

Artwork Trading Platform

Project Write-up

Member: Jing Jiang EE 20', Zheng Liu EE 20'

Background:

The industry of digital arts has developed greatly in the past decade as art production softwares and personal electronic devices became more and more accessible to everyone. The digital format of the artworks has many underlying meanings such as easy-to-transfer and duplicable. However, an artist, if he or she wants to protect the intellectual property of the work seriously, has to go through a registration process with the Office of Copyright, which is now online but still time consuming. On the other hand, internet also has its own looser rules, which usually can be found in 'terms of use', to regulate the use of others' words, code, pictures, music, etc. Although there are different licences, the general standard is that if a permission is not granted, people can not use other's intellectual property either personally or commercially. For example, instagram is a big name in social medias. It is now used not only for sharing people's life, but also for social media marketing. There are many accounts steal others' personal posts and repost in order to attract more followers and promote their product. In practice, if a person saves a post on instagram and use it as his or her wallpaper, it will be no harm. The real problem is the commercial uses as described in the example. A copyright infringement is often not admitted because it is hard to prove the ownership and a lack of permission. Copyright thieves usually argue that the artworks are originated by themselves or they have got permission from the creator to use them, even with forged paperwork.

A big part of the reason of this situation is the lack of a system of trading. Our platform will fill the blank. The importance of the platform is firstly that it builds up the market and awareness of copyright. Blockchain provides trust for the participants as it keeps all the transactions in a distributed ledger. Also, at this stage, because there is no existing traditional systems, our blockchain solution does not have to worry about how to conform with the traditional platforms like many other applications have to deal with. Blockchain's natural advantages and low cost of maintenance will make a good chance for it to prevail. It is better to start now. Speaking of laws, the platform does not replace the current registration system of copyrights but cooperate with it. The platform can keep a clear history of tradings related to an artwork to help tell what is legal and what is not. It will be a good start of the simplification of protecting copyright and a practical approach of digital artwork trading.

Related work:

There is no existing blockchain serves the same purpose of our project as stated in the background section. We recently know that a startup called monograph is entering the business. One of the most successful projects that are similar and related to ours is CryptoKitties. CryptoKitties lives in Ethereum. Artists can create designs of kittens with different colors, decorations, gestures, etc. Each design will only have a small number of copies. Because of that, digital artworks on CryptoKitties has scarcity like physical arts. The purpose of CryptoKitties is very different from our project. It gives value to digital artworks by scarcity, which makes it more like a financial game because there is truly no scarcity in the digital world as everything could be duplicated without damage. The source code of CryptoKitties is in the public domain and can be studied to benefit our design.

Brief introduction to our solution:

As said earlier, trading copyright with blockchain does not automatically give as strong protection as by law but it has its social effect. If an artist manages the copyright of his artworks by blockchain, it will be clear that he owns it and to whom he gives permission to use.

Our project is built with Ethereum as its platform. The whole application live on a smart contract. Copyright is the good in our project and there are various ways of trading it with different smart contracts. We did not use token, which usually is the approach to deal with tradable items on Ethereum because of some unsolvable bugs at the time and the intention was to reduce complexity. Without the charisma of tokens, Ethereum still gives us a lot of reasons to choose it, especially over Hyperledger. First it has ether as incentive. This can be the currency in the trades. An easy handling of double spending issue with the built-in mechanism called nonce comes with the system. Second, it is more friendly for UI building and interactions. And third, it can handle identities and servers easily with Metamask and Ganache. Ganache sets up a personal testing blockchain with set addresses and Metamask is the portal to interact with them. We used truffle for compilation and migration to link the front end, back end and the contract. These are relatively mature development packs with high stability and existing examples.

The users can buy copyright and rent copyright. Buying can be through a direct purchase or an auction. Buying a copyright means that the buyer will have complete ownership and can treat the work however he wants. Rent a copyright means that the copyright can be used for a certain amount of time. Rent is paid when the transaction is initiated, and the start and end time for the lease can be chosen by the client freely. The smart contract serves as the connection between sellers and buyers. Buyers pay money (ether) into the contract and the contract then pays the sellers. In auction, the seller sets a ground price and a duration. When they bid, they need to pay upfront the bidding value to the smart contract. The contract will keep the money

until a higher bid comes in or the auction ends. If there is a higher bid, the money goes back to bidder's account. When auction ends, the contract holds the money of the winner and then transfer to the art seller to complete the deal. Both of these types of deals are initiated by the owner, which means owner decides whether to put the artworks on the market and in which way they are open for trading.

