

CRNL: Commit-reveal numeric lottery

Астахов Сергей



Agenda

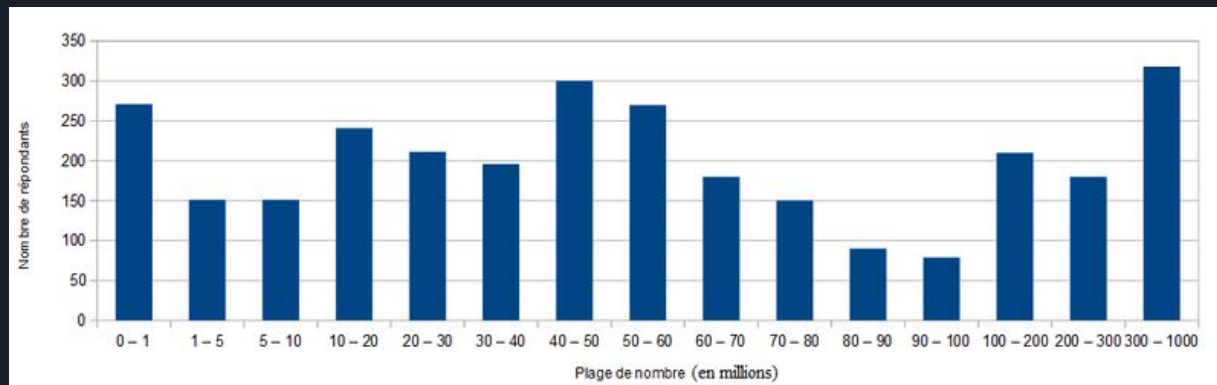
- Введение
- Стадии работы контракта
 - commit
 - reveal
 - rewarding
 - destruction
- Реализация
 - interface & methods
 - data structures
 - modifiers & events
 - tests
- Решенные проблемы безопасности и экономики
- Дополнительные опции
- Возможные варианты наследуемых контрактов



Введение

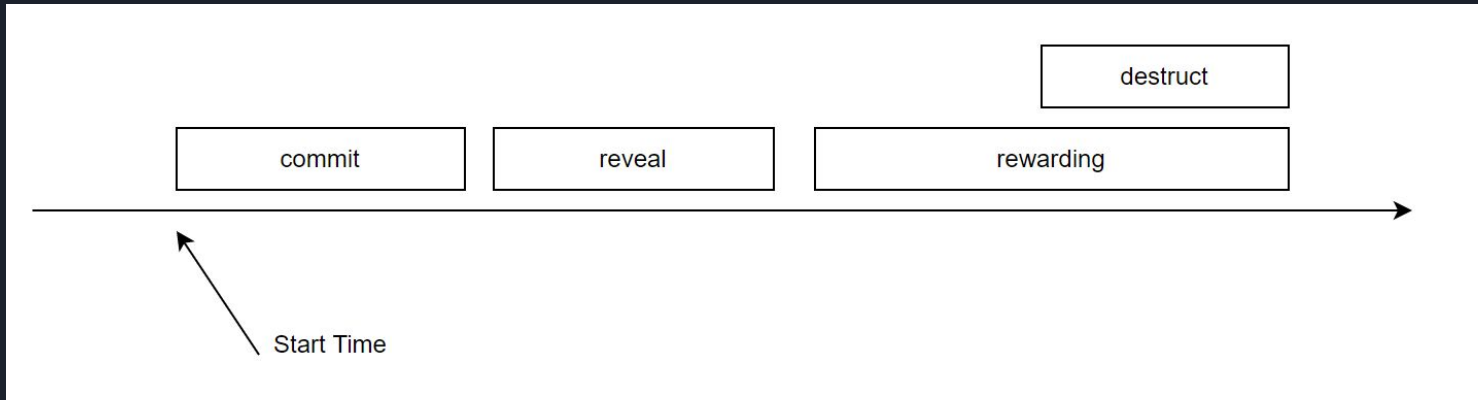
В 2005 году датская газета Politiken предложила своим читателям сыграть в следующую игру: каждый желающий мог прислать в редакцию действительное число от 0 до 100. Тот, чье число оказалось бы ближе всего к $2/3$ от среднего арифметического присланных чисел, выигрывал 5000 датских крон (на тот момент около \$800).

