

Networking

Jogos multiplayer e comunicação em jogos

Slides adaptados por: William Quelho Ferreira (william.quelho.ferreira@usp.br)

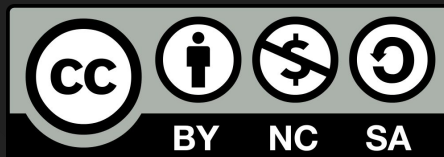
Slides originais: Gustavo Ferreira Ceccon (gustavo.ceccon@usp.br)



Objetivos

- Introduzir alguns conceitos de redes de computadores
- Mostrar problemas comuns de jogos multiplayer
- Mostrar algumas arquiteturas de redes aplicadas a jogos
 - ◆ Peer-to-peer (P2P)
 - ◆ Cliente-Servidor
- Explicar conceitos e protocolos de sockets
 - ◆ TCP vs. UDP
- Mencionar problemas e soluções de transmissão de dados





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/>



Índice

1. Conceitos
2. Problemas Frequentes
3. Arquitetura de rede
4. Sockets
5. Envio de Mensagens
6. Interpolação e Extrapolação



1. Conceitos



1. Conceitos

- Computadores
- Rede de computadores
- Pacotes



1. Conceitos

Sobre “lag”:

- “Lag” devido a processamento intenso no computador do jogador
- “Lag” devido a atrasos da rede, protocolos de comunicação, perdas de pacote, distância entre computadores, ...



2. Problemas Frecuentes



2. Problemas Frequentes

→ Perda de sincronia entre jogadores

- ◆ “Mas eu usei a faquinha e ele não morreu!”

- ◆ Estados conhecido dos jogadores não corresponde ao estado real

→ Lag

- ◆ Queda da framerate e/ou responsividade reduzida

- ◆ Perda de dados, instabilidade de rede



3. Arquitetura de Rede



3. Arquitetura de Rede

→ Como os jogadores vão se comunicar?

◆ Servidor

- Dedicado
- *Hosted*

◆ P2P

3. Arquitetura de Rede

→ Peer-to-peer (P2P)

- ◆ Jogadores trocam informações entre si, sem intermediário
- ◆ Lag com 1 jogador => lag com todos
- ◆ Informações descentralizadas
- ◆ Mais comum em jogos com apenas 2 jogadores

3. Arquitetura de Rede

→ Cliente-Servidor

- ◆ Informações centralizadas, facilita consistência
- ◆ Servidor tem autoridade
- ◆ Mais facilmente escalável
- ◆ *Hosted* vs. dedicado

4. Sockets

4. Sockets

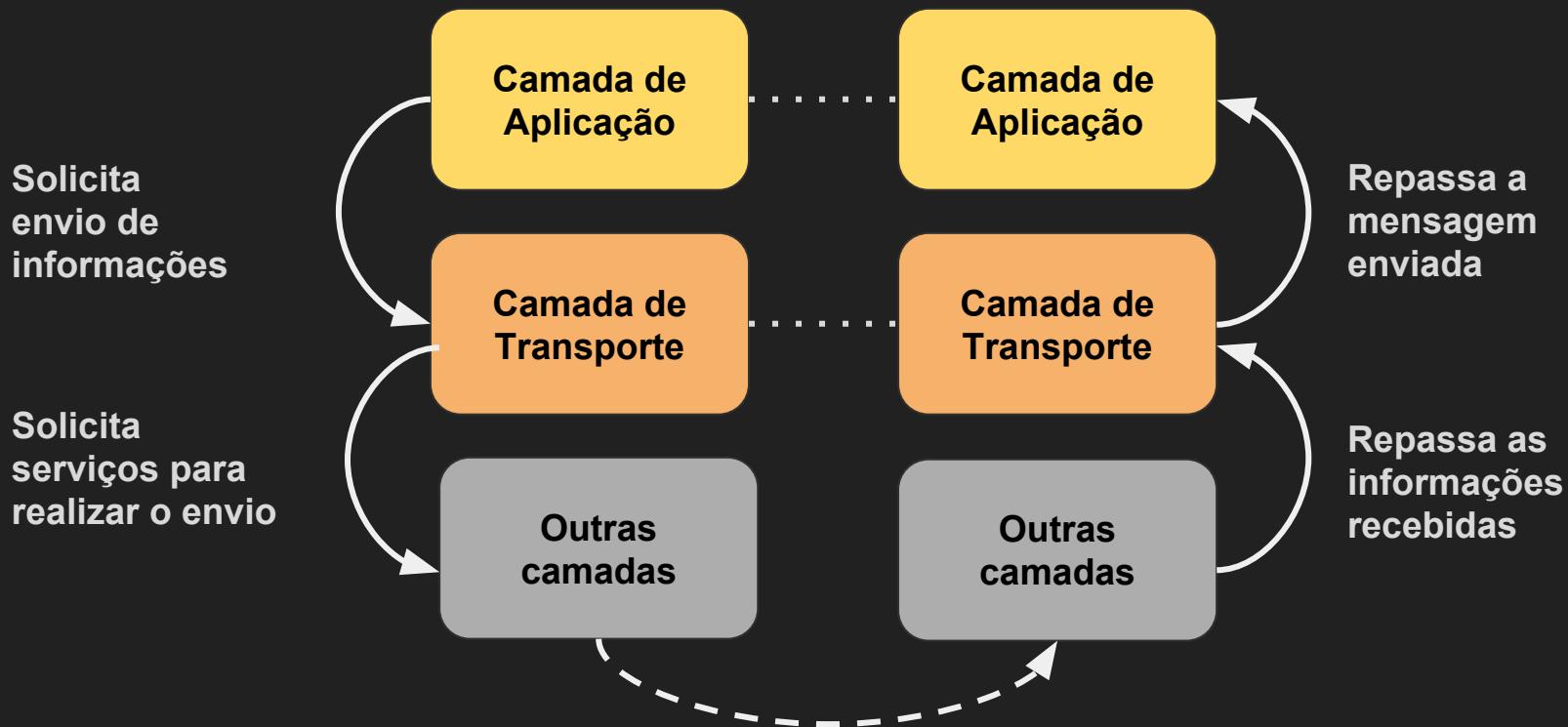
- Estruturas para envio de mensagens entre computadores
- Realizam comunicação por meio de um *protocolo*

