JACK
ICO & Game

Be the first or the last

EtherJack.io is the first fully smart contract based jackpot game. The core game process is safe and secure, running completely on-chain, operated by contracts with disclosed source code, with only optional extensions basing off-chain.

Abstract

EtherJack.io is the first fully smart contract based jackpot game. The core game process is safe and secure, running completely on-chain, operated by contracts with disclosed source code, with only optional extensions basing off-chain.

In EtherJack, the whole game process is controlled by participants, in this sense it's a massively multiplayer community game.

The rules are simple: bids fill up the jackpot, the last to bid wins and gets the huge prize.

The developers (or anyone else) can't affect the game process, can't manipulate or adjust it, or change the rules on the fly. The fixed contract code ensures the game is transparent and fair to all participants.

Smart Contracts in Online Gambling

The advantages of Blockchain systems are apparent when it comes to entrusting currency to some medium. The medium being a smart contract on a blockchain, for which the source code is disclosed, and which allows no backdoors for developers, completely solves the trust issues associated with this kind of games.

Such architecture eliminates the possibilities for meddling and cheating, which the game organizer might employ (like rigged RNGs), or if there are any, makes their presence apparent.

Today, there already exists a number of games and platforms working within blockchains, powered by smart contracts. However, because of certain technical limitations (like the time it takes to have a transaction fully processed), most types of games are effectively incompatible with the architecture: nobody is ready to wait up to a minute before the next action.

This has lead to a situation where on-chain gambling is something everyone is talking about, but actually most people keep playing old-fashioned off-chain games, being susceptible to all the same dirty tricks like rigged RNGs and more. The problem is likely to be resolved in the future when blockchains become faster and most transactions are included in blocks and confirmed within a few seconds, but not today.

EtherJack offers a game, for which quick transaction processing is not critical, and which is latency tolerant. The game is designed in such a way that the presence of the latency does neither break the game nor make it less interactive or engaging.

Thus, EtherJack enjoys all the advantages of an on-chain game (100% fair and transparent, with clearly defined and unchangeable rules) and doesn't suffer from the downsides associated with slow processing speeds.

One Time Offer and One Time Chance

Unlike many ICOs, the EtherJack ICO is not offering a share in a business, which isn't established yet, and does not in fact make any promises that can't be verified.

The game is completed, and is running. Participating in the ICO is the first stage, where the jackpot is accumulated, and a portion of game tokens is distributed. The tokens sold (JACK Tokens) are a part of the game, and their role is defined in the game's smart contracts.

Once the ICO is complete, the Jackpot contract will (with a 2-day delay) switch to the game stage, and it will be the time for JACK holders to be taking advantage of their investment.

The Game

The basic rules are quite simple. There's a jackpot, at every moment it's claimed by a player (the last to bid), and there's a countdown timer running. If someone bids (by sending an amount of Ether to the Jackpot smart contract), that person now claims the jackpot, the bid is added to the jackpot, and the timer is reset.

If nobody claims the jackpot by the moment the timer runs out, it goes to the last person who placed a bid. Over time, the bid price goes up (from 0.005 ETH, up to over 1 ETH), and timer duration goes down (from 3 hours down to 5 minutes).

There's a second level meta-game present: each 20th, 50th, and 100th bid also win "minor prizes": 10% of bids' worth during the corresponding period. For example, each 20th bid wins 10% of Ether used for last 20 bids. If all of these bids were made in Ether, and the bid price didn't change, it will be precisely double the bid price. If some of the bids were made in JACK, the amount will be smaller. If the bid price changed, the amount will depend on when within that interval the price changed.

Everything described above is entirely implemented in smart contracts with published source code, which means it can be verified and audited by the public. All smart contracts' variables are made public for the same purpose.

If a player bids double, triple, etc. amount of bid price, they buy JACK tokens for the extra amount. Tokens can be used to place bids without spending Ether later, for investment, or to be sold later at higher price. See "JACK Token Economy" below for details.

Bid Price

When the game is running, the bid price changes over time. The bid price depends on the total amount of bids made, and is defined by the following formula:

$$E_{bid}(N) = 10 \cdot 10^{15} + \left\lfloor \frac{N}{100} \right\rfloor \cdot 6 \cdot 10^{15}$$

Bid amount formula (in wei).

```
uint256 newBetAmount = 10 finney + (totalBets / 100) * 6 finney;
```

Bid amount rule. Solidity code present in the Jackpot contract.

