

`docs/tonrody-core-concept.md`

TONRODY — Концепция системы и базовые механики

1. Общий обзор

TONRODY — это система честных азартных взаимодействий между пользователями сети TON, в которой:

- * платформа не играет против пользователей;
- * платформа не забирает комиссию с ставок и выигрышей;
- * единственный обязательный расход пользователей — газ сети TON;
- * все игровые средства двигаются **только между пользователями**;
- * доход проекта формируется исключительно за счёт **добровольных донатов** и **платной кастомизации** (режим Donateroom, приватные эффекты, оформление).

Система строится вокруг трёх основных компонентов:

1. **Игровые режимы** (Coin Flip, публичные лобби, Donateroom);
2. **Экономика проекта** (treasury, токен, байбеки);
3. **Модель доступа и статусов** (TONethernet / Pyramidtype как уровни участия и влияния, но не источник дохода).

2. Базовые принципы честности

1. Платформа **никогда** не участвует в играх как отдельный игрок.
2. Платформа **не получает процент** с игровых пулов (Coin Flip, лобби).
3. Контракт:

- * не содержит логики, позволяющей отправить игровой пул на системный/админский адрес;
- * выступает нейтральным арбитром: принимает депозиты, выбирает победителя, выплачивает пул.

4. Любые средства, поступающие проекту, возникают только:

- * из добровольных донатов,
 - * из оплаты кастомизации/визуальных эффектов в специальных режимах (Donateroom).
5. Все правила и условия (ставки, количество игроков, тип лобби, донат-составляющая, условия разблокировки токена) зафиксированы в коде и публично доступны.

3. Игровые режимы

3.1. Coin Flip (режим 1v1)

****Описание:****

Честная дуэль 1 на 1, где два игрока делают равную ставку, система честно определяет победителя, и весь пул получает один из игроков.

****Процесс:****

1. Игрок А:

- * создаёт Coin Flip-лобби;
- * вносит ставку (фиксированная или в рамках допустимого диапазона);
- * выбирает сторону (орёл/решка).

2. Контракт фиксирует:

- * сумму ставки;
- * выбранную сторону игрока А;
- * статус лобби: «ожидает второго игрока»;
- * дедлайн (опционально).

3. Игрок В:

- * заходит в это лобби;
- * вносит **такую же сумму**;
- * автоматически получает противоположную сторону.

4. Контракт:

- * принимает обе ставки;
- * фиксирует их как единый раунд;
- * вызывает честный механизм случайности (VRF / on-chain entropy / commit-reveal);
- * определяет победителя;
- * отправляет **весь пул** (ставка А + ставка В) победителю.

****Особенности:****

- * Платформа не получает никакой части пула.
- * В контракте отсутствуют функции, позволяющие вывести игровые средства на адрес платформы.
- * Единственные комиссионные расходы для пользователей — газ сети TON.

3.2. Публичные лобби (N игроков)

****Описание:****

Режим, в котором N игроков вносят одинаковую ставку, и один из них получает весь пул.

****Процесс:****

1. Создаётся лобби с параметрами:

- * количество слотов `N` ,
- * размер ставки для одного слота.

2. Игроки по очереди занимают слоты, внося одинаковую сумму TON.

3. Когда все N слотов заполнены:

- * лобби переходит в статус «готово к розыгрышу»;
- * контракт запускает механизм случайного выбора победителя.

4. Контракт:

- * определяет 1 победителя;
- * отправляет ему **весь пул** (`N * ставка`).

****Особенности:****

* Платформа не забирает комиссию с пула.

* Никакие средства из игрового пула не могут быть направлены на системные адреса.

* Контракт реализует только:

- * хранение состояния лобби;
- * приём ставок;
- * выбор победителя;
- * выплату выигрыша.

3.3. Donateroom и приватные лобби

****Описание:****

Расширенный режим лобби (как публичные лобби или Coin Flip), в котором:

- * игровая часть (ставки, выбор победителя, выплата) остаётся честной и идентичной базовым режимам;
- * добавляется **донат-составляющая** и расширенная **кастомизация визуальных эффектов**.

3.3.1. Донат-составляющая

Дополнительно к игровой ставке может быть назначена **donationAmount** (фиксированная или настраиваемая величина).

