CS 252:

Advanced Programming Language Principles

Event-based Programming *& cryptocurrencies*

Prof. Tom Austin

San José State University

Inline JavaScript

```
<html>
   <input
      type='button'
      onclick='alert("Hello!");'
      value='Say hi' />
</html>
```

Using in-line event handlers is common, but considered bad practice.

The "better" approach

```
<html>
  <input id='thebutton'
         type='button'
         value='Say hi' />
  <script type="text/javascript">
    var btn = document.
                getElementById('thebutton');
    btn.onclick = function() {
        alert('Groovy');
    };
  </script>
</html>
```

Note the `id` attribute

Perhaps better still

```
<html>
  <input id='thebutton'
         type='button'
         value='Say hi' />
  <script type="text/javascript">
    var btn = document.
              getElementById('thebutton');
    function sayGroovy() {
        alert('Groovy');
    }
    btn.addEventListener('click', sayGroovy);
  </script>
</html>
```

```
function sayGroovy() {
  alert('Groovy');
}
btn.addEventListener('click',
                     sayGroovy);
btn.addEventListener('click',
  function(){ alert("Bogus");}
);
```

```
function sayGroovy() {
  alert('Groovy');
  btn.removeEventListener('click',
                          sayGroovy);
}
btn.addEventListener('click',
                     sayGroovy);
btn.addEventListener('click',
  function(){ alert("Bogus");}
);
```

Note that JavaScript (in a browser) is single threaded.

An event runs to completion before the next event begins.

Importing events in Node.js

Choose whatever name you like

```
var EE =
  require('events').EventEmitter;
```

`events` is a module

```
var EE = require('events').EventEmitter;
var ee = new EE();

die = false;

ee.on('die', function() {
  console.log("I'm melting!!!  Oh, what a world...");
  die = true;
});


setTimeout(function() {
  ee.emit('die');
}, 100);

while (!die) {}

console.log('done');
```

TCP Server example

```
var net = require('net');
var eol = require('os').EOL;

var srvr = net.createServer();

srvr.on('connection', function(client) {
  client.write('Hello there!' + eol);
  client.end();
});

srvr.listen(9000);
```

```
$ node tcpserver.js
$
```

```
$ telnet 127.0.0.1 9000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello there!
Connection closed by foreign host.
$
```

Building a Cryptocurrency

Digital Currency – Take 1

"I Alice agree to pay the bearer of this note $100"

Alice

Bob

Any issues with this approach?

Digital Currency 1 – Repudiation

I did not write that note. I'm not paying.

I want my money.

Alice

Bob

"I Alice agree to pay the bearer of this note $100"

Public Key Cryptography &

Digital Signatures

Public Key Encryption

- Uses two separate keys
  - Public key known by everyone
  - Private key known only by owner
- Analogy: locked mailbox
  - Anyone can put a letter in the mailbox
  - Only the mail carrier can get them

Digital Signatures

- Reverse process for digital signatures:
  - Private key encrypts a message
  - Public key decrypts the message
- Analogy: Enclosed bulletin board
  - Anyone can read
  - Only the owner could have posted the messages there

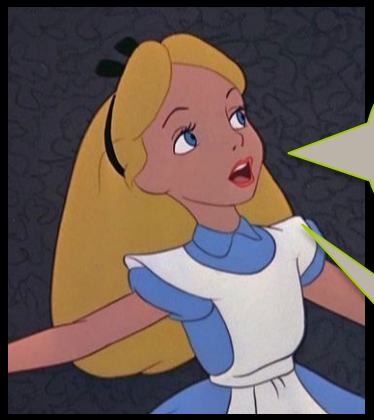
Digital Currency – With Signatures

"I agree to pay the bearer of this note $100"

Alice

Bob

Digital Currency with Signatures Nonrepudiation



I did not write that note. I'm not paying.

Oh… OK.

I want my money.

It has your signature.

