

Создание игры «Угадай слово» на Python

Акентьев Данила Денисович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается процесс разработки простой компьютерной игры "Угадай слово" с использованием языка программирования Python и графической библиотеки tkinter. Игра включает функциональность для угадывания как отдельных букв, так и целых слов, с предоставлением подсказок и ограниченным количеством попыток. Описываются ключевые аспекты создания графического интерфейса, а также интеграции логики игры с элементами пользовательского взаимодействия. Представлен процесс разработки игры, особенности реализации, а также примеры кода и объяснения для студентов и начинающих разработчиков.

Ключевые слова: Python, tkinter, графический интерфейс, разработка игры, "Угадай слово", компьютерные игры, программирование, интерактивность, разработка на Python.

Creating a "Guess the Word" game in Python

Akentev Danila Denisovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the process of developing a simple computer game "Guess the Word" using the Python programming language and the tkinter graphics library. The game includes functionality for guessing both individual letters and whole words, with hints provided and a limited number of attempts. It describes the key aspects of creating a graphical interface, as well as integrating game logic with user interaction elements. The game development process, implementation features, as well as code examples and explanations for students and novice developers are presented.

Keywords: Python, tkinter, graphical interface, game development, "Guess the word", computer games, programming, interactivity, Python development.

1 Введение

1.1 Актуальность

Разработка простых игр с графическим интерфейсом является важной областью в обучении программированию, так как позволяет начинающим разработчикам освоить базовые концепции работы с графическими интерфейсами и алгоритмами. Изучение таких проектов способствует

лучшему пониманию принципов программирования и применению теоретических знаний на практике. Особенно актуально использование Python и tkinter, как одного из наиболее доступных инструментов для создания визуальных приложений.

1.2 Обзор исследований

В своей работе Е.О. Новиков подробно объясняет алгоритм работы игры и проводит глубокий анализ возможностей Pygame для создания более сложных игр. Особое внимание уделено методам, связанным с графикой и взаимодействием с пользователем, что дает возможность разрабатывать динамичные игры с графическим интерфейсом [1]. Другим важным исследованием является работа Д.Ф. Ахметгалиевой и Г.Р. Галиаскаровой, в которой рассматривается создание игры "Змейка" на Python с использованием библиотеки tkinter. В статье акцентируется внимание на необычном подходе к разработке игры, в котором традиционную змейку заменяет персонаж Гомер Симпсон, поедающий пончики. Игра включает элементы геймплея, где с каждым съеденным пончиком персонаж удлиняется, а скорость игры увеличивается. Такое решение подчеркивает возможности tkinter для создания простых, но увлекательных игр с графическим интерфейсом [2]. Ранее вопросы разработки приложений с графическим пользовательским интерфейсом (GUI) с использованием Python и Tkinter привлекали внимание множества исследователей. Например, в своей статье А. С. Матвеева рассматривает применение паттерна «Команда» при создании модульных и структурированных приложений с использованием Tkinter и демонстрирует, как этот паттерн помогает создавать гибкие и расширяемые приложения, где каждая кнопка выполняет отдельную команду. Это позволяет значительно упростить процесс разработки и последующей модификации программ, повысив их надежность и удобство использования [3]. Другим значимым исследованием является работа А. М. Вохидова, Д. А. Вохидова, Р. Ф. Фармоновой и Д. Ш. Хафизовой. В статье рассматривается использование Tkinter и Python в медицине, где описаны ключевые аспекты создания медицинских приложений с графическим интерфейсом. Исследователи подчеркивают удобство использования Tkinter для разработки интуитивно понятных интерфейсов для медицинских работников и пациентов. В работе приводятся примеры практических приложений, использующих эту технологию для создания доступных и удобных инструментов [4]. В своей статье Д. А. Алигулович, З. И. Махмудова и Р. Ф. Сайфуллаев анализируют преимущества и недостатки Python и Tkinter для разработки приложений с GUI. Они подробно исследуют области применения Tkinter в педагогической практике, акцентируя внимание на возможностях создания кроссплатформенных решений, что делает их особенно подходящими для образовательных целей. В работе подчеркивается важность использования современных педагогических технологий с применением этого инструмента [5].

1.3 Цель исследования

Целью данной работы является разработка игры "Угадай слово" с использованием языка Python и библиотеки tkinter, а также анализ основных принципов создания интерактивных приложений для начинающих программистов. В рамках исследования рассматриваются особенности взаимодействия пользователя с приложением, а также решения, связанные с графическим отображением игрового процесса.

2 Материалы и методы

Для разработки игры использовался язык программирования Python и библиотека tkinter, предназначенная для создания графических интерфейсов. Методология работы включает изучение принципов объектно-ориентированного программирования, создание простых пользовательских интерфейсов и обработку событий с использованием встроенных возможностей Python. В статье также рассматривается процесс отладки и тестирования игры.

