

О. А. Шведова¹✉, Е. Ю. Воронкин¹

Разработка программного обеспечения для генерации характеристик дизайна персонажей для видеоигр

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: olya.shvedova.96@gmail.com

Аннотация. В статье рассматривается процесс разработки программного обеспечения для генерации характеристик дизайна персонажей для видеоигр. Актуальность данной темы обусловлена существенными время- и трудозатратами художников по персонажам и концепт-художников при создании дизайна персонажей в сфере разработки видеоигр. В качестве основных инструментов для разработки программного обеспечения были выбраны язык программирования Python с библиотекой TKinter для создания графического интерфейса пользователя, модели искусственного интеллекта GigaChat и Kandinsky 3.1 от Fusion Brain, а также встраиваемая система по управлению базами данных SQLite. Процесс разработки программного обеспечения, получившего название «RandChar», состоял из создания концептуальной модели ПО, проектирования функциональной модели при помощи создания диаграмм UML IDEF0, прототипирования программного обеспечения и его тестирования на соответствие функциональным и нефункциональным требованиям. Реализован функционал работы с базой данных вариантов характеристик, режим тренировки с возможностью установки таймера, а также возможность генерации текстового описания персонажа на основе введенного пользователем игрового сеттинга с использованием искусственного интеллекта. Разработанное программное обеспечение было признано полностью соответствующим заявленным требованиям. Данная работа демонстрирует практическое применение современных программных средств для оптимизации и упрощения процесса работы художников по дизайну персонажей для видеоигр.

Ключевые слова: дизайн, база данных, SQLite, python, tkinter, нейросети для генерации изображений, нейросети для генерации текста, GigaChat, Fusion Brain

О. А. Shvedova¹✉, E. Yu. Voronkin¹

Software development for generating character design characteristics for video games

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: olya.shvedova.96@gmail.com

Abstract. The article discusses the process of developing software for generating character design characteristics for video games. The relevance of this topic is due to the significant time and labor costs of character artists and concept artists when creating character designs in the field of video game development. The Python programming language with the TKinter library for creating a graphical user interface, the GigaChat and Kandinsky 3.1 artificial intelligence models from Fusion Brain, as well as the SQLite embedded database management system were chosen as the main tools for software development. The process of developing the software, called "RandChar", consisted of creating a conceptual model of the software, designing a functional model using UML IDEF0 diagrams, prototyping the software and testing it for compliance with functional and non-functional requirements. The functionality of working with a database of characteristic variants, a training mode with the

ability to set a timer, as well as the ability to generate a text description of the character based on the game setting entered by the user using artificial intelligence were implemented. The developed software was recognized as fully compliant with the stated requirements. This work demonstrates the practical application of modern software tools for optimizing and simplifying the work process of character design artists for video games.

Keywords: design, database, SQLite, python, tkinter, neural networks for image generating, neural networks for text generating, GigaChat, Fusion Brain

Введение

Персонажи являются важной частью большинства современных видеоигр: начиная от героя, с которым себя ассоциирует игрок, и заканчивая неигровыми персонажами, так или иначе погружающие игрока в сеттинг придуманного мира [1]. Упрощение процесса работы над этими задачами позволит повысить общую эффективность, сократить время – и трудозатраты, а также позволить найти новые решения в дизайне персонажей для конкретного проекта. С целью автоматизации вышеуказанных процессов было принято решение о разработке специализированного программного обеспечения.

Актуальность создания программного обеспечения для генерации характеристик дизайна персонажей заключается в необходимости разработки концепции оригинальных дизайнов в сжатые сроки, что не всегда позволяет дизайнерам уделять достаточно времени поискам подходящих референсов и идей, что, в свою очередь, может привести к созданию некачественных или заурядных дизайнов и увеличить время работы над каждым из них за счет большого количества правок и переработок [2].

В ходе выполнения работ по проектированию и разработке программного обеспечения были созданы UML-диаграммы, концептуальная модель программного обеспечения, и база данных, которую использует в одном из режимов работы программное обеспечение (далее – ПО), разработан работающий прототип ПО и проведено тестирование для проверки соответствия ПО всем выдвинутым требованиям [3].

Таким образом, целью данной работы является разработка проекта программного обеспечения для генерации характеристик дизайна персонажей для видеоигр.

