

# BingoEth

Relazione progetto P2P

Leonardo Manneschi

Leonardo Scoppitto

Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Struttura del progetto</b>	<b>1</b>
<b>3</b>	<b>Manuale utente</b>	<b>1</b>
3.1	Utilizzo istanza pubblica . . . . .	1
3.2	Esecuzione da sorgente . . . . .	2

# 1 Introduzione

L'applicazione sviluppata come progetto finale del corso di Peer To Peer & Blockchains si chiama BingoEth e si tratta di un'implementazione del bingo 75 (la versione americana del bingo) realizzata in React/javascript per la parte di front-end e in Solidity per la parte di smart contract. È possibile consultare il repository

## 2 Struttura del progetto

Il progetto è stato organizzato come segue:

```
├── client → Codice del client scritto in React
├── docs → Directory contenente la relazione del progetto
├── truffle → Root del progetto dello smart contract
│   ├── contracts
│   │   └── Bingo.sol → Sorgente del contratto
│   ├── migrations
│   │   └── 1_deploy_contract.js → Script di deploy del contratto
│   └── test → Test del contratto
└── Dockerfile/docker-compose.yml → Container docker per il test e il deploy
```

```
function accuse (int256 _gameId) public {
    require(_gameId > 0, "Game id is negative!");
    require(gameList[_gameId].creator!=msg.sender, "Creator cannot accuse!");
    require(
        containsAddress(
            gameList[_gameId].joiners,
            msg.sender
        ), "Player not in that game!");
    require(gameList[_gameId].accusationTime == 0, "Accusation already made!");
    gameList[_gameId].accusationTime = block.timestamp;
    gameList[_gameId].accuser = msg.sender;
    emit ReceiveAccuse(_gameId, msg.sender);
}
```

## 3 Manuale utente

### 3.1 Utilizzo istanza pubblica

Per rendere più fruibile l'utilizzo dell'applicazione sviluppata, è stato eseguito il deploy di **BingoEth** su un server privato insieme a un'istanza di Ganache, così da non dover configurare nulla se non la rete su Metamask. È possibile accedere al progetto all'url <https://bingoeth.alteregofiere.com/>.

Gli account disponibili con le relative chiavi private sono:

Available Accounts

=====

```
(0) 0xC2F709C582CDe40CA38b108Fb0e639c14e108A8a (10000 ETH)
(1) 0x35301246031343A7d13507b19aa6d4fE40F6587c (10000 ETH)
(2) 0x97f4Ec285360456cb008B71B22cF1eFDd094e766 (10000 ETH)
(3) 0x0fac963Ae1E20De87294A627938da0f04d9eb78E (10000 ETH)
(4) 0x2b614F260BDD54FF6180543C9125A0e69679d04a (10000 ETH)
(5) 0x9307C2e5EB3e6935D6F35bFa13288BFc36aDd846 (10000 ETH)
(6) 0x7670F41114B30a6f06FD629707b193eCe545a59A (10000 ETH)
(7) 0x1923D74eAC7Fde59A9b08bAc5970350644cC2a90 (10000 ETH)
(8) 0xc0748B93286b8DC2D6C2736725413De61da4Bf56 (10000 ETH)
```

(9) 0xe50a5343693edd7F9c52ec66ff7358b5f18300B1 (10000 ETH)

#### Private Keys

=====

(0) 0x99409bd109959b84aae9234ce755a39a6df4c009a53c6a81ea0713d65bd80eb7  
(1) 0x98d1ccd915ef23a2a088dfec9a47960f7de3b71d5f3c2aff53b7cb85e411adc5  
(2) 0xe41c4f1aad19f7238246df7712f748c5cf4d6ac1c8a78e1b3aafb4a74e6ae39d  
(3) 0x42941bc9871700dade0f40b607a8e0528fd9578dbefcf4abe6b4f4b2eea817b0  
(4) 0x7a956b721714c8708ff22b25b193dae7efc7eaec77fe5b40c5c2944fc42feebe  
(5) 0x9d19b8c0ad94ac66122485684d44bfa782dd84def2c2382b991f3632f3b9a294  
(6) 0x8b93665547a073f7055c043ce4e6115a13e0900d18e23d206c9b0b17f8bf27e8  
(7) 0x93c2d0ce2d1398fd5585c348a7befd000fecfebd0c9d5005550e820317ab0a47  
(8) 0xd994441bc950b1d80191c8a0454a7b186a3562a8cf486390d4f740a713384d81  
(9) 0x9f6e155cd2faf86bb96331732d9f5c6727fe0858773d7ad141f67425e7b1fe1b

### 3.2 Esecuzione da sorgente

*Nota: per l'utilizzo del progetto è consigliato avere docker (consultare <https://docs.docker.com/get-docker/>) installato sul proprio pc.*

Steps:

1. Clonare il repository da [https://github.com/leomanne/P2P\\_Project](https://github.com/leomanne/P2P_Project) o da <https://github.com/ski-by7/BingoEth> (il secondo repository è il mirror del primo) o scompattare l'archivio BingoEth.zip
2. Aprire il terminale/powershell e spostarsi dentro la directory del progetto
3. Eseguire il comando `docker compose up`

Eseguiti i passi indicati, inizierà il processo di build del container docker. Dopodiché si avvierà il container che compilerà il contratto, per poi avviare l'applicazione **BingoEth**. Una volta terminato il processo di build, sarà possibile accedere all'applicazione tramite browser all'url <http://localhost:80>.

Gli account configurati su ganache, esposto all'url <https://chain.alteregofiere.com> sono i medesimi indicati nella sezione precedente