Данная игра известна в теории игр под названием «угадать $2/3$ среднего». Она демонстрирует разницу между абсолютно рациональным поведением и реальными действиями игроков.



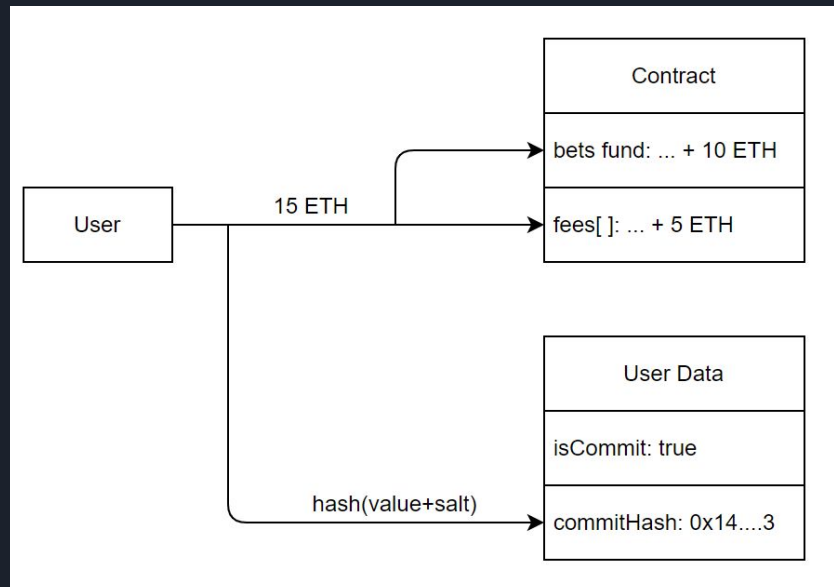
Распределение 2898 ответов на решающий момент конкурса Jeux et Stratégie 1983 года. Википедия

Временные периоды



Commit

- Участник переводит ставку, honor fee (“гарантию честности”) и комиссию на адрес контракта
- Участник присылает $\text{hash}(\text{value}, \text{salt})$
- Участник может поменять хеш, не внося дополнительных средств

