

Foo et al. Project Documentation

Sandeep Muthu

June 5, 2024

Contents

1	Project Overview	2
2	Getting Started	2
2.1	Prerequisites	2
2.2	Installing	2
3	Building the Project	2
4	Running Tests	2
5	Code Structure	2
5.1	Detailed File Descriptions	3
6	Usage Examples	3
6.1	Example Code	3
7	Future Implementations	4
7.1	Simple Calculation Mode	4
7.2	Advanced Calculation Mode	4
7.3	Implementation Steps	4
8	Contribution Guidelines	4

1 Project Overview

Foo et al. is a software package that calculates the volume of a sphere based on the Foo et al. parameterization. This project demonstrates best practices in C++ development, including modular code design, unit testing with Google Test, and comprehensive documentation.

2 Getting Started

These instructions will help you set up the project on your local machine for development and testing purposes.

2.1 Prerequisites

- **CMake** version 3.10 or higher
- **GCC** or another compatible C++ compiler
- **Git** for version control

2.2 Installing

1. **Clone the repository:**

```
git clone https://github.com/d33psan/Foo-et-al.git
cd Foo-et-al
```

2. **Initialize and update submodules** (if using submodules):

```
git submodule update --init --recursive
```

3 Building the Project

1. **Create a build directory:**

```
mkdir build
cd build
```

2. **Generate build files with CMake:**

```
cmake ..
```

3. **Build the project:**

```
make
```

4 Running Tests

1. **Navigate to the build directory:**

```
cd build
```

2. **Run the tests using CTest:**

```
ctest
```

5 Code Structure

The project is structured as follows:

```

Foo_et_al/
├── CMakeLists.txt      # Main CMake configuration file
├── include/            # Header files
│   └── foo.h
├── src/                # Source files
│   ├── main.cpp
│   ├── sphere/
│   │   ├── sphere.cpp
│   │   └── utils.cpp
├── tests/              # Test files
│   └── fooTest.cpp     # Renamed test file
├── googletest/         # Google Test framework (after FetchContent)
└── README.md          # Project documentation

```

5.1 Detailed File Descriptions

- **CMakeLists.txt**: Configures the build process.
- **include/sphere.h**: Header file for the sphere functionality.
- **src/main.cpp**: Main program file.
- **src/sphere/sphere.cpp**: Implementation of sphere volume calculations.
- **src/sphere/utils.cpp**: Utility functions for the sphere calculations.
- **tests/sphereTest.cpp**: Unit tests for the sphere calculations.

6 Usage Examples

Here's how to use the Foo_et_al library in your own project:

6.1 Example Code

```

#include <iostream>
#include "sphere.h"

int main() {
    double radius = 5.0;
    try {
        double volume = foo::sphere::vol(radius);
        std::cout << "The volume of a sphere with radius-" << radius << "-is-"
        << volume << std::endl;
    } catch (const std::invalid_argument& e) {
        std::cerr << "Error:-" << e.what() << std::endl;
    }
    return 0;
}

```

7 Future Implementations

To enhance the functionality of the Foo et al. library, we plan to introduce a new feature that allows users to choose between simple and advanced calculation modes. Below are the details of the proposed enhancements:

7.1 Simple Calculation Mode

In the simple calculation mode, the existing implementation will be used, where the user inputs a single radius and gets the volume of the sphere.

7.2 Advanced Calculation Mode

In the advanced calculation mode, users will have the option to:

- Enter multiple radii and receive the volumes for each radius.
- Choose the output format for the volumes: fixed-point notation or scientific notation.

7.3 Implementation Steps

To implement the advanced calculation mode, the following steps need to be taken:

1. Modify the user interface to accept multiple radius inputs.
2. Implement functions to calculate volumes for multiple radii.
3. Add options for the user to select the output format (fixed-point or scientific notation).
4. Update the main program to handle the new advanced calculation mode and format the output accordingly.
5. Write unit tests to verify the functionality of the advanced calculation mode.

These enhancements will make the Foo et al. library more versatile and suitable for scientific projects.

8 Contribution Guidelines

We welcome contributions from the community! To contribute:

1. Fork the repository.
2. Create a new branch (`git checkout -b feature-branch`).
3. Make your changes.
4. Commit your changes (`git commit -m 'Add new feature'`).
5. Push to the branch (`git push origin feature-branch`).
6. Open a pull request.

Please make sure to update tests as appropriate and follow the coding style of the project.

GitHub Repository: https://github.com/d33psan/Foo_et_al