

UNIVASSOURAS
ENGENHARIA DE SOFTWARE

LEANDRO LIMA CARDOSO (202323366)

DOCUMENTO DE REQUISITOS
DESENVOLVIMENTO DE UMA *GAME ENGINE* 2D EM FORMATO DE BIBLIOTECA
EM C++
PRÁTICAS EXTENSIONISTAS INTEGRADORAS VI

SAQUAREMA
2026

LEANDRO LIMA CARDOSO (202323366)

DOCUMENTO DE REQUISITOS
DESENVOLVIMENTO DE UMA *GAME ENGINE* 2D EM FORMATO DE BIBLIOTECA
EM C++
PRÁTICAS EXTENSIONISTAS INTEGRADORAS VI

Trabalho apresentado como requisito parcial para obtenção de nota da disciplina de **Práticas Extensionistas Integradoras VI** do curso de Engenharia de Software da Univassouras (campus universitário de Saquarema).

Prof. Dr. Altemar Sales

SAQUAREMA
2026

RESUMO

Este documento apresenta a especificação de requisitos para o desenvolvimento de uma *engine* de jogos bidimensionais implementada na linguagem de programação C++. A especificação tem como objetivo definir de forma clara e estruturada as funcionalidades, restrições e características de qualidade que o sistema deve atender durante seu desenvolvimento. O documento descreve os requisitos funcionais e não funcionais da *engine*, estabelecendo as funcionalidades essenciais que deverão ser oferecidas aos desenvolvedores que utilizarem a biblioteca para criação de jogos 2D. Entre essas funcionalidades estão o gerenciamento do *loop* principal do jogo, renderização gráfica, gerenciamento de entidades, controle de entrada do usuário, sistema de colisão e organização de cenas. Além da definição dos requisitos, o documento apresenta critérios de aceitação que permitem verificar se as funcionalidades foram implementadas corretamente, bem como uma matriz de rastreabilidade que relaciona os requisitos definidos com os módulos responsáveis por sua implementação no sistema. A especificação de requisitos serve como base para as etapas posteriores do projeto, incluindo a definição da arquitetura do sistema, o planejamento da implementação e a validação das funcionalidades da *engine*.

Palavras-chave: *game engine*; C++; especificação de requisitos; jogos 2D; desempenho.

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Propósito	5
1.2 Escopo do Sistema	5
1.3 Definições, Acrônimos e Abreviações	6
1.4 Referências	6
1.5 Visão Geral do Documento	6
2 DESCRIÇÃO GERAL DO SISTEMA.....	7
2.1 Perspectiva do Produto	7
2.2 Funções do Produto	8
2.3 Características dos Usuários	8
2.4 Restrições	9
2.5 Suposições e Dependências	9
3 REQUISITOS ESPECÍFICOS	10
3.1 Requisitos Funcionais	10
3.2 Requisitos Não Funcionais.....	12
4 CRITÉRIOS DE ACEITAÇÃO	13
5 MATRIZ DE RASTREABILIDADE DE REQUISITOS.....	15

1 INTRODUÇÃO

1.1 Propósito

Este documento tem como objetivo especificar de forma clara, estruturada e verificável os requisitos funcionais e não funcionais da biblioteca de desenvolvimento de jogos 2D baseada em C++, desenvolvida no contexto do Trabalho de Conclusão de Curso.

A especificação aqui descrita servirá como base para:

- Definição da arquitetura do sistema
- Planejamento de implementação
- Validação técnica
- Testes e verificação
- Evolução futura da engine

1.2 Escopo do Sistema

O sistema consiste no desenvolvimento de uma biblioteca modular em C++ destinada à criação de jogos digitais 2D com estrutura baseada em *tiles* e perspectiva *top-down*.