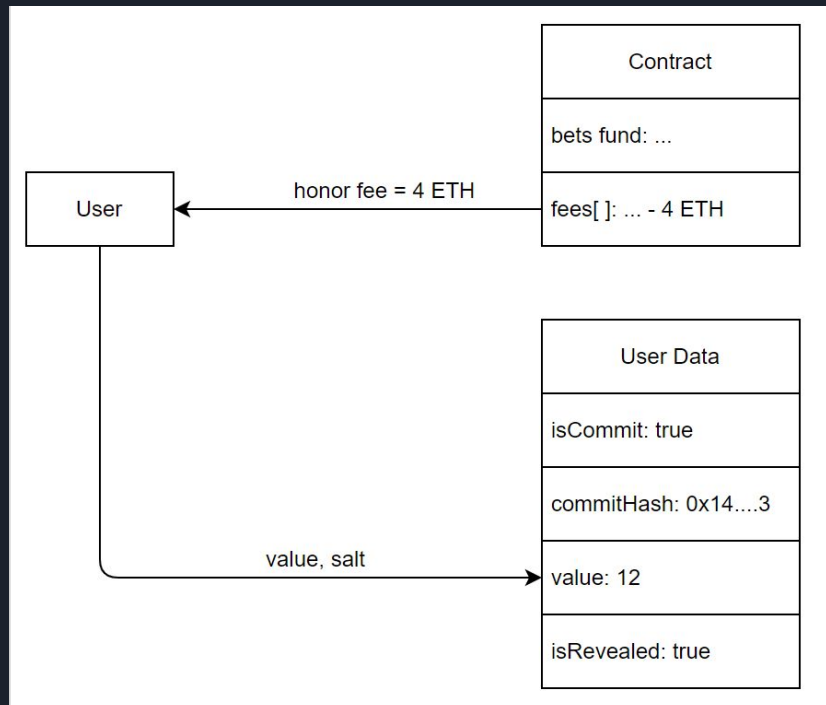


Reveal

- участник присылает value, salt
- если

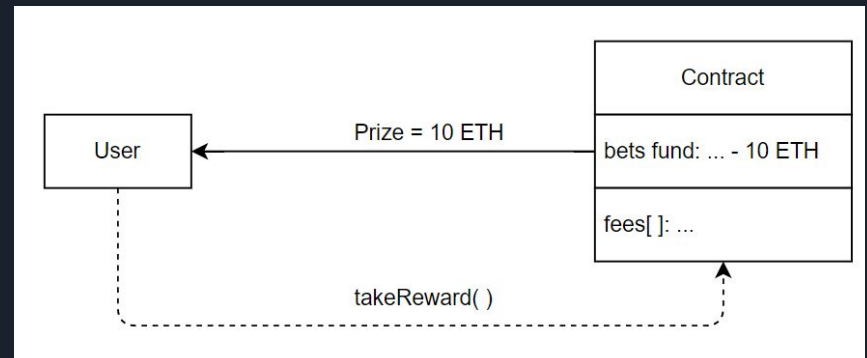
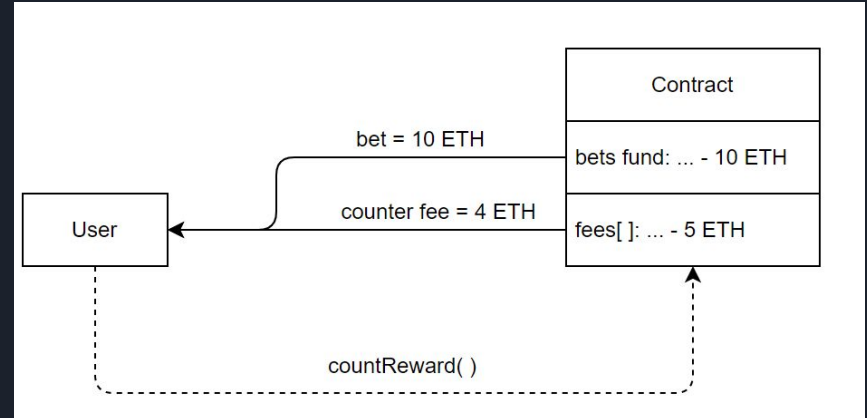
$\text{commitHash} == \text{hash}(\text{value}, \text{salt})$

