**The DiagramManager**

Development document

Author: AleOleOrl,

aleoleorl@gmail.com

July, 2024

# **Change log**

|  |  |  |  |
| --- | --- | --- | --- |
| ## | Date |  | Changes, information |
| 1 | 09.07.2024 |  | Initial version. Added an information about current development version of the application. Added a plan of Development handling. Added a plan for version 0.6. |
|  |  |  |  |
|  |  |  |  |

# **Preview**

The DiagramManager project was conceived as a demonstration project of the common developer skills. However, during the development the idea to improve its possibilities and make it some kind useful tool make the necessarily to have a clear direct plan of the next steps. That’s why this document was created.

# **Preconditions**

# **Repository**

<https://github.com/aleoleorl/diagramMaker.git>

# **Version 0.5**

This version is presented in main branch and should means as a default version of the project.

Useful content of the version:

* main board with main Canvas object appCanvas;
* menu for items creation;
* menu for an item correction;
* menu for items navigation;
* information line;
* tools menu for drawing items;
* possibility to create and delete items: label, paint, multiline, connector for multiline;
* top menu with possibilities to save and load current document, to check the application versions, to handle view of available menus.

# **Version 0.6**

This version should be next in a row and it will be the first version with a plan of development.

Structure of development.

1. Should be created the list of expected changes.
2. Should be provided the structure of git handling of each change.

No necessary of stricter schedule right now.

## **Expected changes list**

There is a row of changes that should be done in the default engine of the project.

1. Replace Variables in all item classes and their data classes by Properties. The main idea is to make it easier and more universal handle for other updates.
2. Rehandle of parameters structure inside Item classes.
   1. Put all default variables into CommonParameter:DefaultParameter
   2. Create list or dictionary of Parameter objects.
3. Item inherited methods using:
   1. Rehandle of using item object inside each item class by using common inherited methods.
   2. The same changes for parameter handling methods.
   3. Delete SetParameters method, should be SetParameter only. Also need to analyze null parameters.
4. Rehandle the Connector idea.
   1. Add an object Connector. Contains a list of connections, an enum object of my connector type, an enum object of the current connector status.
   2. Each Item will contain a Container object. Currently it will be statuses: None, Master, InternalConnector.
5. Should be rehandled Multiline Item. The reason is to make it more clarified because in future it will be the base for other complex tube objects.
6. Add groups in menu panels and possibility to minimize & open back them.
7. Add layers by using SetZIndex method.
   1. Correct navigation menu to show layers and its items.
8. Correction of item menu to get possibility to handle all available parameters
   1. Each parameter property should have a property flag of availableness or not for changes.
   2. Flags should be available in SetParameter menu.
9. Rehandle of save-load serialization/deserialization. It should be more universal. Handling of data must not involve correction of save-load methods each time.
10. Need a test application and a test plan for the testing of the application from user point of view. Means to check the correct behavior of the main presented user functions.

# **Plan of handling**

Each task point will be developed in the personal branch. Here will be created new documents if it will be necessary (i.e., test project and test plans).

After the end of task its branch will be merged into a special branch dev. This branch will be merged into master branch in the end of task list handling for version 0.6.