Alice

Bob

"I agree to pay the bearer of this note $100"

Alice

Signing example
(in-class)

Double Spending

Take this IOU from Alice to square our debt.

OK

Bob

"I agree to pay the bearer of this note $100"

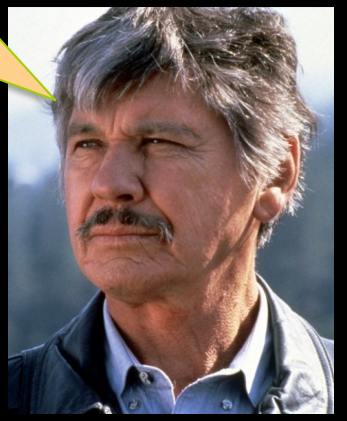
Alice

Charlie

Double Spending

But… I already paid.

I'm here to collect my money.



Alice

Charlie

"I agree to pay the bearer of this note $100"

Alice

A centralized authority could monitor all transactions…

Because everyone trusts banks

Or everyone could keep track of the transactions, and vote whenever a discrepancy arises.

What is a cryptocoin worth?

In our earlier examples, the value of our IOUs is tied to US dollars.

Other cryptocurrencies (such as Bitcoin) are not tied to any other currency. We'll follow this model from now on.

Digital Currency – Ledger

Alice: 20
Bob: 11
Charlie: 5
David: 34

Alice

"I am giving 10 cryptocoins to Bob"

Alice

Alice: 20
Bob: 11
Charlie: 5
David: 34

Bob

Alice: 20
Bob: 11
Charlie: 5
David: 34

Charlie

Alice: 20
Bob: 11
Charlie: 5
David: 34

David

Digital Currency – Ledger

Alice: 5
Bob: 11
Charlie: 20
David: 34

Alice

"I am giving 15 cryptocoins to Charlie"

Alice

Alice: 5
Bob: 11
Charlie: 20
David: 34

Bob

Alice: 5
Bob: 11
Charlie: 20
David: 34

David

Alice: 5
Bob: 11
Charlie: 20
David: 34

Charlie

Digital Currency – Ledger

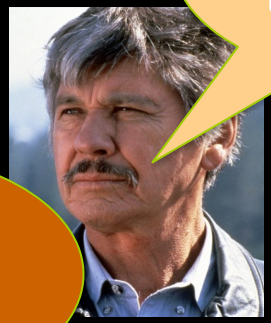
Alice

"I am giving 8 cryptocoins to David"

Alice

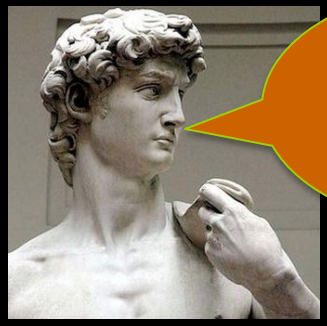
Invalid transaction!

Bob



Invalid transaction!

Charlie



Invalid transaction!

David

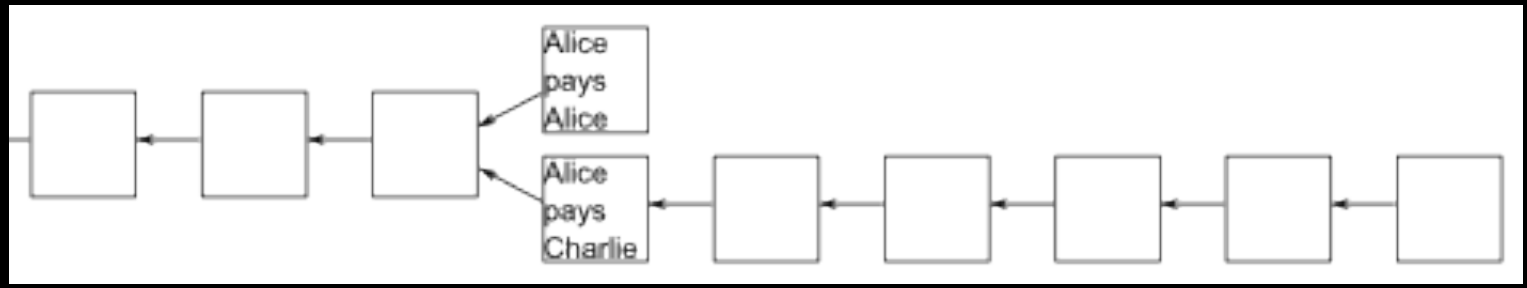

Bitcoin Miners

People are always tempted to cheat. How can we catch them in a peer-to-peer system?

