Software Requirements Specification

for

Nugget 2D Game Engine

Version 1.0 approved

Prepared by (group - 124):
Aniket Inamdar (IIT2023003)
Wasil Iftekhar (IIT2023001)
Vivek Gadhia (IIT2023004)

April 29, 2025

Table of Contents

Revision History . . . . . **iii**

1 Introduction **1**
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 1

2 Overall Description **1**
2.1 Product Perspective 1
2.2 Product Functions 1
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 2

3 External Interface Requirements **3**
3.1 User Interfaces 3
3.2 Hardware Interfaces 3
3.3 Software Interfaces 3
3.4 Communications Interfaces 3

4 System Features **3**
4.1 Rendering Engine 3
4.2 Scene and Entity Management 3
4.3 Input Handling 4
4.4 ImGui Integration 4
4.5 Asset Management 4

5 Other Nonfunctional Requirements **4**
5.1 Performance Requirements 4
5.2 Safety Requirements 5
5.3 Security Requirements 5
5.4 Software Quality Attributes 5
5.5 Business Rules 5

6 Other Requirements **5**

Appendix B: Analysis Models **6**

Revision History

Name	Date	Reason for Changes	Version
Nugget 2D Game Engine	29-Apr-2025	Initial version	1.0

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the requirements for the Nugget 2D Game Engine, a simple, extensible, and cross-platform game engine written in Java using LWJGL for graphics and ImGui for frontend development.

1.2 Document Conventions

- Requirements are uniquely identified as REQ-XX.
- Priorities are marked as [High], [Medium], or [Low].
- Technical terms and acronyms are defined in Appendix A.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, end-users (game developers), and documentation writers.

1.4 Product Scope

Nugget is a lightweight 2D game engine designed for creating 2D games in Java. It provides rendering, input handling, asset management, and a GUI frontend for debugging and development.

1.5 References

- LWJGL: `https://www.lwjgl.org`
- ImGui Java Bindings: `https://github.com/SpaiR/imgui-java`

2 Overall Description

2.1 Product Perspective

Nugget is a standalone, open-source 2D game engine, interacting with the Java runtime, OpenGL (via LWJGL), and ImGui.

2.2 Product Functions

- 2D rendering (sprites, tiles, backgrounds)
- Scene and entity management

- Input handling (keyboard, mouse)
- Asset loading and management
- GUI overlay for debugging and editing (ImGui)
- Basic physics and collision detection

2.3 User Classes and Characteristics

- **Engine Developers**: Extend and maintain engine features.
- **Game Developers**: Use the engine to build games.
- **Testers**: Validate engine features and performance.
- **End-Users**: Play games built with Nugget.

2.4 Operating Environment

- Java 8 or higher
- Windows, Linux, or macOS
- OpenGL-compatible graphics hardware

2.5 Design and Implementation Constraints

- Written in Java
- Uses LWJGL for graphics
- Uses ImGui for GUI frontend
- Cross-platform
- Open-source licensing

2.6 User Documentation

- User manual
- API documentation (Javadoc)

2.7 Assumptions and Dependencies

- Users have Java and OpenGL drivers installed
- Internet access for dependencies
- Relies on LWJGL, ImGui

3 External Interface Requirements

3.1 User Interfaces

- Main game window (OpenGL context)
- ImGui-based GUI overlay
- Console/log output

3.2 Hardware Interfaces

- Keyboard and mouse input
- GPU with OpenGL support

3.3 Software Interfaces

- Java Runtime Environment
- LWJGL
- ImGui Java bindings

3.4 Communications Interfaces

None required for core engine.

4 System Features

4.1 Rendering Engine

Description and Priority: Provides 2D rendering of sprites and tiles. [High]

Stimulus/Response Sequences: User loads a scene → Engine renders all visible entities.

Functional Requirements:

- REQ-1: Support rendering of PNG/JPEG images as sprites.
- REQ-2: Support rendering of tilemaps.
- REQ-3: Allow dynamic resizing of the game window.

4.2 Scene and Entity Management

Description and Priority: Organizes game objects into scenes. [High]

Stimulus/Response Sequences: User switches scene → Engine loads new scene.

Functional Requirements:

- REQ-4: Support adding/removing entities at runtime.
- REQ-5: Allow serialization/deserialization of scenes.

4.3 Input Handling

Description and Priority: Captures and processes keyboard and mouse input. [High]

Stimulus/Response Sequences: User presses a key → Engine triggers event.

Functional Requirements:

- REQ-6: Support customizable key bindings.
- REQ-7: Support mouse clicks, movement, and wheel input.

4.4 ImGui Integration

Description and Priority: GUI overlay for debugging and editing. [Medium]

Stimulus/Response Sequences: User opens debug menu → ImGui overlay appears.

Functional Requirements:

- REQ-8: Allow real-time editing of entity properties.
- REQ-9: Display engine statistics (FPS, memory usage).

4.5 Asset Management

Description and Priority: Handles loading and management of assets. [Medium]

Stimulus/Response Sequences: User loads asset → Engine caches asset.

