– запуск инновационных проектов, направленных на создание устойчивых продуктов и услуг (разработка умных систем управления ресурсами).

Список использованных источников

1 ATLANT [Электронный ресурс]. – Электронные данные. – Режим доступа: https://atlant.by/?ysclid=m26rviw4km648662974 (дата обращения: 18.10.2024).

2 UN.org [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.un.org/sustainabledevelopment/ru/ (дата обращения: 18.10.2024).

3 SDGS.by [Электронный ресурс]. – Электронные данные. – Режим доступа: https://sdgs.by/?ysclid=m2glhnttvb349134744 (дата обращения: 18.10.2024).

УДК 004.89

A.V. Sidorenko, I.A. Samoilova, Ye.A. Spirina,
Buketov Karaganda University
Karaganda, Kazakhstan

CHATBOT DEVELOPMENT TECHNOLOGY USING PYTHON

Abstract. *The Discord chatbot developed in Python is an innovative and multifunctional tool that can significantly improve server management and make the interaction of participants more interesting and exciting.*

А.В. Сидоренко, И.А. Самойлова, Е.А. Спирина
Карагандинский университет им. Е.А.Букетова,
Караганда, Казахстан

ТЕХНОЛОГИЯ РАЗРАБОТКИ ЧАТ-БОТА С ИСПОЛЬЗОВАНИЕМ PYTHON

Аннотация. *Разработанный на Python чат-бот для Discord является инновационным и многофункциональным инструментом, который способен значительно улучшить управление сервером и сделать взаимодействие участников более интересным и увлекательным.*

With the development of digital technologies and social networks, there is an increasing need for innovative solutions for communication and interaction in the online environment. One of the most promising areas in

this area is the development of chatbots (virtual interlocutors) that can automatically process user requests and provide information or perform certain actions. One of the popular platforms for creating and using such interlocutors, chatbots, is Discord - a multi-user online platform for communication, community organization and voice communication (Figure 1).

Fig. 1. Discord platform logo

The technology of developing a chatbot for Discord is a complex and multifaceted task that requires the integration of various technologies and algorithms [1]. The choice of development tools fell on Python, as it is a powerful and versatile programming language that is widely used in various areas of software development, including the development of bots for online platforms. This programming language has become one of the most popular and in-demand programming languages due to its simplicity, flexibility and rich ecosystem of libraries and frameworks. Using Python to develop a virtual interlocutor allows you to reduce the time and cost of development, as well as ensure high performance and reliability of the created solution [2].

The technology of creating a chatbot for Discord includes several key stages, each of which is important for creating a fully functional and reliable bot. This thesis covers the entire development process, from conceptualization to testing and implementation. Each of the development stages is presented in detail below.

The first step is to set the task and define the requirements for the bot. At this stage, it is important to determine what functions the bot should perform, as well as what problems it should solve. This step includes analyzing user needs and researching existing solutions to identify gaps and determine the unique features of the virtual interlocutor being developed.

The second stage involves selecting the appropriate technologies and tools to implement the tasks. The Python programming language and the discord.py library were chosen to develop the bot, as they provide convenient and powerful tools for interacting with the Discord API. Additionally, the following libraries and tools were selected: json: for working with data in JSON format; NLTK or a similar library: for natural language processing; asyncio: for working with asynchronous code.

Designing the bot architecture. At this stage, the bot architecture is developed, which includes a modular structure that allows you to easily add and change functionality. The architecture includes the following components: main module: responsible for connecting to Discord and processing events; greeting module: responsible for greeting new users; moderation module includes commands for moderators; anti-crash module: monitors the stability of the bot and prevents its crashes; entertainment module: includes mini-games and quizzes; RPG module: implements the RPG system.

After the architecture is designed, the implementation of the bot's main functions begins. This stage includes writing code for each module. An example of the implementation of greeting new users is shown in Figure 2.

```
@bot.event
async def on_member_join(member):
    channel = discord.utils.get(member.guild.text_channels, name='general')
    if channel:
        await channel.send(f'Добро пожаловать на сервер, {member.mention}!')
```

Fig. 2. Greeting new users

Figure 3 shows a fragment implementing moderation.

