Requirement Specification

1. Introduction

[Particle Filter Simulator](https://github.com/HenryJaQiu/GRP-17-18-Group6/tree/master/Particle%20Filter%20Simulator) is a lightweight application which is mainly used by college students and teachers. Because Particle Filter Simulator is aimed for teaching or processing image and data, in order to meet the main needs of users and provide positive user experience, a large amount of significant requirements should be carefully considered and implemented. In this part of report, it will illustrate the process of gathering and implementing requirement and provide a list of different types of requirements. For each specific requirement, it will be analyzed and explained in more details.

1. Use case diagram

Use case diagram could describe how users interact with system and it could be the first step from requirement specification to software implementation. The actors, use cases and system boundary will be shown as follow. It could help developers implement the main function of Particle Filter Simulator and provide a clear diagram for users to understand each parts of the software.

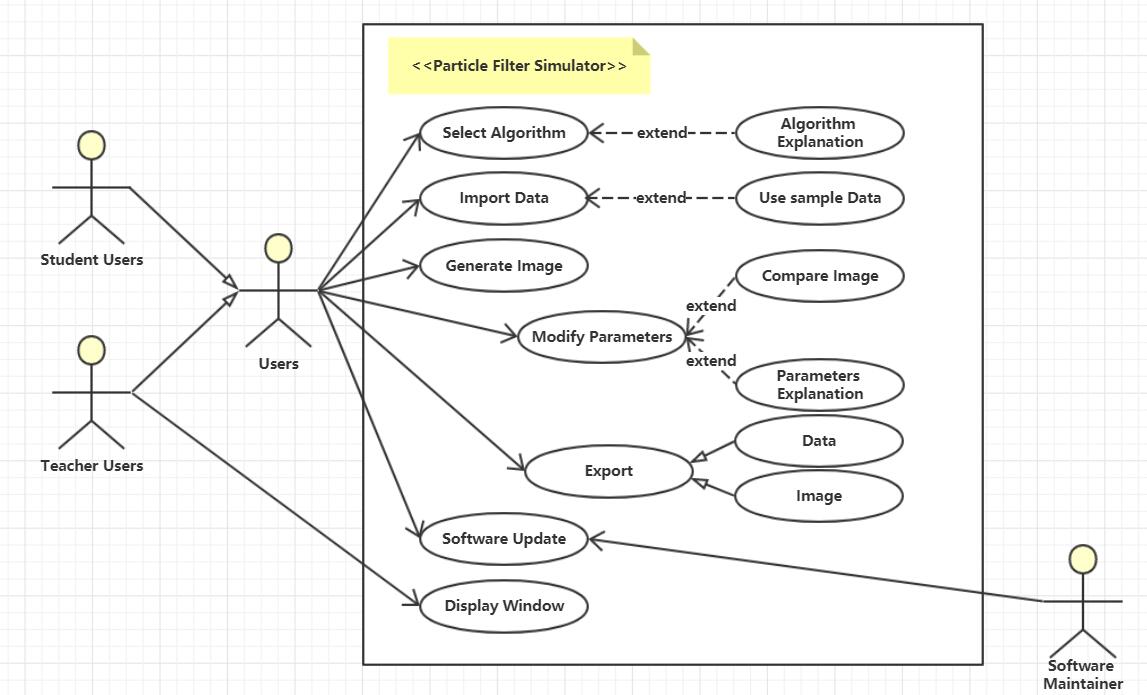


Figure 1: Use case diagram

1. Prototype

Software prototype provides a brief user interface and divides the whole software into several parts. Developer could better implement division and cooperation by focusing on each different component in prototype. It also provides a feasible option to validate if the software design achieves expected user experience.

* 1. Start interface

After open the software, user should first select different algorithms. It c first impression of user experience and help users gets start easily.