Since during the game JACK is purchased or sold at this price, and 1 JACK can replace this amount of Ether in a bid, this price naturally corresponds to the JACK intrinsic value during the game stage.

Price at selected points is listed in the table below.

Bids made	Bid price
0	0.01 ETH (10 finney)
100	0.016 ETH (16 finney)
1,000	0.07 ETH (70 finney)
5,000	0.31 ETH
10,000	0.61 ETH
50,000	3.01 ETH
100,000	6.01 ETH

Table. Bid price during the game stage

Bidding an amount of Ether exceeding the bid price at least twice returns JACK token or tokens to the player (provided there's supply of JACK available for sale). See “JACK Token Economy” for details.

Timeout Duration

The timeout duration after each bid changes over time and depends on the total amount of bids made. It is defined by the following formula:

$$T(N) = 5 + \left\lfloor \frac{19200}{100 + N} \right\rfloor \quad minutes$$

Timeout duration formula.

```
(5 + 19200 / (100 + totalBets)) * 1 minutes
```

Timeout duration rule. Solidity code present in the Jackpot contract.

The duration at selected points is listed in the table below.

Bids made	Duration
1	195 min.
100	101 min.
1,000	22 min.
5,000	8 min.
10,000	6 min.
19,100	6 min.
19,101 +	5 min.

Table. Timeout duration

Bidding an amount of Ether exceeding the bid price at least twice returns JACK token or tokens to the player (provided there's supply of JACK available for sale). See "JACK Token Economy" for details.

Player Messages

Every participant can post messages on the game's website, for self-advertising, or for fun.

Posting a message is accomplished in two steps:

1. Entering the message on the website.
2. Confirming the message with a bid: the system will ask to place a bid for a specific amount (e.g. 0.010012345 ETH when the bid price is 0.01 ETH) and will activate the message when such bid appears on the blockchain.

The confirmation stage also associates the message with a specific Ethereum address (from where the bid was made) and its total sum of of invested Ether.

The more Ether the poster has invested into the game (across both the ICO and game stage), the higher their message will be ranked in the Top Messages section. The "investment weight" is counted just by amount of Ether sent into the game, not by amount of JACK held or stored in the Vault.

The Last Messages section simply lists the latest messages in the order or their activation.

A player can edit a previously published message by posting a new one and confirming it from the same Ethereum account or wallet. The message will retain all weight it has from the invested Ether. Messages can be edited at any time and at any rate.

Technical Implementation

The game has four primary components:

1. JACK Token smart contract
   deployed at address 0xe64d5f33E28387855d33980719f27f0d579392aF
   source code for audit purposes:
   https://etherscan.io/address/0x861825daa0068136a55f6effb3f4a0b9aa17f51f#code
2. Jackpot smart contract
   deployed at address 0xe64d5f33E28387855d33980719f27f0d579392aF
   source code for audit purposes:
   https://etherscan.io/address/0xe64d5f33e28387855d33980719f27f0d579392af#code
3. Croupier bot
   Ethereum address: 0xd3502FCFd30Db5819caB83990FD05661851D605D
4. The website
   https://etherjack.io/

The Smart Contracts

The two smart contracts are written in Solidity. Their source code is available for public review so that everyone can ensure the game is run fairly.

The contracts are proprietary software with disclosed source code and not open source software. All copyright is retained by the authors, any modifications of the source code or its use outside of the intended purpose (the audit) is disallowed.

Token Smart Contract

The Token smart contract:

- Defines the JACK Token and its transfer functions
- Handles deposits and withdrawals (intercepts transfers to Croupier and applies custom logic)
- Handles token bids (intercepts transfers to Jackpot and applies custom logic)
- Handles deposit freezing and unfreezing (API available for Croupier only) for token sales through the Jackpot contract

There's fixed supply of 1 million JACK, which can only decrease. Each time a token is used for a bet, it's burned irreversibly, thus decreasing the total supply.

JACK Tokens can be freely transferred between players or traded at exchanges.

When Tokens are sent to Croupier, they are counted towards the user's deposit with Croupier on-chain. Later, the entire deposit can be withdrawn, on-chain using the deposit data stored in the blockchain.

If a deposit owner wants to be selling tokens through the Jackpot contract they ask the Croupier to do so. The Token contract freezes the deposit by Croupier's request, thus making it available for off-chain manipulations on Croupier's side.

If the user wants to stop selling and return the JACKs back to the deposit, Croupier will request deposit unfreezing, which will return the deposit amount back to the blockchain.

