План Тестирования: Шахматы: Дополнение

Раздел 1: Введение и Цель Тестирования

Основная цель тестирования проекта "Шахматы: Дополнение" – подтвердить, что весь функционал игры работает согласно Техническому Заданию, обеспечить стабильность работы системы, и проверить удобство использования для будущих игроков. Тестирование также направлено на выявление и исправление ошибок (багов) и логических несоответствий.

Элементы системы, подвергаемые тестированию:

Godot-клиент (графический интерфейс, рендеринг, пользовательский ввод)

Python-сервер (игровая логика, обработка матчей, ГСЧ)

Сетевое взаимодействие (корректность обмена сообщениями по API)

Взаимодействие клиента и сервера в целом.

Элементы системы, не подвергаемые тестированию на данном этапе:

Взаимодействие с будущей Базой Данных: тестирование данного аспекта будет отложено до момента фактической реализации и интеграции БД.

Сторонние библиотеки: используются как есть, их внутренняя логика не тестируется.

Раздел 2: Стратегия Тестирования

Применяемые типы тестирования (в порядке приоритета):

Функциональное тестирование: Наивысший приоритет. Проверка корректной работы всех игровых механик: ходы фигур, способности Редких Героев и Легендарных Воинов, эффекты всех типов карт (Аура, Состояние,

Очищение, Переманивание, Изменение типа), влияние смены дня/ночи и материала доски, механика Двойного добора, корректность работы ГСЧ для вероятностных событий, все условия победы/поражения (мат, пат, сдача).

Тестирование удобства использования (UI/UX): Высокий приоритет. Оценка интуитивности интерфейса, удобства управления, понятности игрового процесса и обратной связи для пользователя. Это критически важно для привлекательности игры.

Тестирование безопасности: Средний приоритет. Проверка базовых аспектов безопасности, таких как предотвращение непредусмотренного вмешательства (вся игровая логика на сервере, валидация запросов). Учитывая отсутствие уязвимой личной информации и отсутствие шифрования, фокус на предотвращении манипуляций с игровой логикой.

Тестирование нагрузки: Средний приоритет. Оценка поведения сервера и клиента под умеренной нагрузкой (например, несколько одновременных подключений, интенсивный обмен данными). Цель – убедиться, что система остается отзывчивой. (Нижняя граница нагрузки не устанавливается, но стремимся к стабильной работе).

Системное тестирование: Средний приоритет. Комплексная проверка всей системы, имитирующая реальное использование, включая установление, поддержание и восстановление сетевого соединения.

Методология проведения тестирования:

Гибридное тестирование: Сочетание ручного и автоматизированного подходов.

Ручное тестирование: Будет использоваться для имитации сложного пользовательского ввода, оценки UI/UX, а также для исследования трудновоспроизводимых сценариев и непредсказуемых взаимодействий.

Автоматизированное тестирование: Будет применяться для проверки работоспособности конкретных функций (модульные тесты для серверной логики на Python) и для нагрузочного тестирования. Предполагается использование автоматизированных инструментов для тестирования и логирования.

Регрессионное тестирование: Будет проводиться на всех этапах функционального расширения (т.е. постоянно в течение Фазы 3 "Реализация"), чтобы убедиться, что новые изменения не нарушают уже существующий и корректно работающий функционал.

Критерии входа и выхода из фазы тестирования:

Критерии входа в основную фазу тестирования (Фаза 4):

Все основные функции сервера и клиента, описанные в ТЗ, реализованы.

Спецификации API стабилизированы и имплементированы.

Базовые модульные тесты для серверной логики написаны.

Критерии выхода из фазы тестирования:

Все критические и высокоприоритетные дефекты, выявленные в ходе тестирования, исправлены и проверены.

Весь функционал игры, описанный в ТЗ, проверен и соответствует требованиям.

Тестирование удобства использования не выявило серьезных проблем.

Отсутствие новых критических дефектов в течение заранее определенного периода (например, 1-2 недели).

Раздел 3: Тестовая Среда

Окружение тестирования:

Серверная часть: Тестирование Python-сервера будет производиться на виртуальной машине или локальном сервере, имитирующем предполагаемое развертывание.

Клиентская часть: Тестирование Godot-клиента будет проводиться на физических устройствах и/или виртуальных машинах с различными уровнями выделенных ресурсов (разная производительность), чтобы оценить работу игры на широком спектре конфигураций.