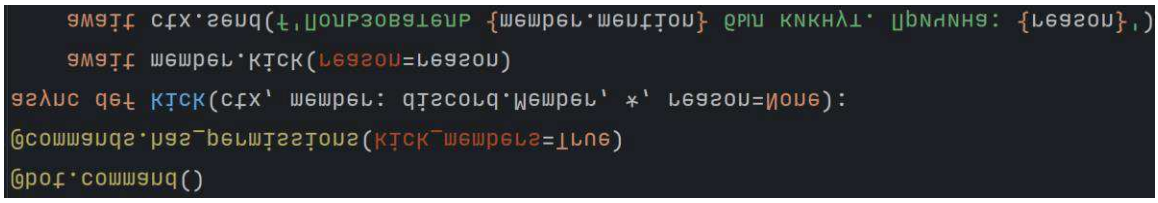

Fig. 3. Moderation

The anti-crash system is necessary so that the built-in mechanisms for protection against attacks and failures ensure stable operation of the server and prevent its shutdown due to possible overloads or malicious actions (Fig.4).

```python
import signal

def handle_crash(signum, frame):
    print("Бот поймал сигнал:", signum)
    # Реализуйте логику восстановления бота

signal.signal(signal.SIGTERM, handle_crash)
signal.signal(signal.SIGINT, handle_crash)
```

Fig. 4. Implementation of the anti-crash system

After all the functions are implemented, the bot is tested and debugged. At this stage, it is important to check that all modules work correctly and interact with each other without conflicts. Testing is carried out both on a local server and on a real Discord server to ensure that all functions work correctly.

Discord server. This stage includes setting up access rights, inviting the bot to the server, and providing the necessary access to channels and roles. Server administrators and moderators are also trained to use bot commands.

Support and update. An important step in the process of developing a virtual chatbot. Once the bot is implemented, it is important to ensure its support and regular updates. This includes monitoring the bot's operation, fixing errors, adding new features, and adapting to changes in the Discord API. Regular updates help keep the bot relevant and stable.

Key results and conclusions obtained during the writing of the work:

1. Developing a chatbot for Discord using Python requires a comprehensive approach and consideration of various aspects, from the choice of technologies and tools to evaluating the effectiveness and usability of the bot.

2. Discord is a powerful platform for communication and interaction, providing users with ample opportunities for creating and managing communities.

3. Using the Python programming language and the Discord.py library allows you to create powerful and effective virtual interlocutors for Discord with minimal investment of time and resources.

4. Developing natural language processing algorithms is a key aspect of developing virtual interlocutors, allowing bots to understand and analyze

user text messages.

5. Evaluating the effectiveness and usability of the bot is an important stage of development and allows you to create a product that meets user needs and achieves its goals.

Thus, the developed chatbot for Discord, created using Python, is an innovative and multifunctional tool that can significantly improve server management and make the interaction of participants more interesting and exciting.

References

1. Discord API Documentation. URL: https://discord.com/developers/docs/intro.

2. Python Documentation. URL: https://docs.python.org/3/.

УДК 629.056.8

Ч. Сейитнепесов, Г. Мелебаева, Г. Атаева, Г. Реджепова
Институт Телекоммуникаций и
Информатики Туркменистана

ТЕХНОЛОГИИ СПУТНИКОВОЙ СВЯЗИ ДЛЯ ГЛОБАЛЬНОГО ПОКРЫТИЯ ИНТЕРНЕТА

Аннотация. *С развитием технологий и увеличением потребности в высокоскоростном доступе в интернет, спутниковая связь становится все более актуальной. Данная статья рассматривает ключевые технологии спутниковой связи, направленные на обеспечение глобального покрытия интернета, а также их преимущества и недостатки. Анализируются существующие проекты и системы, такие как Starlink, OneWeb и Project Kuiper, а также будущие перспективы и вызовы, стоящие перед отраслью.*

Ключевые слова: *Телекоммуникационные системы, широкополосные сети, беспроводная связь, инфраструктура связи, технологии передачи данных, интеграция сетей IoT (Интернет вещей), 5G технологии будущее телекоммуникаций, конвергенция сетей.*

Ch. Seyitnepesov, G. Melebayeva, G. Ataeva, G. Redjepova
Institute of Telecommunications and
Informatics of Turkmenistan

SATELLITE COMMUNICATION TECHNOLOGIES FOR GLOBAL INTERNET COVERAGE