Block chain final project

Aim: Simple cards base gambling based on blockchain

Description: Our system will allow two players to play with each other simple type of gambling game based on cards. Each player gets one card randomly and both users will place 10 coins that are based on our block chain. Whoever has the highest card will get all the money that were placed in a bet automatically in his/her account.

Development Language: Python 3

Block chain platform: Komodo

In order to run this project on your system there are some prerequisite which are as follow

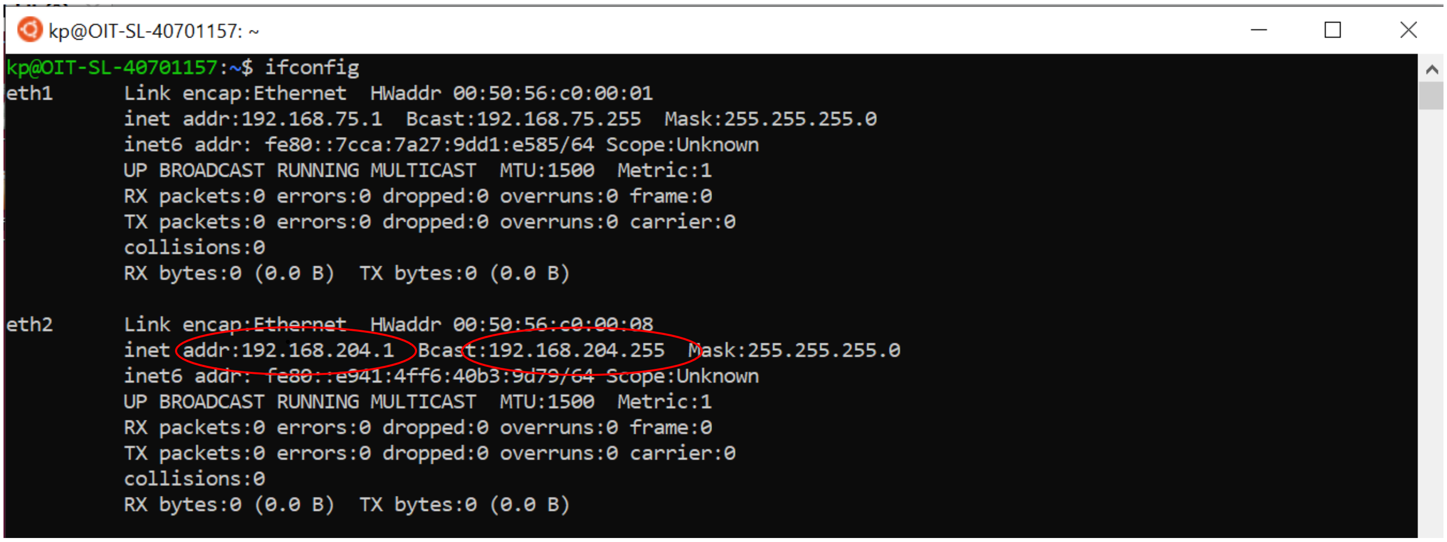
1) Python3 latest version
2) Two machines that are connected to each other using Local Area Network or same router. Alternate to this two machine method is to have a single machine but inside a single machine run virtual machine so technically we will have a two machine in a same system.
3) Installed Komodo on both of the machine
4) Operating system : ubuntu preferable
5) Install Komodo rpc on both of the machine to made a call from our python program to Komodo daemon running on both of the machines.
   Command to install Komodo rpc : pip install komodorpc

Setting up Blockchain on Komodo platform

In order to create a block chain on our local network first we need to bring minimum of two machines on a same network. In order to do that I have created a virtual LAN network between host OS and Guest OS running on virtual box.

After connecting two systems on LAN I obtained the IP address and subnet for both system

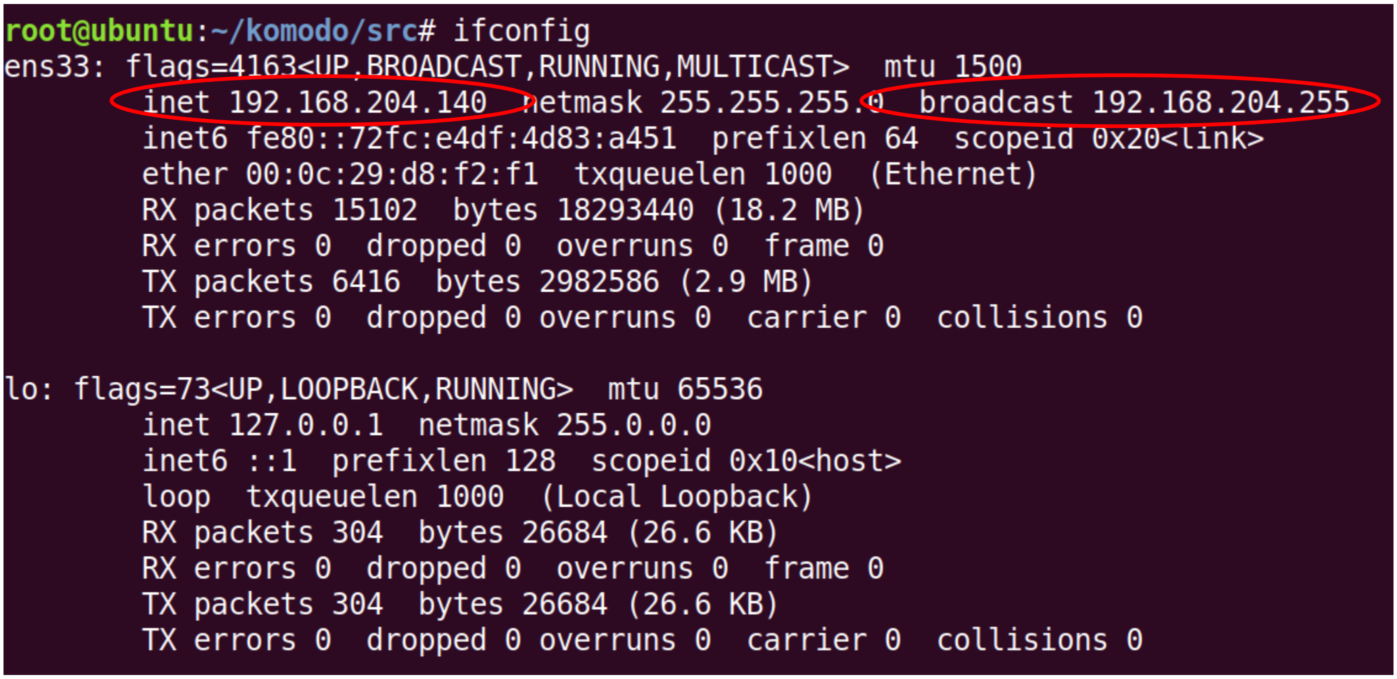
System 1



```
kp@OIT-SL-40701157:~$ ifconfig
eth1      Link encap:Ethernet  HWaddr 00:50:56:c0:00:01
          inet addr:192.168.75.1  Bcast:192.168.75.255  Mask:255.255.255.0
          inet6 addr: fe80::7cca:7a27:9dd1:e585/64 Scope:Unknown
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 00:50:56:c0:00:08
          inet addr:192.168.204.1  Bcast:192.168.204.255  Mask:255.255.255.0
          inet6 addr: fe80::e941:4ff6:40b3:9d79/64 Scope:Unknown
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

System 2



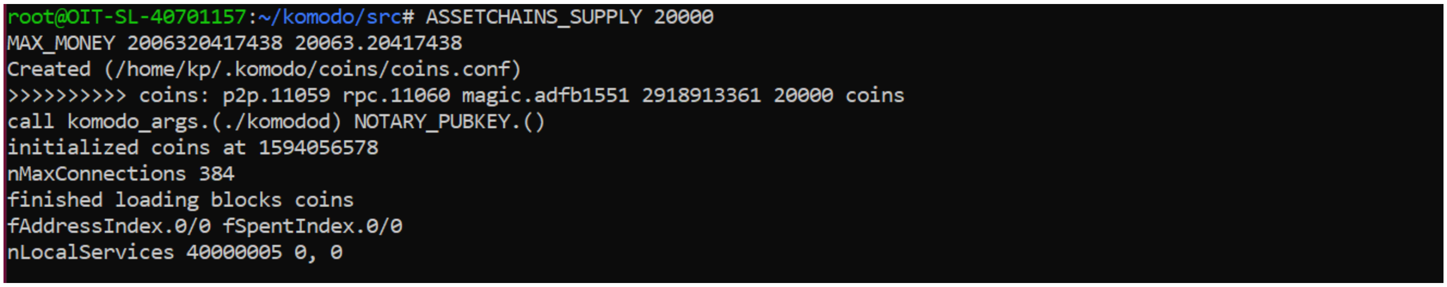
```
root@ubuntu:~/komodo/src# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.204.140  netmask 255.255.255.0  broadcast 192.168.204.255
        inet6 fe80::72fc:e4df:4d83:a451  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:d8:f2:f1  txqueuelen 1000  (Ethernet)
        RX packets 15102  bytes 18293440 (18.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6416  bytes 2982586 (2.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 304  bytes 26684 (26.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 304  bytes 26684 (26.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Starting the blockchain on system 1

