Лабораторная работа №4

Qt

Инструментарий и требования к работе

Работа выполняется на C++. На сервере сборка под C++20. 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.

Задание

Создать приложение с пользовательским интерфейсом (GUI), позволяющее играть в игру Сапёр. Правила игры: Сапёр (игра) — Википедия Онлайн: https://minesweeper.online/		

Детали реализации

Мины расставляются после первой открытой клетки. Первая раскрытая клетка, таким образом, не может быть миной.

В случае открытия клетки, вокруг которой нет мин, все клетки до окружающих непустых включительно должны раскрываться автоматически.

Размер поля по умолчанию: 10 на 10.

Число мин по умолчанию: 10.

Обработка нажатий:

- открытие клетки (ЛКМ);
- установка/снятие флажка на клетке (по нажатию ПКМ);
- возможность быстро открывать клетки: нажатие на СКМ на клетку с номером раскрывает все закрытые поля в этой области, если все мины в этом районе были отмечены флагами и подсвечивает неоткрытые клетки в противном случае.

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

Пользовательский интерфейс описывается кодом (без использования .ui/QML). Использовать иконки/изображения не запрещено (даже приветствуется).

Должно быть реализовано:

1. Запуск новой игры с настройкой размеров поля и числа мин (все параметры должны быть с проверками на корректность введённых данных) в появляющемся диалоговом окне.
2. После задания параметров новой игры должно появится поле с закрытыми клетками, по которым можно нажимать, открывая их содержимое.
3. В случае проигрыша (попадания на мину) всё поле раскрывается и появляется поздравительное сообщение с тем, что игра завершена проигрышем, поле становится неактивным. Ячейка с последним нажатием должна визуально отличатся от других ячеек с минами.
4. В случае выигрыша появляется сообщение о выигрыше.
5. При запуске с аргументом командной строки `dbg` (значение `argv[1]`) должна появиться кнопка/галочка, позволяющая “подглядывать” за скрытым состоянием поля.

6. Приложение должно позволять через пользовательский интерфейс (menu и toolbar) запускать новую игру с теми же параметрами или с заданием параметров.
7. При закрытии программы с активной игрой текущее состояние должно сохраняться в .ini файл рядом исполняемым файлом и восстанавливаться из него при следующем запуске.
8. Отдельно будет оцениваться “правильность” поведения интерфейса: адекватная реакция на масштабирование окна, …
9. На дополнительные баллы можно реализовать локализацию: поддержку нескольких языков, возможность их переключения во время работы программы и автосохранение при выходе.

Полезности

- Qt (памятка)
- Qt Creator (памятка)
- Общая информация: Сапёр (игра) — Википедия
- Онлайн-версия классического Windows-сапёра: https://minesweeper.online/
- QWidget Class | Qt Widgets 6.7.1
- QMessageBox Class | Qt Widgets 6.7.1
- QGridLayout Class | Qt Widgets 6.7.1
- QPushButton Class | Qt Widgets 6.7.1