- * Игровая ставка участвует в формировании пула и разыгрывается между игроками.
- * `donationAmount`:

- * не входит в игровой пул;
- * направляется на адрес **treasury** или специального buyback-кошелька проекта.

Таким образом:

- * игровой режим остаётся честным и без комиссий;
- * добровольные донаты становятся единственным источником дохода проекта.

3.3.2. Визуальные и звуковые эффекты

Фронтенд дополнительно реализует:

- * особые визуальные эффекты:
 - * анимации поражения (например, «снаряд-базука», прилетающий в аватар проигравшего и оставляющий след);
 - * эффекты вокруг аватаров (огоньки, ореолы, шлейфы, glow);
 - * кастомные рамки, карточки игроков и т. п.;
- * звуковые эффекты (свист, взрыв, окружение и т.д.);
- * особый UX:
 - * приватный доступ (по ссылке/инвайту);
 - * автозакрытие лобби через 10 минут после выхода второго участника после раунда (настроочный параметр).

3.3.3. Разделение обязанностей: контракт / фронтенд

Контракт:

- * хранит:
 - * тип лобби: `public | private | donate`;
 - * сумму ставки;
 - * величину `donationAmount`;
 - * статус лобби и список участников;
- * обеспечивает:
 - * приём ставок;
 - * выбор победителя;
 - * выплату **игрового** пула победителю;
 - * перевод `donationAmount` на адрес treasury/buyback;
- * генерирует события (events), например:
 - * `LobbyCreated` ,
 - * `LobbyFilled` ,
 - * `LobbyResolved(lobbyId, winner, isDonate, donationAmount)` .

Фронтенд:

- * отслеживает состояние лобби через бэкенд/прослушивание событий контракта;
- * отрисовывает все визуальные эффекты и анимации;
- * управляет таймерами, закрытием лобби, звуками;
- * реализует приватный доступ и UX Donateroom.

4. Stakes, TONethernet и Pyramidtype

С учётом отказа от комиссий платформы и перераспределения доходов между пользователями, **классический стейкинг (как “вклад под процент”) в системе TONRODY отсутствует.**

4.1. Смысль Stakes

Stakes в данной архитектуре:

- * не являются источником финансовой доходности;
- * не генерируют пассивный доход;
- * не связаны с перераспределением TON между участниками.

Stakes — это:

- * показатель **позиции пользователя в системе**;
- * параметр статуса, доступа и веса в экосистеме;
- * условие допуска к определённым режимам, ивентам, кастомизациям.

4.2. TONethernet (сетевая модель доступа)

TONethernet — сетевая модель, в которой:

- * каждый игрок рассматривается как узел;
- * связи между игроками (совместные лобби, рефералы, история взаимодействий) формируют сетевую структуру;
- * важен:
 - * объём участия,
 - * количество и качество связей,
 - * активность в экосистеме.

Влияние TONethernet:

- * определяет:
 - * доступ к отдельным игровым режимам и лобби;
 - * приоритет/скорость доступа;
 - * “вес” профиля в различных внутренних механиках;

* не влияет напрямую на получение TON:

- * никаких “процентов от чужих ставок”;
- * никаких выплат за счёт других пользователей.

Это **модель доступа и статуса**, а не модель прибыли.

4.3. Pyramidtype (иерархия статусов, а не денег)

Pyramidtype рассматривается как:

- * иерархическая модель уровней статуса (новичок → участник → резонатор → узел → проводник и т.д.);
- * система, описывающая:

- * структуру связей,
- * глубину влияния,
- * вклад в экосистему.

Pyramidtype:

- * не распределяет TON по уровням;
- * не даёт пользователю “процент” от депозитов нижестоящих;
- * используется для:
 - * выдачи косметических элементов (рамки, анимации, бейджи),
 - * доступа к закрытым режимам/ивентам,
 - * учёта метрик профиля.

Таким образом, Pyramidtype — это **иерархия прав и доступа**, а не финансовая пирамида.

5. Экономика проекта: Treasury, токен и байбеки

5.1. Источники дохода

Единственный источник дохода проекта:

- * добровольные донаты и доплаты пользователей в Donateroom/приватных лобби;
- * оплата кастомных визуальных/звуковых эффектов.