After bringing two systems under same network now it is time to create a block chain on system 1 by compiling following command on system 1. Here in the command we have to write the IP address of second node (System 2). Ac_supply in command indicate initial supply of coin. We are going to name our cryptocurrency as coins.

Command : ./komodod -ac_name=coins -ac_supply=20000 -addnode=192.168.204.140&

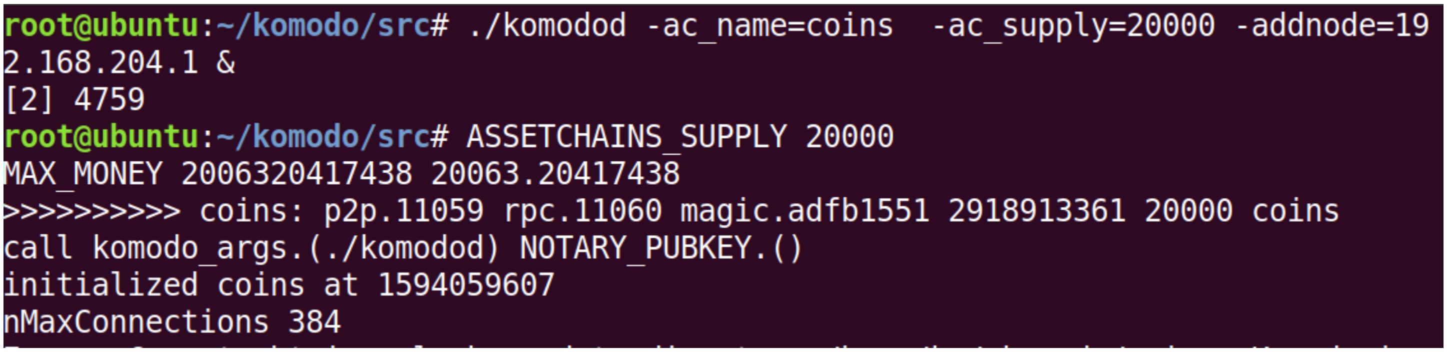


```
root@OIT-SL-40701157:~/komodo/src# ASSETCHAINS_SUPPLY 20000
MAX_MONEY 2006320417438 20063.20417438
Created (/home/kp/.komodo/coins/coins.conf)
>>>>>>>>>> coins: p2p.11059 rpc.11060 magic.adfb1551 2918913361 20000 coins
call komodo_args.(./komodod) NOTARY_PUBKEY.()
initialized coins at 1594056578
nMaxConnections 384
finished loading blocks coins
fAddressIndex.0/0 fSpentIndex.0/0
nLocalServices 40000005 0, 0
```

Adding second node in blockchain using system 2

After successfully starting a block chain on node1 (System 1) we wrote the same command as command written in node 1 (System 1) but just with one change in addnode where we will mention the ip address of node 1(System 1).