Формат сдачи работы

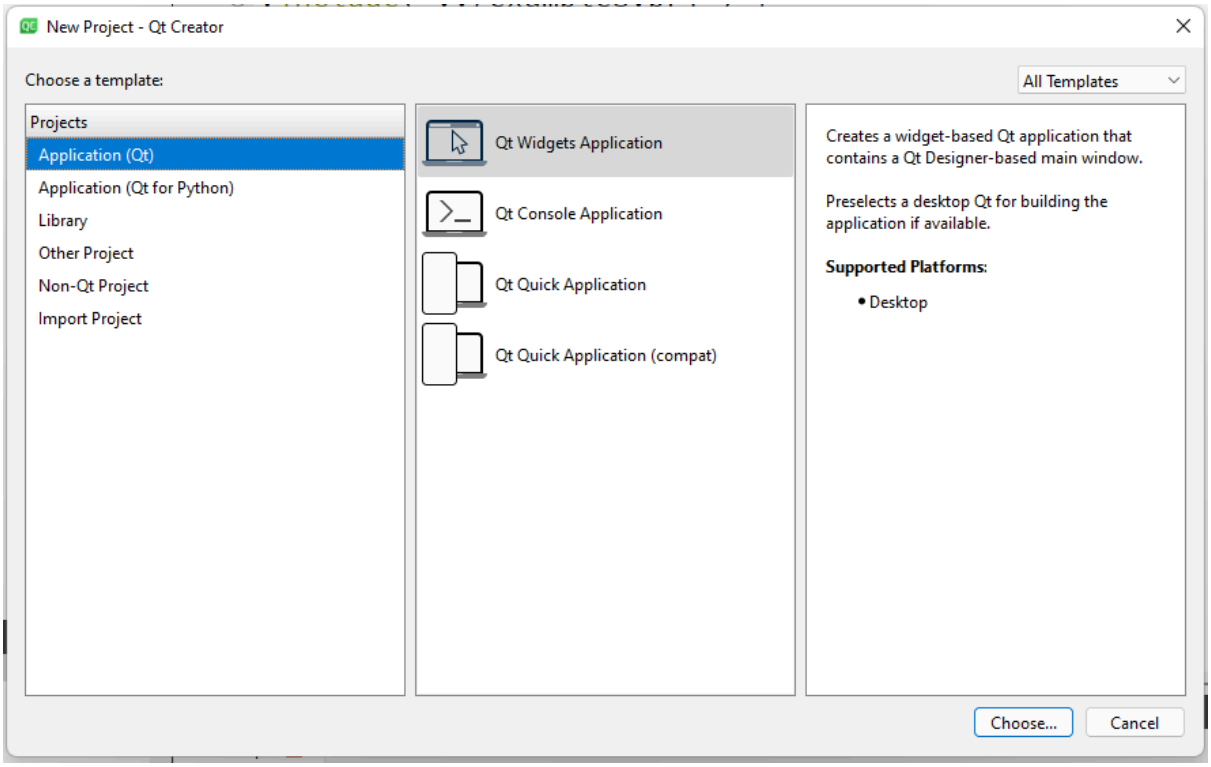
Автотестов на Github не будет. Проверка перед защитой будет заключаться в заполнении проверяющим чек-листа, в котором будет отмечен функционал, выносимый на защиту. На защите код будет собираться на ПК проверяющего.

