Лабораторная работа 4
Qt

Цель работы

Изучить возможности фреймворка Qt для создания приложений с графическим интерфейсом пользователя (GUI).

Стандарт языка и установка

C++17 или новее. Требования к работам и Оформление исходных текстов.

Программа должна собираться Qt 5.12.12. Подробнее: Qt (памятка) и Qt Creator (памятка)

В репозиторий работы обязательно должен быть приложен *.pro файл, система сборки – qmake.

Для выполнения работы необходимо установить модуль Qt Data Visualization и в .pro файле прописать `QT += datavisualization`

Описание

Создать десктоп приложение с пользовательским интерфейсом (GUI), позволяющее отобразить простой 3D-график поверхности и настраивать свойства отображения через виджеты.

Приложение должно позволять сохранять текущие настройки в .ini файл рядом исполняемым файлом и загружать настройки из него.

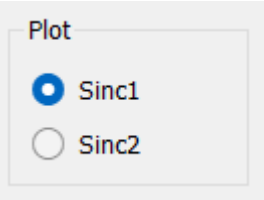
В программе должен быть реализованы **класс MainWindow**, хранящий виджеты приложения, и **класс Plot**, отвечающий за формирование данных для их дальнейшей визуализации. Классы должны быть описаны в .h файлах,а реализация их методов – .cpp файлах. Создавать свои файлы (.h/.cpp) со вспомогательным кодом не запрещено.

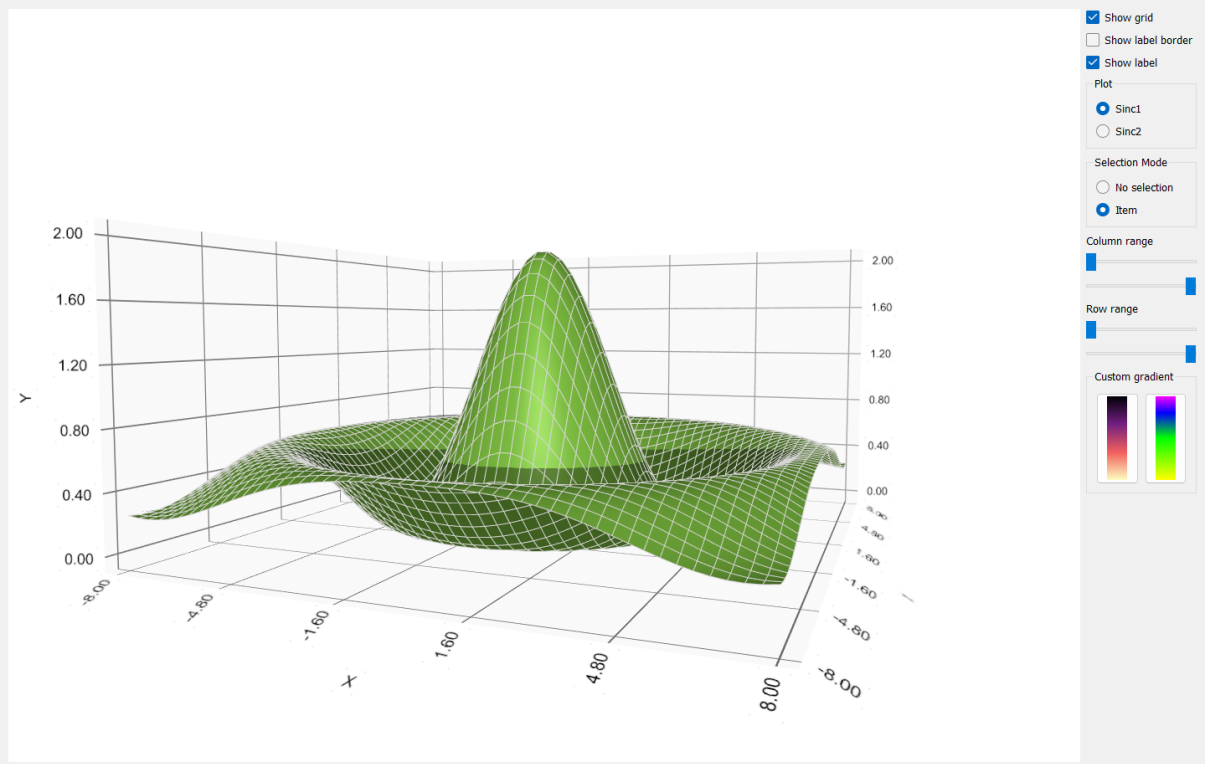
Пользовательский интерфейс описывается кодом (без использования .ui/QML).

