# Постановка

## ссылки

* [Пошаговый игровой процесс](https://docs.google.com/spreadsheets/d/1v8xWFoNcqYjdHWRsi7xbdcn89yH-UzIkvb4ydTvcT88/edit?gid=0#gid=0)
* [API](https://gamemaster.ballrace.online/api/docs/v1)
* [Дизайн](https://www.figma.com/design/toz9X3E315YIwqoM0Xnoik/Untitled?node-id=88-7084&t=rSxqI5kZhr7cMipR-0) *(используем “космический” дизайн)*
* Посмотреть как работают аналогичные элементы в гонке шаров [Демо ballrace](https://ballrace.online/?code=ballrace-x4)

### Компоненты:

* frontend
* генератор случайных чисел (OpenSSL RAND\_bytes)
* компонент анимации гонки (pixi js ?) (можно и на чистом CSS)

### Игровой цикл:

1. До начала гонки отправляется запрос в генератор маршрутов.
2. Генератор отдает ответ с рассчитанными маршрутами движений и результатами гонки.
3. Во время гонки и до окончания таймера принимаются ставки **на следующую гонку**.
4. Гонка начинается, компонент анимации гонки (на клиенте) генерирует анимацию (воспроизводит маршрут каждого таракана) так, чтобы получить необходимые результаты.
5. Когда гонка начинается, **ставки на неё больше не принимаются**.
6. Когда гонка начинается, открывается окно ставок **на следующую гонку**.
7. Когда гонка заканчивается, регистрируются результаты, рассчитывается и начисляется выигрыш.
8. Цикл повторяется.

## User stories

User stories [link](https://docs.google.com/document/d/1VIMv4IlK9lKJ9qB3HUGoKMByoVozH7jazIyhm-KbPyo/edit?tab=t.0)

## Фронтенд

### функциональные кнопки

#### last game

|  | кнопка last game отображает всплывающее окно истории прошедших игр с местом и цветом таракана который пришёл, в случае если таракан не пришёл отображать пустоту  всплывающее окно с историей игр должно отображаться слева внизу если окно ставок закрыто или слева вверху если окно ставок открыто (согласно дизайна)  Релизовано в API платформы. **`GET: /api/v1/gameplay/games/sessions/{code}/last`** |
| --- | --- |

#### make a bet

|  | кнопка make a bet предлагает на выбор пользователю четыре кнопки с помощью которых можно поставить ставки 25 50 100 150 в зависимости от валюты номинала, номинал может меняться  (параметры этих кнопок (как и сами кнопки) реализованы через API платформы и приходят с валютами при запросе данных по игровому инстансу **`GET: /api/v1//gameplay/games/instances/{code}`**  (по тапу на уже сделанную ставку ставка прибавляется к выбранной в нижнем блоке, то есть например если поставлено 25, то по тапу прибавляется еще 25) необходимо учитывать акцент а акцент интерфейса (на клиентской логике)  кнопка отмены под блоком быстрых ставок отменяет все ставки для всех типов ставок (на клиентской логике)  кнопка x2 умножает все поставленные ставки на x2  (на клиентской логике)  калькулятор с цифрами можно использовать для кастомной ставки с шагом в 1  (на клиентской логике)  кнопка возврата отменяет только что сделанную ставку то есть необходимо предусмотреть механизм последовательной отмены ставок  (на клиентской логике) (если vue, то удобнее всего через [useRefHistory](https://vueuse.org/core/useRefHistory/)  кнопка bet закрепляет ставки после установки, после нажатия на кнопку bet происходит списание средств и ставки изменить на игру невозможно  (реализовано в API платформы, **`POST: /api/v1/gameplay/games/bets/{code}`**) |
| --- | --- |

#### history bets

|  | кнопка history bets отображает историю ставок переключателя не будет, самом верху будут актуальные ставки далее история за последние 10 игр  (реализовано через API платформы, **`GET: /api/v1/gameplay/games/bets/{code}/latest`**  + обновление данных через WS (тоже есть в доке по API)  всплывающее окно с history bets должно отображаться справа внизу если окно ставок закрыто или справа вверху если окно ставок открыто (согласно дизайна) |
| --- | --- |

### Ставки

### Общая информация

Игрок делает ставку на исход следующей гонки.

Игрок может делать мультиставки в рамках одного типа ставок или кросс-ставки

В момент отправки ставки (нажатие кнопки BET) сумма всех сделанных ставок списывается со счёта. (API платформы, смотри выше)

Проигравшие ставки остаются в казино (не возвращаются). (под капотом платформы)

Выигравшие ставки рассчитываются по формуле [выигрыш] = [размер сделанной ставки на событие] Х [коэффициент к ставке на этой событие] и переводятся на счёт игрока (сумма по всем выигравшим ставкам). (под капотом платформы)

Коэффициенты статичные. (приходят в API при запросе данных по игровому инстансу)

Например:

* игрок ставит **$1** на то, что в следующей гонке таракан **№3** придёт **под №3,** коэффициент на это событие **6.3**
* ставка выигрывает
* игроку начисляется выигрыш в размере **$1 X 6.3 = $6.3**

|  | ТИП 1 Ставка на победителя, какой таракан придет первым в забеге |
| --- | --- |
|  | ТИП 2 Ставка на то кто займет первое и второе место  по вертикали первое место по горизонтали второе место  выбор пары победителей 1е и 2е место  пример: вертикаль оранжевый то есть если мы ставим на оранжевого слева на то что он займет первое место, а второе место займет жёлтый, это одна ставка |
|  | ТИП 3 Ставка какой таракан в какую ловушку придет  по нажатию и открывается дополнительное окно ставок, здесь можно поставить какой конкретно таракан придёт в какую ловушку, для каждой ловушки можно выбрать любое количество тараканов |

Все ставки (тип, математика) — генерируются на бекенде (платформе), настраиваются через платформу и отдаются в API на фронтенд. Клиент только их рендерит в нужных местах и отправляет на сервак. Дальнейший обсчет — также на бекенде.