We made the seller (artists) have more control and designed the bidding mechanism like that because current difficulties of handling frauds. The bidding system solves the problem of how to make sure one has enough deposit to pay for what he bids. This is the multiagent aspect of our project. In our setup with considerations of that, there is still one possible fraudulent move. A person could rent an artwork and keeps using it after the deadline has passed. This is a minor problem as our platform serves to keep track of copyrights rather than enforcing the rules, which could be an application of AI. However this reminds us of the experience of every trade and the necessity of a reputation system. Therefore, we made a rating system where sellers rate the buyers. It has minimal value for now concerning that human interventions could only be for the rent case and users can switch accounts very easily. The rating system would be more useful for more complex functions such as on-chain communications. And account setup could be made more expensive by a registration process that requires things like cell phone or Email to pair up with accounts.

We intended to make the network permissioned but soon realized that this is a public application and the public should have fair access to it. It can make the blockchain safer as a malicious node then has to compete with a enormous public network.

Roles:

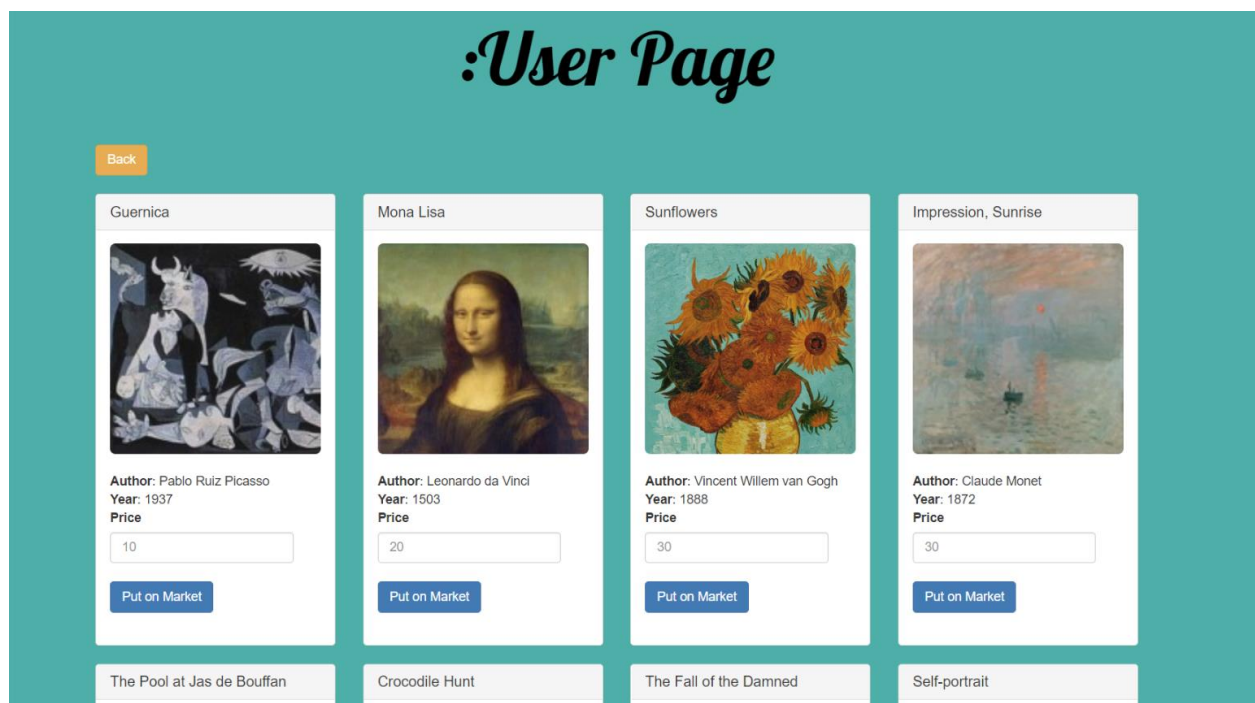
Jing Jiang is responsible for project planning and management of the project. Zheng Liu is responsible for website creation and maintenance and formation of design problems by researching the technical and social aspects of the project. In later stages of the project, Jing will be in charge of construction and Zheng will be in charge of test and evaluation. For actual implementations, specifically, Jing will focus on transactions initialized by sellers (artists) , including creation of artwork, set initial price, copyright open for rent, copyright open for sale, open for subscription, etc. Zheng will be focus on the buyer side and systematic processes, including bid, resell, start auction, close auction, cancel auction, etc.