Jackpot Smart Contract

The Jackpot smart contract runs the game:

- ICO stage: selling tokens for investments
- Starting the game when the ICO goal is complete
- Ether bids
- Token bids
- Minor prize tracking and payouts
- Game end conditions
- Jackpot payout

Technical Implementation

All critical parts of the game are implemented in this smart contract and happen on-chain. This includes tracking bids, last bidder, keeping the jackpot, checking game end conditions, paying out the jackpot.

The smart contract interacts with other parts of the system (Token Contract, Croupier, the website) via method calls and events.

Communication with players is done solely by accepting and emitting transfers (JACK and Ether).

Actions on incoming Ether transfer depend on the stage:

- **ICO:** transfer bought JACK to the investors
- **Game stage:** accept Ether as a bid, if it's double bid price or more, sell JACK; accept JACK as a bid
- **Game over:** if the jackpot hasn't been paid out yet, and there's an incoming transfer for any amount from the winner, transfer the jackpot to the winner. This operation is not really needed, since Croupier will automatically make Jackpot perform the payment when it detects game end, but it is there to guarantee the payout without relying on an off-chain component (Croupier).

Croupier Bot

Croupier is a bot running in protected environment, working with an Ethereum node. It operates with an off-chain database and has the following functions:

- When it detects that the conditions for ending ICO and starting the game are met, it asks Jackpot to start the game. This ability is **not exclusive** to Croupier, any user can do that by performing a call, or just transferring any amount of Ether to Jackpot.
- When it detects the game end conditions, it asks Jackpot to end the game. This ability is **not exclusive** to Croupier, any user can do that by performing a call, or just transferring any amount of Ether to Jackpot.
- When the game is ended, it asks the Jackpot contract to pay out the jackpot. his ability is **not exclusive** to Croupier, the winner can get the jackpot paid out by transferring any amount of Ether to Jackpot.
- Every time an Ether bid is made, the Jackpot contract transfers dividends to Croupier. Croupier distributes 70% of the dividends it receives among JACK deposit holders proportional to their deposit value (off-chain operation).

Croupier Bot functions (continuation):

- Players can request payout (by transferring a specified amount of Ether to Croupier), and Croupier will transfer the Ether it owes to them.
- Players can request JACK deposit withdrawal (by transferring a specified amount of Ether to Croupier), and Croupier will trigger an on-chain deposit withdrawal operation.
- Players can request that Croupier sets their JACK deposit on sale at current token price (by transferring a specified amount of Ether to Croupier), or stop selling their deposit (by transferring a different amount), and Croupier will trigger a combination of on-chain and off-chain operations to make that happen.
- Manages player message activation.

Website

The http://etherjack.io/ website displays the status of the game, instructions for the current stage, list of messages, as well as the game rules.

The site listens to the blockchain status changes and pushes them to users' browsers, updating the information with each new block. Real-time feed of bids placed and minor prize won is displayed, with a link to view each transaction via etherscan.io.

The JACK holders' dashboard allows token holders to see their balances and statuses, and get instructions for interacting with the Croupier, for depositing and selling JACK, or getting paid the earned interest.

While the game is in ICO stage, the site shows how the game is going to look when the game is running, with simulated bids incoming.

House

House is an Ethereum account controlled by the developers. It is the account from which the Token and Jackpot contracts were deployed. House has a very limited set of special abilities:

- **Contract trust set-up.** When the Token and Jackpot contracts were created, House told them the addresses of each other. This operation can only be done once, and it has been performed already.
- **Emergency ICO abortion.** While the game is in ICO stage, House can abort the ICO in case of emergency. If the ICO is aborted, investors can return their investments from the Jackpot (on-chain). If the game is not in ICO stage, House can't abort it.
- **Post-abortion termination.** If the ICO is aborted, House can terminate the Jackpot contract in 2 months after the abortion date.

Apart from this, House has no more power to influence the game than any other Ethereum user.

ICO Mechanics

At contract deployment, 1 million JACK Tokens were created. 2.5% of the amount was transferred to House (the developers) for bounty rewards. The rest is available for sale during ICO.

The ICO lasts until the jackpot (Jackpot contract's balance) has at least 333 ETH. When the threshold is reached, a 48-hour timer starts. When the timer runs out, ICO ends, and the game starts.

During ICO, investors transfer Ether to the Jackpot Contract (0xe64d5f33E28387855d33980719f27f0d579392aF) and receive JACK in return. JACK is the token currency actually used in the game, so it has its intrinsic value just according to the game rules, and a growing price as coded in the contract.