A *engine* deverá oferecer:

- Sistema de renderização 2D
- Gerenciamento de entidades
- Sistema de colisão
- Gerenciamento de recursos (imagens, mapas)
- Estrutura de *loop* principal de jogo
- Controle de entrada (*input*)

- Organização modular com foco em performance

Não fazem parte do escopo:

- Editor gráfico integrado
- Sistema de física avançado (3D)
- Interface gráfica para configuração

1.3 Definições, Acrônimos e Abreviações

Engine — Conjunto de módulos responsáveis por fornecer infraestrutura para desenvolvimento de jogos.

Tile — Unidade gráfica básica que compõe mapas 2D.

FPS — *Frames Per Second* (quadros por segundo).

API — Interface de Programação de Aplicações.

RF — Requisito Funcional.

RNF — Requisito Não Funcional.

1.4 Referências

- Documentação ISO/IEC 25010 (Modelo de Qualidade de Software)
- Sommerville, I. Engenharia de Software.
- Stroustrup, B. The C++ Programming Language.

1.5 Visão Geral do Documento

Este documento está organizado da seguinte forma:

- Seção 2 apresenta a descrição geral do sistema;

- Seção 3 detalha os requisitos funcionais e não funcionais;
- Seção 4 define critérios de aceitação;
- Seção 5 apresenta matriz de rastreabilidade.

2 DESCRIÇÃO GERAL DO SISTEMA

2.1 Perspectiva do Produto

O sistema proposto consiste no desenvolvimento de uma biblioteca de software destinada à criação de jogos digitais bidimensionais, implementada na linguagem de programação C++. A *engine* será projetada como uma solução modular, permitindo que diferentes componentes do sistema sejam desenvolvidos, mantidos e evoluídos de forma independente.

A proposta da *engine* é fornecer uma infraestrutura básica para o desenvolvimento de jogos 2D, oferecendo funcionalidades essenciais como renderização gráfica, gerenciamento de entidades, controle de entrada do usuário e detecção de colisões.

O sistema será utilizado principalmente por desenvolvedores ou estudantes interessados em compreender os fundamentos da arquitetura de motores de jogos e na criação de jogos 2D com maior controle sobre o funcionamento interno da aplicação.

A *engine* não terá como objetivo competir com motores comerciais amplamente consolidados, como Unity ou Unreal Engine, mas sim servir como uma ferramenta educacional e técnica que permita compreender os princípios estruturais envolvidos na construção de um motor de jogos.

2.2 Funções do Produto

A *engine* deverá disponibilizar um conjunto de funcionalidades essenciais que permitam o desenvolvimento de jogos 2D baseados em mapas compostos por *tiles*.

Entre as principais funções do sistema estão:

- Inicialização e gerenciamento do *loop* principal do jogo;
- Renderização de elementos gráficos bidimensionais;
- Gerenciamento de entidades dentro do ambiente do jogo;
- Carregamento e gerenciamento de recursos gráficos;
- Controle de entrada do usuário (teclado e mouse);
- Implementação de sistema de colisão entre entidades;
- Estrutura de organização de cenas ou estados de jogo;
- Atualização e controle do ciclo de execução do jogo.

Essas funcionalidades formam o núcleo da engine e serão disponibilizadas através de uma API acessível ao desenvolvedor.

2.3 Características dos Usuários

O sistema será utilizado principalmente por dois perfis de usuários:

Desenvolvedores iniciantes ou estudantes de programação, interessados em compreender a estrutura e o funcionamento de motores de jogos, bem como em desenvolver jogos 2D simples utilizando a linguagem C++.

Desenvolvedores com conhecimento intermediário em programação, que desejam experimentar arquiteturas de desenvolvimento de jogos com maior controle sobre os componentes internos do sistema.

Os usuários devem possuir conhecimentos básicos em programação orientada a objetos e familiaridade com conceitos fundamentais de desenvolvimento de software.

2.4 Restrições

O desenvolvimento do sistema está sujeito a algumas restrições técnicas e acadêmicas.

