangman\_images[wrong\_guesses], (250, HEIGHT - 400))

else:

#если ошибок больше допустимого количества, отображение изображения поражения и проигрыша

win.blit(hangman\_images[-1], (250, HEIGHT - 400))

pygame.display.update()

pygame.time.delay(1000)

win.blit(defeat\_image, (250, HEIGHT - 400))

pygame.mixer.Sound.play(lose\_sound)

pygame.display.update()

pygame.time.delay(2000)

show\_dialog("You lose! Do you want to continue?", ["Continue", "Close"])

return

#проверка, угаданы ли все буквы в слове

if all(letter in guessed for letter in word):

win.blit(win\_image, (250, HEIGHT - 400))

pygame.mixer.Sound.play(win\_sound)

pygame.display.update()

pygame.time.delay(2000)

show\_dialog("Congratulations! You win!", ["Continue", "Close"])

return

#отображение загаданного слова с угаданными буквами и пропусками

display\_word = ""

for letter in word:

if letter in guessed:

display\_word += letter + " "

else:

display\_word += "\_ "

text = WORD\_FONT.render(display\_word, 1, (0, 0, 0))

win.blit(text, (WIDTH // 2 - text.get\_width() // 2, HEIGHT - 100))

pygame.display.update()

#функция для отображения диалогового окна после завершения игры

def show\_dialog(message, options):

run\_dialog = True

dialog\_font = pygame.font.SysFont('comicsans', 30)

while run\_dialog:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.MOUSEBUTTONDOWN:

x, y = pygame.mouse.get\_pos()

for i, option in enumerate(options):

option\_text = dialog\_font.render(option, 1, (0, 0, 0))

text\_width, text\_height = dialog\_font.size(option)

button\_rect = pygame.Rect(WIDTH // 2 - text\_width // 2, 450 + i \* 50, text\_width, text\_height)

if button\_rect.collidepoint(x, y):

if option == "Continue":

run\_dialog = False

reset\_game()

elif option == "Close":

pygame.quit()

sys.exit()

#диалоговое окно

win.fill((255, 255, 255))

pygame.draw.rect(win, (0, 0, 0), (150, 250, 500, 300), 3)

text = dialog\_font.render(message, 1, (0, 0, 0))

win.blit(text, (WIDTH // 2 - text.get\_width() // 2, 300))

for i, option in enumerate(options):

option\_text = dialog\_font.render(option, 1, (0, 0, 0))

win.blit(option\_text, (WIDTH // 2 - option\_text.get\_width() // 2, 450 + i \* 50))

pygame.display.update()

#следующий уровень

def reset\_game():

global word, guessed

word = random.choice(words).upper()

guessed = []

#главный цикл игры

run = True

while run:

draw(word, guessed)

for event in pygame.event.get():

if event.type == pygame.QUIT:

run = False

if event.type == pygame.MOUSEBUTTONDOWN:

x, y = pygame.mouse.get\_pos()

for i in range(26):

letter\_x = 50 + (i % 13) \* 50

letter\_y = 300 + (i // 13) \* 50

if (letter\_x - 20 <= x <= letter\_x + 20) and (letter\_y - 20 <= y <= letter\_y + 20):

letter = chr(65 + i)

if letter not in guessed:

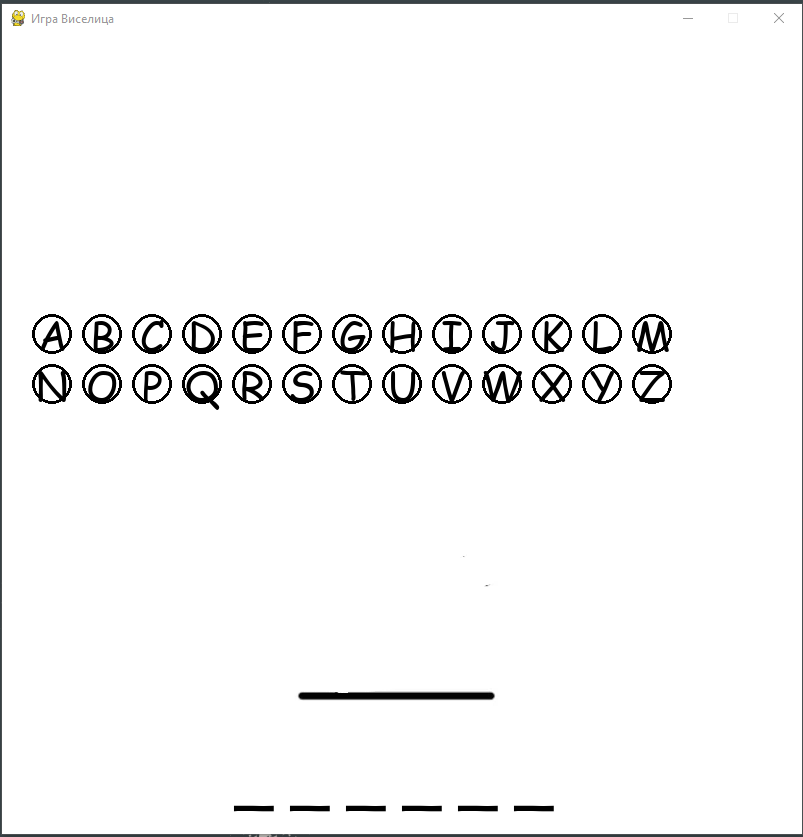
guessed.append(letter)