Должно быть реализовано:

1. Просмотр графика функции:
    a. sinc(distance_from_zero)
    b. sinc(x)*sinc(z)

    Функция выбирается при помощи RadioButton. Пример оформления:

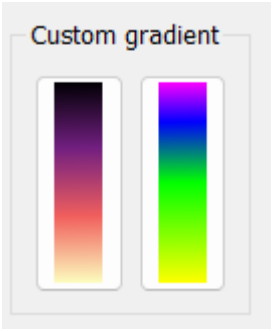




2. Настройки должны отображаться с одной стороны (на скрине выше приведён пример отображения настроек справа от графика).
    a. Все настройки сгруппированы по смыслу.
    b. Все названия поясняют настройки.

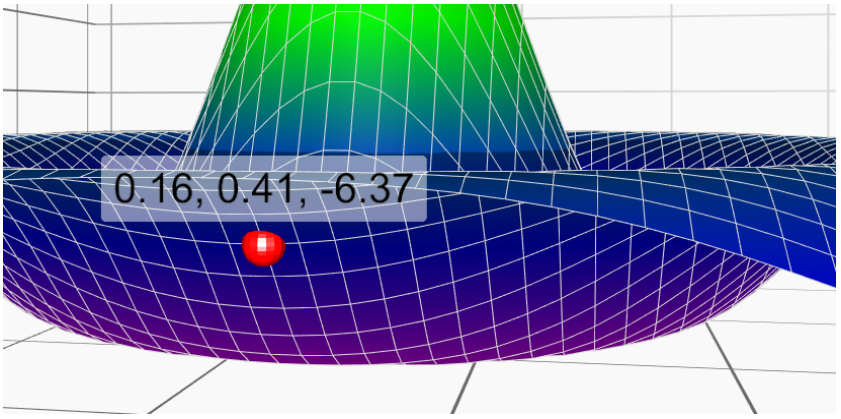
c. При изменении размера окна, настройки не изменяются в размерах, изменяет размеры только график.

3. Должен выбираться градиент цветов, при помощи которых отображается график.
   a. Пользователю должно быть предложено как минимум 2 различных градиента, между которыми он может переключаться. Хотя бы один из градиентов должен быть задан через опорные значения цветов (см. ссылку 1). Пример оформления:



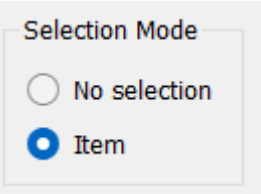
   b. Градиент должен запоминаться для каждого из двух графиков. Если на графике 1 пользователь выбрал градиент1, а для графика2 – градиент2, то при переключении снова на график1 он должен отображаться с градиентом1.

4. Возможность выбирать точку на графике.
   a. Точка, выбранная нажатием левой кнопки мышки по графику, должна быть выделена, и над ней должны выводиться координаты. Пример ниже:

