**基于Fabric区块链的虚拟货币游戏**

虚拟货币规则：

* 每一个虚拟货币都具有唯一标识id，称为bid
* 每个虚拟货币当前价值取决于其交换价值，交换价值假设为某种法定货币，如人民币(单位：元)
* 所有虚拟货币都来自发行方管理员，不存在用户挖矿获得bid

游戏规则：

规则1，任何一个人都不允许注册多个用户。

规则2，所有注册用户都可查看每个虚拟币的交换历史，但不能查看用户真实身份，包括用户名。

规则3，所有注册用户相互平等，交换价格经协商确定，但受最近交易价格影响。

规则4，对已成交的交易，任何人不能更改或反悔。

规则5，任何用户对自己不能进行交易，**任何交易都必须在两个用户之间进行**。

起始条件：发行方管理员(一种)持有n个货币，价格1元。

假设条件：所有用户之间共享一种协商工具，一种法定货币转账工具。

区块链中记载已成交的交易，而且要求买入方(甲方)执行交易。

一次交易可买入多个虚币，但要求卖方(乙方)为同一人。

同一个虚币可在两个人之间来回购买。张三卖给李四，李四卖给张三。

基本原则：所有用户都可以自由买卖，但如果发现违规，就停止其交易，不能买卖。

约定：KV不改不删

解决方案及演示脚本

数据模型，

方案1，简单KV模型，{货币bid : 交换价格cost}

买入方执行一笔交易数据：{货币bid : 交换价格}，系统自动记录成交时间与买入方证书。

好处，演示交易方便

演示过程：

[0]启动Fabric，含CAServer

注册用户，用户“张三”注册、登记

命令行执行

node ksuser ksuser.json zhangsan

node ksuser ksuser.json lisi

node ksuser ksuser.json wangwu

[1]管理员执行初始化：{bid01:1, bid02:1, ...}

执行addjson，加入一个json文件

addjson init.json，公开5个虚拟货币{bid01:1,bid02:1, ...,bid05:1}，成本价格1元

[2] 张三，李四，王五， 用户登录查看

分别新开控制台，模拟登录

node fnc Fabks.json zhangsan

node fnc Fabks.json lisi

node fnc Fabks.json wangwu

登录可查看当前状态

query range bid01 bid99 //范围查

showKeycount //遍历查询

mytx //查看自己的所有交易，买入记录

mykeys bid01 bid99 //查看自己当前持有的虚币，以及他人持有的虚币

showBlocks //查看一个区块

showtx //查看一个交易

showtxlast //查看最后一个交易

query history bid03 //查看bid01所有交易历史

[3]张三交易

2元购买bid01

invoke put bid01 2

mytx //确认自己的交易，添加一个交易，一个区块，所有人可见

[4]李四交易，2元买入bid02

invoke put bid02 2

...

[5]王五交易

invoke addMKey {"bid03":2,"bid04":2,"bid05":2} //2元买下剩余虚币

...

[6]张三向王五购买bid03 3

invoke put bid03 3 //3元成交，所有用户可见

query history bid03 //查看一个虚币的交易历史，1，2, 3

[7]李四操作

query history bid03 //查看bid03历史，认为当前3元还有升值空间，于是:

李四执行, invoke put bid04 4 //4元成交，从王五购买

约束：

对于一次交易，用户a向b购买1个bid，如何确保a != b ?

即如何防止自己与自己交易？

bid在交易前creator证书与当前交易的用户证书应该不一样。如果一样就禁止。

方案1，前端控制，Web服务器控制，区分自己当前持有的虚币集合S与他人持有的虚币集合H。

买入时，只能从H中买入。

如何计算S与H？

执行mykey bid01 bid99，S={isMine==true}，H={isMine==false}

形成一个Web页面，显示当前他人的虚币，作为当前用户下一步交易的虚币。

方案2，中间件控制，对invoke函数进行扩展控制

最安全的方法。

本质上就是为 invoke put key value添加一个前置条件

[1]由key计算当前交易的扩展信息

mykey key ==>{value, txud,timestamp, isMine}

[2] if isMine == true then

非法交易，停止

正常交易

以上方法仅适用于一次交易单个虚币，如果交易多个，就需要再扩展。

链码函数无法实现。链码函数中无法从{key, creator}计算同一个key的前一个交易的creator

计算1：**当前用户如何查询自己的所有交易？**

执行mytx，能看到自己的所有交易，就是所有买入交易，返回一个数组可继续计算。

计算2，当前用户如何计算**当前持有**虚币，以及购买价格？

已卖出的不算。

执行mykey bid01 bid99得到当前自己所持有的虚币{isMine==true}

买入价格为value属性

计算3，当前用户如何计算自己的净利润？

净利润 = 对已卖出的虚币，sum(卖出价 - 购入价)

如果未卖出就没有净利润，但有市值。

此时，张三，李四都没有卖出，净利润为0

admin的利润=5\*(2-1) = 5

王五，卖出2个，bid03(3-2)= 1, bid04(4-2)=2, 净利润=1+2=3

还持有1个虚币bid05

区块链中仅记录买入交易，卖出交易是隐藏的，如果明确表示卖出信息就冗余。乙方是唯一确定的，可计算的。

过程：

[1]mytx==>[{key1，v1},{key2， v2},...]，所有买入

[2]for 其中每个key，计算之后的一个交易，A=tx1

if A ==空， then 无利润，没有卖出

if A != 空， then 利润 += A.tx.value - v1，已卖出

方法：已知一个{key，txid}之后同一个key的下一个txid的value。

也是一种追溯计算，由一个key，history(key)=[{txid1,v1},{txid2,v2},...]

这种计算，后一步历史，可编写新链码函数计算，也可调用history(key)然后客户端计算

计算4，当前用户如何计算自己所持虚币的当前市值？

当前市值 = 当前持有虚币数量 × 最后交易价格

此时，最后交易价格=4

admin的当前市值= 0

张三，有2个，2\*4 = 8

李四，有2个，=8

王五，有1个=4

方法：

[1]showtxlast能看到最后交易价格

[2]当前持有的虚币数量，

mykeys bid01 bid99==>{...}

==>过滤条件isMine==true，聚合count

计算5，当前用户如何计算自己的投入总金额？

投入金额 = 买入总金额

admin = 5\*1 = 5

张三，2\*1 + 3\*1 = 5

李四，2\*1 + 4\*1 = 6

王五，2\*3 = 6

方法：运行mytx，得到当前用户自己执行的交易，即买入记录

区块链计算有以下几种方式：

1. 客户端调用已有通用链码函数；
2. 纯客户端编程；
3. 调用已有通用链码函数，加客户端编程；
4. 添加新的链码函数，不加客户端编程；
5. 新链码结合客户端编程。

当有选择时，按以上次序选择，保持简单性。