Для достижения поставленной цели были выполнены нижеперечисленные задачи:

- изучена предметная область;
- проведен анализ существующих программных решений с выявлением достоинств и недостатков каждого из них;
- создана концептуальная модель ПО;
- создана функциональную модель ПО средствами языка диаграмм UML и работающих с ним современных программных средств;
- создан прототип программного обеспечения для генерации характеристик дизайна персонажей для видеоигр средствами языка программирования Python и искусственного интеллекта.

– проведено тестирование ПО на соответствие выдвигаемым требованиям.

Методы и материалы

В качестве средств моделирования и проектирования разрабатываемого ПО выступили концептуальная модель, а также диаграммы UML и IDEF0.

Концептуальная модель позволяет отразить существенные для достижения цели моделирования элементы ПО и их свойства, а также связи между ними (рис. 1).



Рис. 1. Концептуальная модель программного обеспечения

Далее в процессе проектирования функциональной модели программного обеспечения были созданы диаграммы IDEF0 уровней A-0 и A0, которые предназначены для анализа, документирования и проектирования функциональных процессов в системах [4] (рис. 2, 3).

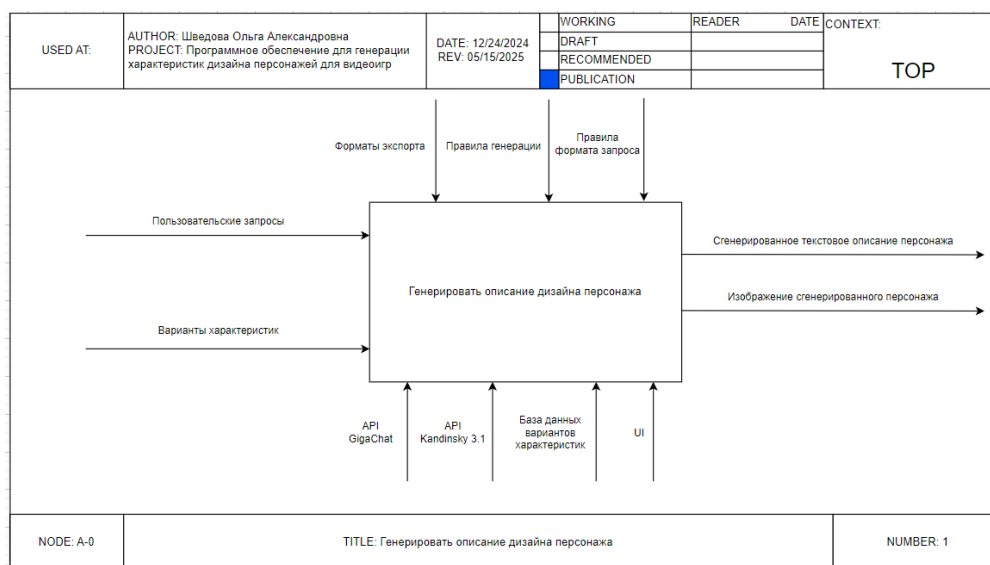


Рис. 2. Диаграмма IDEF0, уровень A-0

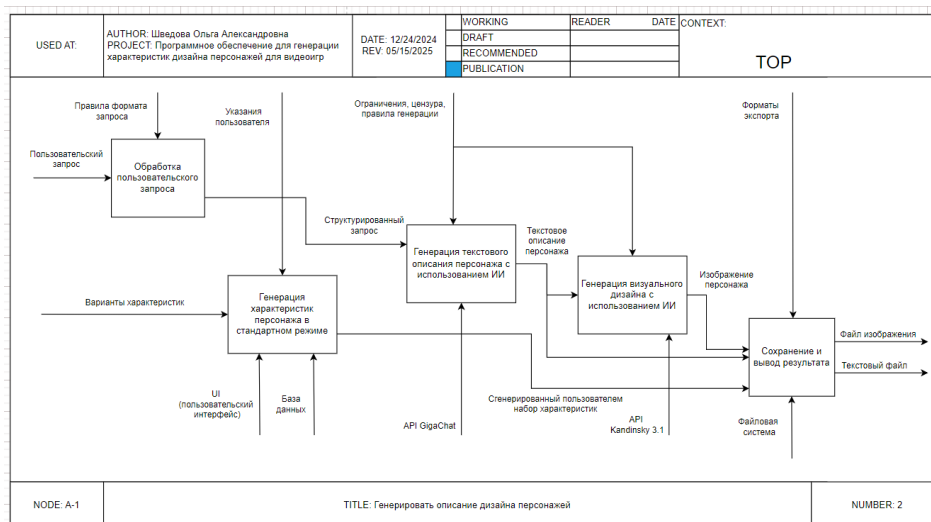


Рис. 3. Диаграмма IDEF0 верхнего уровня

Также была создана диаграмма компонентов, которая используется для моделирования физической структуры программной системы [5] (рис. 4).