3 Результаты и обсуждения

Подключаем нужные библиотеки для работы с интерфейсом и логикой игры (рис.1). Будем использовать tkinter для создания графических элементов, messagebox для вывода уведомлений и random для случайного выбора слова.

```
import tkinter as tk
from tkinter import messagebox
import random
```

Рисунок 1. Импорт библиотек

- tkinter — библиотека для создания интерфейса.
- messagebox — для вывода сообщений о победах или ошибках.
- random — для случайного выбора слова.

Переходим к созданию функции, которая будет отвечать за выбор случайного слова и подсказки (рис.2).

```
def get_word_and_hint(): 2 usages
    """Возвращает случайное слово для игры и подсказку."""
    words = [
        ('python', 'Язык программирования.'),
        ('компьютер', 'Электронное устройство для обработки данных.'),
        ('книга', 'Источник знаний или развлечений.'),
        ('кот', 'Популярное домашнее животное.'),
        ('машина', 'Транспортное средство для передвижения по дороге.')
    ]
    return random.choice(words)
```

Рисунок 2. Функция возвращения случайного слова и подсказки

- `random.choice(words)`: выбирает случайную пару из списка слов и подсказок.

Пишем функцию, которая будет отображать текущее состояние слова в игре, скрывая не угаданные буквы (рис.3).

```
def display_word(word, guessed_letters): 3 usages
    """Отображает текущее состояние угадываемого слова, скрывая неугаданные буквы."""
    return " ".join([letter if letter in guessed_letters else "_" for letter in word])
```

Рисунок 3. Функция отображения состояния

- `" ".join(...)`: соединяет символы в строку с разделением пробелами.

Функция проверяет, была ли буква угадана, и если нет, заменяет её на подчеркивание.

Приступаем к основной логике игры, где будем обрабатывать ввод пользователя (буква или слово) (рис.4).

```
def on_guess(): 1 usage
    """Обрабатывает ввод буквы или слово и обновляет состояние игры."""
    guess = entry.get().lower()

    # Если введено одно слово для угадания
    if len(guess) > 1:
        if guess == word:
            messagebox.showinfo(title="Поздравление", message=f"Поздравляю! Ты угадал слово '{word}'!")
            reset_game()
        else:
            messagebox.showinfo(title="Проигрыш", message=f"Увы, ты не угадал слово. Загаданное слово было '{word}'.")
            reset_game()
    # Если введена одна буква
    elif len(guess) == 1:
        if not guess.isalpha():
            messagebox.showwarning(title="Некорректный ввод", message="Пожалуйста, введите одну букву.")
            return

        if guess in guessed_letters:
            messagebox.showwarning(title="Уже угадано", message="Эту букву вы уже угадали.")
            return

        guessed_letters.append(guess)

        if guess in word:
            result_label.config(text=f"Буква '{guess}' есть в слове!")
        else:
            result_label.config(text=f"Буквы '{guess}' нет в слове.")
            global attempts
            attempts -= 1

    # Обновляем отображение слова и количество попыток
    word_display.config(text=display_word(word, guessed_letters))
    attempts_label.config(text=f"Осталось попыток: {attempts}")

    # Проверка на победу или проигрыш
    if all(letter in guessed_letters for letter in word):
        messagebox.showinfo(title="Поздравление", message=f"Поздравляю! Ты угадал слово '{word}'!")
        reset_game()
    elif attempts == 0:
        messagebox.showinfo(title="Проигрыш", message=f"Увы, ты проиграл. Загаданное слово было '{word}'.")
        reset_game()

    entry.delete(first=0, tk.END) # Очищаем поле ввода после каждого ввода
```

Рисунок 4. Функция основной логики игры

- `entry.get().lower()`: получаем введенную строку и преобразуем её в нижний регистр для упрощения обработки.

После того как игрок вводит букву или слово, обновляется состояние игры, и проверяется, угадал ли он все буквы или проиграл.

Создаем функцию, которая будет сбрасывать игру, начиная новый раунд (рис.5).

```
def reset_game(): 4 usages
    """Сбрасывает состояние игры."""
    global word, guessed_letters, attempts
    word, hint = get_word_and_hint()
    guessed_letters = []
    attempts = 6
    word_display.config(text=display_word(word, guessed_letters))
    attempts_label.config(text=f"Осталось попыток: {attempts}")
    hint_label.config(text=f"Подсказка: {hint}")
    result_label.config(text="")
    entry.delete(first=0, tk.END)
```

Рисунок 5. Функция рестарта игры

- `get_word_and_hint()` выбирает новое слово и подсказку.

Обновляются все компоненты интерфейса, чтобы подготовить игру к следующему раунду.

