

Язык описания пользовательских сценариев в платформе QReal

Автор: Чернов Д.В., 471 гр.

Научный руководитель:

ст. пр. Кириленко Я.А.

Рецензент:

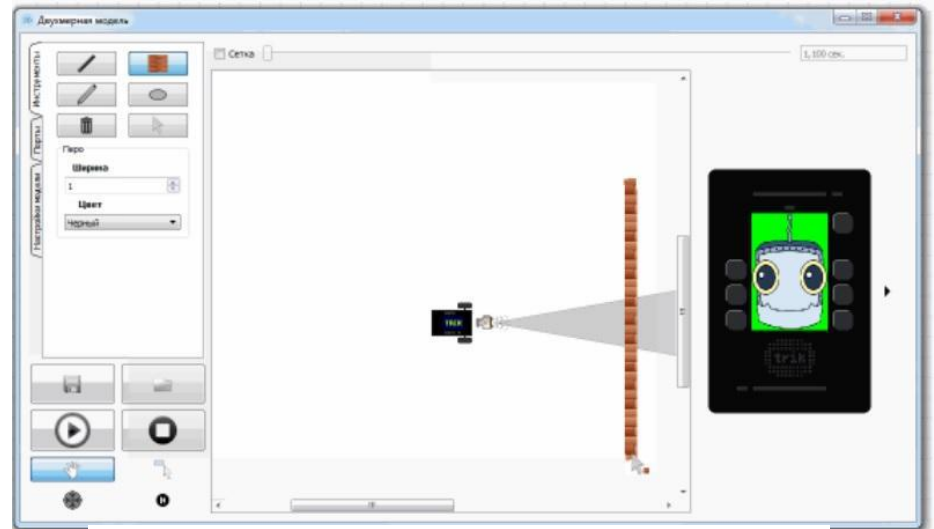
ст. пр. Брыксин Т.А.

DSM-платформа QReal

- Кросс-платформенный инструмент
- Создание предметно-ориентированных языков

TRIK Studio

- Создана на платформе QReal
- Предназначена для визуального программирования роботов
- Целевая аудитория: школьные учителя информатики и их ученики



*окно 2d-модели
робота TRIK Studio*

Обучающие инфраструктуры

- Справка
- Видео-уроки
- *Встроенная в среду*



*Обучение в
компьютерной игре
Warcraft*

API

- Целевой язык: QtScript;
- Функциональные возможности:
 - Управления виртуальным курсором и клавиатурой
 - Отправки информационных сообщений и указательных стрелок
 - Взаимодействие с палитрой, сценой, редактором свойств и другими специализированными элементами интерфейса системы QReal
 - Взаимодействие с различными элементами пользовательского интерфейса (выпадающий список, кнопки и т.д.)

Постановка задачи

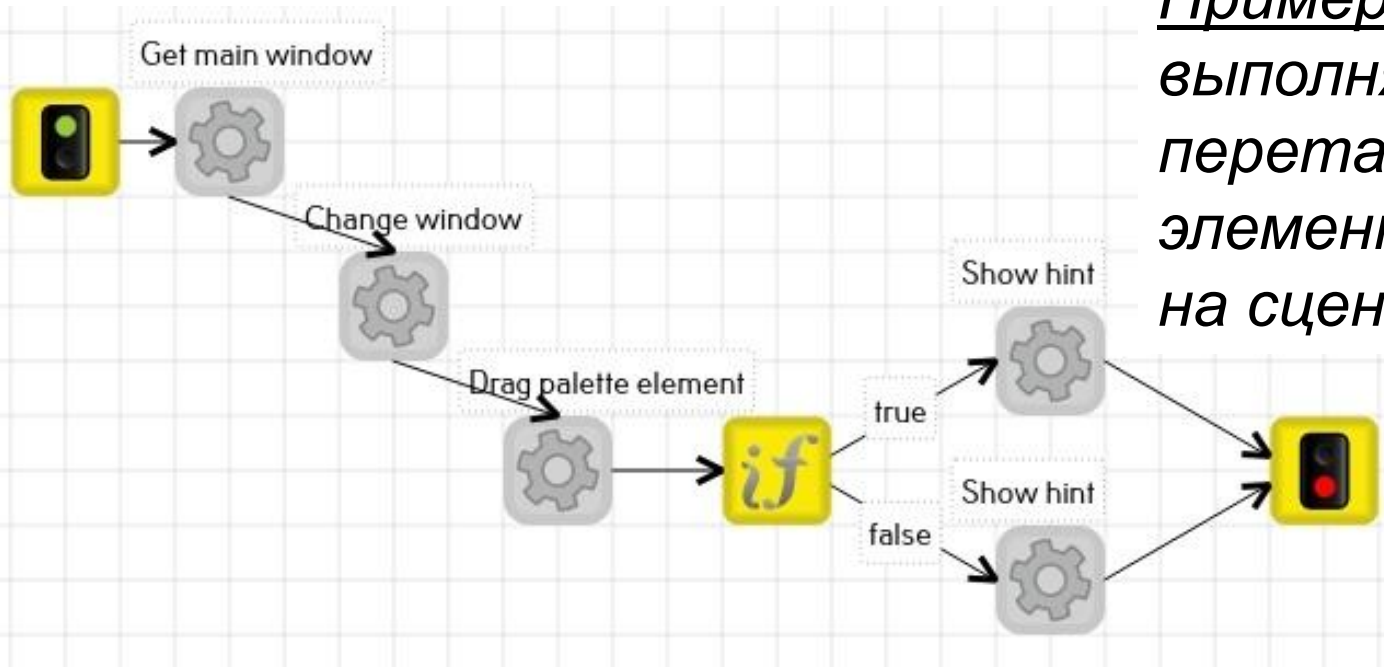
- Реализация языка описания пользовательских сценариев для построения обучающей инфраструктуры в DSM-платформе QReal

