NFT 盲盒demo documentation

For resources, refer to: <https://github.com/winkydes/NFT-Blind-Box-Smart-Contract>

References mostly from: <https://www.youtube.com/watch?v=3vWw9Xt48bs>

**Demo Highlights**

* 根據以下的步驟，你將可以host一個NFT的盲盒發售Event，並透過使用contract的function控制event的玩法

**事前準備**

* 確保chrome上已經安裝了metamask，以及到remix準備開發環境（如果要在local compile code，要先安裝remixd，再透過remix界面進行手動連接，詳情可以到 <https://remix-ide.readthedocs.io/en/latest/remixd.html?highlight=localhost>
* 安裝metamask以後，將metamask連接到Rinkeby測試網絡，並到<https://faucets.chain.link/rinkeby> 獲取0.4-0.5 ETH
* Clone了github repo的話記得run 一次 npm -i 先，確保下載了所有package

**Specification:**

* 本次blind box event有幾個要點：
  + 買家在blind box開箱之前購買NFT，該NFT應該顯示一張封面圖片
  + 活動結束，決定要開箱的時候，NFT的metadata要換成箱子裡的內容
  + NFT的內容決定了究竟最後可以有什麼折扣（需要做account verification）

**基本思路**

* 在opensea上看到的NFT，都一定需要相應的metadata（json file）去儲存該NFT的屬性，例如名稱，image source，描述…，而這些metadata和圖片需要放到pinata cloud，用ipfs 去存取。
* 範例的contract中有兩種URI，分別為notRevealedURI (public variable)和baseURI (private variable), 前者負責顯示盲盒的封面，而後者則為一個裝滿NFT image 的folder
* Contract 中有個叫flipReveal()的function, 可以將revealed 這個variable 由false變true，令一個叫tokenURI() function return不同的URI
* 這個URI將會決定opensea上面究竟會看到什麼NFT
* 當買家mint了一個盲盒，他將會獲得一個token，這個token將會決定開箱之後拿到哪一個NFT

**如何使用demo Contract**

1. **NFT resources setup**

* 首先去pinata註冊一個全新的賬戶，並完成手上的所有NFT
* 將所有NFT image 整合成為一個folder，將所有圖片的名稱以數字命名（由零開始）
  + 這個數字將會決定哪個token得到那個NFT file
  + 當然實際應該自己定義token的數值，就需要改contract中tokenURI的內容，所以這部分可以之後再做
* 一張含有 文字 的圖片

  自動產生的描述為每個NFT image 創建一個新的JSON file, 格式如下：
  + 這裡description是指在opensea上看到的NFT的描述。而image URI是透過pinata中image folder的CID 組成，格式為 ipfs://<folder CID>/<tokenID>.png . Name則為NFT的名稱。
  + 記得將所有相應的JSON file全部跟image file一樣命名（例如0.png的metadata應該以0.json來命名）
  + 當然這個metadata還可以加很多其他的attribute，不過為了簡化流程這次只使用了三個比較常用的field
* 將所有JSON 檔案整合做一個folder，上載到pinata cloud
* 製作盲盒的封面圖片，及為其製作JSON file，同樣上載到pinata cloud
* 到了這一步，pinata應該有四個directory，分別為NFT的圖片，NFT的metadata，盲盒封面圖片及其metadata。

1. **Remix contract setup**

* 請先從github clone repository，如果自己已經完成了step 1，則可以無視metadata的folder。
* 到remix的網頁界面，將remix connect至localhost（詳細請看上面setup）
  + 如果不想connect至localhost，都可以選擇直接從github拿code放到remix
* 打開Contracts folder，打開blindBoxContract.sol
* 這裡先解釋一下這個contract的基本資料：