Когда вы готовы показать работу, то:

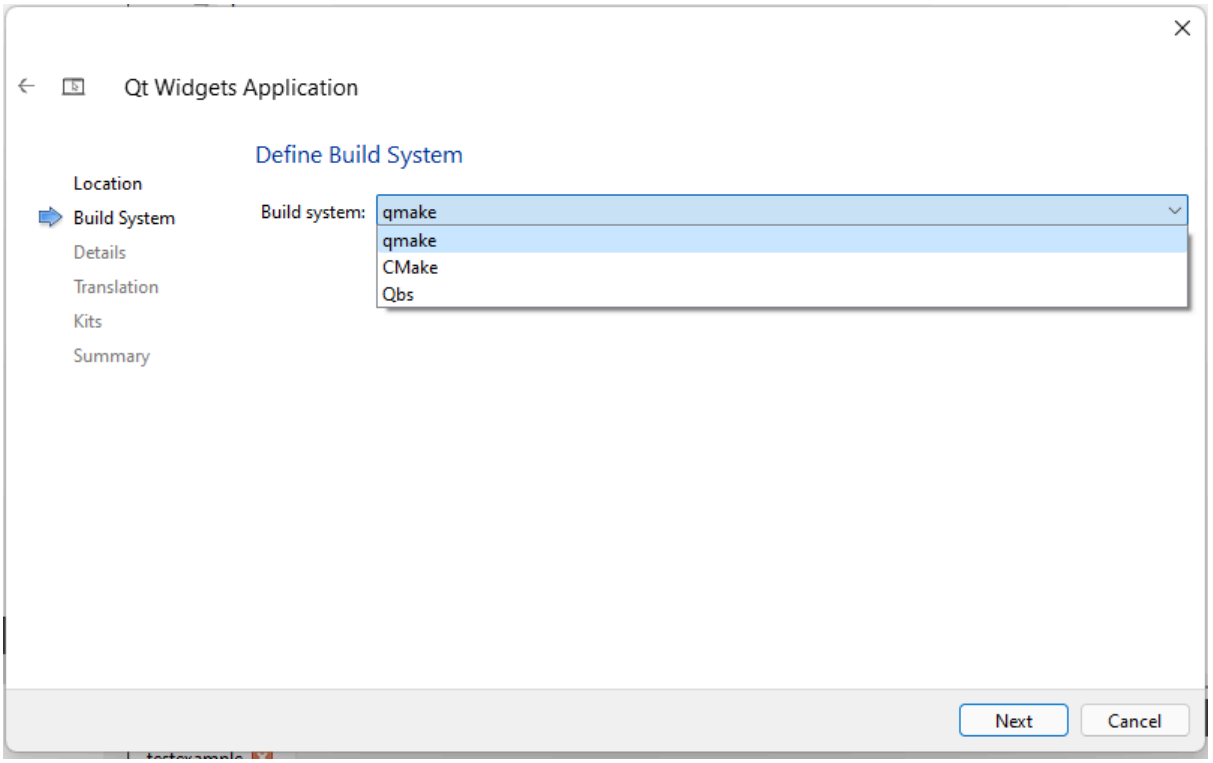
1. загружаете код на github;
2. отправляете на ревью;
3. записываетесь на защиту после получения по результатам ревью чек-листа;
4. приходите по записи;
5. показываете код и собранную программу.

