

УДК 004.7

*Минець О.Р., здобувач,  
Калита Н.І., к.т.н., професор  
Харківський національний університет радіоелектроніки*

## **ІГРОВИЙ DISCORD-БОТ З АВТОМАТИЗОВАНОЮ СИСТЕМОЮ МОНЕТИЗАЦІЇ ТА УПРАВЛІННЯМ ІЄРАРХІЄЮ КОРИСТУВАЧІВ**

Платформа Discord [1] є провідним середовищем для формування закритих та відкритих ком'юніті, де ключовим фактором життєздатності групи є ефективна модерация та внутрішня гейміфікація.

Соціальна взаємодія та економічне регулювання всередині цифрової платформи у Discord ґрунтується на принципах:

- а) соціальна ієрархія та рольова модель доступу;
- б) цифрова мікроекономіка спільнот;
- в) монетизація цифрових послуг.

Управління великою кількістю користувачів базується на розподілі ролей, які визначають права доступу до текстових та голосових каналів. Проблема стандартних засобів платформи полягає у відсутності інструментів для динамічного та автоматизованого створення таких ролей самими користувачами.

Цифрова мікроекономіка включає механізми конвертації часу, проведеного у голосових каналах, у віртуальну валюту або статус. Це вимагає точного обліку активності та запобігання зловживанням.

Монетизація цифрових послуг здійснюється через оренду статусів, інвестиції у створення власних "приватних зон" (категорій) та торгівлю між учасниками, що формує стійку внутрішню економіку сервера.

Розроблений ігровий бот базується на мові програмування Python [2] та бібліотеці discord.py, що дозволяє обробляти події платформи в асинхронному режимі, запобігаючи блокуванню основного потоку виконання. Архітектура боту побудована на базі SQLite з використанням бібліотеки aioredis, що забезпечує неблокуючу взаємодію з БД для кожного окремого сервера (guild-based database isolation). Структура бази даних містить таблиці для обліку балансу користувачів, рівнів, досвіду, ієрархії ролей та детальної історії транзакцій.

Особливістю розробки є надання користувачам можливості створювати власні ролі, інвестуючи ігрову валюту. Система автоматично створює відповідні категорії та канали з налаштованими правами доступу, що реалізує динамічну модель RBAC (Role-Based Access Control). Приклад асинхронної реалізації створення ієрархічної

структури в кодї бота: `category = await guild.create_category(name=f'{role.name}', overwrites=overwrites)`. Бот підтримує не лише створення, а й перепродаж ролей у внутрішньому магазині, де творець ролі може отримувати відсоток від кожної покупки, реалізуючи механізм пасивного доходу.

Гейміфікація реалізована через систему активності у голосових каналах. Бот використовує фонові задачі (`tasks.loop`) для циклічного нарахування досвіду та монет користувачам, враховуючи стан мікрофона та навушників (`self_deaf`). Для оптимізації навантаження на БД застосовано механізм RAM-сесій (`voice_sessions`), де накопичується час перебування до моменту синхронізації з постійним сховищем.

Кожна зміна балансу супроводжується записом у таблицю транзакцій, що дозволяє проводити аудит дій через адміністративну панель. Система також підтримує автоматичне відкликання прав доступу після завершення терміну оренди ролі. Це реалізовано через функцію `check_role_expiration`, яка щохвилини перевіряє `expiration_date` у БД. У разі завершення терміну, бот видаляє роль. Для запобігання втрати даних при раптовій зупинці сервера реалізовано обробник сигналів, який ініціює коректне завершення процесів та збереження активних сесій.

Використання компонентів `discord.ui` (`Modal`, `Select`, `Button`) дозволяє інтегрувати бота в сучасний інтерфейс Discord. Це значно покращує користувацький досвід, оскільки складні дії (переказ монет, вибір кольору ролі, налаштування цін) виконуються через графічні елементи, а не через запам'ятовування складних текстових синтаксисів. Модальні вікна дозволяють збирати структуровані дані від користувача, мінімізуючи помилки вводу.

Розроблений програмний продукт автоматизує складні процеси управління ігровою спільнотою, поєднуючи соціальні та економічні механіки. Використання асинхронного підходу забезпечує стабільну роботу системи на серверах з великою кількістю активних учасників. Перспективою подальшого розвитку є впровадження зовнішніх платіжних систем та розширення ігрових механік (квестів, аукціонів).

### **Список використаних джерел:**

1. Офіційна документація Discord Developer Portal : вебсайт. URL: <https://discord.com/developers/docs> (дата звернення 20.03.2026).
2. Офіційний посібник Python з асинхронного програмування : вебсайт. URL: <https://docs.python.org/3/library/asyncio.html> (дата звернення 20.03.2026).