

Т. О. Ермакова¹✉, Н. С. Головачев¹

Проектирование и программная реализация интерфейсов для системы взаимодействия пользователей с проектом «BlushBound»

¹Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
e-mail: ermakova06tanya@mail.ru

Аннотация. В статье представлены результаты проектирования и программной реализации адаптивной подсистемы пользовательских интерфейсов для игрового проекта «BlushBound». Выполнен анализ проблемы аппаратной фрагментации дисплеев и необходимости интеграции систем динамической локализации в инди-проектах. Для создания клиентского модуля использованы среда разработки Unity 6 и язык C#. Разработана трехуровневая архитектура, базирующаяся на паттерне Singleton и событийно-ориентированной модели взаимодействия компонентов. Реализованы алгоритмы динамического перестроения макетов под сверхширокие экраны, расчет безопасных зон и подсистема инклюзивных настроек с динамическим масштабированием шрифтов. Программное решение позволяет бесшовно интегрировать игровые механики с экранами сюжетной сцены, обеспечивая высокий уровень пользовательского опыта.

Ключевые слова: пользовательский интерфейс, адаптивная верстка, локализация, Unity, паттерны проектирования

T. O. Ermakova¹✉, N. S. Golovachev¹

Design and software implementation of user interfaces for the «BlushBound» project

¹Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: ermakova06tanya@mail.ru

Abstract. The article presents the results of the design and software implementation of an adaptive user interface subsystem for the «BlushBound» game project. An analysis of the hardware displays fragmentation problem and the need for dynamic localization systems in indie projects is performed. The Unity 6 development environment and the C# language were used to create the client module. A three-tier architecture based on the Singleton pattern and an event-driven component interaction model was developed. Algorithms for dynamic layout rebuilding for ultrawide screens, safe area calculation, and an inclusive settings subsystem with dynamic font scaling were implemented. The software solution seamlessly integrates game mechanics with story scene screens providing a high level of user experience.

Keywords: user interface, adaptive layout, localization, Unity, design patterns

Введение

Развитие индустрии интерактивных развлечений диктует новые стандарты качества для разрабатываемых программных продуктов. Графический пользовательский интерфейс и пользовательский опыт перешли в разряд ключевых факторов, напрямую определяющих коммерческую успешность приложения. Современный

графический интерфейс выполняет сложную комплексную задачу: он не только предоставляет интуитивный доступ к базовому функционалу программы, но и обеспечивает глубокое погружение пользователя в виртуальную среду, снижая когнитивную нагрузку при взаимодействии с контентом. Особенностью проектирования интерфейсов для гибридных жанров, к которым относится проект «BlushBound», является необходимость органичного совмещения различных визуальных пространств без перегрузки экрана [1, 2].

Главной трудностью при конструировании визуальных элементов для современного программного обеспечения выступает непрерывная аппаратная фрагментация и физическая разнородность экранов. Матрицы конечных устройств варьируются от традиционных пропорций шестнадцать к девяти до панорамных форматов сверхшироких мониторов. Кроме того, мобильный сегмент отягощен наличием аппаратных вырезов под оптические сенсоры и системные панели. Применение статической верстки и жестко заданных координат в подобной среде неизбежно влечет за собой искажение графики, обрезку текста или полное перекрытие интерактивных элементов. Дополнительным вызовом является обязательная потребность в интернационализации продукта. Проекту требуется централизованный механизм языковой адаптации, позволяющий транслировать тексты без модификации исходного кода. Целью данного исследования является проектирование и программная реализация отказоустойчивой, эластичной и масштабируемой системы пользовательских интерфейсов клиентского модуля для игрового проекта «BlushBound».

Методы и материалы

В качестве основного технологического фундамента для создания визуальной логики интерфейсов применялась программная платформа шестого поколения Unity, а кодовая база и математические алгоритмы формировались посредством объектно-ориентированного языка программирования C# [3]. Использование среды Unity продиктовано широкими встроенными возможностями подсистемы визуализации Canvas, а также наличием библиотеки TextMeshPro. Данная библиотека осуществляет векторное построение текстовых символов на основе вычисления полей расстояний. Внедрение данной технологии имеет критическое значение для интерфейса, так как оно полностью исключает размытие пикселей и потерю качества графики при масштабировании текстов на дисплеях сверхвысокого разрешения [4, 5].