Command : ./komodod -ac_name=coins -ac_supply=20000 -addnode=192.168.204.1 &

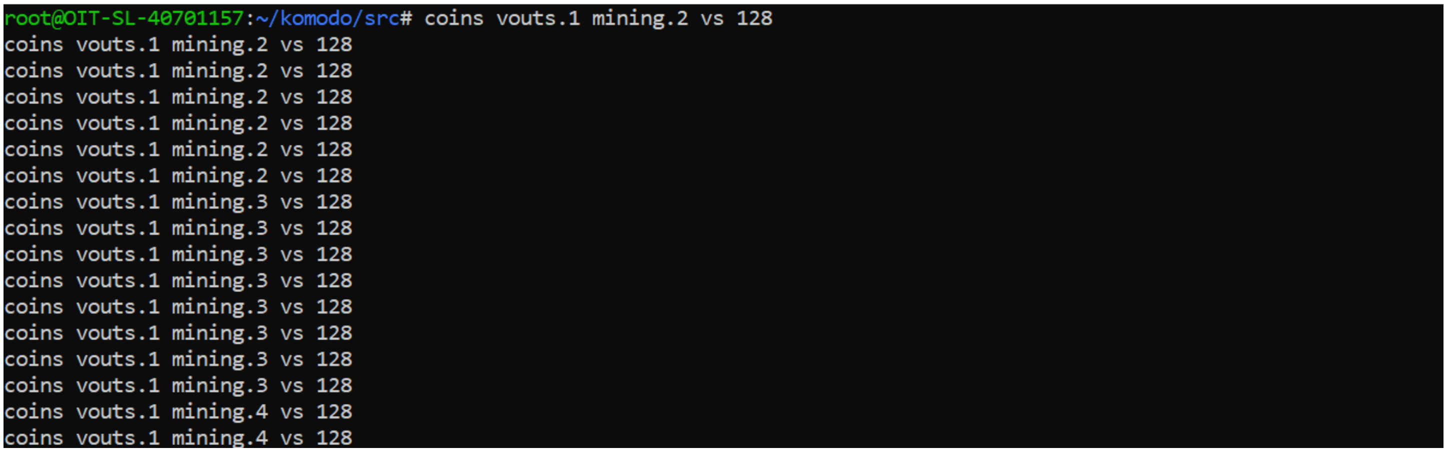


```
root@ubuntu:~/komodo/src# ./komodod -ac_name=coins  -ac_supply=20000 -addnode=19
2.168.204.1 &
[2] 4759
root@ubuntu:~/komodo/src# ASSETCHAINS_SUPPLY 20000
MAX_MONEY 2006320417438 20063.20417438
>>>>>>>>>> coins: p2p.11059 rpc.11060 magic.adfb1551 2918913361 20000 coins
call komodo_args.(./komodod) NOTARY_PUBKEY.()
initialized coins at 1594059607
nMaxConnections 384
```

Starting mining coin

In any od your two systems start mining using following command

Command : ./komodo-cli -ac_name=coins setgenerate true $(nproc)



```
root@OIT-SL-40701157:~/komodo/src# coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.2 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.3 vs 128
coins vouts.1 mining.4 vs 128
coins vouts.1 mining.4 vs 128
```

Generating new address and pubkey on both system

Execute following commands in order to get pubkey and address

```
newaddress=$(./komodo-cli -ac_name=coins getnewaddress)
pubkey=$(./komodo-cli -ac_name=coins  validateaddress $newaddress | jq -r '.pubkey' )
./komodo-cli -ac_name=coins  setpubkey $pubkey
```

After executing following command it will generate address which will be used in future in order to send and receive coins so note the address from both of the system.

Information for system1

```
{
  "ismine": "true",
  "address": "RQomeaQtYHkSshDs7sn6WqBhEEGG8nYsam",
  "pubkey": "02c9d5f21082ab3c801b9763b22739ca52fed07e7d63cff1697b8a529a73377027"
}
```

Information for system2

```
{
  "ismine": "true",
  "address": "RBmHwwtUQDdJmAbGT5XVKiSi7GhE9Zrfym",
  "pubkey": "02e93e778a57dc60504350ba028e728f430dc6e526d0671291c42fb06ab0d7c2c9"
}
```

The only important thing from above information is address so note it down.

Now after this step we are completely done with our setup of block chain on both of the systems. Now its time to switch gears to real application of this in a simple gambling game.

In order to play game both player should have money in their account but here initially we will just have coins in one account so transfer money to other account using following command in order to play the game.

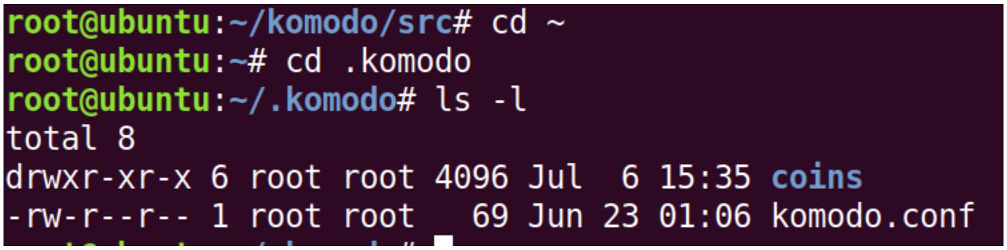
Command : ./komodo-cli -ac_name=coins sendtoaddress <reciepent address> 1000

Here in order to play game and in order to call RPC methods from our frontend we need rpc password, rpc username and rpc port.

Now follow the steps to get the required information to connect frontend with Komodo block chain. Here we have to take care to use a username and password of local machines only so make sure that your are entering a correct information extracted from local machine only.

Step 1) cd ~

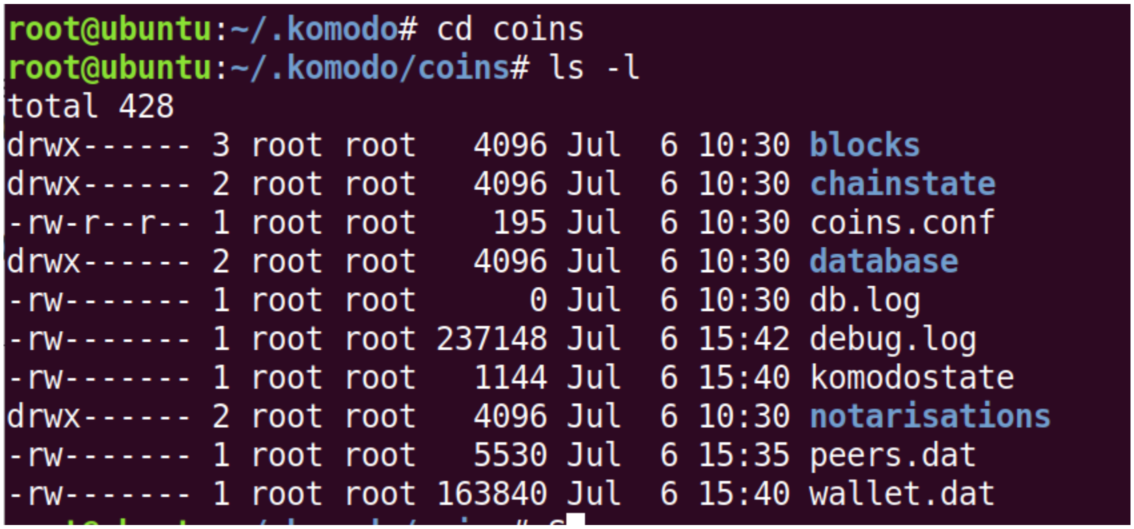
Step 2) cd .komodo



```
root@ubuntu:~/komodo/src# cd ~
root@ubuntu:~# cd .komodo
root@ubuntu:~/.komodo# ls -l
total 8
drwxr-xr-x 6 root root 4096 Jul  6 15:35 coins
-rw-r--r-- 1 root root   69 Jun 23 01:06 komodo.conf
```

