

Система автоматизированного тестирования графического интерфейса в проекте QReal

Новожилов Евгений, 444 группа
Руководитель: Тимофей Брыксин

Проблема

- Желание тестировать основные пользовательские сценарии использования QReal
 - Большая кодовая база
 - Много логики в GUI
 - Недостаточно unit-тестирования
- Текущая реализация использует фреймворк, который работает на сравнении скриншотов
 - Интерфейс постоянно меняется и невозможно так часто обновлять базу скриншотов

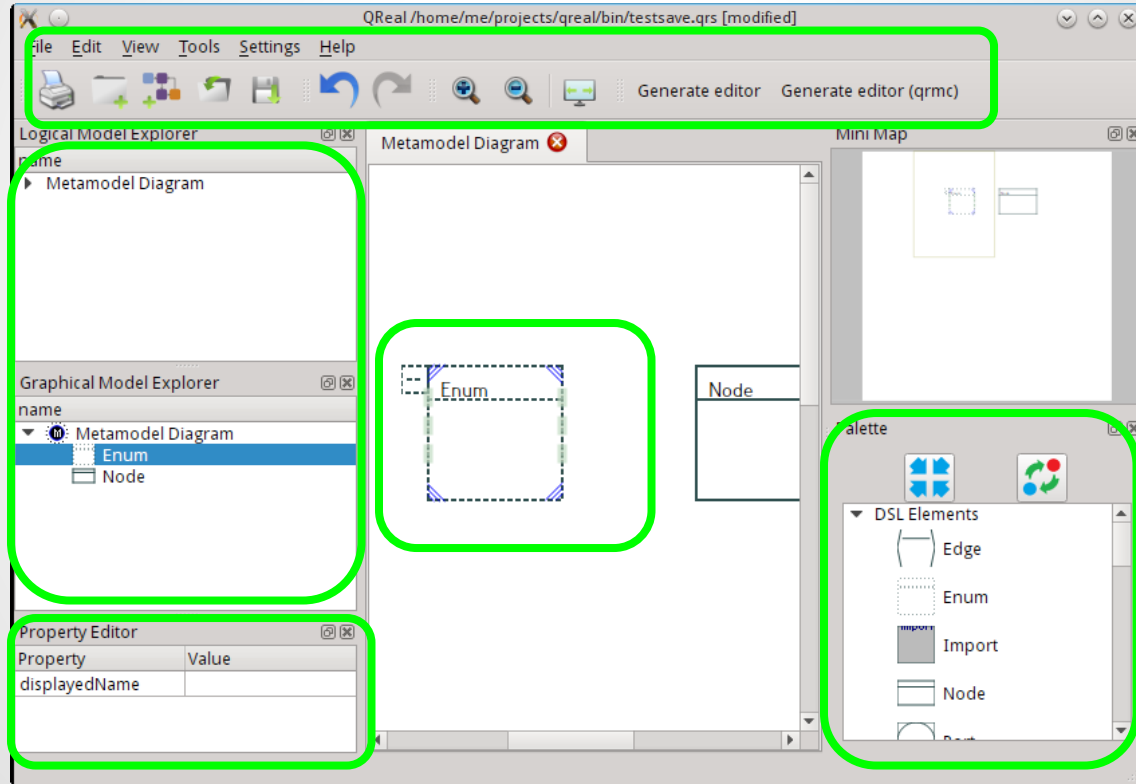
Постановка задачи

- Произвести сравнение существующих фреймворков для тестирования Qt-приложений
- Требования:
 - Кроссплатформенность
 - Сравнение результатов не по скриншотам, а на уровне окон и элементов окон
 - Симуляция событий мыши и клавиатуры
- Реализовать набор тестов для основных сценариев GUI интерфейса проекта QReal

Сравнение

Фреймворк	Описание
Sikuli	Используется на данный момент. Работает на сравнении скриншотов
Froglogic squish	Платный. Поддерживает симуляцию большинства событий Qt.
QTest	Встроенный в библиотеку Qt фреймворк для тестирования. Напрямую вызывает события GUI. Позволяет обращаться к компонентам системы.
Xnee/LDTP/Xlib	Фреймворки и библиотеки для Linux, использующие X11 для симуляции событий

DSM-платформа QReal



Решение

- Реализована библиотека в качестве обертки над QTest для более удобного использования
 - Более высокоуровневые действия по сравнению с QTest
 - Пример:
 - открыть заданное меню и выбрать действие
 - взаимодействие с элементом на сцене
 - изменить свойство объекта в редакторе свойств
 - изменить системные настройки

Пример тестов

- До

```
void DemoGuiTestSet::testPropertyEditor() {
    - MainWindow w("testsave.qrs");
    - QTest::waitForWindowExposed(&w);
    - QTest::qWait(1000);
    -
    - qReal::EditorView *view = qrtestlib::getCurrentEditorView(window);
    - Q_ASSERT(view != nullptr);

    - QList<QGraphicsItem *> l = view->items();
    - foreach (QGraphicsItem *i, l) {
    -     - qReal::Label *label = dynamic_cast<qReal::Label *>(i);
    -     - if (label != nullptr) {
    -         - if (label->toPlainText() == name) {
    -             - QPoint corner = label->parentItem()->scenePos().toPoint();
    -             - QPoint relCenter = label->parentItem()->boundingRect().center().toPoint();
    -             - QTest::mouseClick(view->viewport(), Qt::LeftButton, 0, corner + relCenter);
    -         }
    -     }
    -     - QTest::qWait(2000);
    - }
    - }

    - PropertyEditorView *dock =
    -     - mainWindow->findChild<PropertyEditorView *>("propertyEditor");
    - QtTreePropertyBrowser *tb = dock->findChild<QtTreePropertyBrowser *>();
    - QTreeWidget *treeWidget = tb->findChild<QTreeWidget *>();
    - qDebug() << treeWidget;
    - QList<QTreeWidgetItem *> items =
    -     - treeWidget->findItems(propertyName, Qt::MatchExactly);
    - qDebug() << items.at(0);
    - qDebug() << treeWidget->findChildren<QWidget *>();
    - Q_ASSERT(items.size() == 1);
    -
    - QTest::mouseClick(tw->viewport(), Qt::LeftButton, 0, tw->visualItemRect(wi).center() + QPoint(10, 0));
    - QTest::keyClicks(nullptr, propertyValue, Qt::NoModifier, 200);
    - QTest::keyClick((QWidget *) nullptr, Qt::Key_Enter, Qt::NoModifier, 200);
}
}
```

Пример тестов

- После

```
void DemoGuiTestSet::testPropertyEditor() {
    MainWindow w;
    QTest::qWaitForWindowExposed(&w);
    QTest::qWait(1000);
    qrtestlib::clickObjectOnScene(&w, "Enum");
    Id id = qrtestlib::getElementIdFromFriendlyId(&w, "Enum");
    qrtestlib::setPropertyInPropertyEditor(&w, "displayName", "NewName");
    QCOMPARE("NewName", qrtestlib::getPropertyFromModel(&w, id, "displayName"));
}
```


Результаты

- Произведено сравнение фреймворков для GUI-тестирования
- реализована высокоуровневая библиотека для создания GUI-тестов в QReal
 - Симуляция событий
 - Валидация результатов
- Реализован набор тестов
 - Создание новой диаграммы или проекта
 - Загрузка проекта
 - Изменение системных настроек
 - Изменение свойств элементов