Особое внимание при проектировании клиентского модуля было уделено разработке структуры самих пользовательских интерфейсов на основе принципов гештальт-психологии и модульных сеток. Макет главного меню спроектирован по модульному принципу: графическое пространство логически разделено на функциональные навигационные блоки. Центральную и фоновую часть занимает художественная иллюстрация, создающая эмоциональную привязку. Левая панель интерфейса инкапсулирует клавиши доступа к коллекциям героев и системным настройкам, тогда как правая панель выделена под акцентные элементы запуска бесконечного или сюжетного игровых режимов. Подобное зонирование

снижает время визуального поиска нужного раздела пользователем и формирует четкие паттерны взаимодействия.

Для обеспечения интернационализации графических оболочек была спроектирована подсистема централизованной локализации. Базы данных словарей и переводов опираются на легковесный табличный формат CSV. Структура файла локализации предусматривает наличие заголовочной строки с языковыми кодами, где каждой последующей строке строго сопоставлен уникальный графический ключ-идентификатор и соответствующие текстовые переводы, разделенные символом точки с запятой.

Для обеспечения стабильной работы интерфейсов, алгоритмов адаптивности и подсистемы локализации общая архитектура приложения была разграничена на изолированные слои. Глобальные управляющие классы, такие как контроллер навигации `SceneController`, менеджер параметров `SettingsManager` и менеджер локализации `LocalizationManager`, выстроены с применением порождающего паттерна `Singleton`. Данный управляющий узел инициализируется единожды внутри отдельной стартовой сцены `BootScene` и функционирует полностью автономно на протяжении всей пользовательской сессии. Корректность отображения на экранах различных форматов дополнительно обеспечивалась разработкой вспомогательных скриптов `AspectRatioLayout` и `SafeAreaFitter`. Указанные алгоритмы вычисляют доступные физические габариты монитора и программно сжимают графический интерфейс в пределы гарантированной видимости, защищая интерактивные элементы от перекрытия аппаратными вырезами мобильных устройств.

Результаты

Главным практическим результатом проведенных исследовательских и проектных работ стала комплексная визуальная и программная реализация экранов взаимодействия клиентской среды. Итоговый программный продукт органично объединяет масштабируемое стартовое меню, инклюзивные панели пользовательских конфигураций и производительную систему локализации.

Визуальное представление главного меню и системы навигации демонстрирует интуитивно понятное расположение элементов управления. Для обеспечения высокого уровня доступности интерфейса дополнительно реализован механизм динамического масштабирования типографики. Специализированный модуль `UIFontScaler`, перехватывая системный сигнал об изменении конфигурации из меню настроек, математически умножает исходный размер базового шрифта на заданный пользователем числовой коэффициент. Кроме того, для безопасного завершения работы приложения было реализовано модальное окно `ExitConfirmDialog`. Данный графический узел перекрывает основное меню затемненным фоном, полностью блокируя случайные нажатия на элементы заднего плана, и ожидает явного подтверждения действия от пользователя (рис.1).