### верстка

|  | блок header в нём отображается баланс и кнопка перехода в личный кабинет провайдера  (запрос игрового аккаунта реализован в API платформы **GET: /api/v1/gameplay/games/accounts/{code}/{session}** |
| --- | --- |
| полоса прогресса игры она же отражает окончание текущей игры  (это рендерится и обсчитывается на клиенте, т.к. все данные по маршрутам и длительности игры приходят в начале раунда в настройках игровой сессии в API) |
| фон содержит в себе лабиринт на первом этапе он будет статичен (рассматривается возможность его динамической генерации при каждой сессии игры)  Для реализации лабринта нужно  — разметить все на координатной сетке и составить план лабиринта  — эту координатную сетку (даже при статичном лабиринте в виде картинке) нужно хранить в настройках платформы, чтобы передавать в генератор и корректно обсчитывать маршруты  — на фронт смысла присылать эту координатную сетку не вижу (пока что) |
| финишная черта и ячейки ловушки в которые прибегают тараканы, в каждую ячейку могут прибежать как все тараканы так и ни одного  (должно быть частью координатной сетки + по особому размечено, чтобы понимать, что это финиш) |
| Функциональные кнопки |

## Бэкенд

путь движения тараканов генерируется заранее перед каждой сессией игры с помощью микросервиса с математической моделью расчета маршрута по координатной сетке и передается в бэкенд, после бэк передаёт путь на фронт в виде готового трека для каждого таракана, тараканы на экране должны двигаться согласно трекам

текущий концепт: использование тиков которые будут привязаны ко времени и движению тараканов например рандомно генерируется для каждого тика - шаг таракана, влево вправо вверх вниз, полный путь записывается для каждого таракана и передается на фронтенд, на фронтенде нужно проиграть путь каждого таракана как плеер

тараканы бегут сверху вниз начинают одновременно

Должна быть предусмотрена смена языка и валюты

*(смена языка — локально на клиенте, через query lang=ru/en/…)*

*(список доступных валют (и их настройки для калькулятора ставок) приходит в API)*

Должна быть предусмотрена смена темы с помощью параметров

*(нужен список параметров, которые будут добавлены в админку у игрового инстанса и будут переданы на фронт через API в настройках игрового инстанса — это все уже есть в платформе, нужны только сами параметры, чтобы их добавить в админке)*

Бекенд. Платформа. Тех.реализация:

*Общение через простой POST HTTP запрос. Платформа запрашивает маршруты и получает в ответ готовые треки*

Входные параметры:

* длительность игровой сессии
* количество ходов (с помощью этих двух параметров (которые будут задаваться в админке) сможем регулировать скорость перемещения тараканов
* количество тараканов (игроков) (чтобы не хардкодить в генераторе) (подумать как передавать, просто числом (например, 7) или в виде объекта [{value: ‘RED’}]
* координатная сетка лабиринта с разметкой (например, 0 - пустое место, 1 - препятствие, стена, f1 - финишная черта первой дорожки, f2 - финишная черта второй дорожки и т.д)

Выходные параметры:

* маршруты тараканов (игроков) в виде: {таракан: [[ходN: [x, y]]], …}
* результаты тараканов, кто каким пришел (место), за сколько ходов, пришел / помер по дороге, в какую дорожку в итоге пришел

Платформа получает эти данные по маршрутам и результам.

Сохраняет в игровую сессию.

Рассчитывает время начала и окончания сессии, до какого момента доступны ставки

Отправляем публичные данные на фронт (результаты - не публичные)

Как-то (???) ждем окончания сессии и запускаем все стандартные события, что сессии закончилась, вот результаты, вот обсчет и процессинг ставок

## Вопросы:

**не хватает дизайна блока overtaking**

**переделать дизайн history of bets**

**нет окна win**

**обсудить ставку Overtaking**

**Ставка на обгон - первенство одного таракана к финишу, по отношению к другому таракану.**

**Ставка на секцию:** для каждой ловушки можно выбрать любое количество тараканов или для каждой секции только 1 таракан?

**описать поведение для ставок в зависимости от того, пришёл ли таракан к финишу. т.е. если я поставил на таркана, который не пришёл к финишу, что с моей ставкой:**

ставка на секцию, таракан не пришел к финишу: ставка не сыграла

ставка на место на пьедестале, таракан не пришел к финишу: ставка не сыграла

ставка overtaking “зеленый раньше красного”, зеленый таракан не пришел к финишу: ставка не сыграла

ставка overtaking “зеленый раньше красного”, **красный** таракан не пришел к финишу: ставка **сыграла**

**предложение: начинается гонка, когда прогресс гонки заканчивается и какие-то тараканы не пришли к финишу, не пришедшие к финишу тараканы взрываются,**

**происходит подведение итогов, начинается следующая гонка**

**Бекенд. Генератор маршрутов:**

1. тараканы стартуют в одной координате или в разных?
2. если тараканы в разных, то координата рандомна (в рамках какой-то области) или всегда одна и та же для каждого таракана?