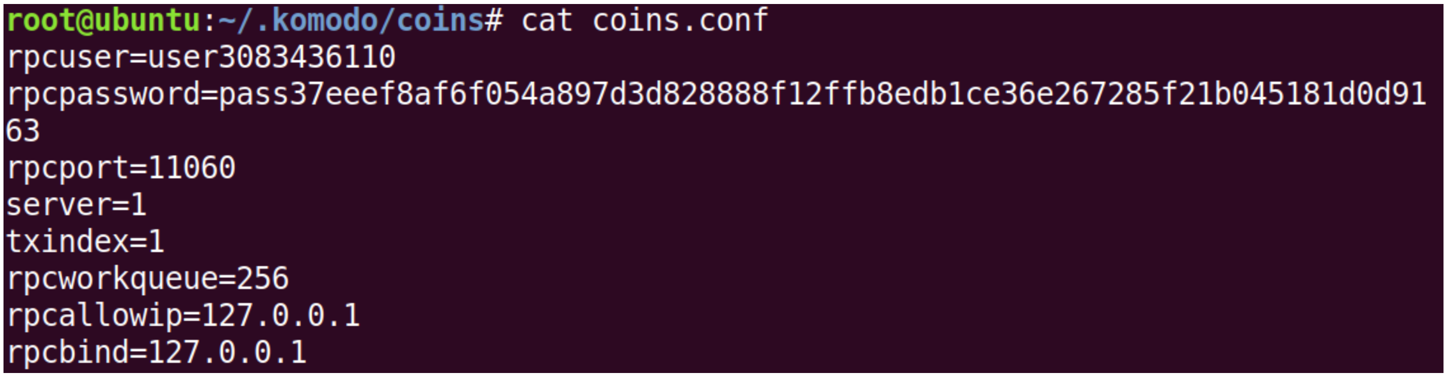
Step -3) Here we will have a directory having a same name as our block chain ac_name

Step -4) cd coins (go into that directory)



```
root@ubuntu:~/.komodo# cd coins
root@ubuntu:~/.komodo/coins# ls -l
total 428
drwx------ 3 root root   4096 Jul  6 10:30 blocks
drwx------ 2 root root   4096 Jul  6 10:30 chainstate
-rw-r--r-- 1 root root    195 Jul  6 10:30 coins.conf
drwx------ 2 root root   4096 Jul  6 10:30 database
-rw------- 1 root root      0 Jul  6 10:30 db.log
-rw------- 1 root root 237148 Jul  6 15:42 debug.log
-rw------- 1 root root   1144 Jul  6 15:40 komodostate
drwx------ 2 root root   4096 Jul  6 10:30 notarisations
-rw------- 1 root root   5530 Jul  6 15:35 peers.dat
-rw------- 1 root root 163840 Jul  6 15:40 wallet.dat
```

Step-5) now display the contain of coins.conf file using cat coins.conf



```
root@ubuntu:~/.komodo/coins# cat coins.conf
rpcuser=user3083436110
rpcpassword=pass37eeef8af6f054a897d3d828888f12ffb8edb1ce36e267285f21b045181d0d91
63
rpcport=11060
server=1
txindex=1
rpcworkqueue=256
rpcallowip=127.0.0.1
rpcbind=127.0.0.1
```

Login information for system 1

rpcuser=user2481781817

rpcpassword=pass452c0ddfd48c9dfff26c600fbd549eb7e93caed96709370d526f528875bfc300
78

rpcport=11060

address=RQomeaQtYHkSshDs7sn6WqBhEEGG8nYsam

Login information for system 2

rpcuser=user3083436110

rpcpassword=pass37eeef8af6f054a897d3d828888f12ffb8edb1ce36e267285f21b045181d0d91
63

rpcport=11060

address= RBmHwwtUQDdJmAbGT5XVKiSi7GhE9Zrfym

After this step we are completely done with the setup process now its time to run our simple betting module.

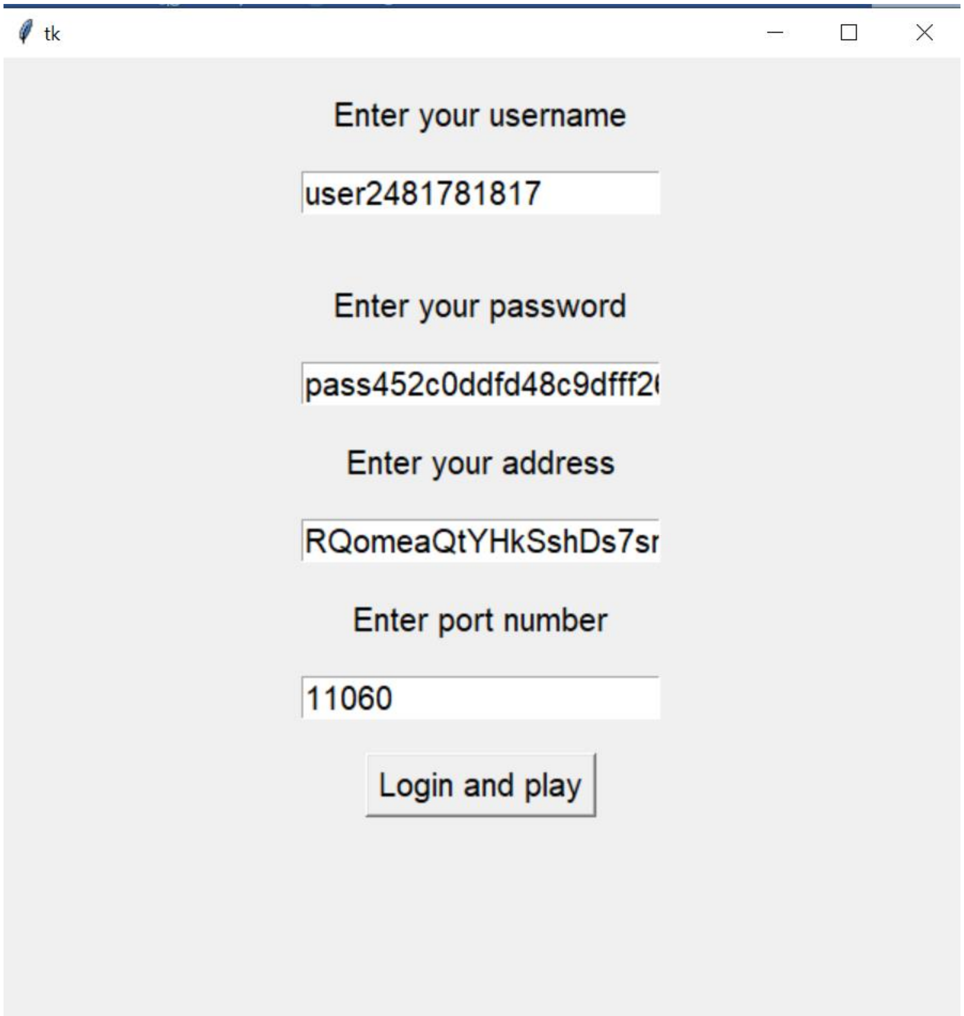
In a project folder there is three python file as follow

