Jackie Chan
jgc11

Code Review 1

1. GameServer::processUnassignedClientMessage(...)

Line 62 of sample1.cpp contains an unnecessary line:

```
bool validInviteCode = gameSessions.find(inviteCode) != gameSessions.end();
```

Here, a variable is assigned which is only used in one ‘if’ statement. An alternative could be to replace:

```
if (validInviteCode) { ...
```

with:

```
if (gameSessions.find(inviteCode) != gameSessions.end()) { ...
```

This would remove some unnecessary code, making the function more concise while maintaining a similar level of readability.

2. class GameSession

The GameSession class seems to violate the Single Responsibility rule for classes, by maintaining current connections while also maintaining a GameObject. An alternative solution would be to abstract out unrelated functionality, such as

```
void receiveMessage(...);
```

into a separate GameSessionConnection class. Another alternative may be to implement GameSession as an interface, with more specialized classes implementing the functionality currently within GameSession.

3. GameServer::run()

The run() function seems well constructed, with a clear purpose for running and updating the game server. However, functionality responsible for getting updates for each game session could be extracted into a separate function, to make the purpose of the run function more clear. Furthermore, there doesn’t seem to be any exception handling for any possible session errors, although that may be handled elsewhere in the codebase.

4. addPlayer(...) Rock Paper Scissors hardcode

The addPlayer() function currently has hard-coded functionality presenting a Rock Paper Scissors game to the players, when the purpose of the project is to build a ‘game engine’ which

supports any game configuration provided by a JSON file, not only Rock Paper Scissors. Consider changing implementation to not only allow Rock Paper Scissors to be played.

5. addPlayer(...) Clarity

The addPlayer() function seems to have code responsible for starting a game, despite the name of the function being ‘addPlayer’, suggesting its responsibility is to add players to a game session. Consider separating game starting functionality into a different function, possible in the GameObject class.