

Тестовое задание на позицию GUI C++ разработчика

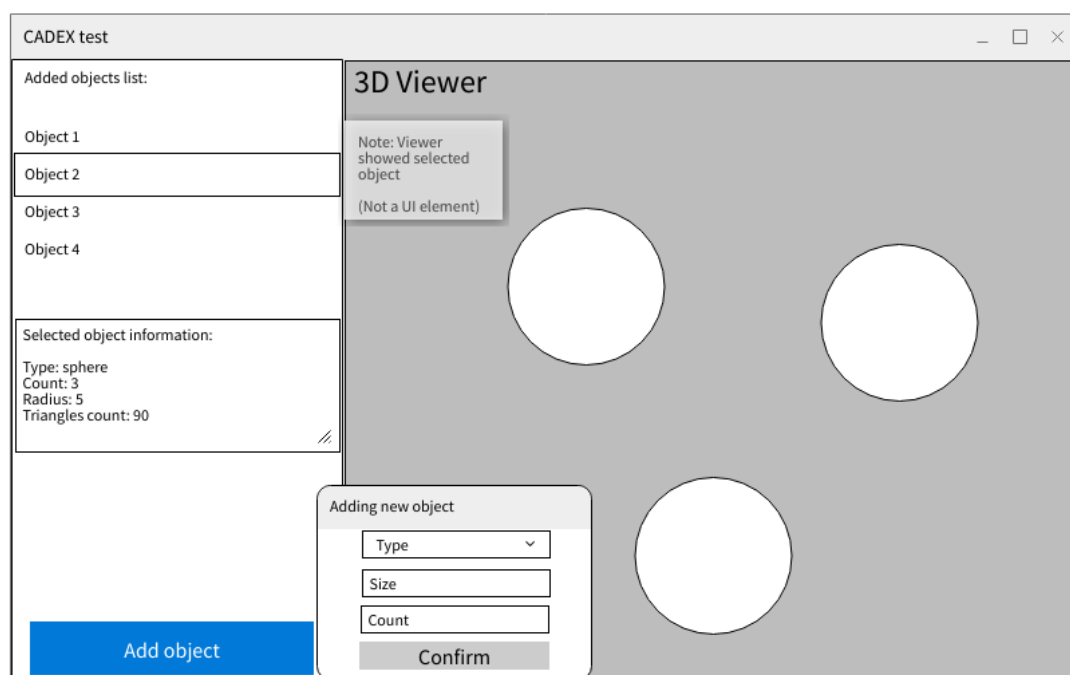
Задание нацелено на оценку уровня владения языком программирования на C++ с использованием Qt библиотек, в частности работе с QML и способности разобраться в новом материале при необходимости.

Задание составлено на английском языке, т.к. большая часть технической информации в будущем будет доступна именно на нем.

Пожалуйста, внимательно прочтите все обязательные требования! Вы также можете выполнить дополнительные задания (под «звездочкой»), это может стать Вашим преимуществом перед другими кандидатами.

Ответ следует направить на адрес jobs@cadexchanger.com, в виде ссылки на проект на github, который должен включать исходный код (*.h, *.cpp, *.qml), а также проектные файлы.

Create a Qt-based GUI application which can display 3D primitives in a 3D view:



1. The main page of the application should contain the 3D viewer, a panel with added 3D objects list and a section with selected object information. The possible design presented above.
2. Application should support ability of adding 3D object of given (input from user) object type, object parameters and object count, e.g. adding 10 boxes with given length, width and

depth. Objects location in the viewer shouldn't be the same and can be provided in random manner.

3. The list of the objects should support selection of the elements. Once an element was selected, the viewer should dynamically update its content - remove previously selected object and display current selected object.

4. Section with selected object information should contain: type, parameters, count and displayed triangles count. Should be dynamically updated after selection change.

(*) Additional optional requirements:

5. Add support of removing selected object from list.

6. Implement support of async displaying - all actions related to object creation and displaying should be happened in additional thread.

Requirements to the implementation:

1. UI of the application should be implemented using Qt QML.

2. 3D view must be OpenGL-based (e.g. using Qt OpenGL or Qt3D). 3D view should be interactive - support rotation, at least.

3. Support of adding at least 2 types of objects, e.g. box and sphere. Other types are at your discretion and are not mandatory.

4. List of added 3D objects should be implemented using c++ based model.

5. The implementation must use virtual methods.

6. Has neither explicit memory deallocation nor memory leaks.

7. The implementation may use constructs of C++11 or higher.

8. The code must compile with any compiler of your choice (gcc, Visual C++, etc).