Dealer.py : This file is used to create or initiate a game. Whoever runs this file is master of game.

Player.py : This file is used to join the room created by player who run the file dealer.py. When we run this file it will ask for ip address here we have to write ip address of machine where the dealer.py file is running

Coins.py : This file made several calls to send and receive coins from our block chain

Now lets run **dealer.py** file first



Enter your username

user2481781817

Enter your password

pass452c0ddfd48c9dfff2

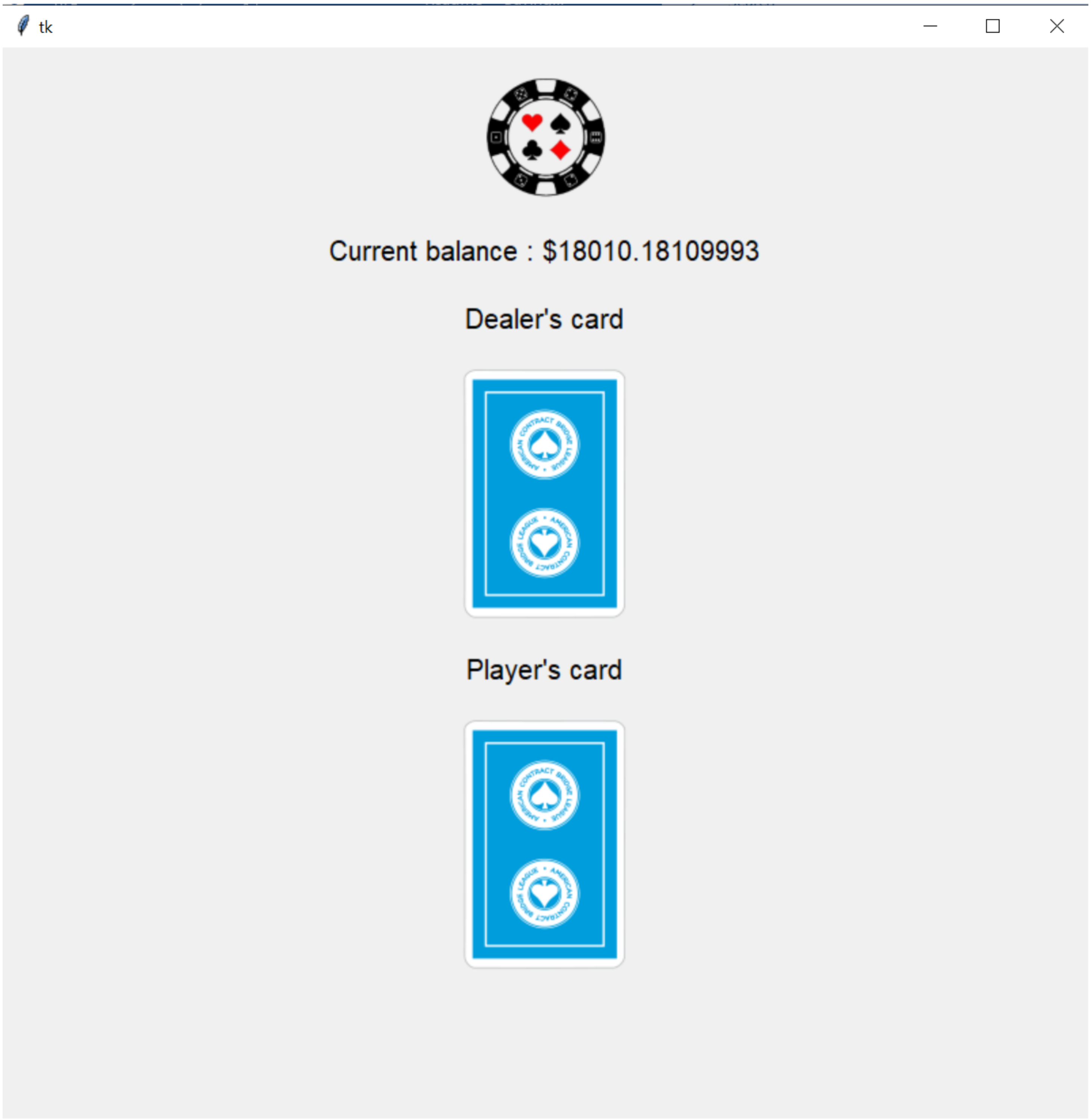
Enter your address

RQomeaQtYHkSshDs7sr

Enter port number

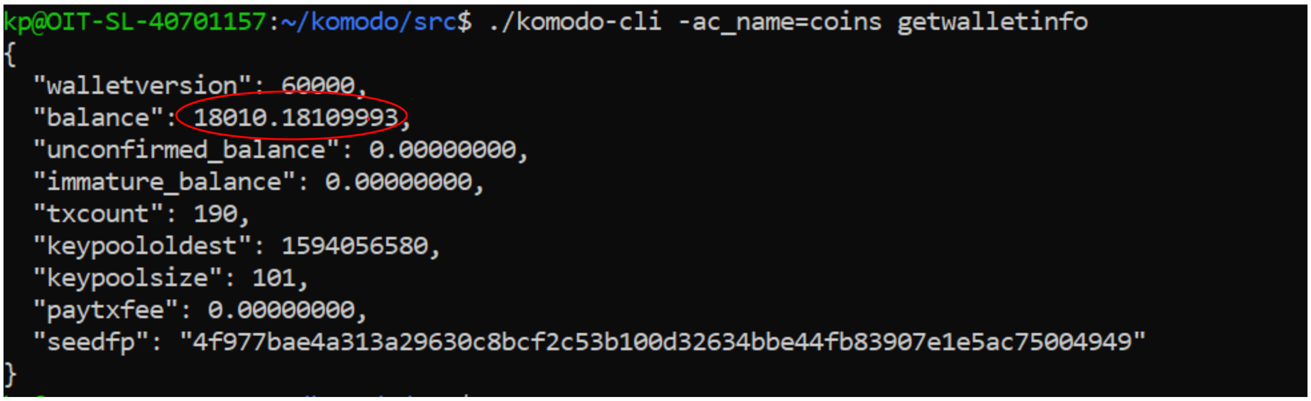
11060

First screen looks like this which will ask for login information



Current balance : $18010.18109993

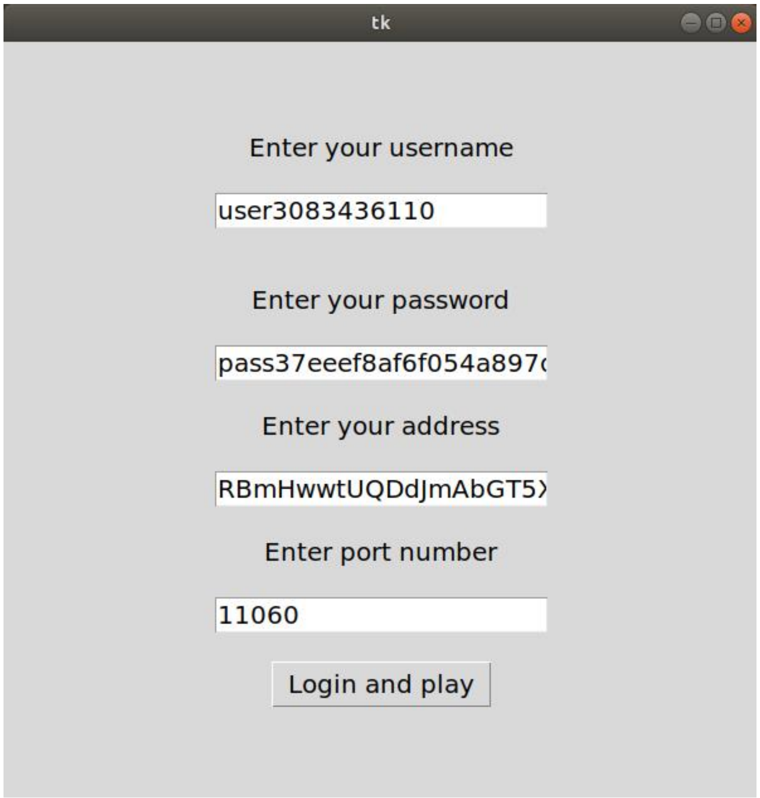
This is the main screen. Right now opponent player has not started the game so all the card are packed right now. Here current balance is latest current balance based on our block chain. We can verify this balance using normal komodo command as follow.



