Ben Hassine A.

  segister.com

Bowling Game Requirement Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VERSION HISTORY** | | | | |
| **VERSION** | **APPROVED BY** | **REVISION DATE** | **DESCRIPTION OF CHANGE** | **AUTHOR** |
| 1.3.1 | ~ | 13-11-2024 | Added content to the document | Ben Hassine A |
| 1.1.1 | ~ | 13-10-2024 | Added Cover and table of contents | Ben Hassine A |
| 1.0.1 | ~ | 12-10-2024 | Create the file | Ben Hassine A |

[1. OVERVIEW 3](#_Toc182397129)

[2. INTRODUCTION 4](#_Toc182397130)

[2.1. Purpose of the system 4](#_Toc182397131)

[2.2. Goals and Objectives 4](#_Toc182397132)

[2.3. Scope 4](#_Toc182397133)

[2.4. Resources 4](#_Toc182397134)

[2.5. Deliverables 4](#_Toc182397135)

[3. STATE OF THE ART 5](#_Toc182397136)

[3.1. Uncle Bob 5](#_Toc182397137)

[4. SOLUTION 6](#_Toc182397138)

[4.1. Functional Requirements 6](#_Toc182397139)

[4.1.1. Game Control 6](#_Toc182397140)

[4.1.2. Score Something 6](#_Toc182397141)

[4.1.3. Strike 6](#_Toc182397142)

[4.1.4. Spare 6](#_Toc182397143)

[4.1.5. Last Frame 6](#_Toc182397144)

[4.1.6. Exit the Game 6](#_Toc182397145)

[4.1.7. Restart the Game 6](#_Toc182397146)

[4.1.8. Graphical User Interface 6](#_Toc182397147)

[4.1.9. Multi Player Mode 6](#_Toc182397148)

[4.1.10. Best Scores 6](#_Toc182397149)

[4.2. Non-Functional Requirements 7](#_Toc182397150)

[4.2.1. Usability 7](#_Toc182397151)

[4.2.2. Implementation 7](#_Toc182397152)

[4.2.3. Reliability 7](#_Toc182397153)

[4.2.4. Performance 7](#_Toc182397154)

[4.2.5. Supportability 7](#_Toc182397155)

[4.2.6. Interface 7](#_Toc182397156)

[4.2.7. Packaging 7](#_Toc182397157)

[4.2.8. Legal 7](#_Toc182397158)

[5. PROPOSED SYSTEM 8](#_Toc182397159)

[5.1. System models 8](#_Toc182397160)

[5.1.1. Architecture 8](#_Toc182397161)

[5.1.2. Use case model 8](#_Toc182397162)

[5.1.3. Component Model 8](#_Toc182397163)

[5.1.4. Object model 9](#_Toc182397164)

[5.1.5. User interface mock-ups 9](#_Toc182397165)

[5.2. Methodology 9](#_Toc182397166)

[5.3. Schedule, timeline, and deadlines 9](#_Toc182397167)

[5.4. Cost and benefit 10](#_Toc182397168)

[6. ASSUMPTIONS 11](#_Toc182397169)

[7. GLOSSARY 11](#_Toc182397170)

[8. REFERENCES 11](#_Toc182397171)

[9. APPENDIX 11](#_Toc182397172)

# OVERVIEW

The results of the requirements elicitation and the analysis activities are documented in this report. This document completely describes the system in terms of functional and non-functional requirements and serves as a contractual basis between the team members. The document must be written in the English language and generalized as possible for the business/expertise domain. Under no circumstances should any "computerese" terminology creep into this document.

# INTRODUCTION

This section provides a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context.

## Purpose of the system

In the frame of the cooperative work of the team members in the idle space, Mr. Smith asked for a project idea. The proposed idea consists to develop a small solution of **the Bowling Game Score calculator**. Mr Smith thinks if we implement a universal solution, maybe he can present it to Elon Musk and integrate it in the Tesla Software, and will be appreciated children in the jam time within the car to be entertained…

## Goals and Objectives

The project involves developing a Bowling game Solution, just for training propose, following the engineering standards and common conventions used in the industry though.

## Scope

* The solution should be used as an application that interacts with user thru Human-Machine-Interface HMI. Therefor there is 2 possible solutions: either through console terminal or graphical user interface GUI.
* The solution should satisfy the following rules:

The game consists of 10 frames. In each frame the player has two rolls to knock down 10 pins. The score for the frame is the total number of pins knocked down, plus bonuses for strikes and spares

* A spare is when the player knocks down all 10 pins in two rolls. The bonus for that frame is the number of pins knocked down by the next roll.
* A strike is when the player knocks down all 10 pins on his first roll. The frame is then completed with a single roll. The bonus for that frame is the value of the next two rolls.
* In the tenth frame a player who rolls a spare or strike is allowed to roll the extra balls to complete the frame. However no more than three balls can be rolled in tenth frame.
* The Solution should be deterministic and exclude undetermined behaviour.