то участник получает назад
honorFee, value участвует в
формировании среднего



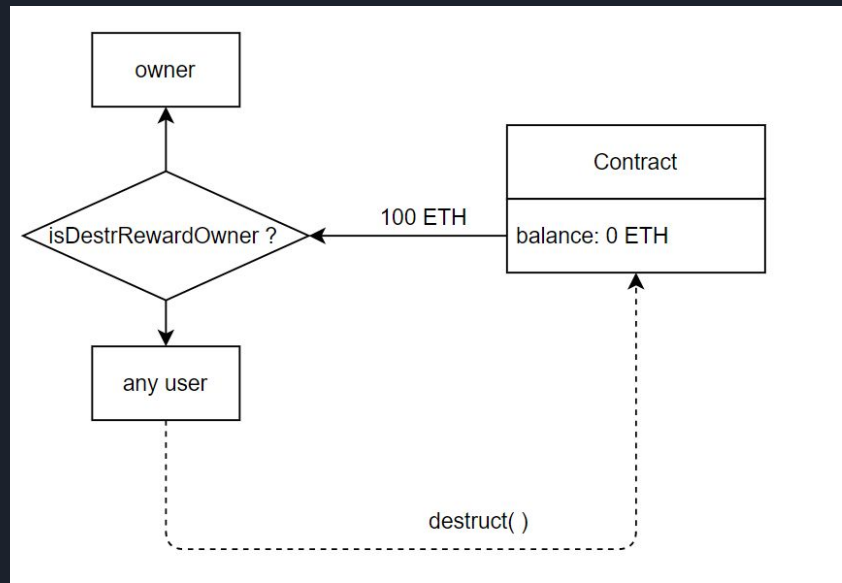
Rewarding

- Один из участников (обязательно revealed) вызывает `countReward()`, находит $\min(|value_i - avg|)$ и количество победителей, за что получает назад свою ставку + комиссии
- Далее участники могут вызывать `takeReward()` и получать награду, если их $|value_i - avg| == \min$



Destruction

- все еще можно вывести ETH
- любой пользователь сети ethereum может вызвать уничтожение контракта, остаток получает sender или owner (зависит от настроек)





Interface

```
interface ICRNL{  
    function commit(bytes32 commitHash_) external payable;  
    function changeCommitHash(bytes32 commitHash_) external;  
    function reveal(uint128 revealNum_, uint128 salt_) external;  
    function countRewards() external;  
    function takeReward() external;  
  
    function isFreePlaces() external view returns(bool isFreePlaces_);  
    function getPhaseId() external view returns(uint8 phaseId_);  
    function getWinnerStake() external view returns(uint256 winnerStake_);  
    function getAvg() external view returns(uint256 avg_);  
}
```



Data strucutres

```
struct UserData {  
    uint256 id;  
    uint256 commitHash;  
    bool isCommitted;  
    bool isRevealed;  
    bool isTookReward;  
}
```

```
mapping (uint256 => uint128) private _reveals; // mapping (id => value)  
mapping (address => UserData) private _users;
```



Modifiers & events

Модификаторы контроля времени

```
modifier commitPhase()
```

```
modifier revealPhase()
```

```
modifier rewardPhase()
```

```
modifier selfDestructPhase()
```

Событие уничтожения контракта

```
event DestructEvent()
```



Tests

- работоспособность “хорошего” сценария и fault tolerance + проверка балансов
- уничтожение контракта + проверка балансов в 2 режимах
- revert неправильного числа на стадии reveal
- revert commit(...) при недостаточном балансе
- смена владельца
- revert commit(...) при превышении числа участников
- revert повторного вызова одной и той же функции одним участником
- проверка view-функций
- revert некоторых view-функций, когда они не могут быть подсчитаны
- проверка модификаторов времени
- работоспособность при 1 участнике
- проверка работы наследуемого контракта
- проверка генерации DestructEvent



Проблемы безопасности и экономики

| <i>Проблема</i> | <i>Решение</i> |
|--|--|
| Превышение gas limit | Ограничение числа участников |
| Множественное получение награды или комиссий | Флаги и require |
| Ошибки при вызове countReward(...) при 0 reveal'ов | countReward(...) может вызывать только revealed пользователь |

Экономическая задача: найти оптимальное соотношение числа участников, размера ставки и комиссий

Возможное решение: подсчитать gas, затраченный для 1 итерации countReward() и умножить на цену(указать в конструкторе):

$\text{counterFee} = \text{gasPrice} * \text{countGasUsage}$



Дополнительные опции


- вывод owner fee возможен в любой момент
- при уничтожении контракта, остаток получает sender или owner (зависит от настроек)
- возможность смены owner'a
- наличие helper.py - программы для расчета commitHash(...)



Варианты наследуемых контрактов

- TrueAvgCRNL - контракт на основе среднего значения values
- MedianCRNL - контракт на основе $\frac{2}{3}$ медианы values
- ModCRNL - контракт на основе $\sum(\text{values}) \% p$
- Contract factory для CRNL

etc.

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with subtle diagonal lines.

Спасибо за
внимание