Инструменты тестирования:

Будут выбраны и использованы автоматизированные инструменты для тестирования (например, фреймворки для модульного тестирования на Python, возможно, инструменты для API-тестирования).

Для логирования событий и ошибок будут использоваться автоматизированные системы логирования.

Для отслеживания дефектов (багов) будет использоваться электронный список (например, Google Sheets, Trello, или простая внутренняя система).

Раздел 4: Задачи Тестирования и Покрытие

Основные функции/механики, подлежащие проверке:

Все функциональные требования, выставленные в ТЗ, будут проверены. Это включает, но не ограничивается:

Базовые шахматы: Ходы всех фигур, взятие, шах, мат, пат, рокировка, взятие на проходе.

Новые фигуры: Способности Архангела (защита, воскрешение), Некроманта (воскрешение, влияние ГСЧ), Морфа (копирование, изменение состояния).

Карточная система: Добор карт, обмен карт, разыгрывание всех типов карт (Аура, Состояние, Очищение, Переманивание, Изменение типа).

Взаимодействия: Сложные взаимодействия между аурами, состояниями, разными типами фигур, эффектами карт, а также влияние смены дня/ночи и материала доски.

Динамические элементы: Корректная работа механики смены дня/ночи, влияния материалов доски.

Механика Двойного добора: Корректное выполнение "взять две" и "обменять одну на две", соблюдение ограничений на 3 использования за игру для игрока.

ГСЧ: Проверка того, что вероятностные события (Некромант, Забывчивость) происходят согласно заявленным шансам и не приводят к непредсказуемым сбоям.

Условия завершения игры: Корректная фиксация мата, пата, сдачи, отключения игрока.

Сетевое взаимодействие: Стабильность соединения, отсутствие рассинхронизаций, корректная обработка отключений/переподключений.

API: Соответствие передачи данных между клиентом и сервером Спецификациям API.

UI/UX: Интуитивность интерфейса, читаемость информации, отсутствие визуальных артефактов.

Покрытие тестированием:

Покрытие будет отслеживаться путем ведения списка всех функциональных и нефункциональных требований, где для каждого будет отмечено, что оно успешно протестировано. Приоритет будет отдаваться покрытию всех критических путей и механик.

Раздел 5: Роли и Ответственность

Ответственные за тестирование:

Основную ответственность за планирование и выполнение тестирования несет самостоятельно инди-разработчик.

Привлечение добровольцев или друзей для проведения пользовательского (игрового) тестирования возможно для получения обратной связи по UI/UX и общему игровому опыту.

Управление дефектами:

Найденные дефекты (баги) будут фиксироваться в электронном списке, который будет содержать: краткое описание, шаги для воспроизведения, ожидаемый результат, фактический результат, приоритет (критический, высокий, средний, низкий) и версию ПО, на которой был обнаружен баг.

Приоритеты багов:

Критический: Блокирует основной функционал, приводит к крашу, делает игру неиграбельной.

Высокий: Серьезно влияет на игровой процесс, но не блокирует полностью.

Средний: Незначительное влияние на функционал или UI, есть обходные пути.

Низкий: Косметические дефекты, опечатки, незначительные улучшения UI/UX.

Раздел 6: Расписание Тестирования

Этапы проведения тестирования:

В течение Фазы 3 (Реализация, Недели 15-22):

Модульное и Интеграционное тестирование: Непрерывное тестирование отдельных модулей сервера и клиента, а также их взаимодействия (API) по мере реализации функционала. Проведение регрессионного тестирования после каждого значительного изменения или добавления новой механики.

Фаза 4 (Полировка, оптимизация и тестирование, Недели 23-28):

Неделя 23-24: Основное функциональное тестирование всех реализованных механик. Начальное тестирование производительности.

Неделя 25-26: Углубленное тестирование всех взаимодействий (ауры-состояния-фигуры-карты), тестирование на предмет логических несоответствий и редких сценариев. Повторное регрессионное тестирование.

Неделя 27-28: Тестирование удобства использования (UI/UX). Финальное системное тестирование. Исправление критических и высокоприоритетных дефектов.

Продолжительность:

Непрерывное регрессионное, модульное и интеграционное тестирование в течение 8 недель Фазы 3.

Интенсивная фаза тестирования (Фаза 4) продолжительностью 4-6 недель.