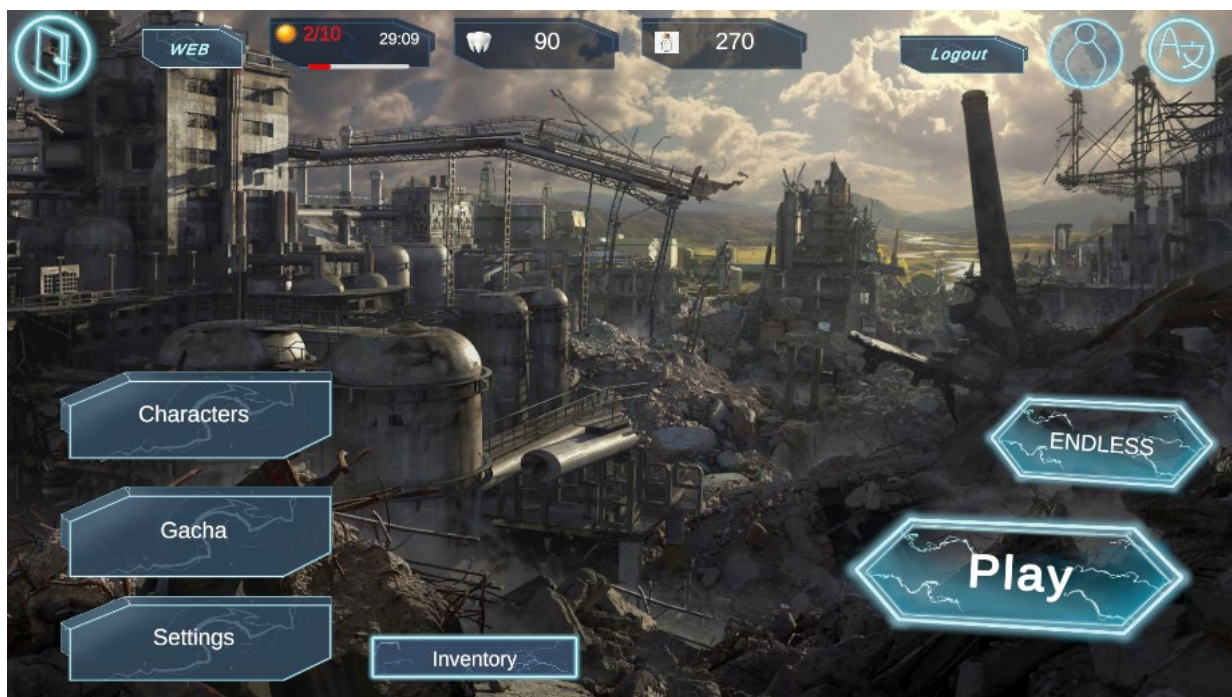


Рис. 1. Визуальная реализация адаптивного пользовательского интерфейса клиентского модуля

Программный механизм локализации графических экранов опирается на ресурсосберегающую событийно-ориентированную архитектуру. Логика работает по следующему алгоритму: элемент-инициатор передает команду в управляющий класс локализации. Класс обрабатывает запрос, сохраняет выбранный язык в локальную энергонезависимую память устройства через интерфейс `PlayerPrefs` и генерирует глобальное системное событие `LanguageChanged`.

Все активные текстовые узлы интерфейса, содержащие скрипт `LocalizedTextTMP`, выступая в роли подписчиков, перехватывают данный сигнал. Получив уведомление, они автономно извлекают корректный перевод из загруженного в оперативную память CSV-массива и мгновенно обновляют свое визуальное содержимое. Указанная программная методология полностью устраняет потребность в ресурсоемком циклическом опросе или программном поиске графических компонентов на сцене (рис. 2).