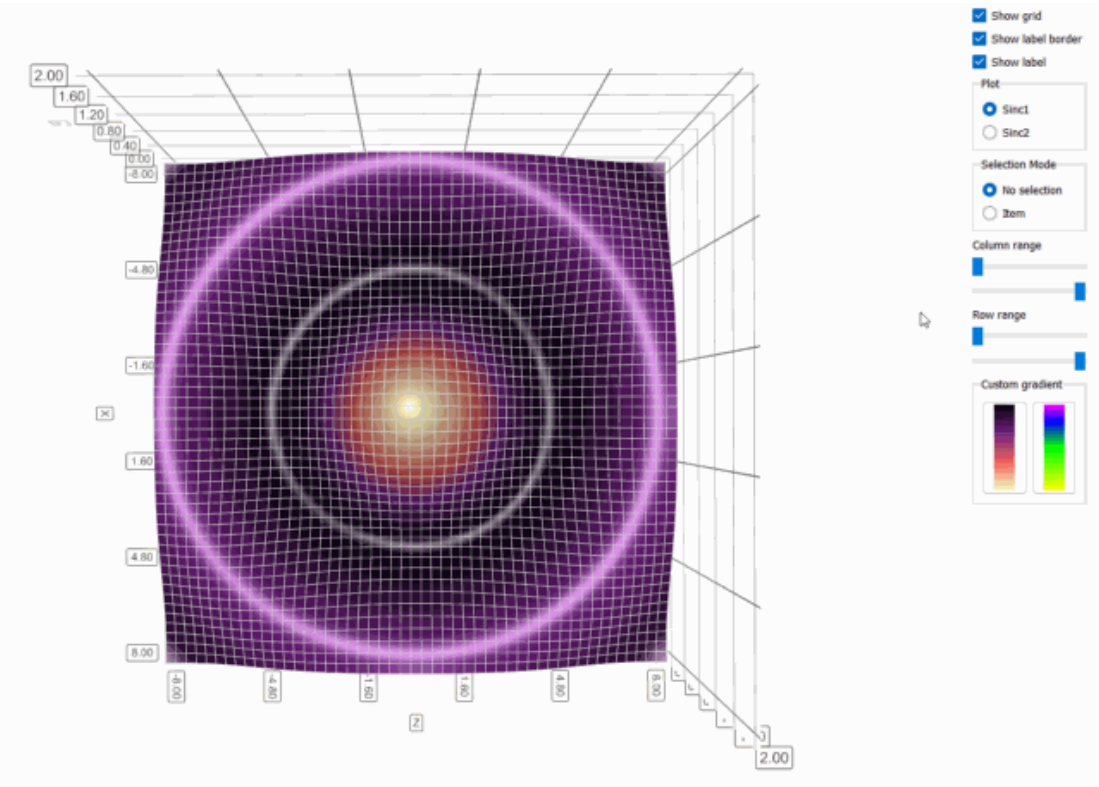


   b. В StatusBar окна должны выводиться эти же координаты.

c. Возможность выделения точки должна быть отключаема через настройки. При отключении этой опции на графике точка не должна выделяться, координаты никуда не выводятся.

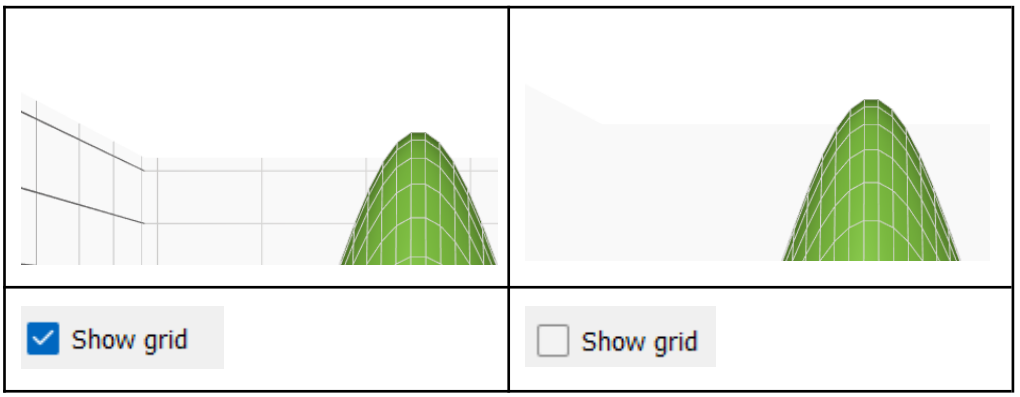


5. Выбор диапазона отображаемых значений и количества шагов.
   a. Начальный диапазон по осям [-10, 10].
   b. Количество шагов: 50.
   c. Значения отображаемого диапазона и шага должны отображаться в пользовательском интерфейсе рядом с соответствующей настройкой (на примере ниже это не реализовано). Поле должно быть редактируемым.
   d. Через GUI должна быть возможность менять диапазон осям x и z с обеих сторон, “урезая” отображаемый график. Это не должно приводить к пересчёту данных.



