|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |
| --- |
| **ОТЧЕТ ПО ВТОРОЙ ЧАСТИ ПРОЕКТА** |
| **по дисциплине** |
| **«Технологии разработки программных приложений»**  **Тема: «Разработка discord бота»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-08-21 | Юнаков И.П. |
| Принял | Туманов М.Б. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

Содержание

[1. Тема проекта 3](#_Toc130759558)

[2. Состав команды 3](#_Toc130759559)

[3. Функциональные требования к проекту 3](#_Toc130759560)

[4. Интерфейс разрабатываемого бота 5](#_Toc130759561)

[5. Репозиторий проекта на GitHub 5](#_Toc130759562)

[6. Вывод 6](#_Toc130759563)

1. Напишите файл README.md с общим описанием проекта, опишите зависимости проекта и команду для его запуска.

Gaembot — это пример, который был создан для изучения командной работы и разработки приложений. Этот проект представляет собой бота для Discord, который позволяет играть в настольные игры, такие как 2048, крестики-нолики и шашки. Разработка для Discord является полезным упражнением для практики программирования и создания приложений. Боты для этой платформы могут значительно упростить процесс общения и взаимодействия между участниками, что делает их популярными в командной среде. Gaembot, предназначен для игры в настольные игры, демонстрирует, как применять программные навыки для создания интерактивных игровых сред.

Для установки бота необходимо выполнить несколько команд, которые описаны в файле [README.md](https://github.com/yunikeil/Gaembot/blob/master/readme.md). Бот использует следующие зависимости: nextcord, aeval, pillow, sphinx и requests. Для разработки бота использовались ресурсы, такие как документация и API от Discord, а также библиотеки и код из открытых исходных кодов на GitHub.

1. Настройте систему сборки вашего проекта.

В проекте используется язык python. Setuptools — это инструмент для упрощения установки, сборки и распространения пакетов Python. Он обычно используется для создания и установки библиотек и фреймворков, но может быть полезен и для проектов, не подразумевающих использование себя в качестве библиотеки или фреймворка. Однако в качестве системы сборки в нашем проекте будет использоваться bash скрипт, который удалит все мусорные файлы кэша, очистит файлы конфигурации, обновит зависимости проекта, подготовит файл-установщик.

1. Продолжайте разработку проекта с использованием системы контроля версий. Снабжайте программный код разработанных классов, методов, функций и т.д. комментариями в формате Javadoc на Java или Docstring на Python.

Docstring — это строка документации, которая расположена сразу после заголовка функции, метода или класса в коде на языке Python. Docstring предназначен для объяснения назначения функции или метода, описания его параметров и возвращаемого значения. Кроме того, в Docstring можно включать примеры использования, описание исключений, которые могут быть возбуждены, и другую полезную информацию.

1. Продолжайте разработку проекта с использованием системы контроля версий.

Sphinx — это инструмент для генерации документации на языке Python, который позволяет создавать красивую и информативную документацию из Docstring-ов в коде. Sphinx поддерживает форматирование документации в разных форматах, включая HTML, PDF, EPUB, и другие. Sphinx автоматически создает генерируемую документацию из исходного кода, а также позволяет настраивать ее содержание и оформление.

Чтобы использовать Sphinx, необходимо настроить проект, добавив специальные файлы конфигурации и инструкции для генерации документации. С помощью инструкций Sphinx можно задать, какие модули и функции следует документировать, а также какие элементы документации нужно включать. С помощью Sphinx можно также добавлять изображения, ссылки и другие элементы в документацию.

1. Вывод.

Использование Docstring-ов и Sphinx позволяет значительно упростить процесс создания документации и сделать ее более информативной и понятной для других разработчиков. Кроме того, использование Docstring-ов и Sphinx улучшает читаемость кода и делает его более понятным и доступным для других разработчиков.