Functional Requirements:

- REQ-10: Support asset hot-reloading.
- REQ-11: Handle asset loading errors gracefully.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- Maintain 60 FPS on standard hardware.
- Asset loading should not block the main thread.

5.2 Safety Requirements

Engine must not access or modify system files outside the project directory.

5.3 Security Requirements

- No network access by default.
- Prevent execution of untrusted code in user scripts.

5.4 Software Quality Attributes

- Portability: Windows, Linux, macOS.
- Usability: Simple API, clear documentation.
- Maintainability: Modular codebase.

5.5 Business Rules

All contributions must follow the project's code of conduct and licensing terms.

6 Other Requirements

Internationalization support for future versions. Optional plugin system for extension.

Appendix A: Glossary

- **LWJGL**: Lightweight Java Game Library
- **ImGui**: Immediate Mode Graphical User Interface
- **Entity**: An object in the game world
- **Scene**: A collection of entities and logic

Appendix B: Analysis Models

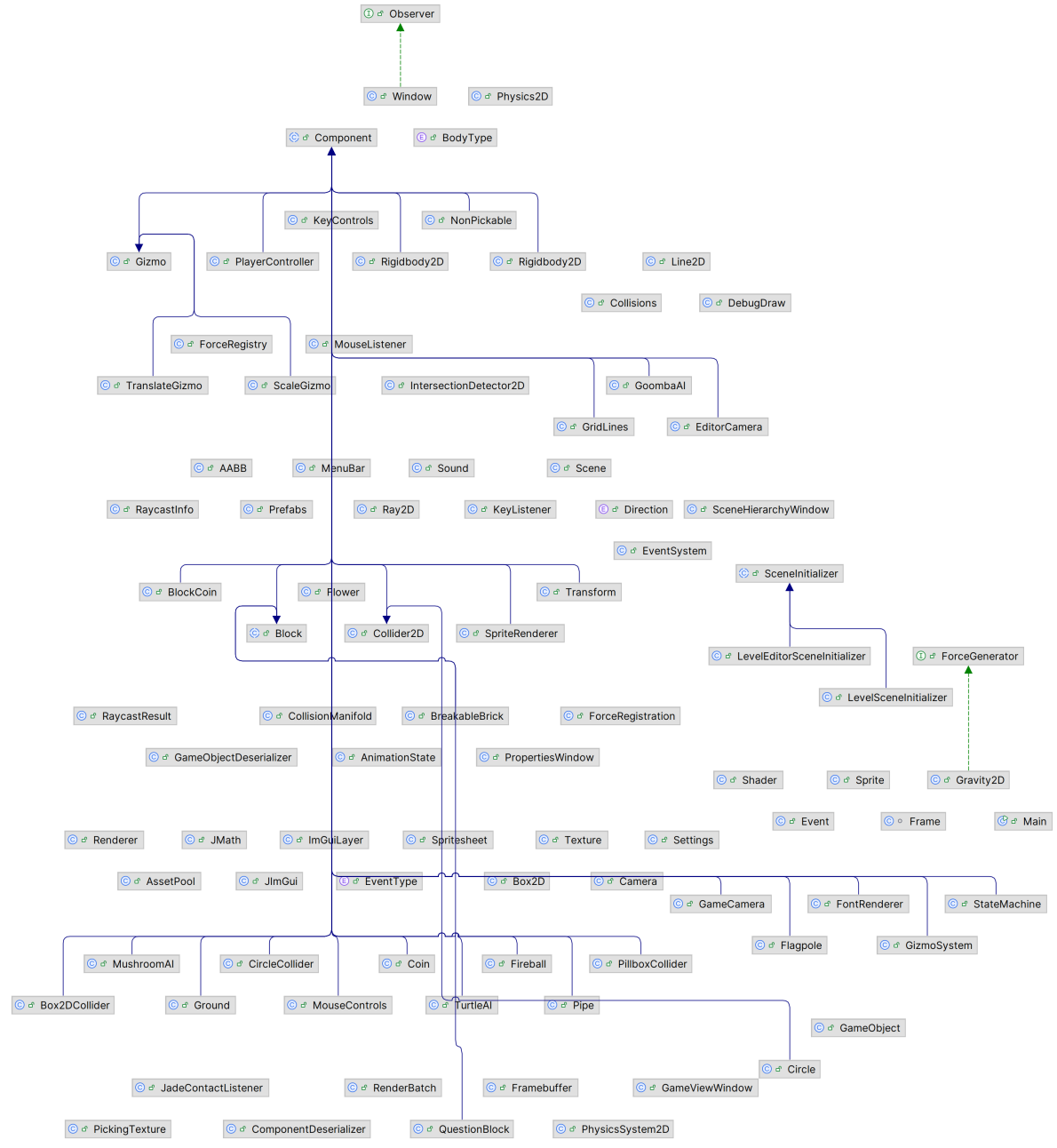

Figure 1: Class Diagram of Nugget 2D Game Engine

Appendix C: To Be Determined List

- Plugin system architecture (TBD)
- Advanced physics integration (TBD)

Project Repository: `https://github.com/Niketw/nugget.git`