

## Protocol Team:

Madhulika Indukuru

Daanyaal Syed

Jonathan Young

## CONSTANTS

100-level codes are reserved for client messages.

200-level codes are reserved for server messages.

300-level codes are reserved for unpaired messages..

### **Request Constants**

CMSG_AUTH	101
CMSG_DISCONNECT	102
CMSG_REGISTER	103
CMSG_CREATE_CHARACTER	104
CMSG_CHAT	105
CMSG_MOVE	106
CMSG_ATTACK	107
CMSG_HEALTH	108
CMSG_CONTROL_POINT_STATE	111
CMSG_CONTROL_POINT_CAP	112
REQ_HEARTBEAT	301

### **Response Constants**

SMSG_AUTH	201
SMSG_DISCONNECT	202
SMSG_REGISTER	203
SMSG_CREATE_CHARACTER	204
SMSG_CHAT	205
SMSG_MOVE	206
SMSG_ATTACK	207
SMSG_HEALTH	208
SMSG_RESOURCE	209
SMSG_CONTROL_POINT_STATE	211
SMSG_CONTROL_POINT_CAP	212
SMSG_RENDER_CHARACTER	310
SMSG_REMOVE_CHARACTER	311
SMSG_SPAWN_GUARDS	312
SMSG_DESTROY_NPC	313
SMSG_SPAWN_GOLEMCP	321
SMSG_DESTROY_GOLEMCP	322
SMSG_SPAWN_GOLEM_NPC	323
SMSG_GOLEM_PIECE	324

Client Side Protocol (Requests)

Type and usage	Format
<b>RequestLogin</b> Client requests to login with a username and password. The server validates this and responds with ResponseAuth.	Short Constants.CMSG_AUTH String Username String Password
<b>RequestLogout</b> Client wishes to log out from the game. No more requests are to be sent after this. The server will respond with ResponseDisconnected, and server will update other users with ResponseRemoveUser.	Short Constants.CMSG_DISCONNECT String message
<b>RequestRegistration</b> Client registers a new account with the server which includes a username and password. The server validates this and responds with ResponseRegistration.	Short Constants.CMSG_REGISTER String Username String Password
<b>RequestCharacterCreation</b> Client wishes to creates a new character tied to their account which includes the character name and the faction which their character is loyal to. The server will respond with CharacterCreationResponse.	Short Constants.CMSG_CREATE_CHARACTER String characterName String factionId int classType
<b>RequestChat</b> The client chats to other clients through the chat function. The client uses an int for which faction that user belongs to and a String for the message being sent.	Short Constants.CMSG_CHAT int factionId // which faction to broadcast to String message
<b>RequestMove</b> Client issues a change to their location, and isMovingflag. This is used when a client wishes to move or stop moving. It is followed by creating a number of ResponseMove and Server will update other users with these ResponseMove	Short Constants.CMSG_MOVE Float x //location vector Float y Float z Float h //facing direction int isMoving // 0 or 1
<b>RequestHeartbeat</b> Client's state has not changed, but enough time has passed that the client would like an update from the server. Each client now is a thread and associated with a specific username, when request comes, server knows which client is sending the heartbeat, so no user id is required. The server will be able to check the client's queued response and send all of them out to the client socket.	Short Constants.REQ_HEARTBEAT

<b>RequestCharacterAttack</b> When the client presses a button bound to a particular attack which is differentiated through attackId. The server responds by animating their attack through ResponseCharacterAttack.	Short Constants.CMSG_ATTACK int attackId // which attack am i using
<b>RequestCharacterChangeHealth</b> When the client receives or deals damage through an attack or receives healing, the health of each player affected is updated. The server responds by updating all health bars involved through ResponseCharacterChangeHealth.	Short Constants.CMSG_HEALTH int healthChange // can be positive or negative
<b>RequestSetControlPointState</b> When the client is interacting with a control point the control point changes colors based on if it's being taken or contested. The server responds with ResponseSetControlPointState.	Short Constants.CMSG_CONTROL_POINT_STATE int controlPointId // which control point int controlPointState // 1 red, 2 blue, 3 purple
<b>RequestControlPointCapture</b> When the client successfully takes a control point they begin to gain resources and have access to healing. The server responds with ResponseControlPointCapture.	Short Constants.CMSG_CONTROL_POINT_CAP int controlPointId int factionId