```
kp@OIT-SL-40701157:~/komodo/src$ ./komodo-cli -ac_name=coins getwalletinfo
{
  "walletversion": 60000,
  "balance": 18010.18109993,
  "unconfirmed_balance": 0.00000000,
  "immature_balance": 0.00000000,
  "txcount": 190,
  "keypoololdest": 1594056580,
  "keypoolsize": 101,
  "paytxfee": 0.00000000,
  "seedfp": "4f977bae4a313a29630c8bcf2c53b100d32634bbe44fb83907e1e5ac75004949"
}
```

Here as we can see that balance value at our GUI and terminal looks exactly same which represents our GUI and komodo block chain are connected to each other.

Now lets run **player.py**



Enter your username

user3083436110

Enter your password

pass37eeef8af6f054a897(

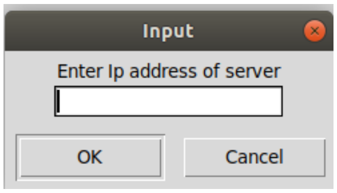
Enter your address

RBmHwwtUQDdJmAbGT5)

Enter port number

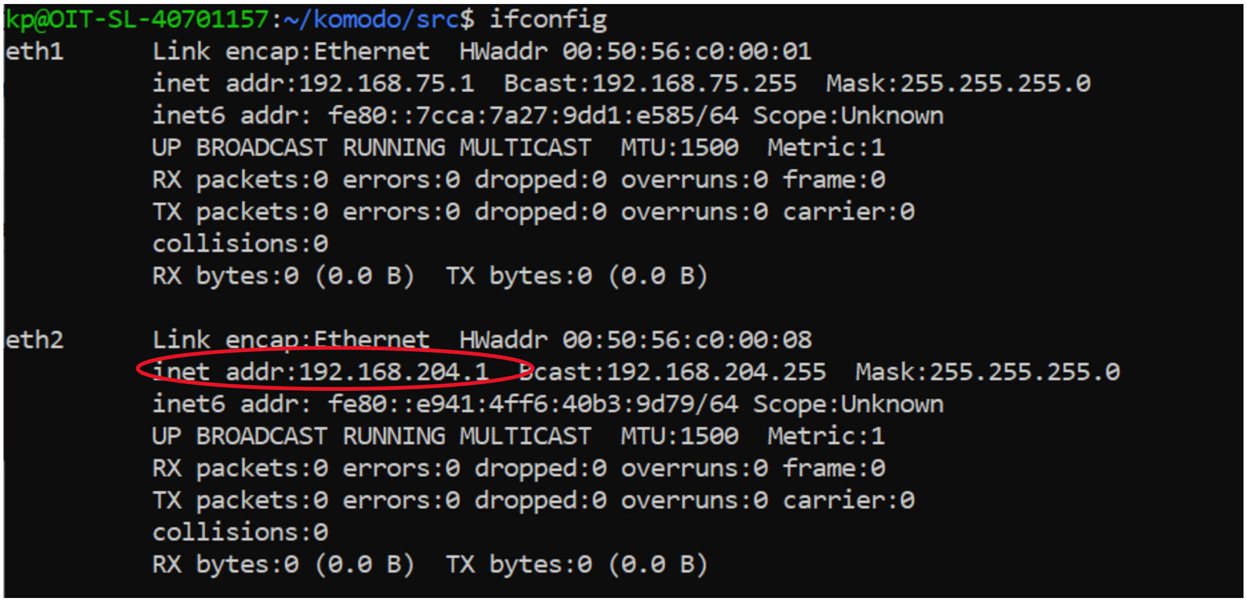
11060

As we have discussed it will ask for some login information first.