ICO Mechanics

The sale price of JACK changes during ICO, so it's more profitable to buy first, rather than last. The price depends on the amount of JACK sold, as indicated in the table below.

JACK sold by the moment of sale	JACK price
< 10,000	0.004 ETH / 4 finney
10,000 – 19,999	0.005 ETH / 5 finney
20,000 – 29,999	0.0053 ETH / 5.3 finney
30,000 – 39,999	0.0057 ETH / 5.7 finney
40,000 +	0.006 ETH / 6 finney

Table. JACK price change during ICO

When buying 200 JACK or more with one transaction, a volume bonus is applied. The bonus discount rules are indicated in the table below. You can find a calculator tool at the website (https://etherjack.io) at the bottom of the page.

JACK bought	Bonus
1 – 199	none
200 – 499	+5%
500 – 999	+10%
1,000 – 4,999	+1/7 (~14.3%)
5,000 – 9,999	+20%
10,000 +	+25%

Table. JACK volume bonus during ICO

House gets 20% of each sale during ICO, for the marketing budget. The rest (80%) goes to fill the jackpot.

The more Ether is accumulated in the jackpot by the game start, the more attractive the game will be for the players. Player activity generates profit for JACK holders and makes the price of JACK grow faster.

Possible Scenarios

After the game has started, nobody knows for sure who is going to get the jackpot and how fast. Players will be trying to claim the jackpot from one another, filling the jackpot even more with their bids, which in turn will make the jackpot more attractive.

JACK holders benefit from continuing game: the more bids are placed, the higher is the value of JACK, and the more dividends are paid out.

Since the JACK price has a fixed minimum price at every point (as hardcoded in the contract, see "Bid Price" and "ICO Mechanics" for specifics), it's possible to estimate baseline Return on Investment (ROI) purely based on JACK price, depending on when they are bought and when they are sold.

This is only baseline ROI, since JACKs also generate dividend payouts provided they are stored in the Vault (see "JACK Token Economy" for details), and that additional revenue is not accounted for in this calculation.

The table below illustrates JACK price changes and corresponding ROI.

ICO 0.004–0.006 ETH / JACK	**1,000 bids** 0.07 ETH / JACK (35 days from start max*)	**5,000 bids** 0.31 ETH / JACK (67 days from start max*)	**10,000 bids** 0.61 ETH / JACK (92 days from start max*)	**50,000 bids** 3.01 ETH / JACK (8.6 mo. from start max*)
1 ETH invested	17.5 ETH ROI: 1650%	77.5 ETH ROI: 7,650%	152.5 ETH ROI: 15,150%	752.5 ETH ROI: 75,150%
–	1 ETH invested	4.4 ETH ROI: 340%	8.7 ETH ROI: 770%	43 ETH ROI: 4,200%
–	–	1 ETH invested	2 ETH ROI: 97%	9.7 ETH ROI: 870%
–	–	–	1 ETH invested	4.9 ETH ROI: 390%

Table. Return on Investment depending on entry stage.

* This is the time it would take to get N bids if each and every bid is made in the last possible moment, that is the longest possible time.

Since in reality bids will be made earlier (risks of blockchain delays, desire to claim minor prizes, etc.), the actual time span is expected to be significantly shorter.

The maximum time from start for N bids is calculated as sum of round times:

$$T_{max}(N) = \sum_{n=1}^{N} T(n)$$

JACK Token Economy

JACK Overview

JACK Tokens are the internal currency of the game. JACK has no fractional denominations, its symbol is "JACK".

JACK Tokens are acquired during the ICO stage, and can be acquired later in the game, and can be used by investors (ICO participants) and players (non-ICO game participants) to extract profit from the game's operation, capitalize on JACK's growing price, and use JACK to place bids without spending Ether.

The JACK Token is an ERC20 token, it can be watched at the following address: 0xe64d5f33E28387855d33980719f27f0d579392aF. Symbol: "JACK", decimals: 0.

The token's Solidity code is available for public review at etherscan.io.

Theory Behind the JACK Value

The JACK token has an intrinsic value because of the two token's functions:

- It can be used for bidding instead of Ether (therefore its value is no less than the bid price at any time);
- Its holder can get a portion of value of all bids made and new JACK tokens sold through the Jackpot contract (dividends).

The value of JACK is backed by the presence of the jackpot prize, since the jackpot is the motivating factor behind players placing bids.

