

Redes

Jogos multijogador e comunicação em jogos através da *Internet*

Slides por: Gustavo Ferreira Ceccon (gustavo.ceccon@usp.br)





Este material é uma criação do
Time de Ensino de Desenvolvimento de Jogos
Eletrônicos (TEDJE)

Filiado ao grupo de cultura e extensão
Fellowship of the Game (FoG), vinculado ao
ICMC - USP

Este material possui licença CC By-NC-SA. Mais informações em:
<https://creativecommons.org/licenses/by-nc-sa/4.0/>



Objetivos

- Introduzir o conceito de *Internet* e redes
- Problemas de um jogo multijogador
- Sockets
 - ◆ TCP vs. UDP
- Segurança e perda de dados
- Arquiteturas
 - ◆ *Peer-to-peer* (P2P)
 - ◆ Cliente-Servidor



Índice

1. *Internet*
2. Problemas Frequentes
3. *Sockets*
4. Dados e Informação
5. Arquitetura



1. *Internet*



1. *Internet*

- Computadores
- Rede
 - ◆ Com/Sem fio
- Estrutura
- Roteadores / *Switcher* / *Hub*
- Servidores e Data Centers



1. *Internet*

→ Protocolos de Rede

- ◆ Garante uma “linguagem” entre os computadores
- ◆ Vantagens e Desvantagens
- ◆ TCP e UDP



1. *Internet*

→ *Internet*

- ◆ Zona de guerra
- ◆ Não segura
- ◆ Não confiável
- ◆ Vulnerável

→ Ataques



1. *Internet*

→ Nomenclatura

- ◆ *Ping - Latência*
- ◆ *Round-trip time*
- ◆ *Interpolation delay*

2. Problemas Frecuentes



2. Problemas Frequentes

- Singleplayer e Multiplayer
- Acompanhar os estados do outro lado
- Transformar dados crus em informação útil
- Perda de dados
- Segurança da informação
- Compatibilidade
 - ◆ 32 ou 64 bits, little endian ou big endian, sistema operacional

3. *Sockets*

3. *Sockets*

- Fazem comunicação entre servidor-cliente
- Ditam o protocolo usado (TCP ou UDP)
- Fazem a passagem de mensagem (pacote)
- Cliente - *Client*, faz (geralmente) 1 conexão
- Servidor - *Listener*, aceita **n** conexões de clientes

3. *Sockets*

- Geralmente usamos um endereço (IP) para conexão
- Podemos usar um nome e pedir o endereço (DNS)

3. Sockets

TCP	UDP
Reliable—monitors message transmission, tracks data transfer to ensure receipt of all packets	Unreliable—no concept of acknowledgment, retransmission, or timeout –
Ordered—buffering provisions to ensure correct order of data packets	Not ordered—data arrives in order of receipt
Heavyweight—dedicated connection, provisions for speed and congestion control	Lightweight—no dedicated end-to-end connection, no congestion control
Streaming	Datagram oriented
Heavy overhead	Light overhead
Lower speed	Higher speed

3. *Sockets*

- Xadrez
- Jogo de tiro
- *MMORPG* - WoW, Ragnarok
- Starcraft

4. Dados e Informação

4. Dados e Informação

→ Input do usuário

- ◆ Útil em alguns casos: FPS em LAN
- ◆ Ruim para outros: MMORPG
- ◆ Servidor sempre tem o controle

→ Informações (posição, vida etc.)

- ◆ Melhor aproveitada

4. Dados e Informação

→ Problemas de mandar informações

- ◆ Criar um protocolo
- ◆ Imagine um MOBA (LoL, Dota)
 - TCP sobre UDP
- ◆ Muitos dados
 - Usando UDP

4. Dados e Informação

- Usando UDP para mandar informações
 - ◆ Não é garantido que eles chegarão
 - ◆ Não é garantido a ordem
- Soluções
 - ◆ Mandar várias vezes ou garantir a chegada
 - ◆ Ordenar ou usar interpolação e extrapolação

4. Dados e Informação

- Esperar o servidor responder é horrível
 - ◆ Acaba com a experiência do jogador
- *Client-prediction*
 - ◆ Faz no cliente o que faz no servidor
 - ◆ Manda informação
 - ◆ Servidor decide se aceita ou não
 - Limite de distância - WoW e outros *exploits*

5. Arquiteturas

5. Arquiteturas

→ *Peer-to-peer* (P2P) Lockstep

- ◆ Espera pela resposta do outro lado
- ◆ Seguro e confiável

→ *Client-Server*

- ◆ Servidor cuida de tudo e ele que manda (mestre)
- ◆ Servidor passa para todos os *peers* as informações

4. Arquiteturas

Exemplo com Pong

Dúvidas?

Referências

Referências

- [1] Game Coding Complete, Fourth Edition (2012) - Mike McShaffry, David Graham
- [2] <http://gafferongames.com/networking-for-game-programmers/>
- [3] <http://gamedev.stackexchange.com/questions/249/how-to-write-a-network-game>
- [4] https://www.reddit.com/r/gamedev/comments/1tvbe0/is_it_just_me_or_is_networking_really_hard/
- [5] <https://docs.unrealengine.com/udk/Three/ReplicationHome.html>
- [6] https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking
- [7] <http://drewblaisdell.com/writing/game-networking-techniques-explained-with-pong/>
- [8] <https://gamedevelopment.tutsplus.com/tutorials/building-a-peer-to-peer-multiplayer-networked-game--gamedev-10074>
- [9] <https://docs.unity3d.com/Manual/UNetConcepts.html>
- [10] <https://unity3d.com/learn/tutorials/topics/multiplayer-networking>