

Техническое задание (ТЗ)

Адаптация Unity 3D игры (Android) под Telegram Games (Telegram Mini App / WebApp)

Версия: 0.1 Дата: 26.12.2025 Таргет окружение: Telegram iOS/Android + Telegram Desktop (Web — best effort). Технологии: Unity (Web), C#, 3D.

Цель работ: получить Web-версию игры, запускаемую внутри Telegram Mini App (WebView), с облачными сохранениями и backend-интеграцией для сервисных функций (например: лидерборды/аккаунт/инвентарь) через валидацию initData.

1. Входные данные (предоставляет заказчик)

- Unity-проект (архив/репозиторий) и текущий Android-билд (при наличии).
- Список сцен/режимов, которые должны быть доступны в Telegram-версии (score/MVP).
- Список внутренних SDK/плагинов проекта (перечень).
- Доступы/контакты по Telegram-боту и BotFather (кто отвечает за настройку/владение ботом).

2. Объём работ (делает исполнитель)

2.1 Аудит проекта под Web-окружение Telegram (WebView)

- Проверка совместимости ассетов/шейдеров/плагинов с Web-окружением.
- Оценка рисков по производительности/памяти и времени загрузки.
- Фиксация версии Unity и рендер-пайплайна проекта (Built-in / URP / HDRP) по факту аудита, с внесением в README.

2.2 Сборка Web-версии (Unity) и запуск внутри Telegram

- Настройка Web-сборки и стабильного запуска внутри Telegram Mini App (WebView).
- Корректная работа при изменении видимой области (viewportChanged), ресайзе/сворачивании/разворачивании Mini App.
- Учёт safe area и content safe area (чтобы UI не перекрывался элементами Telegram/вырезами экрана).

2.3 Интеграция Telegram WebApp SDK (обязательно)

- Подключить telegram-web-app.js в до других скриптов.
- Вызвать Telegram.WebApp.ready() после того, как приложение готово к отображению.
- Обработка событий Telegram: как минимум viewportChanged/safeAreaChanged (если требуется по UI).

2.4 Backend-интеграция (обязательно)

- Передавать initData (Telegram.WebApp.initData) на backend.
- Валидировать initData по алгоритму Telegram (hash/ HMAC; проверка auth_date на актуальность) и только после этого доверять данным пользователя.
- Использовать валидированную идентичность пользователя для функций: лидерборды/аккаунт/инвентарь (конкретный функционал уточняется в проекте).

2.5 Сохранения (обязательно)

- Сохранения реализуются через Telegram WebApp CloudStorage.

- Учитывать ограничения CloudStorage: лимит 1024 items на пользователя на бота; ограничения по ключам (символы/длина).
- Версионирование формата сохранений (чтобы обновления не ломали прогресс).

2.6 Монетизация (TBD)

- Монетизация обсуждается отдельно после подписания NDA.
- Если монетизация будет включена: цифровые товары/сервисы в Telegram продаются через Telegram Stars (currency XTR); использование openInvoice() и обработка invoiceClosed — по требованиям Telegram.

3. Технические требования Telegram Mini Apps, которые должны быть соблюдены

1. Mini App работает внутри Telegram клиентов (iOS/Android/Desktop) и не зависит от установки приложения.
2. Подключение telegram-web-app.js в до других скриптов; использование объекта window.Telegram.WebApp.
3. Корректное использование ready() для уведомления Telegram о готовности приложения.
4. Учёт динамической высоты видимой области Mini App (viewportChanged) и safe area inset / content safe area inset.
5. CloudStorage для сохранений: лимит 1024 items/пользователь/бот; ключи 1-128 символов и только A-Z, a-z, 0-9, _, - .
6. Backend обязателен: валидация initData на сервере (hash/HMAC + auth_date) перед использованием пользовательских данных.
7. Хостинг: для production URL Mini App должен быть HTTPS; в test environment возможны HTTP links без TLS (для разработки).

4. Результат работ и критерии приёмки

4.1 Deliverables

- Web-билд (архив/папка сборки) для размещения по HTTPS URL Mini App.
- Unity-проект/ветка с изменениями под Web + интеграцией Telegram WebApp SDK.
- Инструкция деплоя/обновления: где задаётся URL Mini App в BotFather/настройках бота и как тестировать.
- Документ по backend-валидации initData (какие параметры передаются, где проверяются, требования к auth_date).
- Док по сохранениям: ключи, структура, версия, миграции.

4.2 Критерии приёмки

1. Mini App открывается из Telegram и отображается без залипаний; выполнен ready() после загрузки.
2. Игра стабильна на Telegram iOS/Android + Desktop в согласованном тестовом сценарии.
3. CloudStorage сохраняет прогресс между сессиями; ограничения учтены (лимиты/ключи).
4. initData валидируется на backend; при невалидных/устаревших данных доступ к серверным функциям блокируется.
5. UI не обрезается важными зонами (safe area / content safe area) и корректно реагирует на изменения viewport.

Ссылки на официальную документацию

Telegram Mini Apps (основная документация WebApp API: initData, события, safe area, CloudStorage, openInvoice и т.д.): <https://core.telegram.org/bots/webapps>

Validating data received via the Mini App (алгоритм проверки initData на backend, auth_date):

<https://core.telegram.org/bots/webapps#validating-data-received-via-the-mini-app>

CloudStorage (лимит 1024 items/пользователь/бот, ограничения ключей):

<https://core.telegram.org/bots/webapps#cloudstorage>

Payments for Digital Goods (Telegram Stars, валюта XTR):

<https://core.telegram.org/bots/payments-stars>

Bot features: test environment (HTTP links без TLS для тестов):

<https://core.telegram.org/bots/features>

Test environment (production требует HTTPS, test допускает HTTP/IP) — справочная платформа: <https://docs.telegram-mini-apps.com/platform/test-environment>

Getting App Link (BotFather требует HTTPS — справочная платформа):

<https://docs.telegram-mini-apps.com/platform/getting-app-link>