

Technical challenges of Networking Games

Contents

Introduction	1
Networking Library	1
Server / Hosting	2
Scalability	2
Cheating	2
Databasing	2
Protocols	2
Security	2

Introduction

Networking in games has become a very large part of the gaming industry, it can bring a lot more appeal to a game, the ability to play with friends and share the experience. Networking does come with a good number of problems and forethought to incorporate it into the project. Some of the main issues I will try to cover in this document. You can then make an informed decision on if it's worth implementing.

Networking Library

The incorporation of a networking library would be highly advised as creating one from scratch would take a lot more manpower. It depends on the implementation but in nearly every case using a library is a much smarter idea.

The next question would be what library? There are a vast number of them, some of the main c++ libraries for networking would consist of:

- Boost.Asio (if you are using Boost)
- Asio (standalone)
- ACE
- SFML networking
- SDL_net

Just to name a few, so you would need to do some research to make sure the library you choose to go with has all the features you wish to use. It would be preferable if the library also had future support and good documentation.

Using a library will save a lot of time but it will still take a lot of time to learn the library and how to use it.

Server / Hosting

Depending on if the game is P2P or server based you might need to invest in servers which is another expense that will vary depending on location / size.

There are plenty host server hosting services that you could investigate.

Scalability

If you're making an indie game that is intended to be played by a few hundred people at most. You need to make sure you have a plan if the game gains a lot of popularity will the game be able to handle a large influx of people and if not, what will the plan be?

Cheating

One of the most notable problems that comes with networking is it almost always comes with cheaters. People trying to take advantage of the game for wins, enjoyment, popularity, there's plenty of reasons people cheat but how are you going to detect if people are cheating and how will you deal with it once you find out they are cheating.

Some popular solutions to this would be including an anti-cheat software, and maybe a live cheat review board that consists of some members that deal with reports and manually check to see if they spot anything. It's impossible to stop 100% of cheaters but minimizing the number of them will make the networked experience a lot more fun.

Some well known anti cheat softwares include:

- Easy Anti-Cheat
- Faceit Anti-Cheat
- BattlEye

Each one comes with different levels of protection and different levels of intrusion into the user's computer. Some people will not play the game if the anti-cheat is too invasive so be sure to take that into consideration.

Databasing

You will most likely need to keep track of the players playing the game and for that you will need to have a database to handle all the different users and potentially you can take information from the players to use as playtesting.

Protocols

There are a lot of internet protocols and you would need to decide what one you wish to use for the game, these will vary for the type of game. A lot of games will use a mixture of UDP and TCP protocols inside their games and they will be good enough for most games, but you need to make sure that you consider which one is best for the project.

Security

The players will potentially need to log in to the game and maybe they have items that they can sell for money so the security of the items and the players information is very important and is

Author: Aaron O'Neill (C00241596)

something that would need to be address when networking a game. The information would need to be encrypted and well protected. The specifics for that would need to be addressed in each specific implementation and is worth thinking about