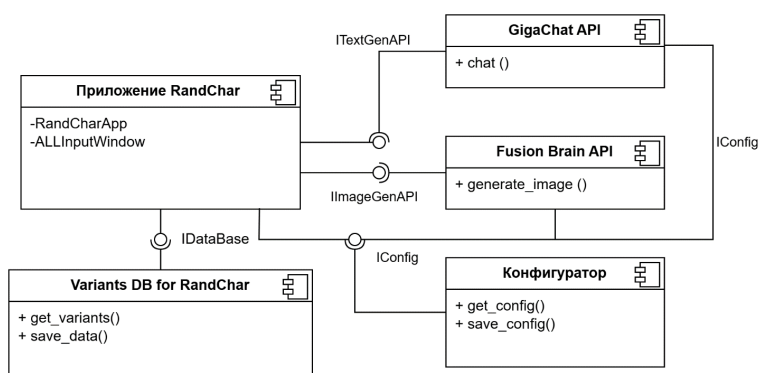


Рис. 4. Диаграмма компонентов

Перед началом разработки ПО путем проведения сравнительного был определен оптимальный стек технологий. В качестве языка программирования для реализации прототипа ПО был выбран Python [6]. GigaChat и Kandinsky 3.1 были использованы в качестве моделей искусственного интеллекта для генерации текстовых описаний и изображений персонажей соответственно [7]. Также для работы с базой данных вариантов характеристик было принято решение использовать систему для управления базами данных SQLite.

Результаты

Учитывая специализацию рабочего процесса художников и концепт-художников по персонажам к программному обеспечению для генерации характеристик дизайна персонажей предъявляются следующие функциональные требования:

- работа с базой данных, содержащей варианты для каждой из характеристик дизайна;
- режим тренировки с возможностью установки таймера;
- работа с генеративной моделью искусственного интеллекта для генерации текста, содержащего описание внешнего вида и краткого бэкграунда персонажа, по игровому сеттингу;
- возможность частичной работы без подключения к сети Интернет.

Время загрузки окон программы не должно превышать 5 секунд. Время генерации текстовых характеристик средствами искусственного интеллекта не должно превышать 10 секунд.

Интерфейс должен иметь русскую локализацию и быть понятным пользователям, знакомым с ПК на базовом уровне. Программное обеспечение должно работать на операционной системе Windows, версия 10 и выше. В случае возникновения ошибок в работе программы пользователь должен быть осведомлен о возникшей проблеме, например, с помощью диалоговых окон с описанием ошибки.

Прототип программного обеспечения разрабатывался средствами языка программирования Python с использованием следующих библиотек и модулей:

- Tkinter и ttk, PIL и ImageTK – для работы с GUI;
- sqlite3, configparser, json – для работы с данными;
- requests, gigachat и fusionbrain – для сетевых взаимодействий, включая работу с API нейросетей;
- os, sys, io, base64, time, random – системные модули.

В данном приложении используется преимущественно объектно-ориентированное программирование [8]. Крупные объекты, такие, как окна и наборы инструкций для взаимодействия с API, объединены в классы, применяется полиморфизм, то есть переопределение методов родительских классов, а также инкапсуляция – приватные методы и атрибуты классов. Каждый класс отвечает за свою функцию: запуск окон, взаимодействие с API, работа с всплывающими сообщениями и основной класс приложения. Модульная архитектура приложения делает его доступным для дальнейшей доработки и масштабирования без необходимости в существенной переработке программного кода.

Интерфейс окна стандартного режима работы программы представлен на рис. 5.