## Resources

* Mr. Müller, from Rotterdam Netherlands, is interested by implementing a kernel for the game as a self-motivation to take the challenge and practise to sharpen his TDD Skills. He is available from January to February 2024.
* Mr. Santos, from Sau Paulo Brazil, confirmed his take in charge of the implementation of the console interface from January to February 2024 to interact with kernel through terminal and insure the functional requirement.
* Ms. Monique, from Paris France, has a design in mind for the GUI and should be able to work on it from February to Mars 2024.

## Deliverables

The solution should be deliverable for a multitude of platforms, consumable from different business Scenarios (reusable), as a release candidate of an automated development pipeline. The solution ensure the correctness of the interaction, trough validations of entries.

# STATE OF THE ART

## Uncle Bob

Based on Uncle Bob TDD Challenge, we can summarize the scoring for the Bowling:

* Each game includes 10 turns, or “frames” for the player.
* In each frame, the player gets up to 2 tries to knock down all the pins.
* If in 2 tries, he fails to knock them all down, his score for that frame is the total number of pins knocked down in his 2 tries.
* If in 2 tries he knocks the 10 down, this is called a “Spare” and his score for the frame is 10 plus the number of pins knocked down on his next throw (in his next turn).
* If on his first try in the frame he knocks down all the 10 pins, this is called a “Strike”. His turn is over, and his score for the frame is 10 plus the simple total of the pins knocked down in his next two rolls.
* If he gets a spare or strike in the last (10th) frame, the bowler gets to throw one or two more bonus balls, respectively. These bonus throws are taken as part of the same turn. If the bonus throws knock down all the pins, the process does not repeat: the bonus throws are only used to calculate the score of the final frame.
* The game score is the total of all frame scores.

# SOLUTION

## Functional Requirements

The functional requirements describes what the system is going to do, through functions that the system should be able to perform.

### Game Control

* Each game includes 10 turns, or “frames” for the player.
* In each frame, the player gets up to 2 tries to knock down all the pins.
* The game score is the total of all frame scores.

### Score Something

* If in 2 tries, the player fails to knock all pins down, his score for that frame is the total number of pins knocked down in his 2 tries.

### Strike

* If on his first try in the frame the player knocks down all the 10 pins, this is called a “Strike”. His turn is over, and his score for the frame is 10 plus the simple total of the pins knocked down in his next two rolls.

### Spare

* If in 2 tries the player knocks the 10 pins down, this is called a “Spare” and his score for the frame is 10 plus the number of pins knocked down on his next throw (in his next turn).

### Last Frame

* If the player gets a spare or strike in the last (10th) frame, the player gets to throw one or two more bonus bowls, respectively. These bonus throws are taken as part of the same turn. If the bonus throws knock down all the pins, the process does not repeat: the bonus throws are only used to calculate the score of the final frame.

### Exit the Game

* Possibility to interrupt the game and exit.

### Restart the Game

* Allow the player to restart the game.

### Graphical User Interface

* Possible use through GUI.

### Multi Player Mode

* Manage many players at the same Game.

### Best Scores

* Save and Show the ascendant best 10 Players scored.

## Non-Functional Requirements

The non-functional requirements describes how the system should function. The non-functional requirements are vital for how the users will perceive the final system since these requirements. This includes usability, reliability, performance, supportability, implementation, interfaces, operational, packaging, and legal requirements.

### Usability

* An intuitive application to enable users to play and interact quickly.

### Implementation

* The Application should be implemented using a native language, C++ will be used as developing language.

### Reliability

* The solution should be covered by unit tests, therefor Google Test is used as testing framework.

### Performance

* The solution should be as fast and as light as possible.
* The solution ensure the correctness of the interaction, thru input check.

### Supportability

* The solution should be platform independent. CMake is a good match for cross platform project management and generation.

### Interface

* The solution should offer different type of interfaces:
  + A console application, using a terminal.
  + A graphical user interface, using Qt.

### Packaging

* As deliverable product the solution should have an easy to deploy application installer

### Legal

* The solution is developed under the MIT License.