Игровые режимы (Coin Flip, публичные лобби):

- * не содержат комиссии;
- * не генерируют доход платформе.

5.2. Treasury

Treasury (казначейский кошелёк/контракт):

* получает:

- * все `donationAmount` из Donateroom;
- * иные добровольные поддержки/донаты.

* использует TON:

- * для байбеков токена проекта на DEX;
- * для добавления ликвидности в пул;
- * для будущих активностей и внутренних механик (например, ивенты, награды).

5.3. Токен проекта (условно TNRD)

Роль токена:

- * служит знаком участия и резонанса в экосистеме TONRODY;
- * выдаётся за активность:

- * участие в играх,
- * донаты,
- * рекуррентную активность, ивенты, кампании;
- * не выдаётся как “процент на депозит” или “гарантированный доход”.

6. Модель блокировки и разблокировки токенов (Locker + оракул)

Токен проекта не становится сразу ликвидным на кошельках пользователей. Для обеспечения честности и долгосрочности используется связка:

1. **Токен (Jetton/стандарт)**
2. **Locker-контракт (замок)**
3. **Оракул цены (price oracle)**

6.1. Locker-контракт

Locker:

* хранит заложенные токены и учётные записи:

- * `locked[user]`
- * `unlocked[user];`

* получает токены от treasury или минтера токена;

* при начислении награды:

- * вызывается функция вида `reward(user, amount)`;
- * токены переводятся в Locker;
- * значение `locked[user]` увеличивается на `amount`;

* пока токены в состоянии `locked`:

- * пользователь не может их перевести/продать;
- * баланс существует как запись в контракте.

После выполнения условий разблокировки:

- * `locked[user]` частично или полностью переносится в `unlocked[user]`;
- * пользователь может вызвать `claim()` или аналогичную функцию;
- * Locker отправляет реальный баланс токена на адрес пользователя.

6.2. Оракул цены

Оракул:

* хранит текущую цену токена:

- * в TON,
- * или в эквиваленте \$ через стейблкоин;
- * обновляется:

- * либо бэкендом проекта (off-chain → on-chain),
- * либо сторонней оракул-сетью (если интегрирована).

Locker использует данные оракула для проверки условий разблокировки.

6.3. Условие “разблокировка при цене \$1”

Пример глобального условия:

* при достижении ценой токена значения `>= \$1` и удержании этого уровня в течение заданного периода (N блоков / минут / часов):

- * в Locker выставляется флаг `isGlobalUnlock = true`.

После этого:

* либо:

- * контракт автоматически переносит все `locked[user]` → `unlocked[user]`;
- * либо:

* каждый пользователь сам вызывает `unlockMyTokens()` / `claim()`, и контракт по запросу:

- * проверяет `isGlobalUnlock == true`;
- * переводит его токены в разблокированное состояние и отправляет их на его адрес.

Все параметры (порог цены, источник цены, период удержания) фиксируются при деплое контракта.

Изменение этих параметров возможно только через прозрачные механизмы (например, мультиsig, timelock), либо не возможно вовсе.

7. Роли в системе

* **Игроки**

Единственные субъекты, которые:

- * вносят депозиты в игровые режимы;
- * выигрывают игровые пулы.

* **Платформа**

Обеспечивает:

- * интерфейс и UX;
- * бэкенд-сервисы;
- * связь фронтенда с контрактами.

Не участвует в распределении игровых средств.

* **Контракты игровых режимов**

Отвечают за:

- * приём ставок;
- * хранение состояния лобби/дуэлей;
- * честный выбор победителя;
- * выплату игровых пулов.

* **Treasury**

Принимает:

- * донаты и `donationAmount` из Donateroom;

Использует:

- * TON для байбеков токена и других целей развития экосистемы.

* **Locker**

Управляет:

- * залоченными и разблокированными балансами токена;

* процессом разблокировки по условиям.

* **Оракул***

Поставляет:

* актуальную цену токена для проверки условий разблокировки.

* отдельно описать **сущности контрактов** (типы, поля, структуры сообщений);

* сделать `TNRD-01: Game Core Spec`, `TNRD-02: Token & Locker Spec`, `TNRD-03: Donateroom UX & Events` .