**Задание 1.**

Получив задание подобного плана я бы приступил к его разбиению на блоки, руководствуясь принципом от большого к малому, от сложного к простому. Для игры Royal Match, я бы выделил следующие блоки:

**1.** **Area** (архив пройденных areas).

В данном блоке я бы проверял локализацию, соответствие текущего и пройденных этапов, соответствие макету, возможности зайти на пройденный этап, анимация появления объектов на пройденных этапах, возможность пролистать заблокированные и пройденные этапы, невозможность открыть заблокированные этапы, работоспособность кнопок top и bottom их появление и исчезновение.

**2.** **Leaderboard**

Данный блок я бы разбил на под блоки для удобства работы:

* **Friends** (все друзья игрока)

В данном под блоке я бы выделил возможность подключиться к профилю ФБ и после этого появление двух вкладок:

**Friends List** (все друзья из соцсети ФБ)

Проверить сортировку друзей по уровням, рамки и места соответствуют золоту, серебру и бронзе. Аватарки тянутся из соцсети ФБ. Эмблема группы каждого игрока актуальная. Сам игрок выделен цветом, возможность скролить список, возможность посетить профиль друга в игре и посмотреть его прогресс и достижения. Возможность удалить его из друзей итд

**Add Friends** (добавление друзей)

Проверить, что игра предлагает нам друзей которые находятся у нас в соцсети ФБ и они еще не играют в игру. Проверить возможность добавление друга из предложенного списка. Возможность найти и добавить друга по ID, возможность удалить друга из рекомендуемых итд.

* **Players** (рейтинг игроков)

Данный блок также разбиваем на под блоки:

**World** (мировой рейтинг)

**Belarus** (локальный рейтинг)

Здесь проверки будут аналогичны **Friends List.**

* **Teams** (команды и их рейтинг)

**World** (команды и их рейтинг мировой)

**Belarus** (рейтинг локальных команд)

Здесь проверки частично будут похожи на блоки с рейтингами и списками описанными выше и будут некоторые новые проверки. Как пример: возможность кликнуть на любую команду, посмотреть описание команды, ее достижения, активность и тип доступности команды. Проверки возможности добавиться в команду если лимит игроков превышен\не превышен по балансу. Проверка кейсов по добавлению игрока не достигшего уровня итд. (согласно документации)

**3.** **Home**

* **Profile** (все что связано с игровым профилем)

Проверки изменения аватарки, рамки, темы, изменение имени, проверка на соответствие urban dictionary, сохранение изменений.

* **Shop + Offers** (название говорит само за себя, покупки игровых ресурсов в магазине и офферах, он же счетчик монет. Монеты зарабатываем за прохождение уровня в match3 и тратим на прохождение уровня, если не хватило ходов для того чтобы завершить его).

Все проверки связанные с покупками (прерываниями, отложенными платежами) и системами оплаты. Старт тех или иных офферов по балансу, контент магазина и акций для сегментов, региональные цены, начисление ресурсов после покупки итд.

* **Lives** (счетчик жизней, необходимых для игры)

Трата жизней (после проигрыша) и их накопление. Максимальное количество согласно баланса, накопление по таймеру, невозможность играть если их ноль.

* **Stars** (счетчик звезд необходимых для улучшения объектов на area, зарабатываются после успешного прохождения уровня в match3)
* **Settings** (все настройки игры)

Отключение\включение звуков, музыки, хаптик эффекта, нотификации. Возможность прочтения пользовательского соглашения, возможность привязать игровой прогресс к внешним аккаунтам (FB, Google, AppleID).