Entre as principais restrições estão:

- O sistema deverá ser implementado utilizando a linguagem de programação C++;
- A arquitetura da *engine* deverá priorizar modularidade e organização estrutural;
- sistema deverá ser executado em ambiente *desktop*;
- projeto deverá ser desenvolvido dentro do período estabelecido para o Trabalho de Conclusão de Curso;
- A *engine* terá foco educacional, priorizando clareza estrutural em vez de complexidade excessiva.

2.5 Suposições e Dependências

Durante o desenvolvimento do sistema, algumas suposições são consideradas para orientar o projeto.

Entre elas destacam-se:

- usuário da *engine* possuirá conhecimento básico de programação em C++;
- Os jogos desenvolvidos utilizando a *engine* terão escopo limitado a jogos 2D;
- A execução do sistema ocorrerá em computadores pessoais com capacidade gráfica básica;

- projeto poderá depender de bibliotecas externas para suporte à renderização gráfica ou manipulação de janelas.

Essas suposições permitem definir limites práticos para o desenvolvimento da *engine* e orientar decisões técnicas durante o projeto.

3 REQUISITOS ESPECÍFICOS

Esta seção apresenta os requisitos específicos do sistema, divididos em requisitos funcionais e requisitos não funcionais. Os requisitos funcionais descrevem as funcionalidades que a *engine* deve fornecer ao usuário, enquanto os requisitos não funcionais estabelecem critérios de qualidade e restrições que devem ser atendidas pelo sistema.

3.1 Requisitos Funcionais

Os requisitos funcionais descrevem as operações e serviços que o sistema deve oferecer aos usuários e desenvolvedores que utilizarem a *engine*.

RF01 – Inicialização da *Engine*

A *engine* deve permitir a inicialização do sistema através de uma interface de programação que configure os componentes principais necessários para execução do jogo.

RF02 – Gerenciamento do *Loop* Principal

A *engine* deve fornecer um mecanismo responsável por controlar o *loop* principal do jogo, incluindo atualização lógica, processamento de entrada e renderização gráfica.

RF03 – Renderização de Elementos Gráficos

A *engine* deve permitir a renderização de *sprites* e outros elementos gráficos bidimensionais na tela.

RF04 – Gerenciamento de Entidades

A *engine* deve permitir a criação, atualização e remoção de entidades dentro do ambiente do jogo.

RF05 – Sistema de Colisão

A *engine* deve fornecer mecanismos para detecção de colisões entre entidades presentes no ambiente do jogo.

RF06 – Gerenciamento de Recursos

A *engine* deve permitir o carregamento e gerenciamento de recursos utilizados pelo jogo, como imagens e arquivos de mapas.

RF07 – Sistema de Entrada do Usuário

A *engine* deve capturar e processar eventos de entrada do usuário provenientes de dispositivos como teclado e mouse.

RF08 – Estrutura de Cenas

A *engine* deve permitir a organização do jogo em diferentes cenas ou estados, possibilitando a transição entre diferentes contextos do jogo.

RF09 – Atualização de Entidades

A *engine* deve permitir que entidades sejam atualizadas a cada ciclo do loop do jogo, possibilitando o processamento de lógica e comportamento.

RF10 – Encerramento do Sistema

A *engine* deve fornecer mecanismos para encerramento seguro da execução do jogo e liberação adequada dos recursos utilizados.

3.2 Requisitos Não Funcionais

Os requisitos não funcionais definem atributos de qualidade e restrições técnicas que o sistema deve atender durante sua execução e manutenção.

RNF01 – Desempenho

A *engine* deve ser capaz de manter uma taxa mínima de 60 quadros por segundo em condições normais de execução.

RNF02 – Modularidade

A arquitetura da *engine* deve ser organizada de forma modular, permitindo que diferentes componentes do sistema sejam desenvolvidos e mantidos de forma independente.

RNF03 – Manutenibilidade

O código-fonte do sistema deve seguir boas práticas de organização e documentação, facilitando sua manutenção e evolução.