Demonstration and Evaluation:

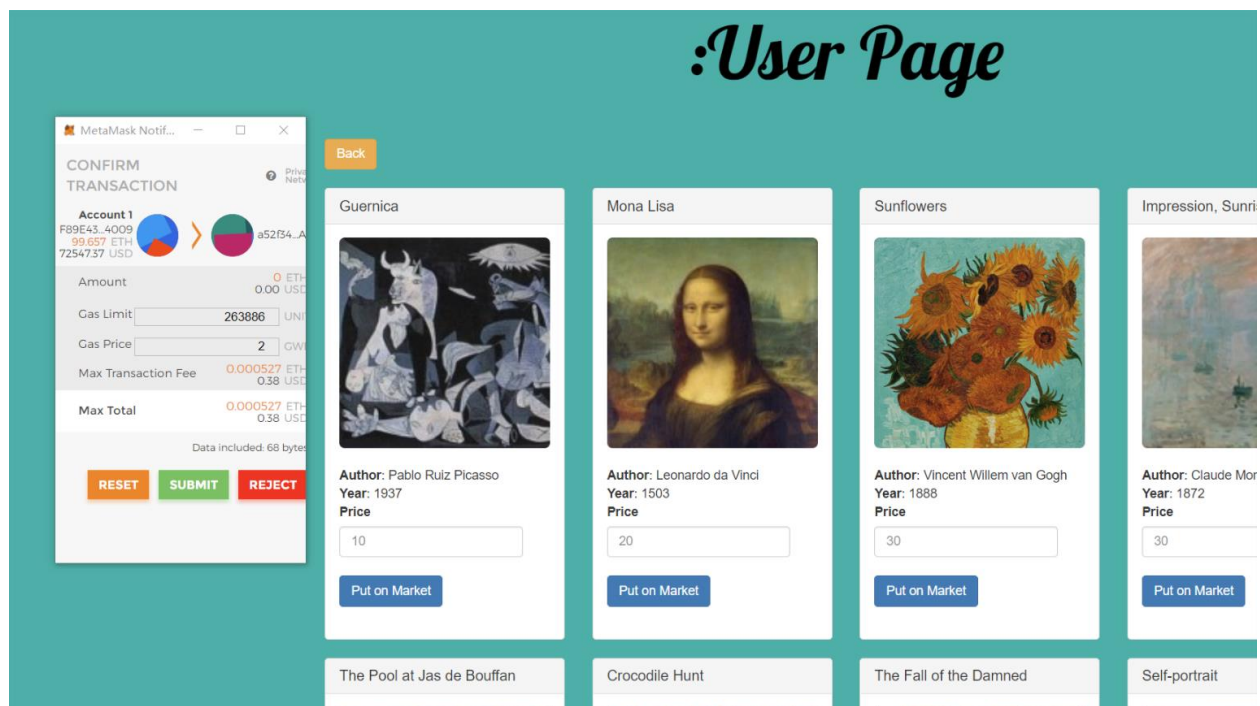
The demonstration contains two parts. The functions of submit art and buy art are demonstrated through our website. Given limited time, the functions of bid and rate art were not implemented on the front end. So we decide to demonstrate them through the Solidity IDE.



At first, there are no artworks for sell on the main platform because no one has uploaded yet.