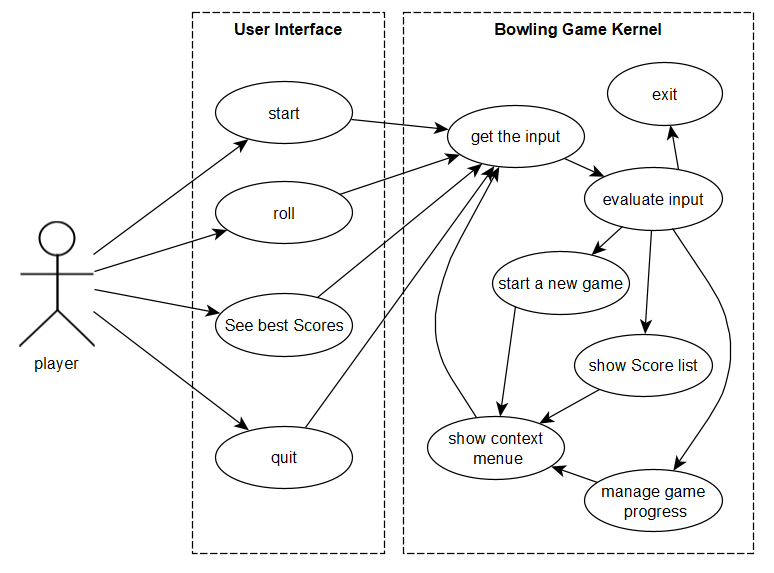
# PROPOSED SYSTEM

## System models

### Architecture

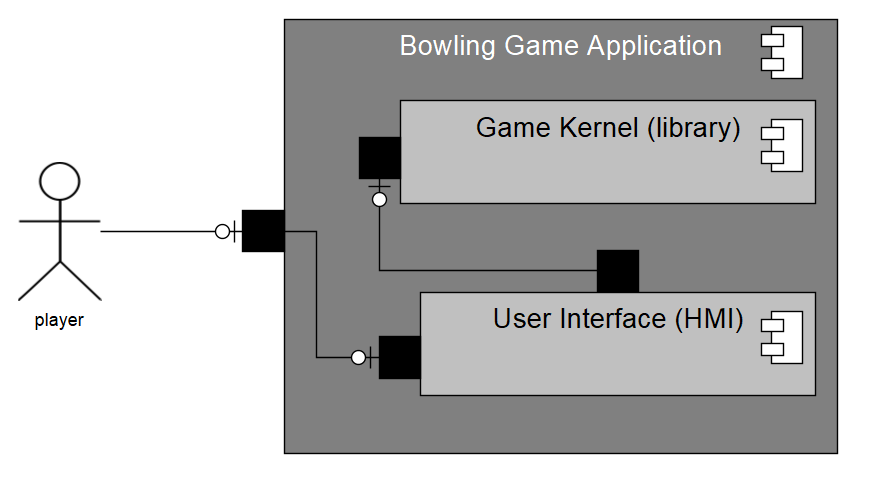
The team performs an object oriented approach in the development progress, and focuses on the modularity of the structure to provide an efficient development in scalability and maintainability.

### Use case model

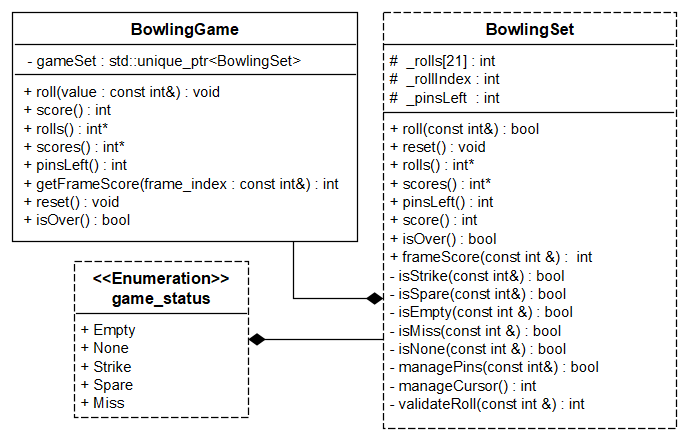
Based on the description of the deployment scenarios, the application covers the following business use cases:

### Component Model

The system can be structured by 2 main components:

* + The kernel: a dynamic library that wraps all the business logic.
  + An interface extension to interact with the user

### Object model

To ensure the development criteria’s described above, the following static diagram describes completely the solution:

### User interface mock-ups

The interfaces should present the frame grid usually used in Bowling. More than that the global score should be shown, and a restart/quit usability (buttons, controls).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Methodology

The Scrum design model is promoted to be used in the project, what is commonly used with agile software development methodology. It involves iterative and incremental development. Since it defines strict and irreversible steps, Waterfall model is not advised to be applied.

## Schedule, timeline, and deadlines

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Mr. Müller |  |  | |
| Ms. Monique |  | |  |
| Mr. Santos |  |  | |
| A. Ben Hassine |  | | |
| Legend: |  | | |

## Cost and benefit

The team has performed the realisation of the solution using the following resources:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Performance | Cost per unit | Total |
| Mr. Müller | 1 x (8 hours) | 50 A$ / h | 400 A$ |
| Ms. Monique | 2 x (8 hours) | 35 A$ / h | 560 A$ |
| Mr. Santos | 1 x (8 hours) | 45 A$ / h | 360 A$ |
| A. Ben Hassine | 5 x (4 hours) | 25 A$ / h | 500 A$ |
|  |  |  | 1820 A$ |

# ASSUMPTIONS

There are some assumptions that facilitate the development of the project:

* The player know how to play Bowling: The rules of game listed in the description is assumed that well known by the player, which exclude the need of introduction session to the game.
* The development should avoid as much as possible third party dependency for the sake of robustness and ease of maintenance.

# GLOSSARY

# REFERENCES

* <https://kata-log.rocks/bowling-game-kata>

# APPENDIX