RNF04 – Portabilidade

A *engine* deve ser desenvolvida de forma que possa ser executada em diferentes sistemas operacionais compatíveis com compiladores C++ modernos.

RNF05 – Escalabilidade

A arquitetura do sistema deve permitir a adição de novos módulos e funcionalidades sem necessidade de alterações estruturais significativas.

RNF06 – Usabilidade da API

A API da *engine* deve apresentar uma interface clara e consistente, facilitando sua utilização por desenvolvedores.

RNF07 – Organização do Código

O código da *engine* deve ser estruturado em módulos ou componentes claramente definidos, com responsabilidades bem delimitadas.

4 CRITÉRIOS DE ACEITAÇÃO

Os critérios de aceitação definem as condições necessárias para que os requisitos especificados neste documento sejam considerados atendidos. Esses critérios permitem validar se as funcionalidades e características do sistema foram implementadas de acordo com o planejado.

A verificação dos critérios de aceitação poderá ser realizada por meio de testes funcionais, execução de protótipos e análise do comportamento do sistema durante o desenvolvimento do projeto.

Para cada requisito funcional estabelecido, deve ser possível demonstrar sua implementação através da execução prática da *engine* ou por meio da análise do código desenvolvido.

Entre os principais critérios de aceitação do sistema estão:

- A *engine* deve inicializar corretamente, permitindo a criação de uma janela de execução do jogo;
- sistema deve executar o *loop* principal do jogo de forma contínua, permitindo atualização lógica e renderização gráfica;
- A *engine* deve ser capaz de renderizar *sprites* e elementos gráficos bidimensionais na tela;
- sistema deve permitir a criação e gerenciamento de entidades dentro do ambiente do jogo;
- A detecção de colisões entre entidades deve funcionar conforme esperado durante a execução do jogo;
- A *engine* deve permitir o carregamento de recursos gráficos utilizados no jogo;
- O sistema deve capturar e processar entradas do usuário provenientes de teclado ou mouse;
- O sistema deve permitir a organização do jogo em diferentes cenas ou estados;
- A execução da aplicação deve manter desempenho adequado durante o funcionamento normal;
- A *engine* deve permitir o encerramento seguro da aplicação sem ocorrência de erros críticos.

O atendimento a esses critérios permitirá confirmar que os requisitos funcionais e não funcionais descritos neste documento foram implementados com sucesso.

5 MATRIZ DE RASTREABILIDADE DE REQUISITOS

A matriz de rastreabilidade de requisitos tem como objetivo estabelecer a relação entre os requisitos definidos neste documento e os módulos ou componentes responsáveis por sua implementação no sistema.

Essa matriz auxilia no acompanhamento do desenvolvimento do projeto, permitindo verificar se todos os requisitos foram devidamente considerados durante a implementação da engine.

Além disso, a rastreabilidade facilita a manutenção e evolução do sistema, possibilitando identificar rapidamente quais partes do software são impactadas por possíveis alterações nos requisitos.

A Tabela a seguir apresenta a relação entre os requisitos funcionais definidos e os principais módulos da arquitetura do sistema.

Requisito	Descrição	Módulo Responsável
RF01	Inicialização da <i>engine</i>	<i>Core</i>
RF02	Gerenciamento do <i>loop</i> principal	<i>Core</i>
RF03	Renderização de elementos gráficos	<i>Renderer</i>
RF04	Gerenciamento de entidades	<i>Entity System</i>
RF05	Sistema de colisão	<i>Collision System</i>
RF06	Gerenciamento de recursos	<i>Resource Manager</i>
RF07	Sistema de entrada do usuário	<i>Input System</i>
RF08	Estrutura de cenas	<i>Scene Manager</i>
RF09	Atualização de entidades	<i>Entity System</i>
RF10	Encerramento do sistema	<i>Core</i>

A utilização dessa matriz permite acompanhar de forma estruturada a implementação das funcionalidades previstas no sistema, garantindo que cada requisito seja contemplado na arquitetura e no desenvolvimento da *engine*.