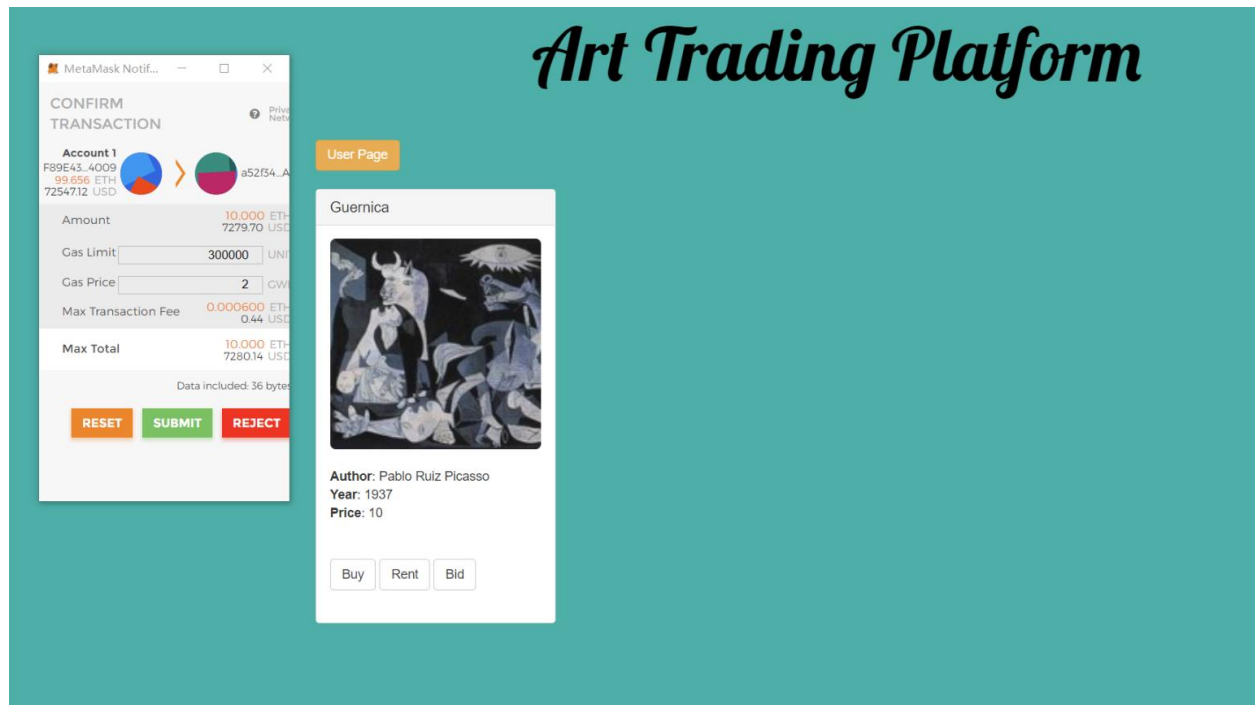
We have a preset user holding several artworks, as shown above. Now the user clicked “Put on Market” button.



Metamask shows the transaction window and the user clicked “Submit”.



Back to the main page, there is one artwork available for sell, as shown above.



The user hit “Buy”, and a metamask transaction window showed up. The user then clicked “Submit”.

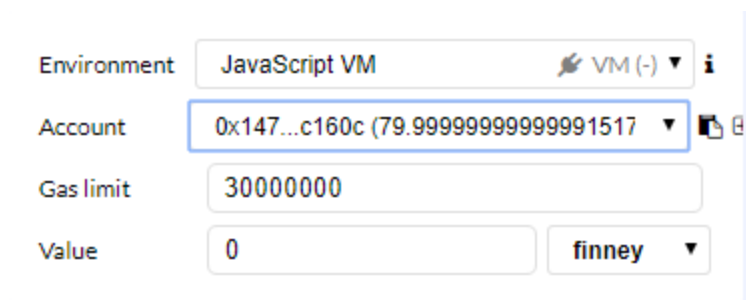


As shown above, the status of this piece of artwork becomes “Sold”.

Below is the demonstration of auction using Remix IDE:

<code>decoded input</code>	<pre>{ "uint256 ID": "0", "uint256 _price": "10000", "uint256 duration": "5" }</pre>
----------------------------	--

The auction is initiated. The auctioned item is at ID 0, start price 10000 and duration is 5 minutes.



