하위 설계

1 스마트 컨트랙트

1.1 Struct Player

|  |  |  |
| --- | --- | --- |
| **Description** | 게임플레이어의 재화 관리를 위한 객체를 저장하는 부분. | |
| **type** | **식별자** | **description** |
| int | High\_score | 이전 게임에서의 최대 기록 |
| Int[3] | Item\_list | 보유중인 아이템의 개수 |
| int | reward | player의 재산 |

1.2 Player[] players

|  |  |
| --- | --- |
| **Description** | index값으로 접근할 수 있는 player데이터 |

1.3 Int[3] item\_price

|  |  |
| --- | --- |
| **Description** | 아이템 가격 |

1.4 GetFunctions

|  |  |  |
| --- | --- | --- |
| **Description** | index값으로 접근할 수 있는 player데이터 | |
| **type** | **식별자** | **description** |
| int | getHigh\_score | 이전 게임에서의 최대 기록을 가져오는 함수 |
| Int[3] | getItem\_list | 보유중인 아이템의 개수를 가져오는 함수 |
| int | getreward | player의 재산을 가져오는 함수 |

1.5 CreatePlayer

|  |  |  |
| --- | --- | --- |
| **Description** | 신규 유저일 경우 호출하는 함수 | |
| **type** | **식별자** | **description** |
| int | index | -1이 들어올 경우 신규 유저라는 의미 |

1.6 Update

|  |  |  |
| --- | --- | --- |
| **Description** | game이 끝나고 유저가 사용한 아이템, 점수들을 갱신하는 함수 | |
| **type** | **식별자** | **description** |
| int | index | 유저 구분을 위한 UID |
| int | score | 유저가 갱신한 점수 |
| Int[3] | Remain\_itemList | 사용하고 남은 아이템 개수 |

1.7 Send

|  |  |  |
| --- | --- | --- |
| **Description** | 다른 유저에게 선물하는 함수 | |
| **type** | **식별자** | **description** |
| int | from | 선물을 보내는 유저의 UID |
| int | Int to | 선물을 받는 유저의 UID |
| int | itemIndex | 선물하고자 하는 아이템의 번호 |
| int | itemCount | 선물하고자 하는 아이템의 개수 |

1.8 Buy

|  |  |  |
| --- | --- | --- |
| **Description** | 게임에 필요한 아이템을 구매하는 함수 | |
| **type** | **식별자** | **description** |
| int | index | 아이템을 구매하고자 하는 유저의 UID |
| int | itemIndex | 구매하고자 하는 아이템의 번호 |
| int | itemCount | 구매하고자 하는 아이템의 개수 |

2 Communication Module

2.1 Attribute

Rest통신에 필요한 절차를 구현한 클래스

|  |  |  |
| --- | --- | --- |
| **type** | **식별자** | **description** |
| CommunicationModule | instance | singleton패턴으로 구현 |
| string | Api\_key | 통신에 필요한 api key값을 저장 |
| string | walletAddress | 통신에 필요한 wallet address값을 저장 |

2.2 Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Return type** | | **식별자** | **description** |
| CommunicationModule | | getInstance | singleton패턴으로 구현 |
| int | | createPlayer | 생성된 UID를 반환하는 함수 |
| void | | update | 게임간 유저가 얻은 보상, 사용 아이템 정보를 갱신하는 함수 |
| **parameter** | | | |
|  | int | userIndex | 유저의 UID |
| int | score | 게임간 얻은 점수 |
| Int[] | itemRemain | 게임 후 남은 아이템 개수 |
| string | | getProperties | 유저의 재화 정보를 읽어오는 함수 |
| **parameter** | | | |
|  | int | userIndex | 유저의 UID |
| bool | | send | 선물 전달 함수 |
| **parameter** | | | |
|  | int | from | 선물을 보내는 유저의 UID |
|  | int | to | 선물을 받는 유저의 UID |
|  | int | itemIndex | 선물하고자 하는 아이템 번호 |
|  | int | itemCount | 선물하고자 하는 아이템 개수 |
| bool | | buy | 유저가 게임에 필요한 아이템을 구매하는 함수 |
| **parameter** | | | |
| int | | userIndex | 유저의 UID |
| Int[] | | itemCount | 구매하고자 하는 아이템의 개수 |

3 Properties

3.1 Attribute

Communication Module을 통해 받아온 데이터를 게임 모듈에서 사용하기 쉽게 가공하는 단계.

|  |  |  |
| --- | --- | --- |
| **type** | **식별자** | **description** |
| Properties | instance | singleton패턴으로 구현 |
| int | userIndex | 유저의 UID |
| int | reward | 유저가 가지고 있는 돈 |
| Int[] | itemList | 유저가 가지고 있는 아이템의 개수 |
| int | highestScore | 유저가 이전에 갱신한 최고 기록 |

3.2 Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Return type** | | **식별자** | **description** |
| Properties | | getInstance | singleton패턴으로 구현 |
| void | | update | 게임간 유저가 얻은 보상, 사용 아이템 정보를 갱신하는 함수 |
| **parameter** | | | |
|  | int | score | 유저가 게임에서 얻은 점수 |
| string | | getProperties | 유저의 재화 정보를 읽어오는 함수 |
| void | | send | 선물 전달 함수 |
| **parameter** | | | |
|  | int | to | 선물을 받는 유저의 UID |
|  | int | itemIndex | 선물하고자 하는 아이템 번호 |
|  | int | itemCount | 선물하고자 하는 아이템 개수 |
| bool | | buy | 유저가 게임에 필요한 아이템을 구매하는 함수 |
| **parameter** | | | |
|  | Int[] | itemCount | 구매하고자 하는 아이템의 개수 |
| int | | getUserIndex | 자신의 UID 반환 |
| Int | | getReward | 자신이 보유한 돈의 양 반환 |
| int | | getItemCount | 가지고 있는 아이템의 개수 반환 |
| **parameter** | | | |
|  | int | itemIndex | 조회하고자 하는 아이템의 번호 |
| int | | getHighestScore | 자신이 갱신한 최고 기록값 반환 |