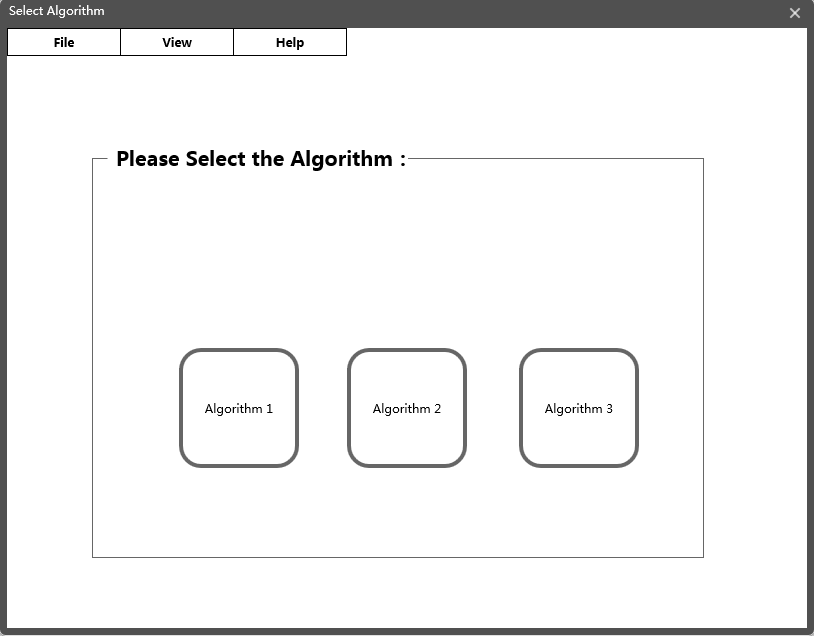


Figure 2: Start Interface

* 1. Navigation bar

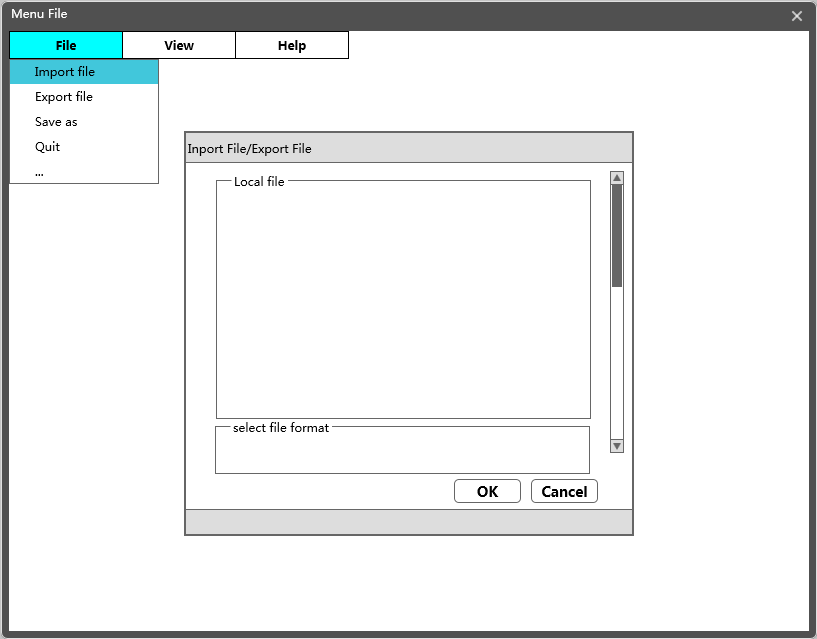
Navigation bar is placed on the top of window and it contains three main options: File, View and Help. File options include the function of operating input and output file. View options include the function of setting window on the top or changing the software appearance. Help options include the help document and software update. 