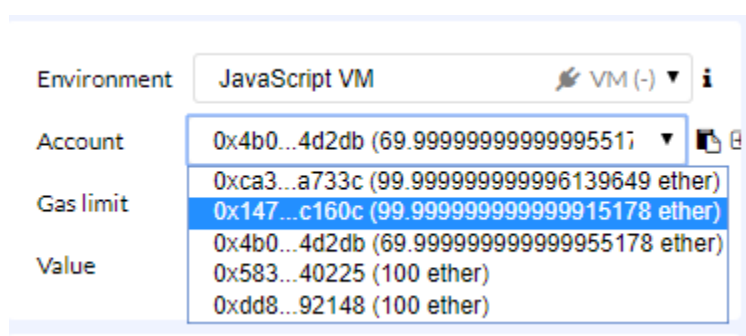
Environment: JavaScript VM

Account: 0x147...c160c (79.9999999999991517 ether)

Gas limit: 30000000

Value: 0 finney

The account (0x147...) submit a bid whose value is 20000 finney. (20 ether)



Environment: JavaScript VM

Account: 0x4b0...4d2db (69.9999999999995517 ether)

Gas limit: 0xca3...a733c (99.9999999999996139649 ether)

Value: 0x147...c160c (99.99999999999915178 ether)

0x4b0...4d2db (69.99999999999955178 ether)

0x583...40225 (100 ether)

0xdd8...92148 (100 ether)

Another person bid for 30000 finney (30 ether). We can see the third account paid the correct value (original amount 100). And the money goes back to account 2.

Unfortunately, there is a bug in the remix IDE that many people reported online about payable functions. We did not make the `claosAuction` function work for now. I have submitted a github issue to the group but no response yet.

Rating is done in a simple way. When users buy artworks, because they just pay the money, the score will automatically be 500 (max), and for other functions, the score is a user input.

```
[
  {
    "from": "0x0de37dce8154ce54d895bd16942c86d568ddb5fc",
    "topic": "0x31bc52c97b0c9df927d5bb961bc48c4257ac95b554b7cbc12b1a68919fdef5d4",
    "event": "Transfer",
    "args": {
      "0": "buy",
      "1": "1525862072",
      "2": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
      "3": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
      "4": "10000",
      "txType": "buy",
      "time": "1525862072",
      "from": "0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c",
      "to": "0x14723A09ACff6D2A60DcdF7aA4AFf308FDDC160C",
      "price": "10000",
      "length": 5
    }
  }
]
```

Environment	JavaScript VM	VM (-) ▼	i
Account	0xca3...a733c (109.9999999999964631 ether) ▼		
Gas limit	0xca3...a733c (109.999999999996463187 ether)		
Value	0x147...c160c (89.9999999999988782 ether)		
	0x4b0...4d2db (100 ether)		
	0x583...40225 (100 ether)		
	0xdd8...92148 (100 ether)		

Above we can see a buy is completed. And we use checkRating function.

decoded output	{
	"0": "uint256: 500"
	}

The returned value is the rating, 500, as expected.

Maintenance:

Our project is maintained on <https://github.com/jassaiy/Digi-Art-Platform>

Easy and Hard Parts:

The easy part for our project includes setting up the development environment, building smart contracts, and constructing the user's interface. However, we met huge difficulties in the interactions between the smart contracts and the front end. For example, if a transaction was put through on the blockchain, we should let the front end website know and update the data and

styles. That means we should have an ID management system to solve this problem. But we struggled in building such a system. Finally, we have to simplify the functions shown on the website in order to compromise with our goal. Also, we have to implement the simplest algorithm, in which we used the arrays to store IDs, as the solution to the interaction issue. Trying to solve this problem, we constructed data structures to pair IDs and artworks and did much research on JavaScript. Jing and I jointly worked on this problem and proposed several solutions. Besides interaction and ID managing, there are other difficulties including manipulating Ganache with Metamask, setting gas amount, and debugging the code.

Reference:

https://eco.copyright.gov	Office of Copyright
https://www.cryptokitties.co/	Official website of CryptoKitties
https://ethfiddle.com/09YbyJRfiI	Source code of CryptoKitties
https://help.instagram.com/478745558852511	Instagram: Term of use
https://solidity.readthedocs.io/en/v0.4.23/	Solidity 0.4.23 documentation
https://cryptozombies.io/	CryptoZombies

Schedule:

Get familiar with Ethereum framework and coding								
--	--	--	--	--	--	--	--	--

Analyzing the project and divide it into sub-tasks								
Building the project/Coding								
Test and debugging								
Update the project process on the website and github								
Finalizing documentation and deliverables								
4/9	4/12	4/16	4/19	4/23	4/26	4/30	5/3	5/7