Далее занимаемся созданием самого графического интерфейса игры, который будет отображать все элементы управления и состояния игры (рис.6).

```

# Настройки игры
word, hint = get_word_and_hint()
guessed_letters = []
attempts = 6

# Создание главного окна
root = tk.Tk()
root.title("Игра 'Угадай слово'")

# Отображение подсказки
hint_label = tk.Label(root, text=f"Подсказка: {hint}", font=("Arial", 12))
hint_label.pack(pady=10)

# Отображение текущего состояния слова
word_display = tk.Label(root, text=display_word(word, guessed_letters), font=("Arial", 18))
word_display.pack(pady=10)

# Ввод буквы или слова
entry = tk.Entry(root, font=("Arial", 14))
entry.pack(pady=10)

# Кнопка для отправки буквы
guess_button = tk.Button(root, text="Угадать букву или слово", font=("Arial", 14), command=on_guess)
guess_button.pack(pady=10)

# Сообщение о текущем состоянии
result_label = tk.Label(root, text="", font=("Arial", 12))
result_label.pack(pady=10)

# Отображение количества оставшихся попыток
attempts_label = tk.Label(root, text=f"Осталось попыток: {attempts}", font=("Arial", 12))
attempts_label.pack(pady=10)

# Запуск игры
root.mainloop()

```

Рисунок 6. Код графического интерфейса игры

- `root = tk.Tk()`: создаем основное окно для игры.

Все элементы интерфейса: `hint_label`, `word_display`, `entry`, `guess_button`, `result_label`, `attempts_label` — настраиваются и размещаются на экране.

Внешний вид игры выглядит следующим образом (рис.7).

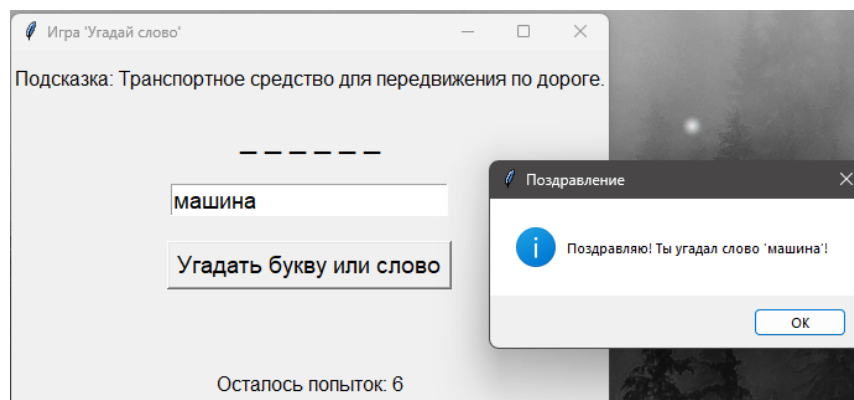


Рисунок 7. Внешний вид игры

4 Выводы

В рамках данной работы была разработана игра "Угадай число" с использованием языка программирования Python и библиотеки tkinter. Реализация игры продемонстрировала основные принципы создания графических интерфейсов, обработку пользовательского ввода и взаимодействие с пользователем через виджеты. Программа позволяет пользователю угадывать случайно сгенерированное число, при этом реализована система подсказок и обработка ошибок ввода, что обеспечивает удобство использования.

Процесс разработки показал, что библиотека tkinter является удобным инструментом для создания простых графических приложений, что делает её хорошим выбором для начинающих программистов. В процессе работы были рассмотрены основные принципы работы с графическими интерфейсами, включая создание и размещение элементов управления, обработку событий, а также взаимодействие с пользователем через диалоговые окна.

Разработка игры показала, что простые приложения, такие как "Угадай число", могут служить эффективным средством для обучения основам программирования, а также для практического применения теоретических знаний. В будущем, данную игру можно расширить, добавив новые функциональные возможности, такие как уровни сложности, ограничение по времени или статистику игр, что сделает её более интересной и полезной для пользователей.

Таким образом, разработка игры "Угадай число" на языке Python с использованием библиотеки tkinter является хорошей основой для изучения принципов создания графических приложений и взаимодействия с пользователем, а также служит примером эффективного применения Python в области разработки интерактивных программ.

Библиографический список

1. Новиков Е.О. Разработка программы «Игра морской бой» на языке Python // Прогрессивные технологии и экономика в машиностроении: сборник трудов XIV Всероссийской научно-практической конференции для студентов и учащейся молодежи. 2023. С. 207-210
2. Ахметгалеева Д.Ф., Галиаскарова Г.Р. Создание игры-змейка на языке программирования Python «Гомер ест пончики» // Colloquium-journal. 2019. № 17-2(41). С. 24-27
3. Матвеева А. С. Реализация паттерна поведения «Команда» в графическом интерфейсе пользователя на Python // Постулат. 2025. №1.
4. Вохидов А.М., Вохидов Д.А., Фармонова Р.Ф., Хафизова Д.Ш. Разработка Графическим Пользовательским Интерфейсом -Программ В Пакете Tkinter С Использованием Современных Педагогических Технологий В Области Медицины // Роль Информационных Технологий В Управлении Ресурсами Персонала Здравоохранения 2022. Т. 34.
5. Aligulovich D.A., Mahmudova Z.I., Sayfullayev R.F. Tibbiyot yo'nalishida

zamonaviy pedagogik texnologiyalarini qo'llab tkinter paketida gui dasturlarini tuzish // Евразийский журнал математической теории и компьютерных наук. 2022. №2(12). С. 31-35.