Faculdade projeção

DOD para otimização server-side em jogos

Documento de Visão

Versão 1.0.0

Histórico de Revisão

Data	Versão	Descrição	Autor
06/09/2020	1.0.0	Criação do Documento de Visão para descrição geral através do tema “DOD para otimização server-side em jogos”.	Nádio Dib

Sumário

Introdução

Ao decorrer de vários anos de trabalho informal na área de desenvolvimento de jogos para criação, manutenção e otimização de aplicações server-side ou *back-end* para servidores através da utilização do *framework* .NET, percebi que sobrecargas de execução nas operações de rotina repetitiva ou *loop* eram de intensa interação entre instâncias de objetos quando se tratava de comunicação por meio de encapsulamento de mensagens com a abordagem do paradigma *Object-Oriented Programming* (OOP) eram muito constantes e necessitavam e atenção.

Logo, a proposta deste documento é coletar, analisar e definir as necessidades e funcionalidades gerais do projeto "DOD para otimização server-side em jogos" para fins acadêmicos e de estudo prático.

Seu foco está nas necessidades da devida compreensão de conceitos e fundamentos computacionais para uma melhor adequação de operações exaustivas exigidas por aplicações de servidor de jogos através de uma nova abordagem feita pelos materiais levantados de DOD e os motivos da existência destas necessidades para mitigação dos problemas contextuais oriundos deste mesmo tema.

Termos e abreviaturas específicos podem ser encontrados no Glossário do respectivo projeto.

Contextualização

Dentro desta perspectiva, rotinas repetitivas em servidores de jogos tornaram-se regiões críticas e sensíveis durante suas operações quando em funcionamento no modo de produção ou *release*. Estas rotinas se caracterizaram como regiões críticas devido a implementação de conceitos de *multithreading* para utilização de recursos de programação paralela concorrente, nos quais, muitas das vezes eram caracterizados por recursos compartilhados entre processos adjuntos das rotinas para fins de otimização.

A utilização de conceitos e fundamentos dos assuntos de *concurrent programming* e *parallel programming* auxiliaram até de certa forma na confecção de uma solução para mitigação de sobrecargas de trabalho nestas operações exigidas por essas rotinas, que se caracterizavam por requisitar alto uso de recursos da CPU, no qual a aplicação estava em funcionamento, naquele ambiente computacional.

Entretanto, após investigar com mais detalhe sobre como possibilitar uma melhor adequação para estas rotinas, a simples utilização dos fundamentos de *Data-Oriented Design* (DOD) juntamente com o devido discernimento do funcionamento do sistema operacional, possibilitavam menor utilização dos recursos computacionais, pelo qual a aplicação está em funcionamento naquele ambiente operacional, e proporcionar confidencialidade no tratamento de dados para procedimentos de concorrência dos recursos, pois estes problemas recorrentes da utilização de conceitos e implementações de funcionalidades *multithreading* seriam ausentes, nos quais procedimentos assíncronos passariam a ser síncronos e pertencentes da mesma *thread* que caracteriza rotina.

De acordo com os assuntos abordados previamente na Introdução deste Documento de Visão e Escopo (DVE) de forma superficial, há a necessidade de apresentar esta abordagem oriunda de conceitos existentes das áreas de desenvolvimento de software como Sistemas Operacionais, Arquitetura de Software, Estrutura de Dados e Algoritmos, Modelagem de Software Orientado a Objeto e Lógica de Programação, nos quais fazem parte da formação acadêmica do curso de Engenharia de Software.

Problemática

Operações em rotinas que desempenham processos de longa duração e iterações entre inúmeras instâncias de objetos através do paradigma OOP, não são eficientes quando suscetível a uma grande coleção de instâncias de objetos para iteração resultando no aumento do tempo de resposta da rotina que desempenha um papel crítico para toda a aplicação.

Este problema afeta a integridade e confiabilidade de aplicações de jogos para servidores, comprometendo o seu devido funcionamento sem a necessidade de alocação de melhores recursos computacionais, apenas trabalhando com os já existentes.

A necessidade de alocação e investimento de máquinas mais robustas para o devido funcionamento da aplicação de jogos em servidores, bem como alto custo para manutenção e mitigação deste problema abordando apenas o paradigma OOP, onde é necessário o contrato de profissionais mais experientes para apoio técnico de *tunning* ou utilização de outras tecnologias para serem acopladas na aplicação e assim possibilitar uma falsa sensação de solução para este problema.