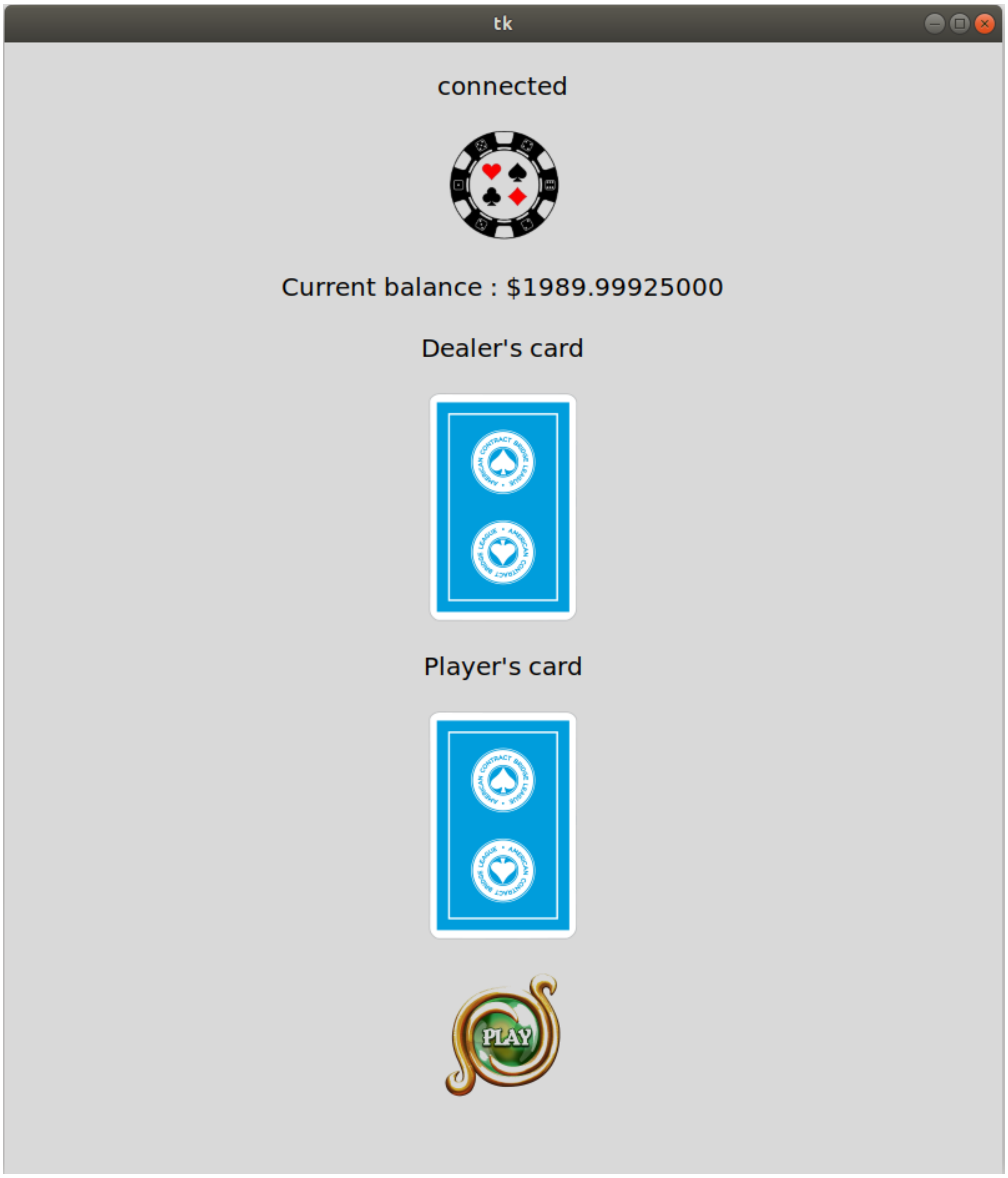
Enter Ip address of server

In second step it will ask for ip address here we have to write the ip address of system where dealer.py is running.



```
kp@OIT-SL-40701157:~/komodo/src$ ifconfig
eth1      Link encap:Ethernet  HWaddr 00:50:56:c0:00:01
          inet addr:192.168.75.1  Bcast:192.168.75.255  Mask:255.255.255.0
          inet6 addr: fe80::7cca:7a27:9dd1:e585/64 Scope:Unknown
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 00:50:56:c0:00:08
          inet addr:192.168.204.1  Bcast:192.168.204.255  Mask:255.255.255.0
          inet6 addr: fe80::e941:4ff6:40b3:9d79/64 Scope:Unknown
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```



tk

connected

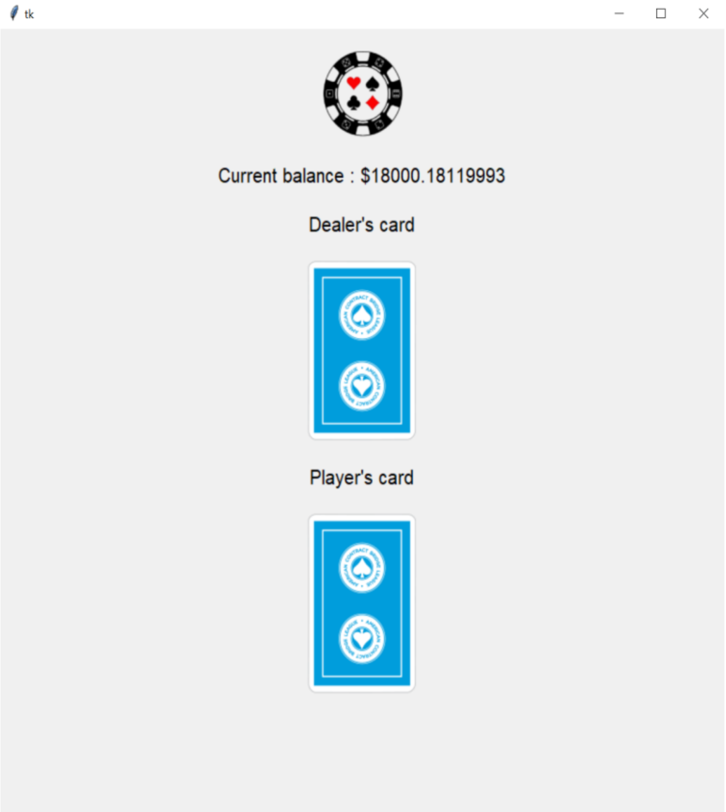
Current balance : $1989.99925000

Dealer's card

Player's card

After entering ip address we will get screen like this. Now if we want to play we can press play button which will randomly draw two card one for dealer and one for player.

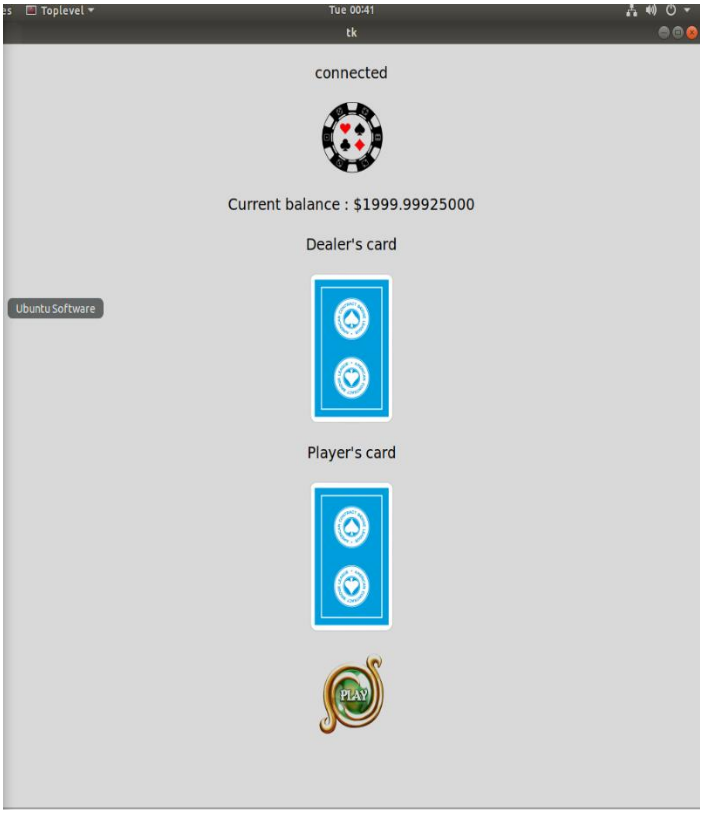
Before the play



Current balance : $18000.18119993

Dealer's card

Player's card