Figure 3: Navigation bar-File

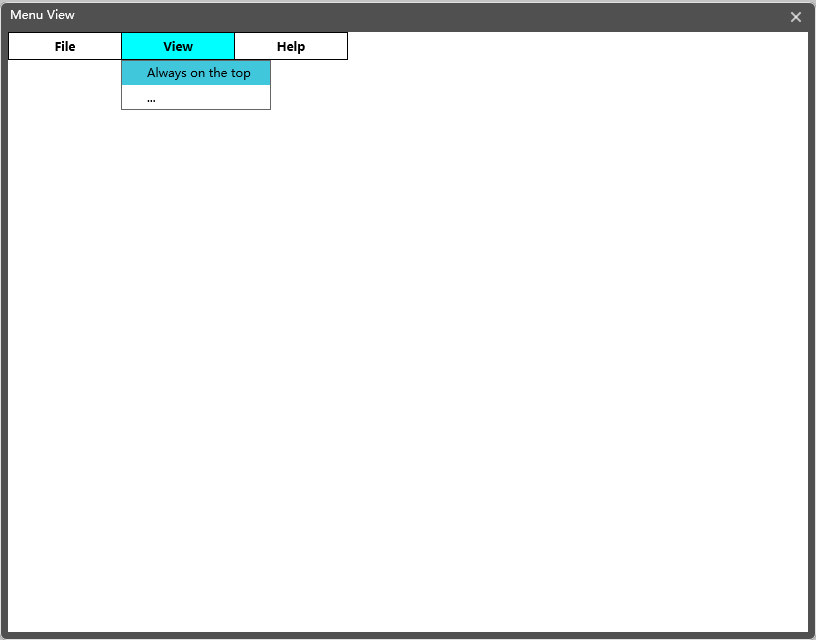


Figure 4: Navigation bar-View

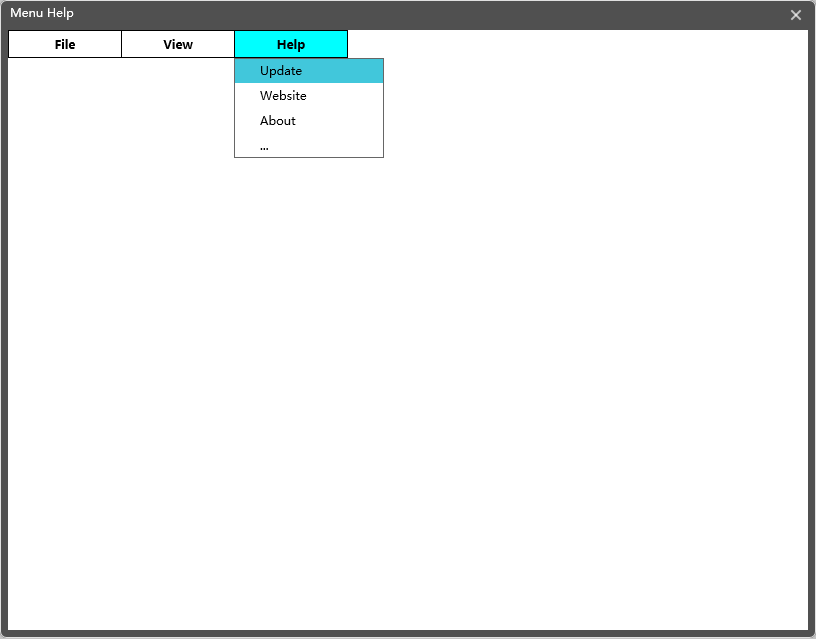


Figure 5: Navigation bar-Help

* 1. Main interface

Main interface of generating data and image which include three components: function button, outcome image and parameter controller. Function button include several functional buttons to deal with data and image. Outcome image include an area to generate image with coordinate axis and two scroll bar. Parameter controller contain a list contain parameter name, several slider used to adjust parameter value and a start and refresh button.

