**matgoClient.py 스크립트**

**A. global 변수**

|  |  |
| --- | --- |
| **objectTable** | ( index, 맞고 클라이언트 객체 )를 저장하는 맵( 딕셔너리 ) |

**#global 함수 – default (게임툴에서 스크립트에 요청하는 함수)**

|  |  |
| --- | --- |
| **cCallbackMakeClient**( index ): | 클라이언트 객체 생성 |
| **cCallbackOnReceive**( index, dataStr ): | 서버로부터 메시지를 받았을 때 툴에서 스크립트로 전달 |
| **cCallbackOnConnect**( index ): | 서버와 접속되었을 때 툴에서 스크립트로 알림 |
| **cCallbackOnDisconnect**( index ): | 서버와 접속 종료되었을 때 툴에서 스크립트로 알림 |
| **cCallbackGetIsPlaying**( index ): | 현재 게임중인지 여부 출력 |
| **cCallbackGetLocation**( index ): | 현재 위치(ex:채널, 게임방 등) 출력 |
| **printLastMsg**( index ): | 마지막으로 받은 메시지 출력 |

**#global 함수 - userGateway send func ( 유저 게이트웨이 send 함수)**

|  |  |
| --- | --- |
| **ReqConnect**( index, iSvcID, iGameID, iLastCloseTime,  iProgramVerMajor, iProgramVerMinor, iProgramVerSub,  iProtoVerMajor, iProtoVerMinor, iProtoVerSub,  iMemberInfoGameId, sMemberInfoAvatarUrl,  sMemberInfoNickName ): | 유저 게이트웨이 접속 요청 |
| **ReqChannel**( index ): | 채널 요청 |
| **ReqStore**( index ): | 상점 요청 |
| **ReqInfo**( index ): | 유저 정보 요청 |
| **ReqAutoJoin**( index, iCategoryID, iChannelID ): | 방 자동입장 요청 |
| **ReqLeaveGame**( index, iGameID, bKick ): | 방 퇴장 요청 |
| **ReqClose**( index ): | 유저 게이트웨이와 접속을 끊을 때 요청 |
| **ReqPing**( index, iSeq ): | Ping |

**#global 함수 - matgo send func ( 맞고 게임서버 send 함수 )**

각 함수는 주어진 객체에 따라서, 해당 객체의 멤버 함수를 호출 한다.

|  |  |
| --- | --- |
| **ReqGameReady**( index ): | 게임을 시작할 준비가 되었음을 알림 |
| **ReqGameInitGame**( index ): | 시작하기 버튼을 눌렀을 시, 게임 시작을 요청 |
| **ReqGameReset**( index ): | 게임 리셋을 요청 (안 쓰임) |
| **ReqGameStart**( index ): | (안 쓰임) |
| **ReqGameSelectSunCard**( index, nCard ): | 선 카드 선택 시, (nCard)선택을 요청 |
| **ReqGameSelectCard**( index, nCard ): | 손에 든 카드중 (nCard)를 낼것을 요청 |
| **ReqGameAskGoStopAns**( index, bGo ): | 고 / 스톱 결정 여부(bGo) 요청 |
| **ReqGameAskShakeAns**( index, bShake ): | 흔들지 결정 여부(bShake) 요청 |
| **ReqGameChoiceCardAns**( index, nCard, nType): | 여러장의 카드중 어떤카드(nCard)를 먹어올지 요청 |
| **ReqGameMoveTen**( index, bMove ): | 국진열끗을 쌍피로 옮길지 여부(bMove) 요청 |
| **ReqGameChongTong**( index, bYes ): | 총통인 경우 게임을 진행할지 여부(bYes) 요청 |
| **ReqGameEnd**( index ): | (안 쓰임) |
| **ReqResultOk**( index ): | 결과창의 ok버튼을 눌렀을 시의 요청 |
| **ReqGameAutoPlay**( index, bAuto ): | 자동치기 여부(bAuto) 요청. ( 자동치기 해제용으로 쓰임 ) |
| **ReqGameEmoti**( index, nType ): | 이모티콘 사용 요청 |
| **ReqExitReservation**( self, bExit ): | 나가기 예약(bExit) 요청 |
| **ReqGameTestPacket**( index, testInt, testString ): | 테스트 패킷 보내기 |

