

Milestone 1 Description

1. Game Description

This game is called Look and Guess. It's a multi-player game. Two player are assigned to the same team. Multiple teams competes to win the game for 5 rounds. Every round, the game uses NASA satellite view images that drops each team in a same random location and challenges them to work out where they are by inputting the latitude and longitude they guess. The score is calculated based on the distance between the actual position and guessed position. If the position a team guesses is the closet to the specified location, it will get the highest score in this round and vice versa. After 5 rounds, the game is terminated. The game server will calculate the final score for each team. The team with highest score wins the game. If a team doesn't have a choice within a limited time period, the team will get 0 in this round and be forced to continue the next round.

Teamwork is pretty important for this game. Two players in a team can discuss where is the location and make a most reasonable guess.

2. Design Description

Generally speaking, the final project can be split into three modules: app module, chat module and game module. The app module is mainly responsible for connecting to players, joining chatrooms, creating chatroom, joining games and creating games. The chat module is used for sending messages between players. The game module provides players with a user interface to play games.

The app module can get `InitUser` from other clients. Through the `InitUser`, the app module handles a series messages in the `common.message.init` package, like `ChatroomListRequest`, `ChatroomListResponse`, `InitUserInfoRequest`, `InitUserInfoResponse` and `Invitation2Chatroom`. Any request and response messages with the same prefix is a pair, which should be used together. Take getting chatroom list as an example. Suppose client A has connected to client B.

When A needs B's chatroom list, A will send a ChatroomListRequest to B. B has corresponding data packet algorithm to handle the message type ChatroomListRequest. What B does is initializing a message ChatroomListResponse embedded with its chatroom list and sending this message to A. After A gets this message, A will parse the message and get B's chatroom list. The discussion above is how request and response messages work in the app module.

The chat module is similar to HW08, so we are not going to repeat it any more. Here we are going to give more details about game module. The process of initializing a game through CommandRequest and Command Response which are both related to unknown data type is shown below:

1. The server sends a StartGame message, which is an unknown data type to clients, to all of connected clients;
2. When a client receives this unknown data type, it will send back CommandRequest to the server.
3. The server receives CommandRequest, then send CommandRequest with game's UI to the client.
4. The client gets corresponding DataPacketAlgo from CommandRequest and show the game's UI to user.