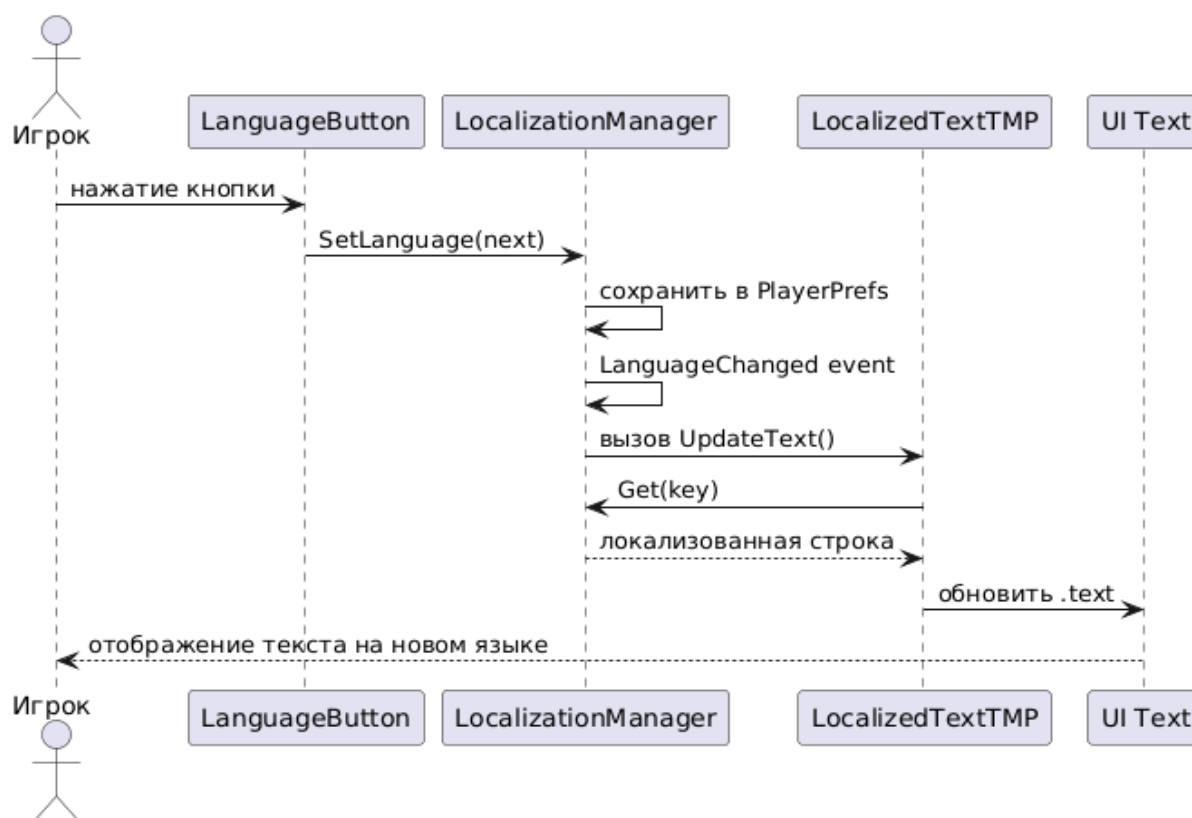


Рис. 2. Алгоритм событийного обновления локализованного текста в интерфейсе

Дополнительно в рамках работы сюжетный нарративный кластер был спроектирован как изолированная интерактивная среда «StoryScene_FirstMeeting». В текстовый интерфейс визуальной новеллы внедрен алгоритм графической фокусировки для управления вниманием игрока: художественный портрет активного спикера отображается на экране с максимальной непрозрачностью, тогда как графическая модель слушающего персонажа программно затемняется путем снижения альфа-канала до значения ноль целых пять десятых.

Заключение

В результате проведенного исследования и программной разработки была создана отказоустойчивая подсистема пользовательских интерфейсов для проекта «BlushBound». Использование алгоритмов расчета безопасных зон и скриптов динамического переключения макетов позволило обеспечить полную кроссплатформенную адаптивность интерфейса. Разработанная архитектура на базе паттерна Singleton и системы событий успешно решает задачи централизованного управления настройками и мгновенной локализации. Бесшовная интеграция модуля визуальной новеллы подтверждает надежность спроектированной навигационной системы. Предложенные архитектурные решения могут быть использованы при разработке масштабируемых интерфейсов для коммерческих кроссплатформенных приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сондерс, К. Проектирование игровых интерфейсов / К. Сондерс, П. Новак. – Москва : ДМК Пресс, 2020. – 312 с.
2. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – Санкт-Петербург : Питер, 2021. – 368 с.
3. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5: основные принципы и практики программирования / Э. Троелсен, Ф. Джепикс. – Москва : Вильямс, 2022. – 1328 с.
4. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг. – Санкт-Петербург : Питер, 2023. – 400 с.
5. Unity User Manual 2023. User Interfaces (UI) [Электронный ресурс] // Unity Technologies. – URL: <https://docs.unity3d.com/Manual/UIToolkits.html> (дата обращения: 12.05.2026).

© Т. О. Ермакова, Н. С. Головачев, 2026