* **Level** (вкладка перехода в лобби игры, в котором есть информация цели уровня, возможности выбора бустеров и кнопки «Play»)

Этот блок я бы разбил на большое количество мелких задач (под блоков) которые связаны именно с правильностью работы механики match3, соответствие балансу, анимация, соответствие макету, проверка шаблонов игровых полей, «жизни» клеток и наносимый им урон бустерами или просто удалением клеток, подсказки и туториалы, проверка бустеров на нижней панели (залоченность и разлоченность), отработки бустеров в связках и по отдельности. Работа кнопки настроек, зачисление targets и списание ходов (так же возможность их докупить), не списание ходов если не была сложена комбинация для удаления, проверка комбинаций плиток для удаления с появлением и не появлением бустеров итд.

Необходимо так же проверить кейсы с «супер игрой» (Королевские сны, где игра идет на время.

Это был бы один из самых больших блоков.

* **Area** (текущий уровень который улучшает игрок)

На этой вкладке нужно проверить отображение маркеров (указывают количество звезд которые можем потратить на покупку\улучшение объекта данного уровня Area). Прогресс бар - который указывает на количество улучшено\осталось. При клике на вкладку, появляется окно с списком объектов, которые можем купить за звезду, сверху справа количество доступных звезд. После покупки за звезду объекта, мы видим анимацию появления его в текущей area.

**4. Teams** (вкладка для добавления\удаления из команды игроков. Создание своей команды, передача роли, редактирование описания, лого итд. Чат команды для общения и помощи в запросах на жизни).  
 Проверки по добавлению\удалению игроков, ответ на запросы, отправка и получение сообщений, создание и удаление союза, отправка и получение запроса на вступление игрока в союз, проверка лимита игроков в союзе согласно баланса, просмотр профиля игроков (союзников), вступление и выход из союза итд

**5.**  **Collection** (еще не дошел, но могу предположить что это как альбом карточек наградных, возможно ивентовых. Возможно тут предусмотрен обмен между игроками команды)

Возможно где-то не указал, но везде нужно проверить соответствие балансу, локализации, макету. В отдельный блок вынес бы тестирование звуков, туториалов (как правило большой кусок работы), нотификаций, аналитики (в рамках некоторых задач можно проверить клиентскую аналитику). Блок проверок по установке, удалению и накатки билдов для iOS и android платформ.

**Задание 2.**

Для тестирования сценария орисанного в задании я бы использовал следующие техники тест-дизайна:

**1. Эквивалентное разбиение**.

Основная задача, это проверка работы игры на разных уровнях сложности (предположим у нас их три) необходимо выбрать любой из них (мы предполагаем что остальные будут работать так же.

**Тест-кейсы:**

Проверка корректного отображения доступных уровней для игры.

Проверка невозможности начать игру без выбора уровня.

Проверка начала игры на уровне сложности "Легкий" ("Средний" или "Сложный").

Проверка того, что выбранный уровень соответствует сложности по балансу.

Проверить игровой процесс выбранном уровне сложности.

Игрок соединяет 3 плитки на игровом поле.

Проверить, что плитки удаляются при сборе их в ряд (3 шт).

Проверка сохранения прогресса при переходе между уровнями сложности.

Проверка завершения игры на выбранном уровне.

**2. Граничные значения.**

Основная задача блока - проверка поведения игры при достижении критических значений (min-max) количества ходов, целевого количества очков, игровой валюты, таймера времени, бустеров итд.

**Тест-кейсы:**

Проверить, что значения жизней, количество очков, игровой валюты, ходов, бустеров не может быть отрицательное значение.

Проверка начала игры при нулевом значении жизней.

Проверка начала игры при max значении жизней.

Проверить, что количество жизней не может быть больше max значения.

Проверка завершения игры при исчерпании всех ходов.

Проверить, что плитки не удаляются если их 2.

Проверить, что плитки удаляются если их больше 3.

Проверить, что таймер накопления жизней не может вести отсчет с отрицательными значениями.

Проверить, что таймер накопления жизни дает жизнь игроку и не продолжает отсчет выше балансового значения.

Проверка завершения игры при достижении минимального целевого количества очков.

Проверить, что игра не завершается до достижения цели при невозможности ее выполнения.