Bitcoin Proof of Work

- The Bitcoin protocol uses proof of work to verify *block chains*, which determine transaction history.
- This strategy prevents *double spending*, where Alice tries to spend the same coin with both Bob and Charlie.

Blockchains and Proof-of-work



Cryptographic hash functions

- Hash functions compress input to a small, fixed-length output.
  - hash("Some long input") = 87d9417d8cd12635
- A good hash function should be:
  - **one-way**: given a value y it is infeasible to find an x such that h(x) = y
  - **collision-resistant**: infeasible to find ***any*** x and y, with $x \neq y$ such that h(x) = h(y)
  - **efficient**
  - **compression**

Hashes and cryptocurrencies

- Consider `h = hash(someValue)`, where `h` is represented as a binary string.
- What are the odds that the first character of `h` is a 0? What about the 2nd character?
- What are the odds that `h` will start with 2 zeroes? With 3 zeroes?
- And how does this relate to cryptocurrencies?

Mining

- Miners hash transaction details plus a "proof" of spending computational resources
  - Reward: some bitcoins are generated plus there can be transaction fees
- Cost to discover proof – 2^N hashes
- Cost to verify proof – One hash
- The Bitcoin protocol is designed to make mining more profitable than cheating
  - https://bitcoin.org/bitcoin.pdf

Digital Currency – Proof of Work

Alice: 5
Bob: 11
Charlie: 20
David: 34

Alice

"I am giving 2 cryptocoins to Charlie"

Alice

Alice: 5
Bob: 11
Charlie: 20
David: 34

Bob

Alice: 5
Bob: 11
Charlie: 20
David: 34

David

Alice: 5
Bob: 11
Charlie: 20
David: 34

Charlie

Digital Currency – Proof of Work

Alice: 3+1
Bob: 11
Charlie: 22
David: 34

Searching for proof of work…

Alice

"I am giving 2 cryptocoins to Charlie"

Alice

Alice: 3
Bob: 11+1
Charlie: 22
David: 34

Searching for proof of work…

Bob

Alice: 3
Bob: 11
Charlie: 22
David: 34 +1

Searching for proof of work…

David

Searching for proof of work…

Charli

Alice: 3
Bob: 11
Charlie: 22+1
David: 34

Digital Currency – Proof of Work

Alice

"I am giving 2 cryptocoins to Charlie"

Bob

Found proof!

David

Charlie

Digital Currency – Proof of Work

Alice: 3
Bob: 11
Charlie: 22
David: 35

Alice

"I am giving 2 cryptocoins to Charlie"

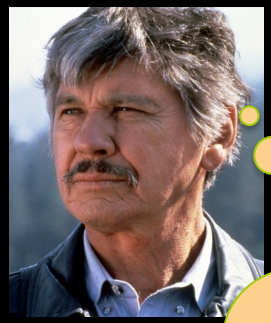
Alice

Alice: 3
Bob: 11
Charlie: 22
David: 35

Bob

Alice: 3
Bob: 11
Charlie: 22
David: 35

David



Charli

Alice: 3
Bob: 11
Charlie: 22
David: 35

Homework and Lab

- Lab: Write a chat server
- Homework: Build a cryptocurrency
  - ***For this homework only***, you may work with **one** partner (if you want).
- More details on Bitcoin: *https://**bitcoin**.org/**bitcoin**.pdf*, *by Satoshi Nakamoto*