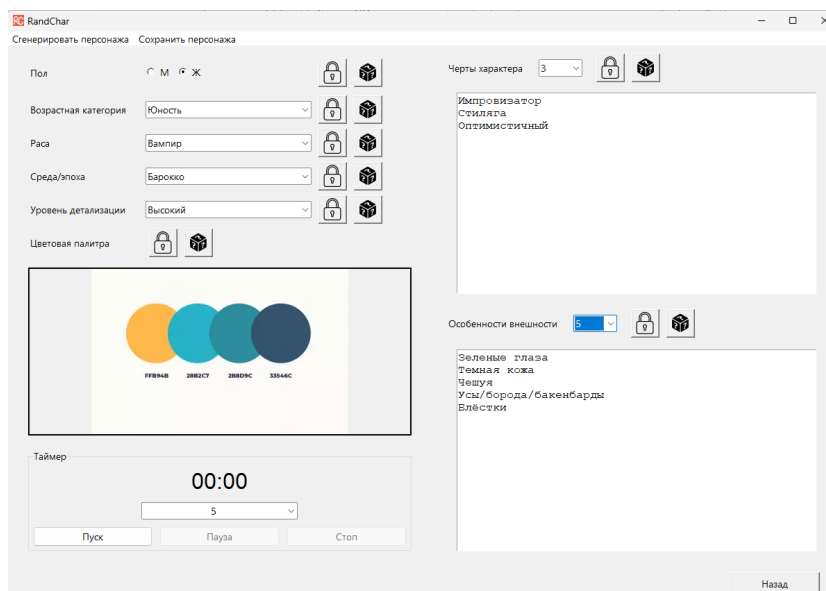


Рис. 5. Окно стандартного режима работы RandChar

RandChar использует API GigaChat для отправки запросов пользователей и получения сгенерированного описания персонажей в структурированном виде. Результат генерации текстового описания персонажа представлен на рис. 6.

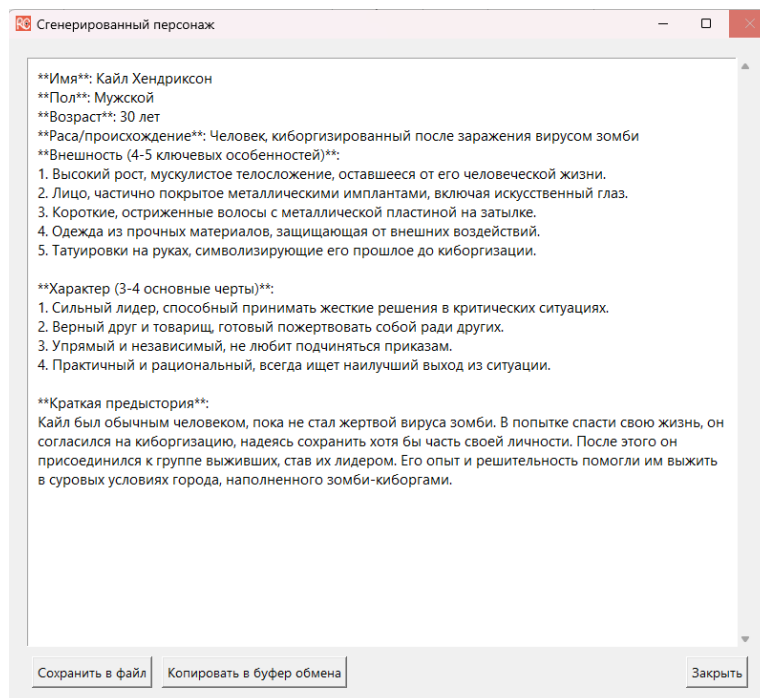


Рис. 6. Результат генерации описания персонажа нейросетью GigaChat

Для удобства пользователя реализованы всплывающие окна, информирующие его в случае возникновения ошибок или, напротив, успешного выполнения той или иной операции.

Для проверки ПО на соответствие функциональным требованиям было принято решение провести ручное тестирование [9]. Каждая из функций была протестирована разработчиком вручную на предмет возникновения ошибок или некорректной работы. В результате тестирования были выявлены и исправлены следующие ошибки в работе ПО:

- при отсутствии данных в базе данных не выводились сообщения об ошибке;
- не воспроизводился звуковой сигнал при завершении работы таймера;
- не появлялось сообщение об отсутствии подключения к сети Интернет.

Для проверки ПО на соответствие требованиям производительности были созданы автоматические тесты с использованием стандартной библиотеки `unittest` и модуля `time` [10]. Результаты автоматического тестирования производительности отражены в таблице 1.

Таблица 1

Результаты автоматического тестирования производительности RandChar

Тестируемый процесс	Время выполнения
Загрузка окон программы	от 0,67 до 3,06 с.
Выполнение запроса GigaChat	от 0,95 до 1,12 с.

Заключение

В статье рассматривается процесс разработки программного обеспечения для генерации характеристик дизайна персонажей для видеоигр, получившего название «RandChar». Целью данной работы было создание удобного инструмента для концепт-художников и дизайнеров персонажей, которое позволило бы сократить время на поиск интересных идей, автоматизировать процесс генерации новых решений в дизайне и дало возможность тренироваться в создании быстрых концептов.