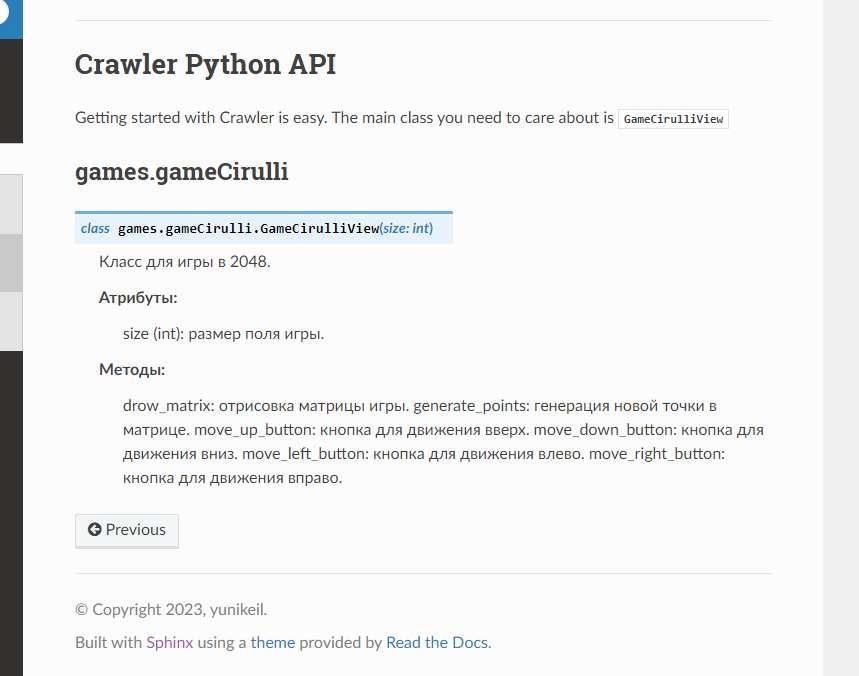


Рисунок 1 – Пример документации Sphinx с изменённой темой

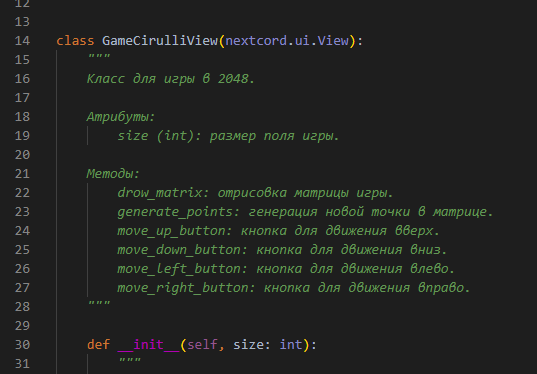


Рисунок 2 – Пример docstring в python файле

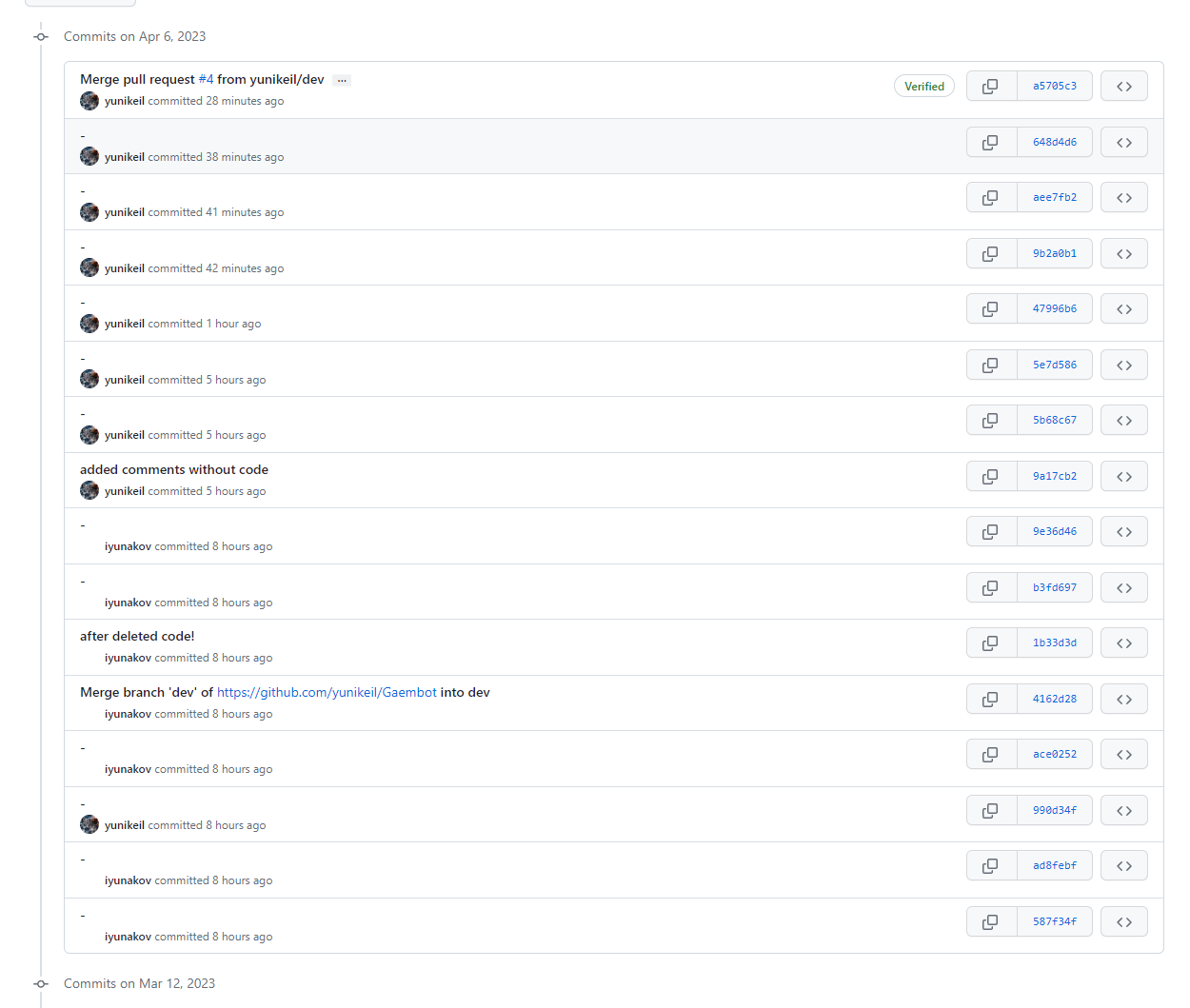


Рисунок 4 – Скриншот из истории коммитов проекта на GitHub

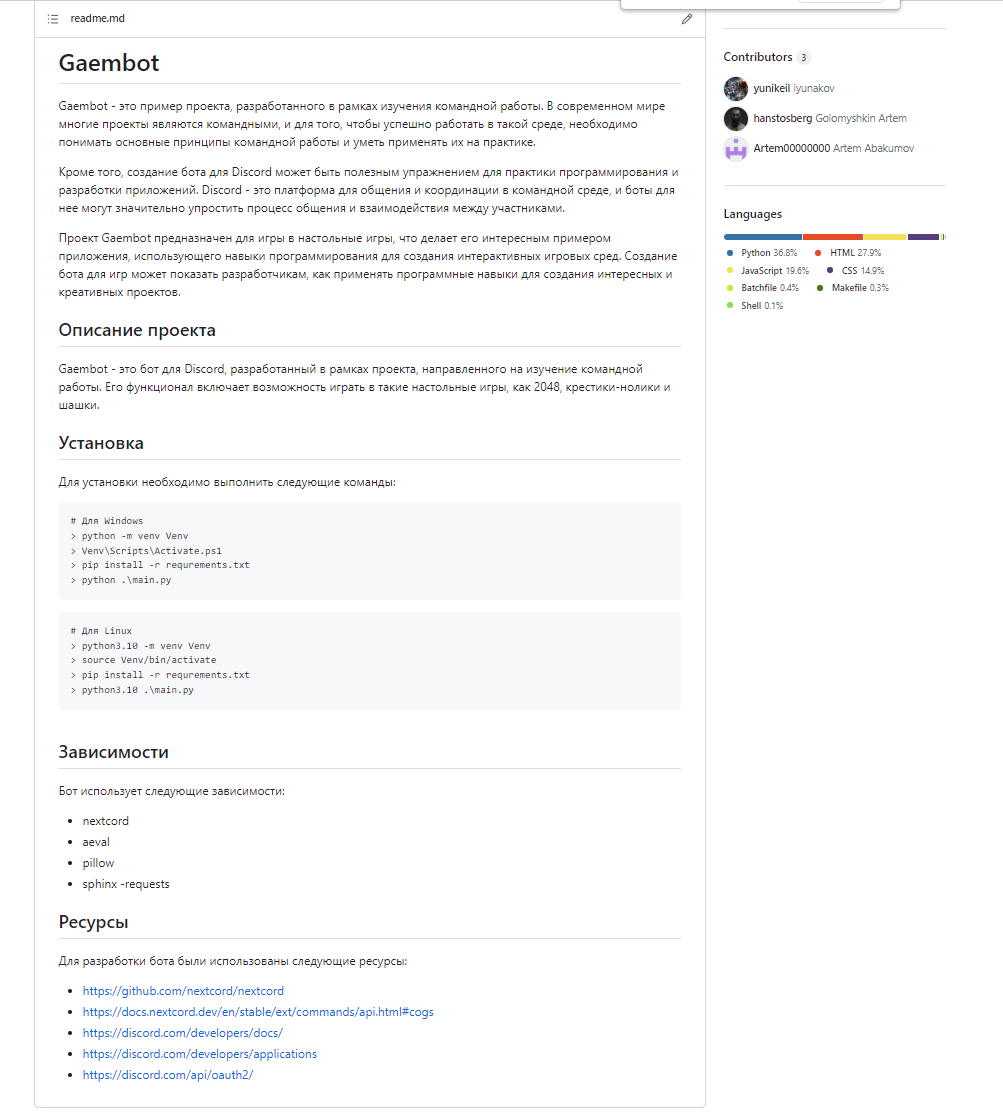


Рисунок 5 – Скриншот с главной страницы проекта

Листинг 1 – Markdown файл readme.md

1. **# Gaembot**
2. Gaembot - это пример проекта, разработанного в рамках изучения командной работы. В современном мире многие проекты являются командными, и для того, чтобы успешно работать в такой среде, необходимо понимать основные принципы командной работы и уметь применять их на практике.
3. Кроме того, создание бота для Discord может быть полезным упражнением для практики программирования и разработки приложений. Discord - это платформа для общения и координации в командной среде, и боты для нее могут значительно упростить процесс общения и взаимодействия между участниками.
4. Проект Gaembot предназначен для игры в настольные игры, что делает его интересным примером приложения, использующего навыки программирования для создания интерактивных игровых сред. Создание бота для игр может показать разработчикам, как применять программные навыки для создания интересных и креативных проектов.