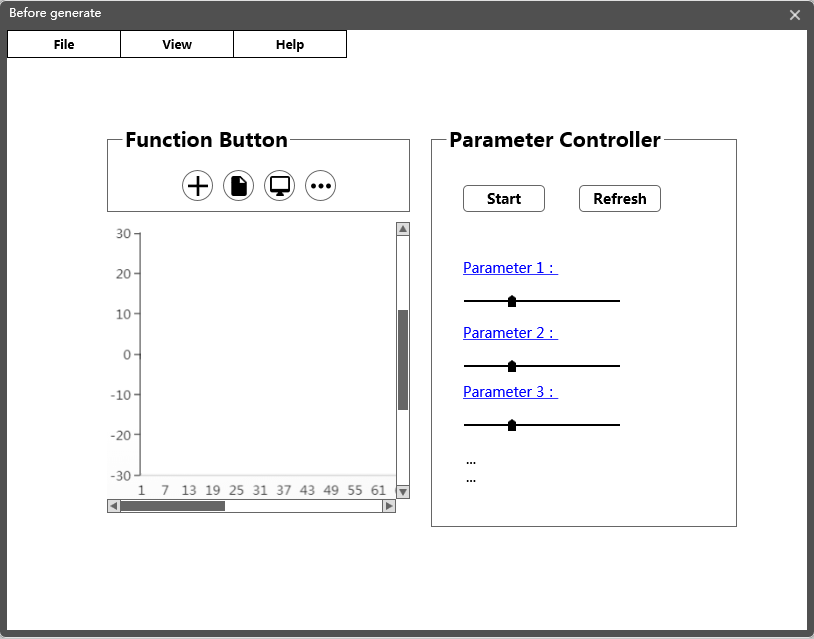


Figure 6: Main interface-Before generate

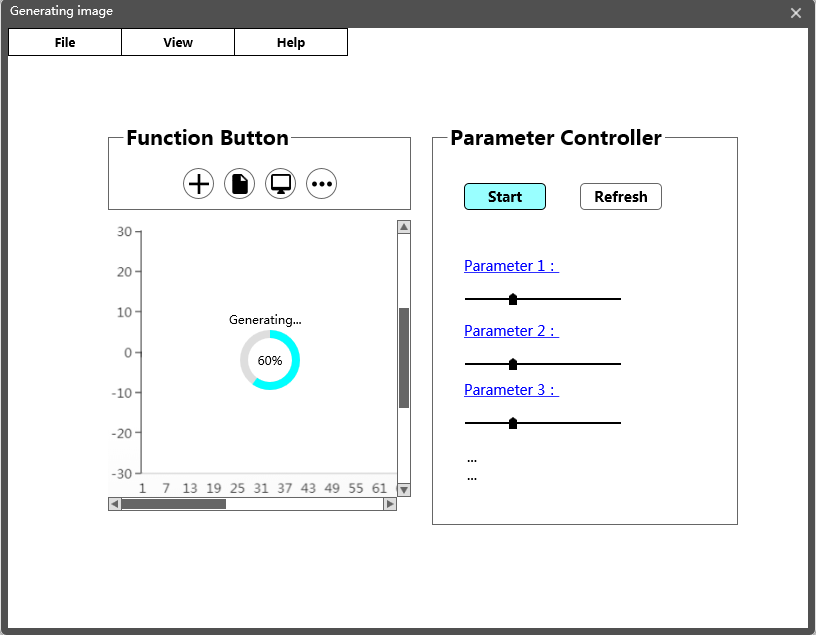


Figure 7: Main interface-generating image

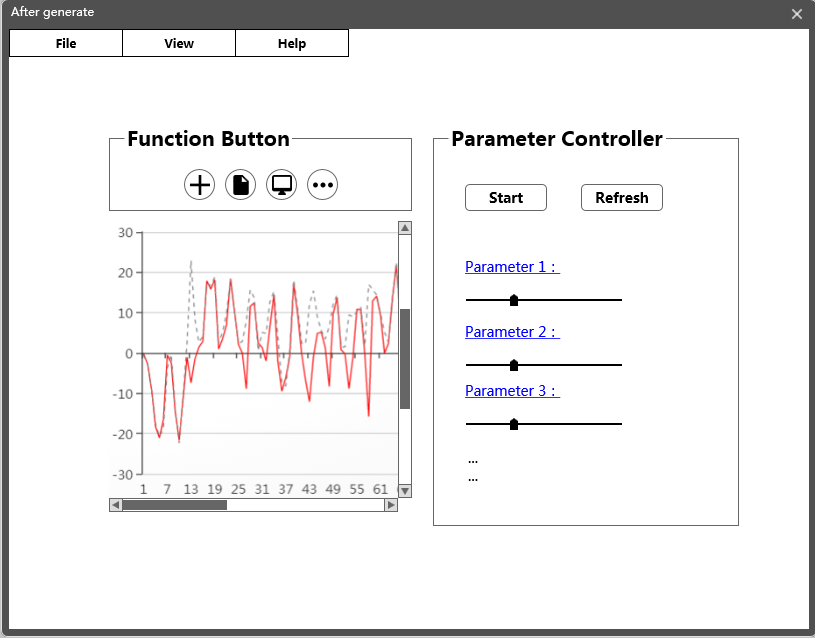


Figure 8: Main interface-After generate

1. Requirements list

During the process of software design and requirements Implementation, developers found that because of the cost limitation of the software, some requirements is difficult to implement and there are also some new requirements need to be considered. In this part, mainly of the requirements that appear in the software design process will be given and which of them have been implemented or not will be introduced.

* 1. Functional requirements

The main function of Particle Filter Simulator is using the algorithm selected by user to generate the outcome image and provide several operations to image or data. These are the main functional requirements appearing during the software design process:

1. The software should illustrate the correctly usage method in help document.

2. The software should provide the function of selecting different algorithms.

3. The software should display the brief explanation of algorithms when user is selecting an algorithm.

4. The software should provide some existing sample data to users without other specific data.

5. The software needs prompt and check the format of the file when users import specific sample data.

6. The software should use the algorithm which is selected by user to deal with the import sample data and generate the outcome image.

7. The software should provide modifiable parameter list and brief explanations of each parameter.

8. The software should have the function of modifying parameters which are initially set as default value.

9. The software should implement the function of exporting images, data, or both of them according to the user’s decision.

10. The software should permit users to combine two images together to compare the differences between them.

11. The software should provide the function of clearing the current data and results if users want to import another sample data or select another algorithm.

12. The software should provide a function of putting the software window into the top.

13. A restart button should be provided to handle the situation of software crashing.

14. The software need to check version update automatically and submit bugs which are collected by users to software maintainer.

15. When the software is using algorithm to generate image, a progress bar or a waiting hint should be provided which is used to display the running state of software.

16. The range of permanents should be limited to avoid the situation that increasing running time of algorithm and software crashing.

4.2 Non-functional requirements

The non-functional requirements are given as follow which include the environment requirements, performance requirements and data requirements:

1. The window size of the software should be suitable for demonstration

2. The user interface should be convenient for user activity.

3. The running time of using the algorithm to deal with the sample data should be as quick as possible.

4. The system resources occupied by software should be limited.

5. The software needs to run on major platforms and operating systems.

6. The software should be convenient for update and extension.

7. The outcome image should have positive distributing accuracy when the value of several parameters is limited.

8. The software requires appropriate software capacity to be convenient for users download and use in a short period of time.

9. It should be convenient to passing parameter data between each component in different part of software.

10. The software is free, it is better that the development and maintenance costs should be properly reduced.