4. Sockets

→ Protocolo de Rede

- ◆ Define a estrutura da comunicação
- ◆ Análogo a uma linguagem natural
 - Bits ~ letras
 - Mensagens ~ frases
- ◆ Dividido em *camadas* responsáveis por diferentes etapas da comunicação

4. Sockets



4. Sockets

→ TCP

- ◆ Transmissão lenta, mas confiável

→ UDP

- ◆ Transmissão não confiável, mas rápida
- ◆ Na prática, usado para qualquer comunicação em tempo real
 - Livestreams
 - FPS



4. Sockets

- Para o TCP, usamos um endereço IP e criamos e mantemos uma conexão, que só é fechada quando solicitada
- Para UDP, mandamos pacotes independentes para o endereço IP e não abrimos uma conexão

5. Envio de Mensagens



5. Envio de Mensagens

- O que enviar?
 - ◆ Input dos jogadores?
 - ◆ Informações (posição, vida etc.)?

5. Envio de Mensagens

→ Input?

- ◆ Se o jogo depender de física ou simulações, pode gerar inconsistência
 - Para jogos em tempo real, **geralmente** uma má ideia
- ◆ Para jogos por turnos, pode simplificar ou reduzir a quantidade de dados enviados
 - Ainda seria necessário conferir sincronia periodicamente

5. Envio de Mensagens

→ Informações?

- ◆ “Sempre” funciona, mas pode ser necessário enviar mensagens muito maiores
 - Restrição de tempo no envio da mensagem
- ◆ Pode ser otimizada enviando apenas informações cruciais
 - Calcular ou estimar informações não importantes

5. Envio de Mensagens

→ Envio via TCP

- ◆ Garantido que as mensagens cheguem e que todas estarão em ordem de envio
- ◆ Lenta (muito lenta)

→ Soluções

- ◆ Usar UDP
- ◆ Fazer checagem de chegada na mão por cima do UDP

5. Envio de Mensagens

→ Envio via UDP

- ◆ Não é garantido que os dados chegarão
- ◆ Mesmo se chegarem, não há garantia da ordem em que uma sequência de mensagens chegará

→ Soluções

- ◆ Mandar várias vezes ou garantir a chegada, como no TCP
- ◆ Ignorar pacotes antigos e apenas atualizar
- ◆ Ordenar ou usar interpolação e extrapolação

5. Envio de Mensagens

- Esperar o servidor responder => lag
- Client prediction
 - ◆ Cliente faz parte do trabalho do servidor para prever o estado do jogo
 - ◆ Manda informação e atualiza os estados locais como posição
 - ◆ Servidor decide se aceita ou não, além de ajeitar quando preciso
 - Limite de distância - WoW e outros exploits
 - Inconsistência, por exemplo, hitbox em FPS

6. Interpolação e Extrapolação

6. Interpolação e Extrapolação

- Extrapolação é prever o estado futuro de outros jogadores
 - ◆ Aproximar estado de outros jogadores com base nos estados passados
 - Chutar um valor plausível para a próxima posição
 - ◆ OK com ping baixo
 - ◆ Com ping alto, jogadores começam a se teleportar

6. Interpolação e Extrapolação

- Interpolação feita pelo jogador para simular a trajetória
 - ◆ Cliente não atualiza todos os valores assim que recebe uma mensagem
 - Interpola entre o valor conhecido e o valor recebido
 - ◆ Tenta melhorar continuidade, mesmo que a um passo inconstante

6. Interpolação e Extrapolação

Netcode do Overwatch

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>
- [11] https://www.reddit.com/r/Overwatch/comments/3u5kfg/everything_you_need_to_know_about_tick_rate/