一張含有 文字 的圖片

自動產生的描述

\_isSaleActive = 決定這個盲盒是否開始販賣，如果為true，則下面的mint function可以allow mint

\_revealed = 決定這個盲盒是否已經打開，如果為true，則所有盲盒都會打開

MAX\_SUPPLY = NFT的總數量（請確保跟pinata上的NFT數量一致）

mintPrice = NFT的價錢

maxBalance = 一個用戶最多可以擁有多少個同類型NFT

maxMint = 一個用戶一次最多可以Mint多少個NFT

一張含有 文字 的圖片

自動產生的描述  
Solidity 中的constructor是這個contract被deploy時只會run一次的code，主要用作initialize 所有基本variable，例如這裡就需要填寫initBaseURI。用戶可以到remix中的deploy界面輸入initBaseURI和initNotRevealedURI。這個URI是以ipfs://<JSON folder/file CID>/ 來填寫，記得一定要輸入metadata的URI，否則opensea上無法顯示資料。（Deploy的setup過程稍後會提到，現階段請先不要按deploy）

一張含有 文字 的圖片

自動產生的描述

這個mintKeithMeta的function主要用作檢查買家是否符合資格，例如\_isSaleActive是否為true，買家是否有send足夠的ETH等資料。如果所有條件符合，就會run \_mintKeithMeta()的 function，開始mint NFT。這個\_safeMint() 的function是openZepplin內建的contract的function。

* 由於我們這個contract是建基於openZepplin內的ERC721 contract來撰寫，所以可以使用入面一些function

一張含有 文字 的圖片

自動產生的描述

這個就是tokenURI的function。只要買家完成mint，那麼這個tokenURI function將會return一個URI，決定究竟買家買到哪一個盲盒。如果這裡\_revealed是false,那麼買家只能看到盲盒封面圖片，但如果\_revealed是false,每個tokenID都會透過這個function generate一個指向相應NFT的URI，從而達到盲盒的效果。

1. **Deploying a contract**

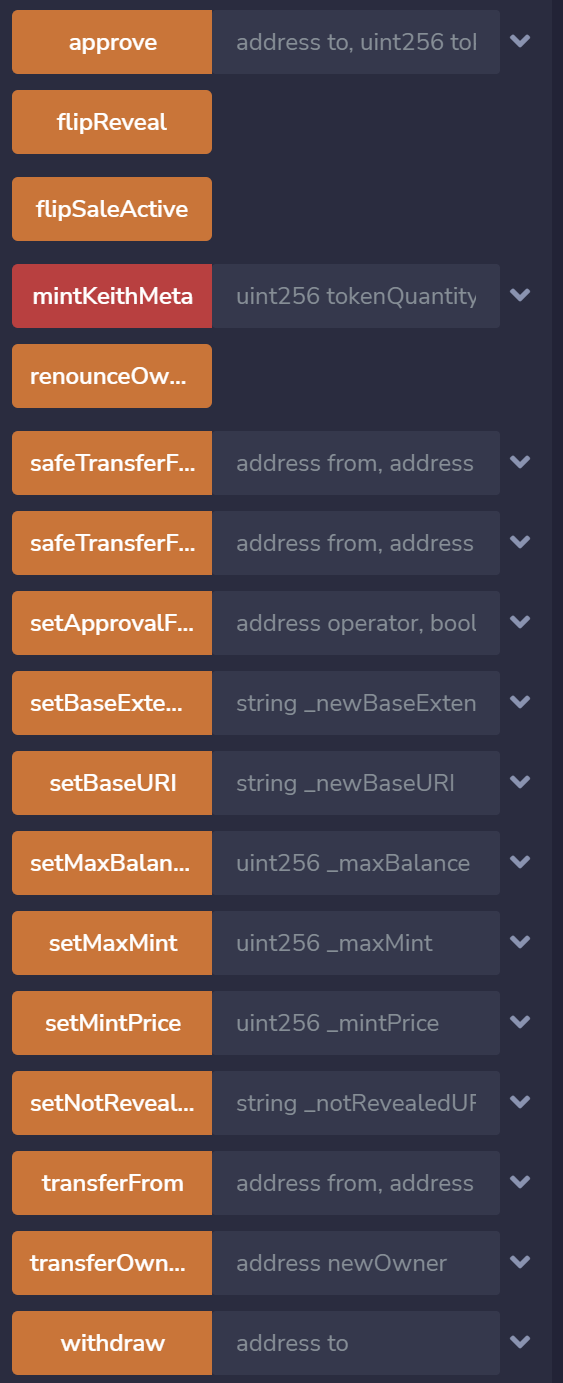
* 這部分將會解說如何Deploy contract:

一張含有 文字, 監視器, 螢幕擷取畫面, 螢幕 的圖片

自動產生的描述

這個是Deploy的界面（記得先compile之後再嘗試以下部分）：

1. 將Environment set 做injected web3
2. 將你的metamask wallet address 填入account的部分
3. Value是每次你進行transaction的時候send的eth數量，可以上網找ETH to GWei 的converter（因為填這部分不能填小數點，所以建議轉GWei做單位）
4. Contract部分記得選擇我們的blindBoxContract
5. Deploy 的按鈕旁邊點展開，將step 2 的initBaseURI 和initUnrevealedURI填入
6. 按Deploy，並支付deploy 的gas fee
7. 完成後應該在下方deployed contract 看到我們的contract，並能跟contract進行互動
8. 只要按了mint的按鈕（當然需要符合所有checking，並send 足夠的ETH才能成功mint），在opensea testnet上就能看見你的NFT。

一張含有 文字, 監視器, 螢幕擷取畫面 的圖片

自動產生的描述

這個是contract deploy之後的界面，會見到有兩種function可以操控（這是模擬實際contract上線後owner可以運行的function）所有橙色的按鈕都是需要消耗gas fee的function，會對contract的內容產生影響。而藍色的按鈕則是check variable的value，不需要消耗gas fee就能查看某個variable 的內容，方便進行debug。而紅色的按鈕就是需要支付ETH的function，所以記得充值足夠的ETH才來按這個按鈕。此時你也應該能找到這個contract的address，你也可以上etherscan.io查看這個contract的所有交易記錄。

此時如果想模擬買家mint NFT，可以使用紅色按鈕進行transaction，耐心等候幾分鐘，就能在opensea testnet上看到你的NFT已經加入你的個人profile內。

此時如果\_revealed是false的話，我們應該只能看到盲盒的封面。用remix將\_revealed變成true的話，只要transaction完成後到opensea 按一下refresh metadata，就應該能看到盲盒打開。

**附錄：注意事項**

這部分將會講述技術上有機會遇到的困難以及解決方法，提供參考：

Q: Deploy contract後，我發現contract有code需要修改，但是修改後卻無法見到改變？

A: Smart contract一經deploy之後，裡面的code就再無法修改，因為已經在blockchain運行中。所以任何時候請確保你的smart contract沒有任何error/bug，以及一定要使用testnet來進行debug （否則將消耗真正的ETH，增加成本）

Q: 我該如何查看IPFS內的檔案？

A: 只要用pinata 的IPFS gateway，貼上相應的CID和file path就能存取你在IPFS的檔案，方便進行debug. (格式：https://gateway.pinata.cloud/ipfs/<your CID>/)

Q: 為何在opensea testnet可以見到我的NFT卻沒有任何照片，描述等資料？

A: 請檢查你的baseURI和notRevealedURI是否指向正確的JSON file，而各JSON file內的image link是否正確（可以用上一題的格式到ipfs看看link是否有效）opensea上如果沒有顯示資料，代表metadata出錯

Q: 打開盲盒之後NFT沒有任何改變，為什麼？

A: 由於整個跟區塊鏈的互動都需要時間，請耐心等候幾分鐘再試試。記得要按NFT的refresh metadata的按鈕才能看到metadata的更新。如果都是沒有用，那麼請再檢查你的baseURI和JSON file有沒有出錯