break

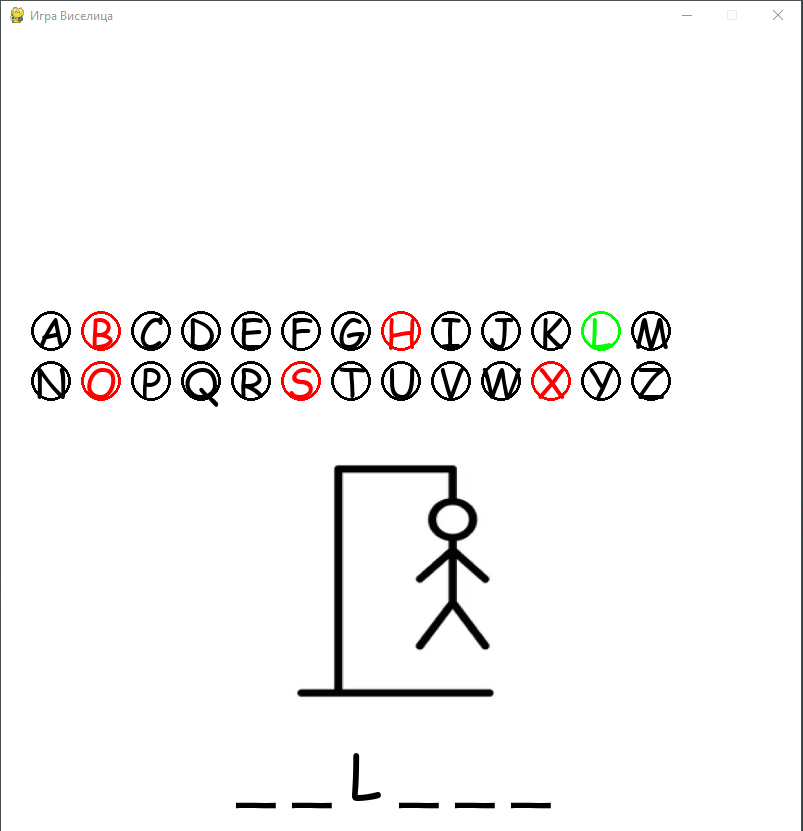
#завершение

pygame.quit()

sys.exit()



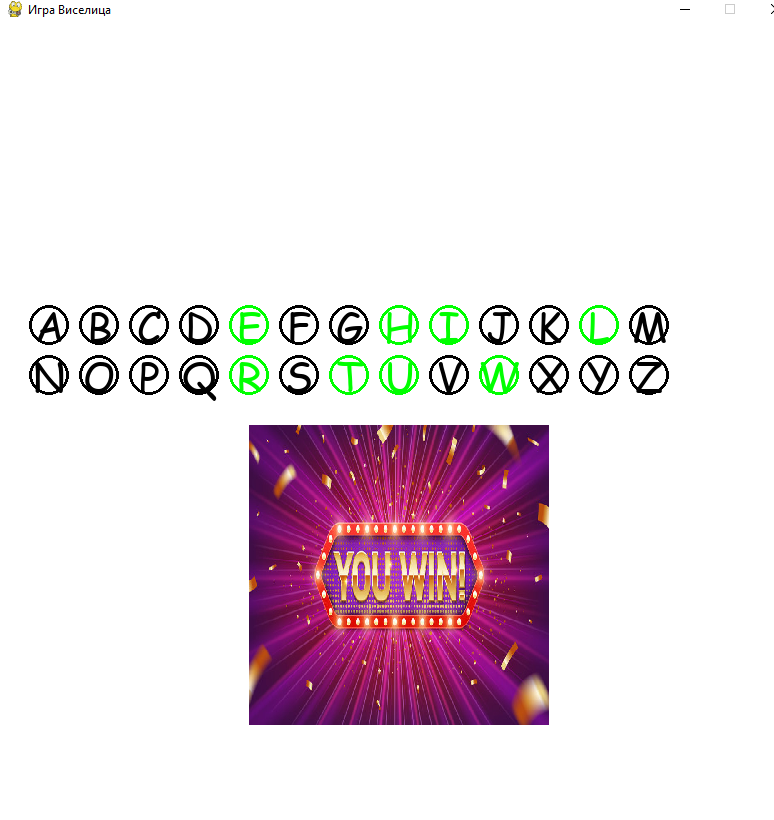
Pilt 1 Внешний вид



Pilt 2 Изображение виселицы



Pilt 3 Проигрыш



Pilt 4 Победа



Pilt 5 Диалоговое окно при проигрыше

****

Pilt 6 Диалоговое окно при победе

# Заключение

Я сделала игру, где нужно угадать название функции или цикла на английском языке. Я довольна своим результатом, игра получилась достаточно хорошей, но иногда не срабатывает звук при победе и проигрыше. Надеюсь, в будущем я это исправлю. Также в будущем я хочу усовершенствовать игру – добавить больше слов.

# Источники

Информация об истории создания языка Python

<https://ru.wikipedia.org/wiki/Python>

# Библиография

Rossum, G. v.