Players placing bids:

- Generate dividends to be paid out to JACK deposit holders;
- Confirm the current growing price of JACK (jackpot is worth bidding at the current price, and JACK can be used for bidding, therefore JACK is worth at least the current price).

Every time a bid is made, jackpot grows in size, dividends are paid out, and JACK supply stays the same. This drives the growth of the value of JACK.

JACK Minting

There is a fixed initial supply of 1 million JACK. The total supply never exceeds the initial supply and can only decrease due to JACK tokens being burned when used for bidding.

All ever existing JACK were minted on contract creation. No new JACK will ever be minted.

When JACK were minted, House got 2.5% of the supply for bounty rewards; the other 97.5% were credited to the Jackpot contract, so that it can sell them during ICO and the game.

Acquiring JACK

JACK Tokens can be acquired in the following ways:

- Participating in the game's ICO: cheapest option (see “ICO Mechanics” for details);
- Bidding at least double bid price during the game phase (see below);
- Buying from exchanges or directly from other people.

When a player bids at least double bid price of Ether, they get JACK tokens back from the Jackpot contract, provided there are JACK tokens available for sale (from Jackpot itself or from selling investors). The number of JACK they get is how many times the extra amount (transaction value less bid price) is greater than the bid price. That is, if the current bid price is 0.01 ETH, and a player bids 0.1 ETH (10 times the price), they get 9 JACK back.

When getting at least 20 JACK, a volume bonus is applied. The bonus discount rules are indicated in the table below. You can find a calculator tool at the website (https://etherjack.io) at the bottom of the page (it calculates ICO investment during the ICO phase and switches to calculating JACK for bid when the game phase is entered).

JACK bought	Bonus (amount is always rounded down)
1 – 19	none
20 – 49	+5%
50 – 99	+10%
100 – 299	+⅐ (~14.3%)
300 – 999	+20%
1,000 +	+25%

Table. JACK volume bonus during the game phase

Every time Jackpot sells JACK Tokens (from its own reserve) during the game phase, 50% of the value goes to fill the jackpot, and the other 50% are distributed as dividends among JACK deposit holders (see "Vault Deposit").

Selling JACK

JACK holders can sell JACK at its face value in the following ways:

- Through the Jackpot contract (see below);
- Selling through exchanges or directly to other people.

When selling tokens to bidders, the Jackpot contract sells them from:

1. JACK Tokens put on sale by holders via Croupier, or
2. Jackpot's JACK balance.

The Jackpot contract sells tokens from its own balance only if there are no tokens put on sale through Croupier. As long as there are tokens put on sale via Croupier, Jackpot sells only them as first priority.

Any player who has at least one JACK token, can ask Croupier to put their JACK tokens on sale. To do this one has to log in to their JACK holder's dashboard at http://etherjack.io/ and follow the instructions, which would be:

1. Transfer the JACK tokens they would like to sell, to Croupier. This will place them in the Vault.
2. Transfer a specific amount of Ether to Croupier as a signal that they would like to have their deposit put on sale. Croupier will then put the tokens on sale with the Jackpot contract.

Every time when there are tokens on sale through Croupier, and somebody buys them:

- Jackpot transfers the full sale price of the tokens to Croupier (on-chain);
- Jackpot emits an event (on-chain) to notify Croupier that tokens were sold from Croupier's pool, and for how much;
- Croupier receives the event, randomly selects seller for each of the sold token, and credits them (off-chain) with a proportional fraction of the sale value (which corresponds to JACK face value for each token sold, or slightly less if a volume bonus was applied).

If a player wants to stop selling their JACKs, they can send another specific amount of Ether to Croupier (per instruction in the JACK holder's dashboard) to have the remaining JACKs moved back to Vault.

When a player is credited with Ether for sale of their tokens, it's done within Croupier records and doesn't generate an immediate on-chain transaction (to avoid unreasonable gas expenses on many micro-transactions). A player can request payout from Croupier at any time by following instructions in the JACK holder's dashboard.

Vault Deposit

Any player who has at least one JACK can become a deposit holder with Croupier and start receiving dividends: a share of every bid made, and every token sold from the Jackpot contract. To make a deposit to the Vault, one has to simply transfer the desired amount of JACK to Croupier (0xd3502FCFd30Db5819caB83990FD05661851D605D). The Token smart contract handles such transfers in a special way and converts them to deposits. The Token smart contract keeps track (on-chain) of all Vault deposits.

