

Проект: Game Launcher

Основы алгоритмизации и программирования

СТУДЕНТ

Иванен Макар

Группа РПО 23/2

ПРЕПОДАВАТЕЛЬ

Рычихин Ярослав Алексеевич

ЛОКАЦИЯ

Академия ТОП

Санкт-Петербург, 2025

Введение в проект

Game Launcher — это десктопное приложение на Python для поиска, фильтрации и отображения информации о компьютерных играх.

Ключевые особенности:

- Модульная архитектура (ООП).
- Интеграция с внешним REST API.
- Поддержка онлайн и офлайн режимов.
- Адаптивный GUI на базе tkinter.
- Оптимизированные алгоритмы поиска.

600 × 400

Цели и Задачи



Основная цель

Создать полнофункциональное приложение для управления каталогом игр, демонстрирующее передовые практики программирования.



Алгоритмы

Реализация линейного, бинарного и расширенного поиска. Разработка комплексной системы фильтрации и сортировки.



Архитектура

Проектирование модульной структуры, разделение логики и представления, интеграция с API и обработка ошибок.

Архитектура Проекта

Слои Архитектуры

- **Data Layer (Данные):**
game.py, game_api.py
- **Business Logic (Логика):**
search_engine.py, filter_engine.py, network_utils.py
- **Presentation (Представление):**
launcher_gui.py, main.py

Принципы проектирования

- **SRP:** Принцип единственной ответственности.
- **Open/Closed:** Открытость для расширения.
- **Liskov Substitution:** Принцип подстановки.
- **Interface Segregation:** Разделение интерфейсов.
- **Dependency Inversion:** Инверсия зависимостей.

Ключевые Модули

game.py & game_api.py

game.py: Инкапсулирует данные игры (название, жанр, рейтинг). Использует геттеры и магические методы.

game_api.py: Взаимодействие с FreeToGame API, парсинг JSON и обработка сетевых ошибок.

search_engine.py

Реализует алгоритмы поиска:

- Линейный поиск ($O(n)$)
- Бинарный поиск ($O(\log n)$)
- Расширенный поиск по нескольким полям

filter_engine.py

Инструменты фильтрации по жанру, платформе, рейтингу. Поддержка множественной фильтрации ($O(n*k)$) и сортировки Timsort.

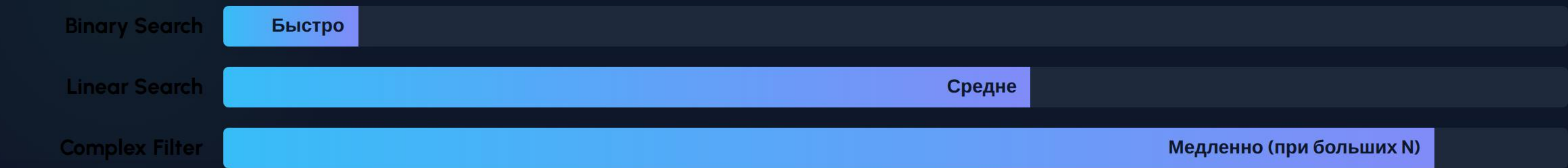
launcher_gui.py

Графический интерфейс на tkinter. Многопоточная загрузка данных, информационная панель с пингом и списки игр.

Сложность Алгоритмов

Алгоритм	Временная сложность	Применение в проекте
Линейный поиск	$O(n)$	Небольшие списки, поиск по описанию
Бинарный поиск	$O(\log n)$	Быстрый поиск по названию (отсортированные)
Расширенный поиск	$O(n \times m)$	Поиск по нескольким критериям одновременно
Сортировка (Timsort)	$O(n \log n)$	Упорядочивание списка игр по рейтингу

Визуальное сравнение эффективности:



Сеть и Интерфейс

Network Utils

Модуль `network_utils.py` обеспечивает диагностику соединения. Реализован пинг через TCP-сокеты (порт 53) для оценки качества связи в реальном времени.

GUI (Tkinter)

Класс `LauncherGUI` объединяет все компоненты. Используется многопоточность для фоновой загрузки данных, предотвращая "зависание" интерфейса при сетевых запросах.

- Индикация пинга цветом (Зеленый/Желтый/Красный)
- Обработка исключений и fallback-данные

600 × 400

Руководство пользователя

Сценарии работы

- **Поиск:** Ввод названия или части слова. Автоматический запуск алгоритма поиска.
- **Фильтрация:** Выбор жанра ("RPG", "Shooter") и платформы ("PC"). Установка минимального рейтинга ползунком.
- **Детали:** Клик по игре в списке открывает панель с полным описанием, датой релиза и рейтингом.
- **Офлайн режим:** При отсутствии сети загружаются локальные резервные данные.



Интерфейс приложения

Разработка и Расширение

1

База

Реализация классов Game и GameAPI. Базовая загрузка JSON.

2

Логика

Внедрение движков поиска (SearchEngine) и фильтрации (FilterEngine).

3

GUI & Сеть

Создание интерфейса на Tkinter и утилит сетевой диагностики.

4

Будущее

Добавление "Избранного", кэширование и индексация данных.

Тестирование

Проект покрыт модульными тестами (unittest) для проверки корректности создания объектов и методов поиска.

Заключение

Итоги реализации

Проект Game Launcher успешно демонстрирует применение ключевых парадигм программирования.

- **ООП:** Глубокое использование инкапсуляции и полиморфизма.
- **Алгоритмы:** Оптимизированная работа с данными.
- **UX:** Отзывчивый интерфейс с обратной связью.

Преимущества

- ✓ Чистый, документированный код
- ✓ Устойчивость к ошибкам (Fail-safe)
- ✓ Легкая расширяемость

Список источников

- **Python Tkinter Documentation:** docs.python.org/3/library/tkinter.html
- **FreeToGame API Docs:** freetogame.com/api-doc
- **Хабр:** "Реализация алгоритмов поиска и сортировки"
- **Хабр:** "Многопоточность в Tkinter приложениях"

г. Санкт-Петербург, 2025г.

Спасибо за внимание!

Готов ответить на ваши вопросы