Создание обучающей инфраструктуры в DSM-платформе QReal

- API для доступа к графическому интерфейсу среды
- Язык описания пользовательских сценариев
- Запись и воспроизведение последовательностей пользовательских действий

Язык описания пользовательских сценариев

- Функциональные возможности: QtScript + API
- Визуальный язык



Пример: диаграмма, выполняющая перетаскивание элемента из палитры на сцену.

Язык описания пользовательских сценариев

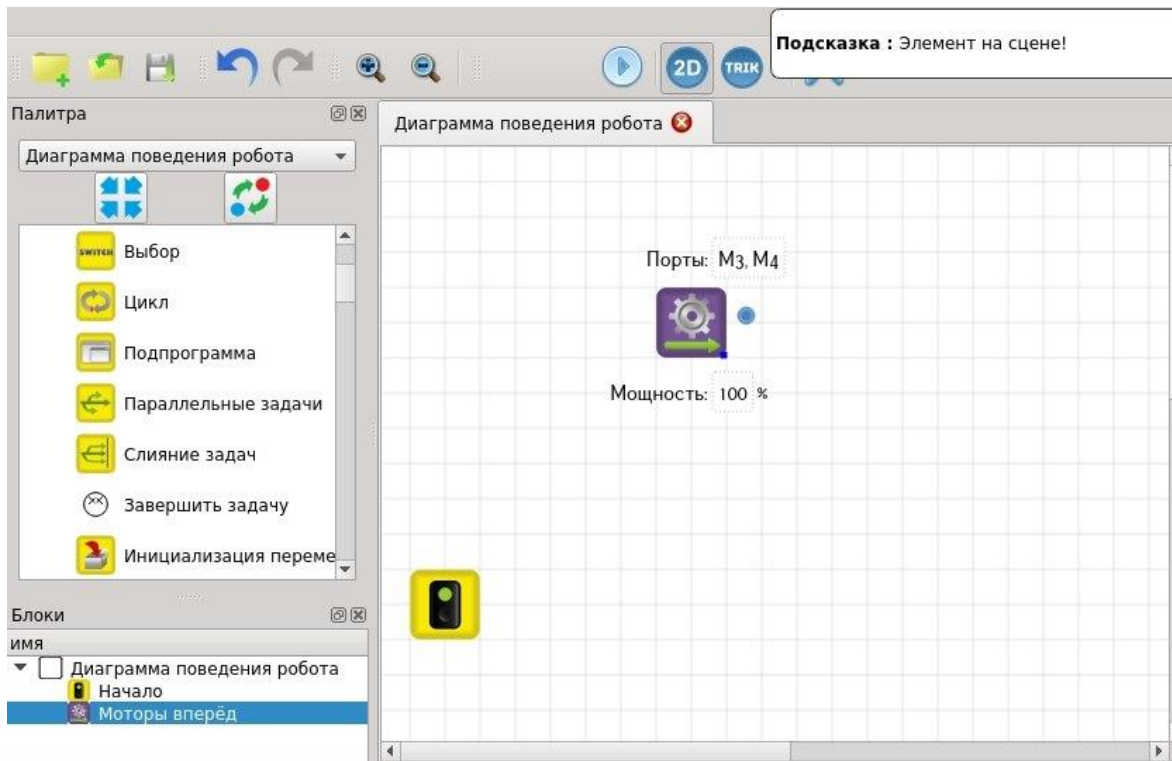
- Генерация в QtScript-код

```
var x;  
  
var main = function()  
{  
    api.changeWindow(MainWindow);  
    var x = api.palette().dragPaletteElement(  
        "qrm:/RobotsMetamodel/RobotsDiagram/TrikV6EnginesForward"  
        , 1000, 100, 100);  
  
    if (x) {  
        api.hints().addHint("Element now on scene.", 1000, MainWindow);  
    } else {  
        api.hints().addHint("Element not found!", 1000, MainWindow);  
    }  
    return;  
}
```