**#MatgoClient class**

|  |  |
| --- | --- |
| **index** | 클라이언트 인덱스 |
| **matgoGameID** | 게임 id ( index + 1 로 임의 생성) |
| **matgoGameIndex** | 게임방내에서의 인덱스 ( 먼저 들어온 유저 1, 나중 들어온 유저 2 ) |
| **turnCount** | 현재 턴수 |
| **userTurnState** | 패를 낸 상태 ( 0 : 안 냈다, 1: 일반 패를 냈다, 2: 보너스 패를 냈다 ) |
| **isPlaying** | 현재 게임중인지 여부 |
| **openCardList** | 바닥 패 리스트 (현재 미사용중) |
| **playerHandCardList** | 유저 손 패 리스트 |
| **playerGainCardList** | 유저 먹은 패 리스트 |
| **enemyGainCardList** | 상대 먹은 패 리스트 |
| **lastMsg** | 마지막으로 받은 메시지 |
| **\_\_init**\_\_( self, \_index ): | 클래스 생성자 |
| **onConnect**( self ): | 접속 성공시 툴로부터 올 콜백 |
| **onDisconnect**( self ): | 접속 해제시 툴로부터 올 콜백 |
| **getIsPlaying**( self ): | 현재 플레이 여부 리턴 |
| **getLocation**( self ): | 현재 위치 리턴 |

**#userGateway rcv** **func** **( 유저 게이트웨이 관련 rcv callback )**

|  |  |
| --- | --- |
| **AnsConnect**( self, iSvcID, iGameID, iStatus, iArg1, iArg2 ): | 접속 성공시 콜백. |
| **ErrConnect**( self, iErrorCode, sErrorInfo ): | 접속 실패시 콜백 |
| **AnsChannel**( self, listChannelInfo ): | 채널 요청 성공시 콜백 |
| **ErrChannel**( self, iErrorCode, sErrorInfo ): | 채널 요청 실패시 콜백 |
| **AnsStore**( self, listItemInfo ): | 상점 요청 성공시 콜백 |
| **ErrStore**( self, iErrorCode, sErrorInfo ): | 상점 요청 실패시 콜백 |
| **AnsInfo**( self, iGameID, iGameIndex, sLevel, sMoney, iWinCnt,  iLoseCnt, iAllInCnt ): | 유저 정보 요청 성공시 콜백 |
| **ErrInfo**( self, iErrorCode, sErrorInfo ): | 유저 정보 요청 실패시 콜백 |
| **AnsAutoJoin**( self, iRoomID, iPointMoney, listGameUserInfo ): | 방 자동입장 성공시 콜백 |
| **ErrAutoJoin**( self, iErrorCode, sErrorInfo ): | 방 자동입장 실패시 콜백 |
| **AnsLeaveGame**( self, iGameID, iNType ): | 게임 종료 성공시 콜백 |
| **ErrLeaveGame**( self, iErrorCode, sErrorInfo ): | 게임 종료 실패시 콜백 |
| **AnsClose**( self ): | 유저 게이트웨이와 접속이 끊어질때 콜백 |
| **AnsPong**( self, iSeq ): | Ping에 대한 ans 콜백 |