=====

## Server Side Protocol (Responses)

=====

Type and usage	Format
<b>ResponseRegistration</b> Affirmative response to RequestRegistration. flag = 1 means registration was successful.	Short Constants.SMSG_REGISTER int flag (0 invalid, 1 valid)
<b>ResponseAuth</b> Affirmative response to RequestLogin. It is followed by creating a number of ResponseRenderCharacter. All ResponseRenderCharacter will be queued up into all OTHER users' update queue.	Short Constants.SMSG_AUTH int flag (0 invalid, 1 valid)
<b>ResponseLogout</b> Client sends out request to disconnect. Server replies and removes user from current list also inform other clients to remove the user. Followed by creating a number of ResponseRemoveUser. All ResponseRemoveUser will be queued up into all OTHER users' update queue.	Short Constants.SMSG_DISCONNECT
<b>ResponseCharacterCreation</b> Affirmative response to RequestCharacterCreation. flag = 1 means that the character was successfully created.	Short Constants.SMSG_CHARACTER_CREATE int flag (0 invalid, 1 valid)
<b>ResponseRenderCharacter</b> Create one character in client game world representing one existing user who already logs in. The new character in the world will be associated with username.	Short Constants.SMSG_RENDER_CHARACTER String characterName int factionId
<b>ResponseRemoveUser</b> One client disconnects. Server informs other clients to remove the user in their game worlds.	Short Constants.SMSG_REMOVE_CHARACTER String characterName
<b>ResponseChat</b> The server responds to the chat function from client through Requeschat. The server uses an int for which faction that user belongs to and a String for the message being sent.	Short Constants.SMSG_CHAT String username int factionId String message

<b>ResponseCharacterMovement</b> server response a change to their location, and isMovingflag. This is used when server moves as the client wishes to move or stop moving. Server will update other users with the Requestcharactermove from the client.	Short Constants.SMSG_MOVE String username Float x //location vector Float y Float z Float h //facing direction int isMoving // 0 or 1
<b>ResponseCharacterAttack</b> The server tells all users that characterName must run the animation for attackId.	Short Constants.SMSG_ATTACK String characterName int attackId
<b>ResponseCharacterChangeHealth</b> The server increases or decreases characterName's health.	Short Constants.SMSG_HEALTH String characterName int healthChange
<b>ResponseControlPointState</b> The server informs all clients that a control point has changed state.	Short Constants.SMSG_CONTROL_POINT_STATE int controlPointId // which control point int controlPointState // 1 red, 2 blue, 3 purple
<b>ResponseControlPointCapture</b> When the server successfully takes a request from client, they begin to gain resources and have access to healing.	Short Constants.SMSG_CONTROL_POINT_CAP int controlPointId int factionId
<b>ResponseChangeResourcePoints</b> Modifies factionId's resource points by resourceAmount.	Short Constants.SMSG_RESOURCE int factionId int resourceAmount
<b>ResponseSpawnGuards</b> When the client successfully takes a control point five guards of type guardId spawn at control point controlPointId. They are aligned to factionId.	Short Constants.SMSG_SPAWN_GUARDS int controlPointId int factionId int guardId
<b>ResponseSpawnGolemControlPoint</b> Spawn the golem control point at locationId (each of the possible golem control point spawn locations have a locationId).	Short Constants.SMSG_SPAWN_GOLEMCP int locationId
<b>ResponseDestroyGolemControlPoint</b> Destroy the golem control point at locationId (locationId is needed in case there are multiple golem control points spawned).	Short Constants.SMSG_DESTROY_GOLEMCP int locationId
<b>ResponseSpawnGolem</b> Create the golem NPC. It should spawn at the control point identified by controlPointId.	Short Constants.SMSG_SPAWN_GOLEM_NPC int controlPointId
<b>ResponseGolemPiece</b> Adds a golem piece to factionId's golem piece counter (Golem spawns when number of golem pieces is 3).	Short Constants.SMSG_GOLEM_PIECE int factionId

<b>ResponseDestroyNPC</b> When the hit points of a certain NPC falls to zero, remove that NPC from the server.	Short Constants.MSG_DESTROY_NPC int npcId // which NPC to destroy
----------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------