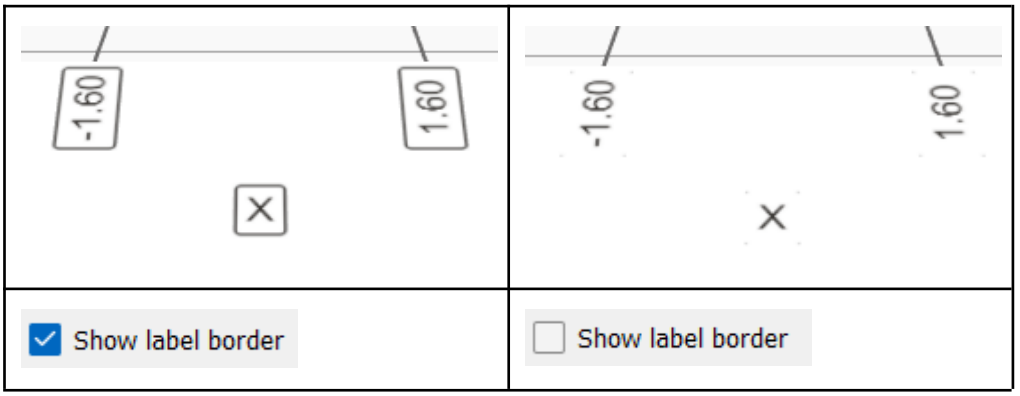
e. Через GUI должна быть возможность менять количество шагов по каждой оси графика. Данные в этом случае пересчитываются.

6. Настройки отображения графика.
   a. Должен быть CheckBox, отвечающий за отображение сетки



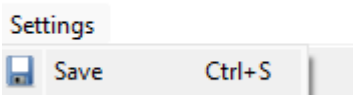
   b. Должен быть CheckBox, отвечающий за отображение надписей

   c. Должен быть CheckBox, отвечающий за отображение границ надписей (активный только при включенном отображении надписей)



7. Возможность сохранения настроек.

a. Настройки должны сохраняться в .ini файл, расположенный рядом с исполняемым файлом.
b. Настройки автоматически загружаются при запуске программы.
c. Программа должна проверять прочитанные настройки, и игнорировать некорректные значения (а не падать).
d. Должна быть возможность сохранения и загрузки настроек через: MenuBar, Toolbar и Shortcuts



e. При наведении на опции (как в MenuBar, так и в ToolBar) в StatusBar должна выводиться подсказка StatusTip.

8. [Бонусные баллы] Перевод всего отображаемого в интерфейсе текста на несколько языков. Должны быть реализованы хотя бы: английский и русский. Должна быть возможность переключить язык через меню, в меню язык всегда называется на своём языке и не переводится. Язык сохраняется вместе с другими настройками. Язык по умолчанию – выбирается в соответствии с языком системы.

Использовать STL запрещено, вместо этого у вас есть классы Qt (QVector и т.д.), которые вы хотите использовать в коде.

Полезное:

1. Viridis Palette Generator
    a. Color gradient - Wikipedia
    b. Turbo, An Improved Rainbow Colormap for Visualization
2. https://doc.qt.io/qt-5/qlineargradient.html
3. https://doc.qt.io/qt-5/qmainwindow.html
4. https://doc.qt.io/qt-5/signalsandslots.html

Формат сдачи работы

Автотестов на Github не будет. Отправок на проверку перед защитой также не будет. На защите код будет собираться на ПК проверяющего.

