Ethereum Vienna
1st Workshop
Contract Development And Deployment

Workshops

- 1st Workshop:
  - Solidity basics
  - Geth basics
- 2nd Workshop: (probably March)
  - Designing a contract
  - Advanced Contract Development
  - Testing
- 3rd Workshop: (TBD)
  - Frontend Stuff

Agenda

- EVM
- Ether Camp
- Solidity
- Geth
- Solidity

ETHEREUM

- Transaction
  - Wraps a message
  - Signed by a private key
  - Only transactions appear in blockchain
  - Gasprice

ETHEREUM

- Message
  - Sender
  - Recipient
  - Value (can be 0 wei)
  - Data
  - Return Value
  - Gaslimit
  - Executes either completely or not at all

ETHEREUM

- Contract
  - 160 bit address
  - Balance
  - EVM Bytecode
  - Runs at every received message
  - Has a persistent 256-bit to 256-bit storage
    - Private
    - Expensive
  - Can spawn new messages

EVM

- Stack machine
- 256 bit words
- Usual instructions
- Special instructions
  - Block data access
  - Tx data access
  - Msg data access
  - Contract data access

EVM

- Storage
  - expensive
  - persistent
- Memory
  - cheaper
  - byte-level-access
- Stack
  - Inaccessible in solidity

EVM

- Out-of-gas Exception
- Logs
  - For Uis
  - Light clients
  - Logging
- Self Destruct / Suicide

ETHER.CAMP

- Solidity.read-the-docs.org

- http://austria-{GROUP}.on.my.ether.camp/ide.html
- User: u{GROUP}{USER}

SOLIDITY

- Coin Contract
  - Creater can issue coins
  - Coins can be sent between users
- Developer writes contract with functions
- Compiler generates init code and dispatcher
- At deployment the contract constructor is executed

SOLIDITY

- Types (with the usual operators)
  - Bool
  - Int: Signed Integer 256-bit (Other sizes available)
  - Uint: Unsigned Integer 256-bit
  - Array (static and dynamic)
  - String
  - Enum

SOLIDITY

- Other Types
  - Address: 160-bit for ethereum address
    - balance
    - send
    - call / callcode
  - Mapping (hashtable-like)
    - from one solidity type to another
    - contains already all keys
  - Contract
    - Like address, but with functions of a specific contract

SOLIDITY

- Control Structures
  - If
  - for
  - While
- this (only for balance and functions) and super
- Automatic getter generation for public variables
- Special variables for blockchain interaction

SOLIDITY

- Global variables
  - msg: specific to one message
    - sender
    - value
    - gas
  - tx: shared by all messages in a transaction
    - origin: creater of the transaction
    - gasprice

SOLIDITY

- Global variables
  - block: shared by all transactions in the same block
    - coinbase
    - difficulty
    - timestamp
    - blockhash
    - Number
- Some special functions related to cryptography (e.g
  sha3)

SOLIDITY

- Events for writing to the Log
-
- Import for importing contracts from other source file
-
- Standard contracts
- Contract Inheritance
  - Code from ancestor copied into contract
  - Still only 1 contract

SOLIDITY

- Modifier for code reuse
- Throw
  - creates exception
  - execution aborts, state reverts
  - cannot be caught on contract functions
  - all gas is used up

#1 Trusted Data Feed

- Contract contains 1 field
- Can only be changed by creator
- Change fires Event
- Field can be read by other contracts
-
- relevant: msg.sender

#2 Subscription

- ONLY 1 Subscription per Contract!
- Recipient
  - Can withdraw PRICE wei per TIME
- Creator
  - Can cancel if there are no outstanding payments
  -
- relevant
  - address.send
  - block.timestamp (unix timestamp / seconds)

SOLIDITY

- To call another contract:
  - Coerce address into the contract type
  - Call the function on that
  - Call .value on function to sent wei
  - Call .gas on function to limit gas

SOLIDITY

- Structs
- Arrays
  - Static
  - Dynamic (push function)

SOLIDITY

- delete
  - Reset a variable to its default value
- functions can have multiple return values
  - Retrieve values by deconstruction:
  - var (x,y,z) = f();

SOLIDITY

- Function modifiders
  - External
    - Can only be called by a message
  - Public
    - Can be called by anyone
  - Private
    - Can only be called by the contract itself
  - Internal
    - Cannot be called by a message

#3 Store

- Add offer (with name and price)
- Take offer (by sending the right amount)
- Confirm the offer (and release the money)