Создание проекта в QtCreator

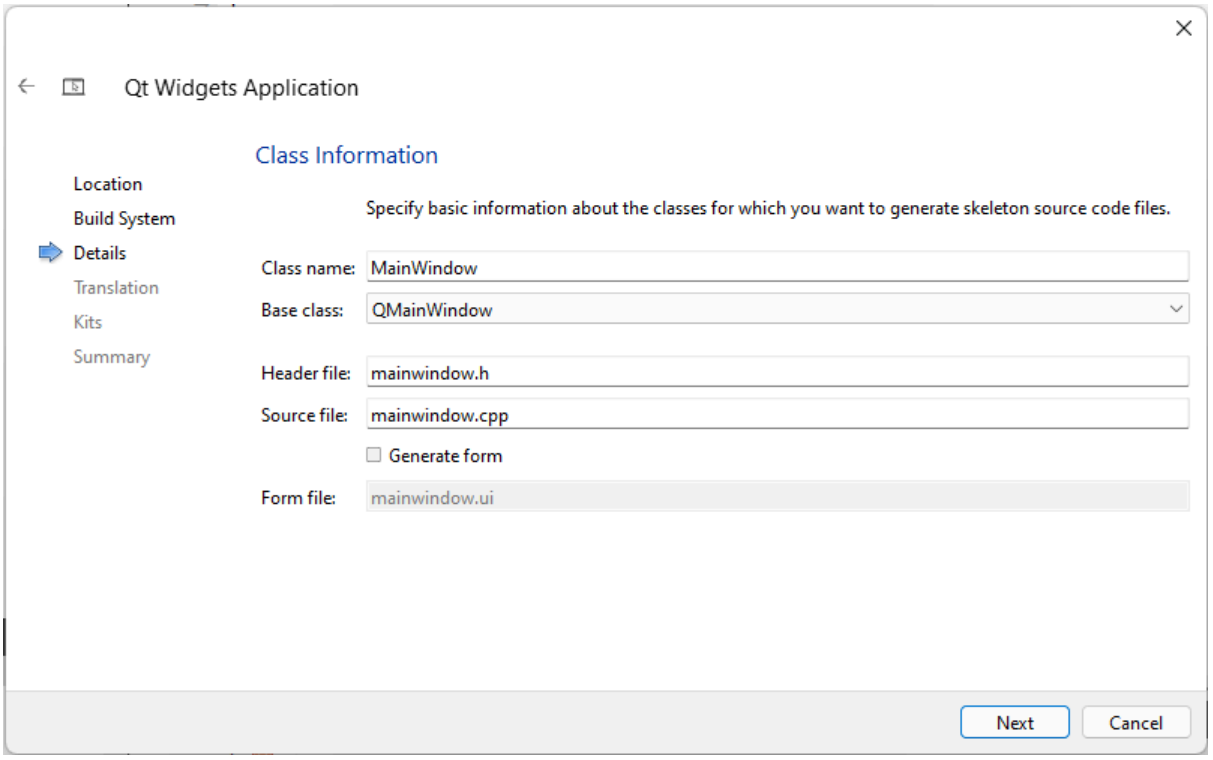
Qt Widget Application



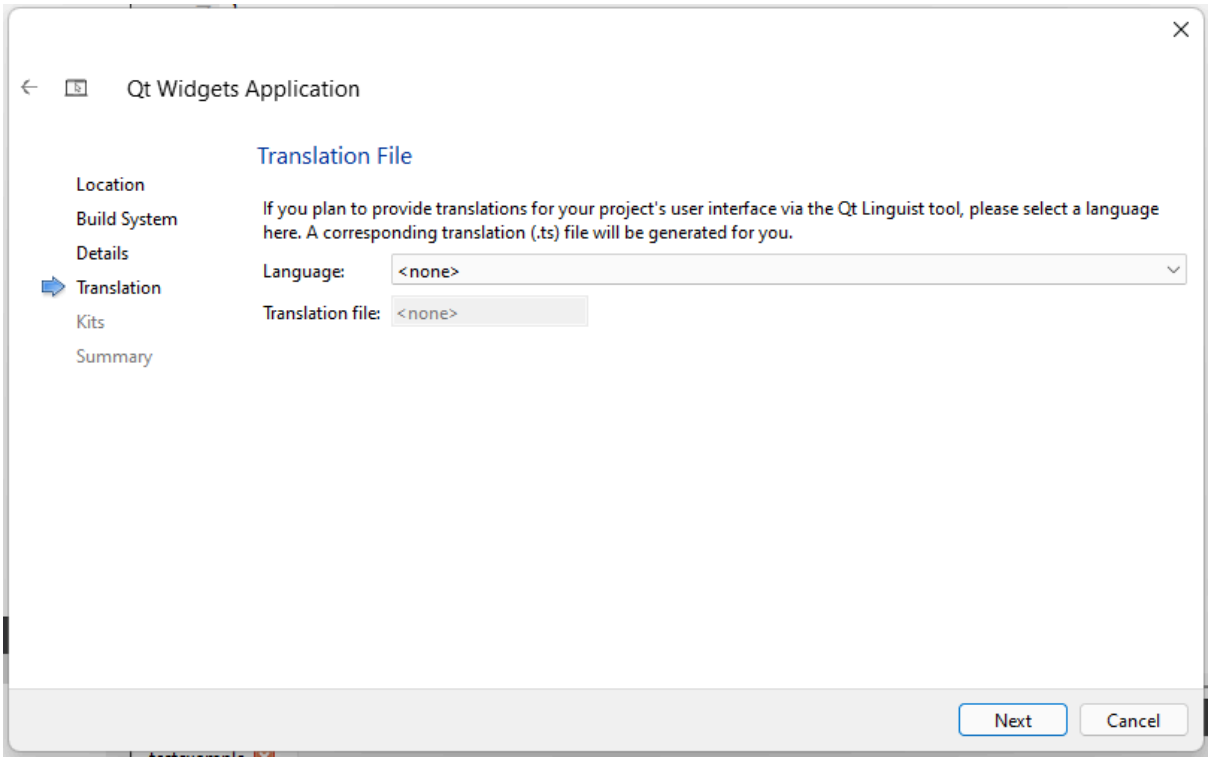
Система сборки: qmake



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.9 не делается, то оставляем None.



Kit выбираете любой из установленных.