**#gostop rcv function ( 맞고 게임 관련 rcv callback )**

|  |  |
| --- | --- |
| **RcvGameStartReady**( self, iGameID, iGameIndex ): | 게임 시작버튼을 표시 |
| **RcvGameinit**( self, iGameID, iGameIndex ): | 첫번째 판인 경우, 게임의 초기화(선고르기)를 알림 |
| **RcvGameStart**( self, iGameID, iGameIndex ): | 첫번째 판인 경우, 게임의 시작을 알림 |
| **RcvGameReset**( self, iGameID, iGameIndex ): | 두번째 이후 판인 경우, 게임의 시작을 알림 |
| **RcvGameNotiSun**( self, iGameID, iGameIndex ): | 유저(iGameID)가 선 임을 알림 |
| **RcvGameSelectSun**( self, iGameID, iGameIndex, iSelectIndex ): | 유저가 선카드를(iSelectIndex) 선택했음을 알림 |
| **RcvGameNotiSelectSun**( self, iGameID, iGameIndex, iListCard ): | 사용하지 않는다. |
| **RcvGameSharing**( self, iGameID, iGameIndex, iListOpenCard,  iListHoldCard ): | 패 분배 ( 바닥 패, 유저 손 패 ) |
| **RcvGameNagari**( self, iNagariType, iListCard ): | 나가리를 알림 |
| **RcvGameChongTong**( self, iGameID, iGameIndex, iListCard ): | 총통을 알림 |
| **RcvGameSelectCard**( self, iGameID, iGameIndex, iListCard ): | 유저가 패(iListCard )를 냈음을 알림 |
| **RcvGameBonus**( self, iGameID, iGameIndex, iCard, iCard2 ): | 유저가 보너스패(iCard)를 내고, 새 패(iCard2)를  얻어감을 알림 |
| **RcvGameTakePee**( self, iGameID, iGameIndex, iCard ): | 유저가 상대 먹은패에서 특정패(iCard)를 뺏어감을  알림 |
| **RcvGameBbuck**( self, iGameID, iGameIndex, iListCard, iCnt ): | 유저가 뻑임을 알림 |
| **RcvGameSeriseBbuck**( self, iGameID, iGameIndex, iListCard,  iCnt, sMoney ): | 유저가 연뻑임을 알림 |
| **RcvGameBbuckEat**( self, iGameID, iGameIndex, iCardMonth,  iBBuckType ): | 유저가 뻑을 먹음을 알림 |
| **RcvGameBomb**( self, iGameID, iGameIndex, iCardMonth, iCnt ): | 유저가 폭탄패를 낸 것을 알림 |
| **RcvGamePan**( self, iGameID, iGameIndex ): | 유저가 판쓸인 것을 알림 |
| **RcvGameDadak**( self, iGameID, iGameIndex, iCardMonth ): | 유저가 따닥인 것을 알림 |
| **RcvGameJjok**( self, iGameID, iGameIndex, iCardMonth ): | 유저가 쪽임을 알림 |
| **RcvGameUpset**( self, iGameID, iGameIndex, iCard ): | 유저가 카드(iCard)를 뒤집었음을 알림 |
| **RcvGameAllEatUp**( self, iGameID, iGameIndex, iListCard ): | 유저가 카드(iListCard)를 먹어옴을 알림 |
| **RcvGameNotiTurn**( self, iGameID, iGameIndex ): | 유저의 턴임을 알림 |
| **RcvGameMoveTen**( self, iGameID, iGameIndex ): | 유저에게 국진열끗을 쌍피로 옮길지 여부를 물음 |
| **RcvGameMoveTenAns**( self, iGameID, iGameIndex, bMove ): | 유저가 국진열끗을 쌍피로 옮김을 알림 |
| **RcvGameChoiceCard**( self, iGameID, iGameIndex, iListCard, iType): | 유저에게 카드중 어떤 것을 먹을지를 물음 |
| **RcvGameShake**( self, iGameID, iGameIndex, iListCard ): | 유저에게 흔들지 여부를 물음 |
| **RcvGameShakeAns**( self, iGameID, iGameIndex, iListCard, bShake): | 유저가 흔들었음을 알림 |
| **RcvGameShakeCount**( self, iGameID, iGameIndex, iCnt ): | 유저의 흔든 횟수(iCnt)를 알림 |
| **RcvGameCapType**( self, iGameID, iGameIndex, iJocboState ): |  |
| **RcvGameGodori**( self, iGameID, iGameIndex, iJocboState ): |  |
| **RcvGameKusa**( self, iGameID, iGameIndex, iJocboState ): |  |
| **RcvGameRed**( self, iGameID, iGameIndex, iJocboState ): |  |
| **RcvGameBlue**( self, iGameID, iGameIndex, iJocboState ): |  |
| **RcvGameBak**( self, iGameID, iGameIndex, iBakType ): |  |
| **RcvGameBakFail**( self, iGameID, iGameIndex, iBakType ): |  |
| **RcvGameNotiMission**( self, iMissionMulti, iListCard, sTitle,  iMissionType ): | 미션 정보를 알림 |
| **RcvGameNotiMissionCard**( self, iGameID, iGameIndex, iListCard): | 유저가 미션 카드를 먹음을 알림 |
| **RcvGameMissionState**( self, iGameID, iGameIndex, sMoney,  iMissionState ): | 미션 상태를 알림 |
| **RcvGameGostop**( self, iGameID, iGameIndex, iCnt, sMoney ): | 유저에게 고/스톱 여부를 물음 |
| **RcvGameGostopAns**( self, iGameID, iGameIndex, iCnt ): | 유저가 고를 선택했음을 알림 |
| **RcvGameStopAns**( self, iGameID, iGameIndex ): | 유저가 스톱을 선택했음을 알림 |
| **RcvGameResult**( self, gameResult ): | 게임 결과를 보여준다 |
| **RcvGameNotiPoint**( self, iGameID, iGameIndex, iPoint ): | 현재 점수를 알림 |
| **RcvGameEnd**( self ): |  |
| **RcvGameKick**( self, iGameID, iGameIndex, iKickType ): |  |
| **RcvAnsGameEmoti**( self, iGameID, iGameIndex, iEmotiType ): |  |
| **RcvAnsGameAutoPlay**( self, iGameID, iGameIndex, bAuto ): |  |
| **RcvAnsExitReservation**( self, iGameID, iGameIndex, bExit ): |  |