Concorrência e Distribuição

TRIVIALIDADES MULTIPLAYER

ALANY GABRIELY - DESIGNE
ARTUR LAPOT - FRONTEND
FILIPE PAÇO (LÍDER) - BACKEND
RANDERSON - DOCUMENTAÇÃO

Escopo

Introdução

O projeto proposto é um jogo de perguntas e respostas projetado para ser jogado por três jogadores simultaneamente. O jogo é implementado em um ambiente distribuído, onde um servidor central coordena a interação entre os jogadores e gerência o fluxo do jogo. Este projeto visa explorar conceitos de software concorrente e distribuído.

Funcionamento do Jogo

O jogo consiste em várias rodadas onde o servidor seleciona aleatoriamente uma pergunta da lista e a envia para os jogadores. O primeiro jogador a pressionar um botão especificado tem a oportunidade de responder à pergunta. Se a resposta estiver correta, o jogador ganha um ponto; se estiver incorreta, ele perde um ponto. O jogo continua até que um jogador atinja cinco pontos.

Tópicos da Ementa Abordados

- Software Concorrente e Distribuído: O projeto explora a criação de um sistema distribuído onde múltiplos jogadores interagem simultaneamente.
- Middleware e Distribuição: O servidor atua como um middleware que coordena a comunicação entre os jogadores e gerencia o fluxo do jogo.
- Concorrência e Paralelismo: O jogo requer gerenciamento de interações concorrentes, como a resposta dos jogadores e atualização de pontuações.

Situações de Concorrência

Aperto do Botão
Quando um jogador pressionao botão e envia uma resposta, o serverside precisa processar essa resposta e atualizar a pontuação do jogador correspondente. Novamente, é fundamental garantir que apenas uma thread do servidor esteja processando as respostas dos jogadores por vez para evitar conflitos e condições de corrida. Algo que pode ser feito através do uso de um mutex para bloquear o recurso quando o jogador pressionar o botão

WebSocket para exibição da situação captada no server na interface do client

O projeto requer o uso de mecanismos de sincronização adequados para garantir a consistência e a integridade dos dados compartilhados entre o servidor e os clientes. Isso pode incluir o uso de estruturas de sincronização para garantir a atualização dos dados de maneira simultânea e justa.

Modelos de Concorrência:

- O projeto pode ser implementado usando diferentes modelos de concorrência, como modelos baseados em threads, processos ou até mesmo modelos baseados em eventos. Cada modelo tem suas próprias vantagens e desvantagens em termos de desempenho, escalabilidade e complexidade de implementação.

Situações de Distribuição

Mais de Um cliente

O sistema é distribuído, pois envolve múltiplos dispositivos físicos interconectados através de uma rede de comunicação. O servidor e os três clientes estão operando em diferentes dispositivos e se comunicam através de uma rede.

Simultânedade de Informações aos Clientes

As perguntas são transmitidas para todos os jogadores simultaneamente, indicando uma distribuição de dados entre os dispositivos conectados.

Responsabilidade de Cada Cliente

Cada cliente é responsável por receber as perguntas do servidor, enviar suas respostas de volta e manter sua própria pontuação. Essa distribuição de responsabilidades entre os dispositivos constitui um aspecto distribuído do sistema.

Tecnologias Utilizadas

FRONTEND	Para o frontend moderno e consistente escolhemos o React para confecionar as telas dos usuários.
SERVERSIDE	O backend será desenvolvido em python, utilizando websockets para a ágil interação com o front. Além disso para lidar com a concorrencia sera feito o uso de mutex e e a biblioteca _thread que facilita para a criação das threads.
DEPLOY	Nosso deploy será feito em servidor próprio