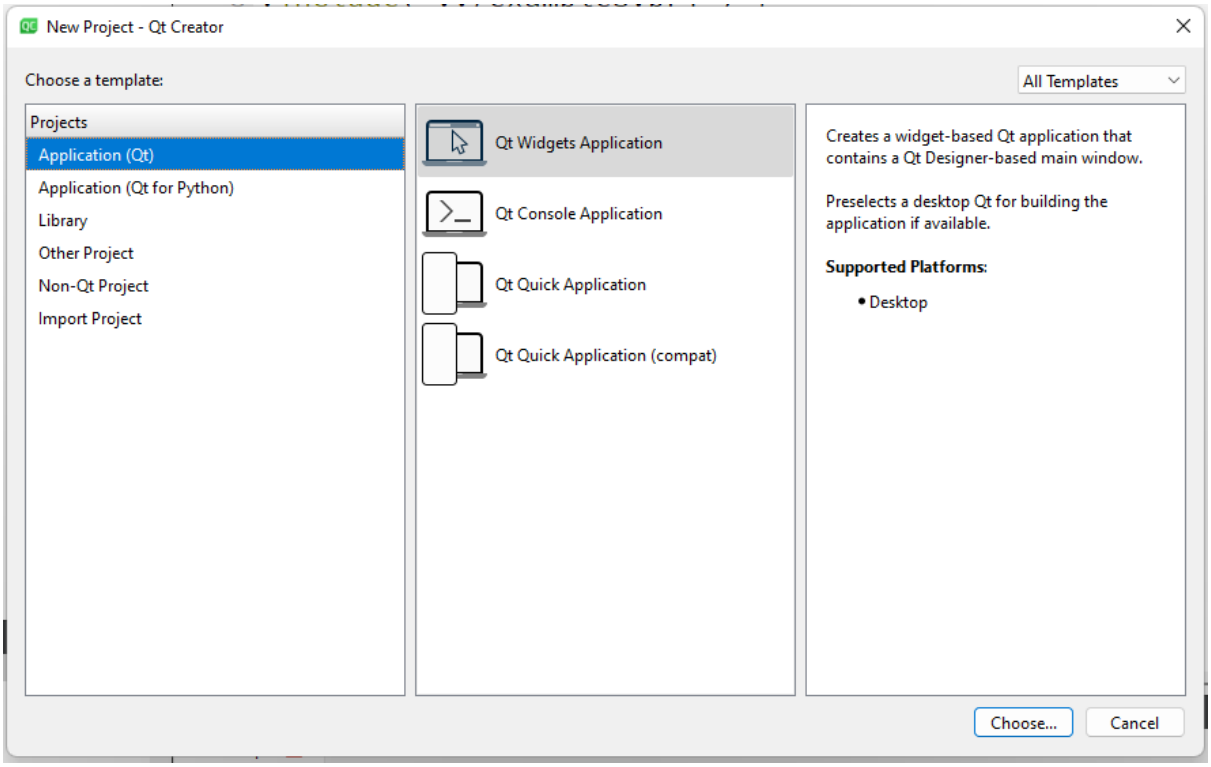
Когда вы готовы показать работу, то:

1. загружаете код на github,
2. записываетесь на защиту,
3. приходите по записи,
4. показываете код и собранную программу.