В качестве языка программирования для разработки ПО был выбран Python. Данный язык хорошо подходит для работы с нейросетями, так как обладает большим количеством соответствующих библиотек, а также является достаточно производительным и популярным среди современного сообщества программистов. Среди множества нейросетей для генерации текста и изображений были выбраны GigaChat и Kandinsky. Не менее важным этапом подготовки к разработке программного обеспечения стал процесс выбора системы для управления базами данных для хранения вариантов генерируемых характеристик, в результате которого было принято решение об использовании СУБД SQLite.

Для визуального представления структуры, поведения и взаимодействия компонентов ПО были разработаны следующие диаграммы и схемы: диаграмма прецедентов, диаграммы IDEF0, диаграмму прецедентов, а также физическую модель базы данных. Завершающим этапом стала разработка прототипа программного обеспе-

чения для генерации характеристик дизайна персонажей «RandChar», которое полностью соответствует всем предъявляемым к вновь разрабатываемому ПО требованиям.

Итог: программное обеспечение полностью соответствует всем предъявляемым требованиям и является готовым к дальнейшему использованию в сферах концепт-арта и дизайна персонажей для видеоигр.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ларицкая М.Г. Дизайн и архитектура как инструмент повествования в игровом виртуальном мире // Архивариус. 2021. №3 (57). URL: <https://cyberleninka.ru/article/n/dizayn-i-arhitektura-kak-instrument-povestvovaniya-v-igrovom-virtualnom-mire> (дата обращения: 18.05.2025).
2. Тамулевич С.В. Концепт-арт компьютерной игры в формировании профессиональных компетенций графических дизайнеров // Проблемы современного педагогического образования. 2024. №84-1. URL: <https://cyberleninka.ru/article/n/kontsept-art-kompyuternoy-igry-v-formirovanii-professionalnyh-kompetentsiy-graficheskikh-dizaynerov> (дата обращения: 18.05.2025).
3. Шатова А. Д. Использование UML-диаграмм для оптимизации бизнес-процессов // Инновационная наука. 2024. №6-2. URL: <https://cyberleninka.ru/article/n/ispolzovanie-uml-diagramm-dlya-optimizatsii-biznes-protsessov> (дата обращения: 16.05.2025).
4. Лапина М. А., Ржевская Н. В., Медведева А. С., Золотова А. Г. Применение диаграммы idef0 для наглядного представления структуры проекта // Auditorium. 2023. №1 (37). URL: <https://cyberleninka.ru/article/n/primenenie-diagrammy-idef0-dlya-naglyadnogo-predstavleniya-struktury-proekta> (дата обращения: 20.05.2025).
5. Диаграмма компонентов // Википедия URL: https://ru.wikipedia.org/wiki/Диаграмма_компонентов (дата обращения: 15.05.2024).
6. Ельсуков Д. А. Python - язык программирования // Экономика и социум. 2021. №11-1 (90). URL: <https://cyberleninka.ru/article/n/python-yazyk-programmirovaniya> (дата обращения: 08.05.2025).
7. Алексеева Л. Г., Алексеев П. С. Язык промптов, или особенности формулирования запросов к генеративным нейросетям для создания изображений // Verba. Северо-Западный лингвистический журнал. 2024. №3 (13). URL: <https://cyberleninka.ru/article/n/yazyk-promptov-ili-osobennosti-formulirovaniya-zaprosov-k-generativnym-neyrosetyam-dlya-sozdaniya-izobrazheniy> (дата обращения: 18.05.2025).
8. Ремазанов И., Гулджанова Д., Ходжамырадов Р. Особенности объектно-ориентированного программирования // Вестник науки. 2024. №5 (74). URL: <https://cyberleninka.ru/article/n/osobennosti-obektno-orientirovannogo-programmirovaniya> (дата обращения: 20.05.2025).
9. Царев Ю. В., Сильянова Е. Ф., Кисельников С. А. Особенности ручного и автоматизированного тестирования программного обеспечения // Вестник науки. 2021. №6 (39). URL: <https://cyberleninka.ru/article/n/osobennosti-ruchnogo-i-avtomatizirovannogo-testirovaniya-programmnogo-obespecheniya> (дата обращения: 20.05.2025).
10. Сарычева Ю.Р., Липатова С.Е. Автоматизированное тестирование API // StudNet. 2021. №7. URL: <https://cyberleninka.ru/article/n/avtomatizirovannoe-testirovanie-api> (дата обращения: 20.05.2025).

© О. А. Шведова, Е. Ю. Воронкин, 2025