Пример:
соответствующий
диаграмме QtScript-код

Язык описания пользовательских сценариев

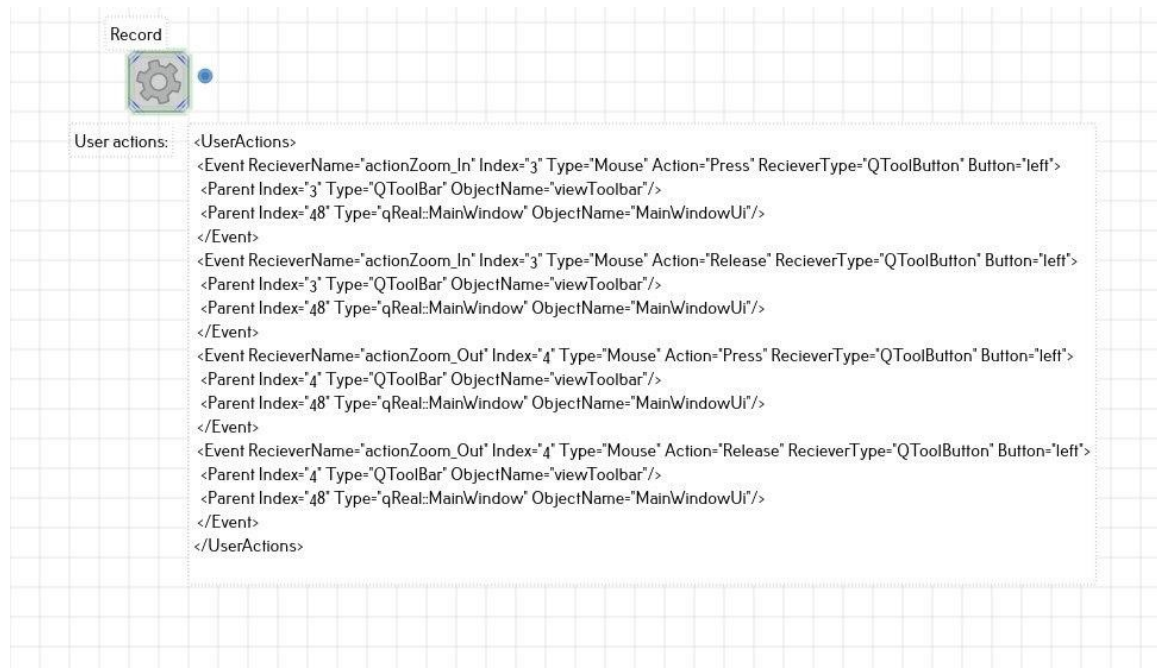
- Интерпретация с помощью API



*Пример:
интерпретация
диаграммы*

Запись пользовательских действий

- Сериализация значимых низкоуровневых событий, спровоцированных пользователем, в формат .xml
- Полученный файл транслируется в программу на языке QtScript



The screenshot shows a Qt Recorder interface with a 'Record' button (gear icon) and a 'User actions:' label. Below the label, a text area displays the recorded actions in XML format:

```
<UserActions>
<Event RecieverName="actionZoom_In" Index="3" Type="Mouse" Action="Press" RecieverType="QToolButton" Button="left">
<Parent Index="3" Type="QToolBar" ObjectName="viewToolBar"/>
<Parent Index="48" Type="qReal:MainWindow" ObjectName="MainWindowUi"/>
</Event>
<Event RecieverName="actionZoom_In" Index="3" Type="Mouse" Action="Release" RecieverType="QToolButton" Button="left">
<Parent Index="3" Type="QToolBar" ObjectName="viewToolBar"/>
<Parent Index="48" Type="qReal:MainWindow" ObjectName="MainWindowUi"/>
</Event>
<Event RecieverName="actionZoom_Out" Index="4" Type="Mouse" Action="Press" RecieverType="QToolButton" Button="left">
<Parent Index="4" Type="QToolBar" ObjectName="viewToolBar"/>
<Parent Index="48" Type="qReal:MainWindow" ObjectName="MainWindowUi"/>
</Event>
<Event RecieverName="actionZoom_Out" Index="4" Type="Mouse" Action="Release" RecieverType="QToolButton" Button="left">
<Parent Index="4" Type="QToolBar" ObjectName="viewToolBar"/>
<Parent Index="48" Type="qReal:MainWindow" ObjectName="MainWindowUi"/>
</Event>
</UserActions>
```

Результаты

- Реализован язык описания пользовательских сценариев и генератор кода в QtScript
- Реализован процесс записи и воспроизведения последовательностей пользовательских действий
- Технология апробирована созданием набора обучающих демонстраций