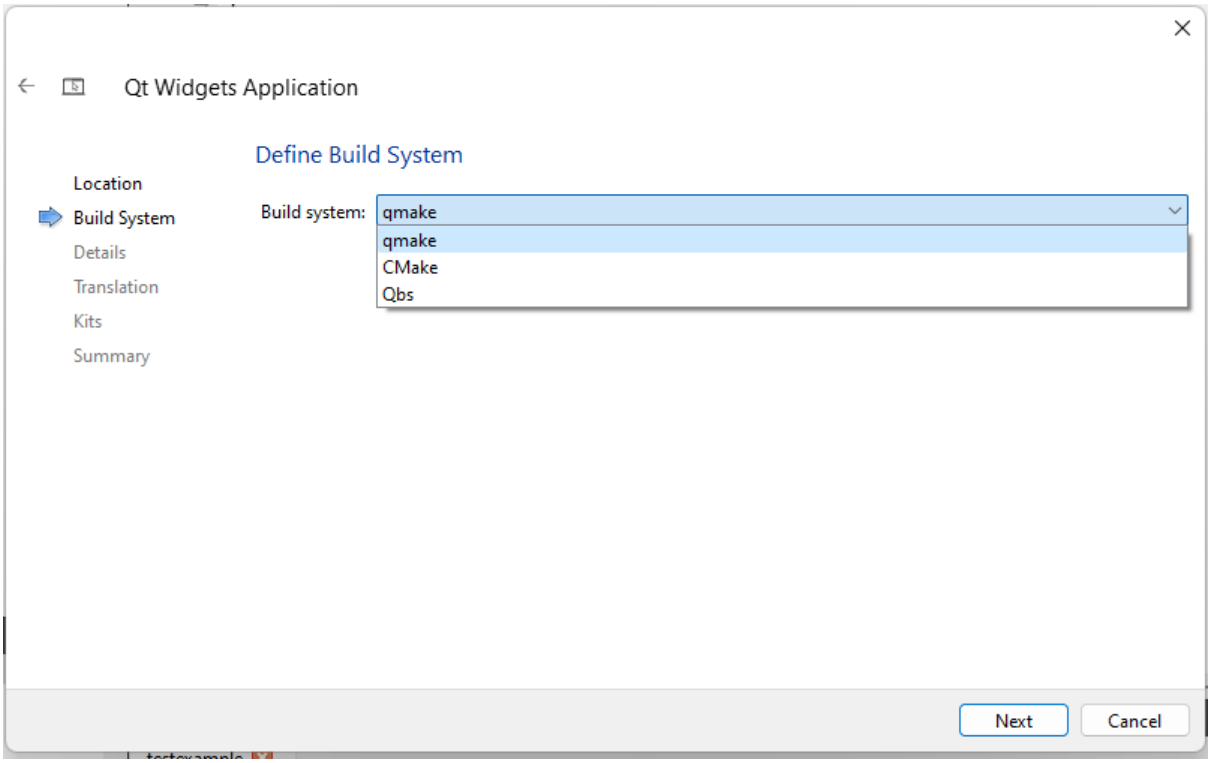
Если работа требует явных доработок и вы пришли не в последний день, то вам будет дана ещё одна попытка защиты-сдачи.