Проверка завершения игры при достижении максимального целевого количества очков.

Проверить начисление и отображение при достижении max значения игровой валюты.

Проверить, что игра не завершается до исчерпания всех ходов, если цель уже достигнута.

Проверить возможность применять бустеры у которых значение равно ноль.

Проверить, что игра завершается при истечении времени игры (если этап игры завязан на время).

Проверить, что игрок может докупить дополнительные ходы, если у него хватает валюты.

Проверить невозможность докупить дополнительные ходы, если валюты не хватает

**3. Тестирование состояний.**

Основная задача - проверка корректности переходов между экранами и состояниями игры.

**Тест-кейсы:**

Проверить вход в приложение.

Проверка перехода от экрана выбора уровня к игровому полю.

Проверка перехода от игрового поля к экрану завершения игры.

Проверка перехода от экрана завершения игры к экрану выбора уровня.

Проверка возможности перехода к другим экранам во время игры.

Проверка возможности прерывания игры в любой момент и возврата к главному меню.

Проверка перехода по внутри игровой нотификации (если такая предусмотрена).

Переход из игрового окна в режим паузы.

Переход из игрового окна в окно инвентаря выбора бустеров.

Переход из игрового окна в окно покупки дополнительных ходов за игровую валюту.

Переход в окно магазина (форсирование окна магазина или оффера) при недостаточном количестве ресурсов при попытке начать игру.

Появление окон подсказок во время игры.

**4. Предугадывание ошибок.**

Основываясь на опыте предыдущих проектов, попытаться предугадать и составить кейсы, где могут быть потенциальные проблемы.

**Тест-кейсы:**

Быстрый клик при входе в приложение.

Быстрый клик или долгое нажатие на выбор уровня сложности.

Беспорядочные клики на поле игры.

Прокликивание анимации при работе бустеров и удалении плиток.

**Задание 3.**  
  
Если предположить, что такой баг-репорт оформил пользователь, то честь ему и хвала. Он выглядит достаточно полным. В нем есть многие атрибуты хорошего баг репорта (номер бага, описание проблемы, шаги воспроизведения, ожидаемый результат, версия игры, платформу итд).

В каждой компании так или иначе существуют свои правила заведения баг-репорта, его оформление, регистрация итд. и возможно где-то это был бы идеально заведенный баг. Но я бы мог добавил некоторые элементы, которые очень важны в любом баг-репорте и мой репорт имел бы вид:

**Номер бага:** RM001

**Название бага:** [Royal Match] Игрок получает ошибку при попытке пройти 50-й уровень

**Приоритет:** Критикал.

**Серьезность:** Критикал.

**На какой отдел и на кого заведен баг**: Сервер, исполнитель Иванов.

**Ссылка на тред обсуждения в Slack:** ……………………..

**Описание проблемы от игрока:** «при прохождении уровня 50 в игре "Royal Match", возникает ошибка и игра зависает, не позволяя продолжить игру. Проблема возникает каждый раз, когда я пытаюсь пройти этот уровень.»

**Шаги для воспроизведения:**

1. Игрок запускает игру "Royal Match".

2. Игрок заходит в свою учетную запись или создает новую.

3. Игрок доходит до уровня 50.

5. Игрок собирает фишки последнего хода (на поле не остается фишек).

**Актуальный результат:** Игрок получает ошибку при попытке пройти 50-й уровень.

**Дополнительная информация:**

Логи: (Если есть).

Видео или скрин (хотя тут шаги простые)

Время падения ошибки: (чтобы найти ее в серверных или клиентских ошибках)

ID игрока: 777888999

Версия игры: 2.1.0

Платформа: Android 10

Марка девайса: Samsung Galaxy A31

У меня достаточно свободного места на устройстве.

**Ожидаемый результат:** Игрок проходит 50-й уровень и переходит на 51-й.

Потраченное время примерно 6 часов.