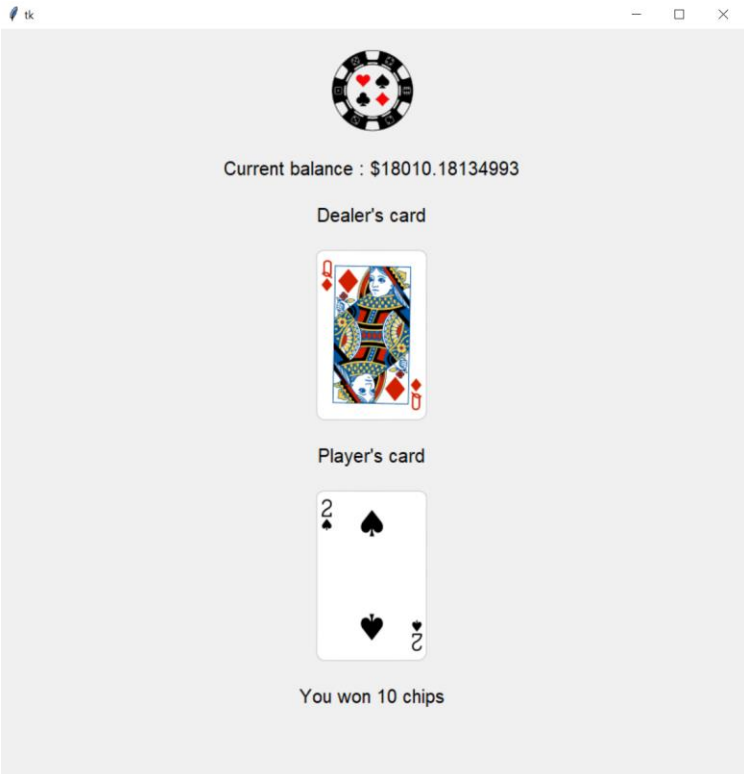
connected

Current balance : $1999.99925000

Dealer's card

Player's card

After one round of play

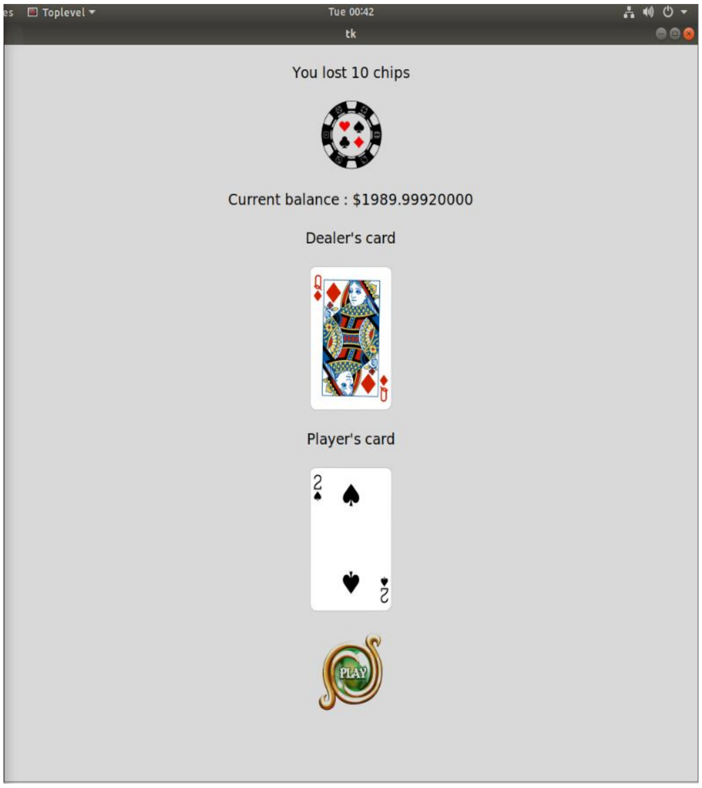


Current balance : $18010.18134993

Dealer's card

Player's card

You won 10 chips



You lost 10 chips

Current balance : $1989.99920000

Dealer's card

Player's card

After the successful draw of cards we can see that winner automatically gets 10 chips and 10 chips just automatically got deducted from other player’s account. Here we have to keep in mind that this balance is real time balance from our own block chain after playing play button we balance will not immediately reflects on the GUI it will take some time to confirm the transaction so we have to wait for 10-20 seconds for transaction to get reflected on screen and it is advisable to play another round after the balance get successfully reflected.

If you want to check weather this changes are perfectly reflected on our komodo block chain or not then you can run following command and check the balance on both of the system.

Command : ./komodo-cli -ac_name=coins getwalletinfo