Создание проекта в QtCreator

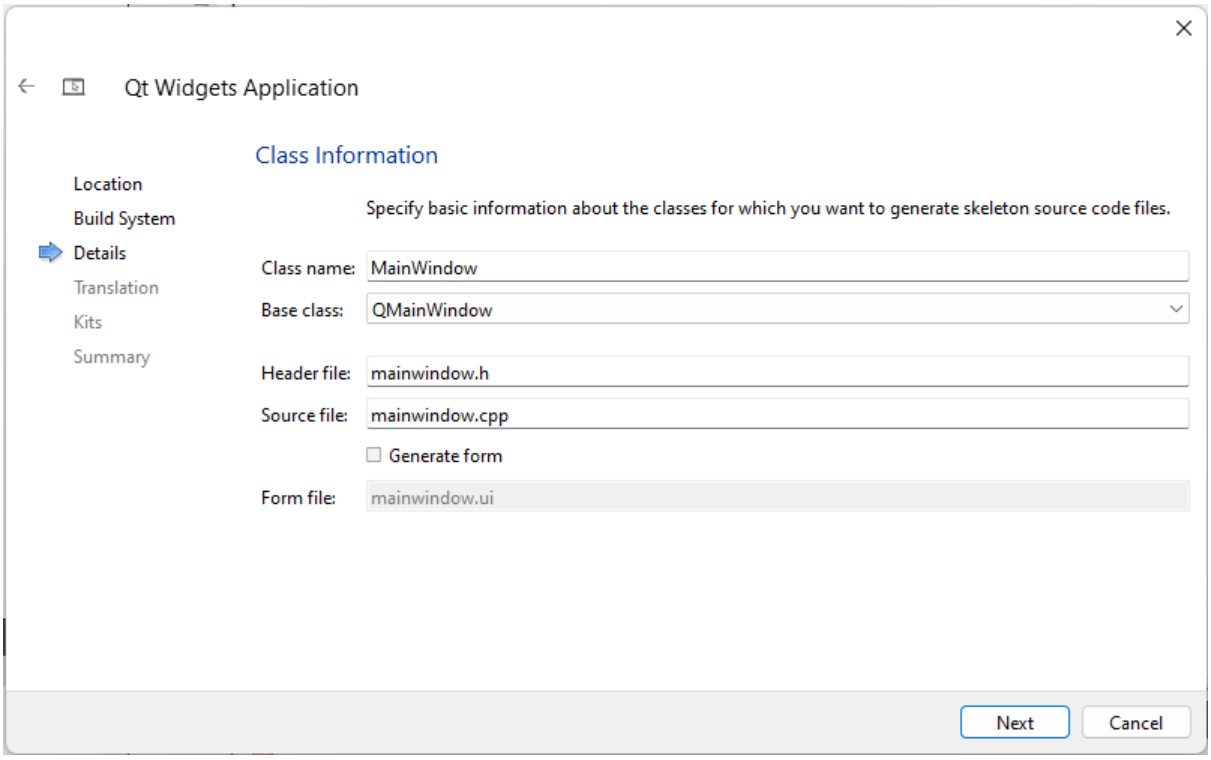
Qt Widget Application



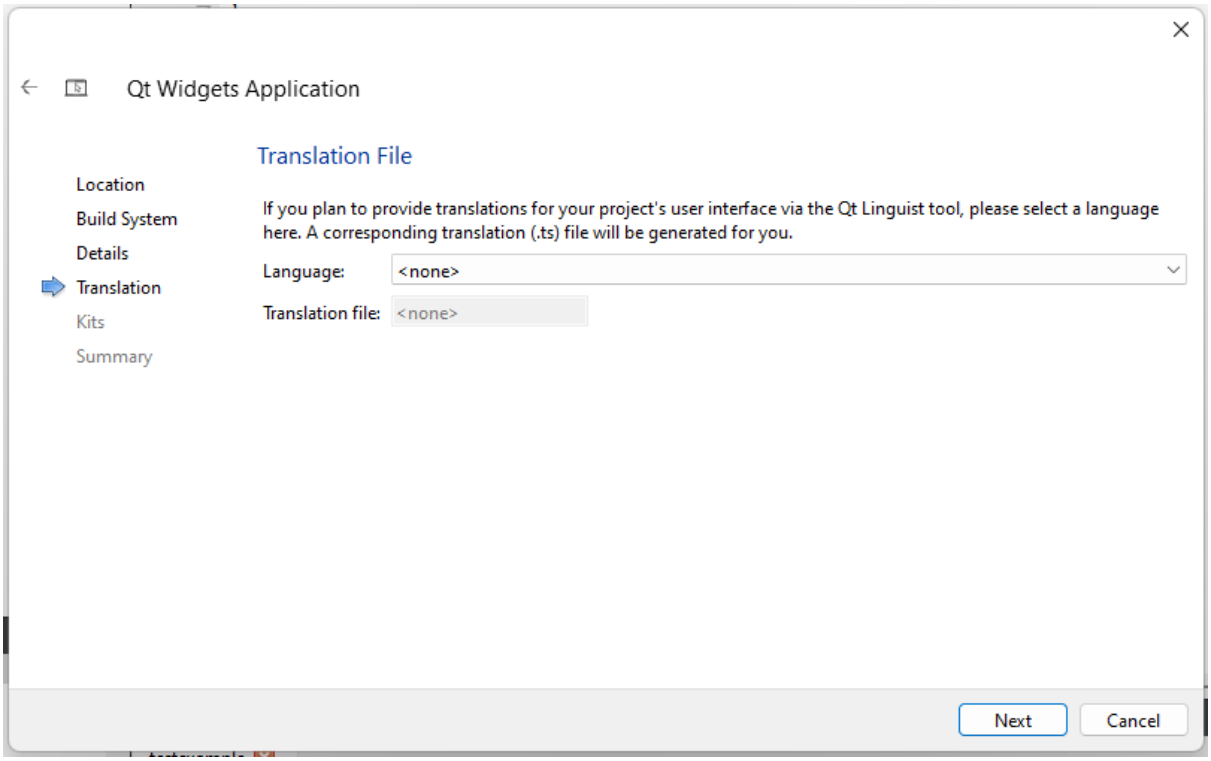
Система сборки: qmake



Автогенерируемый класс: можно выбрать MainWindow, так у вас сразу будет основа для одного из требуемых классов. Обязательно отключаем Generate form, дабы не создался .ui файл.



Файл перевода. Если п.8 не делается, то оставляем None.



Kit выбираете любой из установленных.