Dividends are:

- 25% of every Ether bid made during the game;
- 50% of sale price of every JACK sold from the Jackpot contract (but not from other players through Croupier: when tokens are sold from other players, the seller gets 100% of the sale value).

Dividends are calculated on-chain (logged in events emitted by the Jackpot contract) and transferred to Croupier. Croupier then uses off-chain lookups in its internal database to properly distribute them among deposit holders.

Every time dividends are paid to Croupier, it distributes them in the following way:

- 70% is split among deposit holders, proportional to the size of their deposit;
- 30% goes to House (developers) to cover running costs and ongoing marketing support.

When a player is credited with Ether for sale of their tokens, it's done within Croupier records and doesn't generate an immediate on-chain transaction (to avoid unreasonable gas expenses on many micro-transactions). A player can request payout from Croupier at any time by following instructions in the JACK holder's dashboard.

A deposit holder can also withdraw their JACK deposit from the Vault at any time by asking Croupier to trigger an on-chain deposit withdrawal transaction (by following instructions in the JACK holder's dashboard).

JACK Holder’s Dashboard

The state of the deposit, token sales, and profit can be viewed at any moment in the JACK holder's dashboard. The dashboard is accessible from the game's website at https://etherjack.io/ and shows:

- Current deposit in Vault
- Current amount of JACK on sale with the Jackpot contract
- Total Ether profit from deposit (dividends) over all time
- Total Ether profit from token sale with the Jackpot contract over all time
- Amount of Ether available for immediate payout

The dashboard shows (continuation):

- Detailed instructions for:
    - Depositing JACK
    - Withdrawing JACK
    - Putting JACK on sale
    - Removing JACK from sale
    - Getting a payout

Since the game including its investment related part runs inside the blockchain, all game status is public at all times, and the only possibility to authenticate a user is by their ability to emit transactions from their controlled addresses (the blockchain way). For that reason, no operations can be performed from the dashboard directly. For any action with the deposit to be performed, a user has to initialize a transaction on the Ethereum blockchain from their address. This transaction will then be received by Croupier and processed. The results will be reflected in the dashboard as soon as all the transactions involved get at least 12 block confirmations.

Winning the Jackpot

If the bid timer runs out and nobody makes a bid, the last bidder wins the jackpot. As soon as the conditions for that are met (they rely on the timestamp of the last block), anyone can ask Jackpot to check game end conditions. Croupier will perform the call automatically as soon as it detects that the conditions are met.

Alternatively, any amount of Ether sent to Jackpot when the game end conditions are met, will also trigger game end, and the Ether sent will be refunded.

When the game is finished, Jackpot emits the corresponding event and awards the jackpot:

- 80% goes to the winner;
- 20% goes to House.

Croupier will ask Jackpot to pay out the jackpot to both the winner and the house. If that doesn't happen for any reason, both the winner and House can send any amount of Ether to Jackpot to have it pay out their portion.

This ensures that even in case of a global apocalypse, the jackpot transfer is guaranteed to be performed, provided the Ethereum network lives in any capacity (in basements, powered by diesel generators), without relying on any off-chain components.

Everything described in this section happens entirely on-chain. Croupier (which is an off-chain component) participation is optional and is introduced just for convenience. It can be verified in the contract code that Croupier cannot affect the termination and payout in any way other than anyone else.

Once the jackpot is won, the game is ended, and Jackpot will refuse to accept any Ether or JACK (except from winner or House for payout).

Promotion Plan

It is in both ICO investors' and developers' interest to have as many bids as possible placed, and as many as possible JACK Tokens sold during the game phase.

Therefore it is the creators' task to attract as many people as possible into the game. Funds collected via the ICO as well as the House' share of running dividends will be used for acquisition of new players.

The main directions of marketing are:

- Bounty program on forums. 2.5% of minted JACK will be used for this purpose. The bounty program description is available at https://etherjack.io
- Targeted ads on general purpose platforms: Facebook, Instagram, Google AdWords.
- Paid advertisement on Cryptocurrency, ICO, and Ethereum centric portals.

Bounty Campaign

Additionally, we are launching a bounty campaign at bitcointalk.org for additional visibility and spread across the cryptocurrency community.

The bounty campaign consists of:

- Localization (website and documentation) to major languages;
- Social promotion: sharing on social networks;
- Bitcointalk.org signatures and avatars;
- Articles, YouTube videos and other media

Link to the ANN thread: https://bitcointalk.org/index.php?topic=2810343.0
Bounty campaign: https://bitcointalk.org/index.php?topic=2810352.0