5. **## Описание проекта**
6. Gaembot - это бот для Discord, разработанный в рамках проекта, направленного на изучение командной работы. Его функционал включает возможность играть в такие настольные игры, как 2048, крестики-нолики и шашки.
7. **## Установка**
8. Для установки необходимо выполнить следующие команды:
9. ```shell-session
10. # Для Windows
11. > python -m venv Venv
12. > Venv\Scripts\Activate.ps1
13. > pip install -r requrements.txt
14. > python .\main.py
15. ```
16. ```shell-session
17. # Для Linux
18. > python3.10 -m venv Venv
19. > source Venv/bin/activate
20. > pip install -r requrements.txt
21. > python3.10 .\main.py
22. ```
23. **## Зависимости**
24. Бот использует следующие зависимости:
25. - nextcord
26. - aeval
27. - pillow
28. - sphinx
29. - requests
30. **## Ресурсы**
31. Для разработки бота были использованы следующие ресурсы:
32. - https://github.com/nextcord/nextcord
33. - https://docs.nextcord.dev/en/stable/ext/commands/api.html#cogs
34. - https://discord.com/developers/docs/
35. - https://discord.com/developers/applications
36. - https://discord.com/api/oauth2/

Листинг 2 – Файл с зависимостями requirements.txt

1. aeval==0.1.3
2. aiohttp==3.8.4
3. aiosignal==1.3.1
4. alabaster==0.7.13
5. appdirs==1.4.4
6. async-timeout==4.0.2
7. attrs==22.2.0
8. autopep8==2.0.2
9. Babel==2.12.1
10. cattrs==22.2.0
11. certifi==2022.12.7
12. charset-normalizer==3.0.1
13. colorama==0.4.6
14. docutils==0.18.1
15. esbonio==0.16.1
16. exceptiongroup==1.1.1
17. frozenlist==1.3.3
18. idna==3.4
19. imagesize==1.4.1
20. Jinja2==3.1.2
21. lsprotocol==2023.0.0a1
22. Mako==1.2.4
23. Markdown==3.4.3
24. MarkupSafe==2.1.2
25. multidict==6.0.4
26. nextcord==2.3.3
27. packaging==23.0
28. pdoc3==0.10.0
29. Pillow==9.5.0
30. pycodestyle==2.10.0
31. pygls==1.0.1
32. Pygments==2.14.0
33. pyspellchecker==0.7.1
34. requests==2.28.2
35. snowballstemmer==2.2.0
36. Sphinx==6.1.3
37. sphinx-rtd-theme==1.2.0
38. sphinx\_autodoc\_typehints==1.22
39. sphinxcontrib-applehelp==1.0.4
40. sphinxcontrib-devhelp==1.0.2
41. sphinxcontrib-htmlhelp==2.0.1
42. sphinxcontrib-jquery==4.1
43. sphinxcontrib-jsmath==1.0.1
44. sphinxcontrib-qthelp==1.0.3
45. sphinxcontrib-serializinghtml==1.1.5
46. tomli==2.0.1
47. typeguard==2.13.3
48. typing\_extensions==4.5.0
49. urllib3==1.26.15
50. yarl==1.8.2