Solução Proposta

Utilizar os conceitos e fundamentos multidisciplinares do curso de graduação de Engenharia de Software com o padrão de desenvolvimento DOD, para auxiliar em uma melhor compreensão deste problema sem a necessidade de alto investimento financeiro para migração de outras tecnologias, pois este estudo tem como base providenciar uma sugestão de solução para este problema específico nos quais grandes corporações recentemente utilizaram para outros fins, porém de mesma complexidade e necessidade.

Sendo assim, proponho a devida interação multidisciplinar para fundamentar os conceitos que serão relacionados no decorrer deste projeto.

O que se esperar

Ao final deste projeto, é esperado alcançar um resultado satisfatório provenientes de testes, citações científicas, referências bibliográficas acadêmicas, bem como aplicação dos fundamentos multidisciplinares citados na Contextualização deste DVE em prática.

Objetivo Geral

O objetivo deste documento é contribuir para o aprimoramento e aperfeiçoar a qualidade das informações ofertadas, provenientes de pesquisas e estudos científicos. Sendo assim, este projeto tem como objetivo geral englobar várias áreas de conhecimento que estão interligadas de modo a atender as necessidades essenciais para o devido atendimento destes requisitos propostos na Solução Proposta deste DVE.

Objetivos Específicos

Utilizando como referência principal, mas não somente analisando de forma genérica certos fundamentos e conceitos teóricos, este projeto tem como ênfase específica nos seguintes objetivos:

I. Compreender o devido funcionamento de procedimentos de encapsulamento da mensagem através do paradigma OOP dentro do ambiente computacional;

II. Analisar de forma conceitual partições da memória gerenciadas pelo sistema operacional de modo a atender as necessidades de compreensão para o padrão de desenvolvimento DOD;
III. Interpretar as vantagens e desvantagens entre o paradigma OOP e o padrão de desenvolvimento DOD;
IV. Proporcionar através de uma nova perspectiva a devida relação entre o padrão de desenvolvimento DOD com recursos computacionais, sendo estes o processador e procedimentos de alocação de memória;
V. Capacitar possibilidade de generalização do padrão de desenvolvimento DOD para outros fins que atendem as mesmas características publicadas através dos relatórios de teste e demandam mesma aplicabilidade.

Estrutura do Projeto

Este projeto foi alocado nos seguintes capítulos de modo a auxiliar o seu devido entendimento e compreensão com a linha de pesquisa adotada, sendo estes:

- Capítulo 1 – alocação de memória durante procedimentos instanciação de objetos;
- Capítulo 2 – quando o paradigma OOP se torna um problema;
- Capítulo 3 – funcionamento da memória dentro do ambiente computacional;
- Capítulo 4 – operações assíncronas para otimização de processos através dos fundamentos *multithreading*;
- Capítulo 5 – OOP *versus* DOD;
- Capítulo 6 – os motivos por trás do DOD;
- Capítulo 7 – a devida utilização DOD em operações de alta performance e baixa latência de resposta em aplicações *server-side* de jogos;
- Capítulo 8 – as aplicações do padrão de desenvolvimento DOD pela empresa Unity em sua nova tecnologia, o Unity DOTS;
- Conclusão;
- Referências Bibliográficas;
- Glossário;
- Anexos;
- Apêndices.

Capítulo 1 – alocação de memória durante procedimento de instanciação de objetos

Texto capítulo 1.

Capítulo 2 – quando o paradigma OOP se torna um problema

Texto capítulo 2.

Capítulo 3 – funcionamento da memória dentro do ambiente computacional

Texto capítulo 3.

Capítulo 4 – operações assíncronas para otimização dos de processos através dos fundamentos *multithreading*

Texto capítulo 4.

Capítulo 5 – OOP *versus* DOD

Texto capítulo 5.

Capítulo 6 – os motivos por trás do DOD

Texto capítulo 6.

Capítulo 7 – a devida utilização DOD em operações de alta performance e baixa latência de resposta em aplicações *server-side* de jogos

Texto capítulo 7.

Capítulo 8 – as aplicações do padrão de desenvolvimento DOD pela empresa Unity em sua nova tecnologia, o Unity DOTS

Texto capítulo 8.

Conclusão

Texto conclusão.

Referências Bibliográficas

Texto referências bibliográficas.

Glossário

Texto glossário.

Anexos

Texto anexos.

Apêndices

Texto apêndices.