

Выступление

Здравствуйте, уважаемая комиссия, я Романов Ярослав, студент группы РК9-121. Представляю дипломный проект на тему: разработка кроссплатформенной версии системы имитационного моделирования RAO-studio.

- Подойти к листу 1

РДО предоставляет обширные возможности для моделирования дискретных систем и в настоящее время активно используется в учебных целях. Однако система могла работать только под ОС Windows из-за ограничений, накладываемых библиотекой MFC (Microsoft Foundation Classes), разработанной Microsoft. Данная библиотека очень давно не получала крупных обновлений и фактически устарела. Было принято решение разработать новую версию системы на современной библиотеке разработки, чтобы добиться кроссплатформенности системы и возможности ее запуска под ОС Linux. Была выбрана библиотека Qt, которая в настоящий момент имеет огромные возможности и активно развивается. Приложения, написанные на данной библиотеке, могут работать в Win, Linux, MacOS. Данная библиотека обладает подробной документацией, что упрощает ее использование. Еще одной причиной для разработки новой версии послужила необходимость перехода на Unicode. (кодировка UTF-8). В старой версии РДО используется кодировка codepage 1251, разработанная специально для windows. При работе системы с такой кодировкой модели, открытые в другой операционной системе, открывались неправильно и не могли работать. Unicode – универсальная кодировка, которая отображает символы одинаково во всех операционных системах, поэтому переход на нее необходим, чтобы модели, написанные в Windows, запускались в Linux.

- Подойти к листу 2

Исходя из поставленной задачи, можно вывести основные цели, которые представлены на диаграмме целей (рассказать по листу).

При разработке новой версии системы необходимо понять, за какие функции отвечает операционная система и непосредственно сама RAO-studio. Для этого была разработана диаграмма использования, представленная на листе 2. (рассказать по диаграмме – за данные функции отвечает РДО, за данные функции отвечает ОС, эти функции было необходимо перевести на кроссплатформенность).

- Подойти к листу 3

На диаграмме компонентов отображен состав изменяемых модулей системы. Представлены основные исполняемые файлы, в которые вносились изменения, а также файлы, которые были добавлены при разработке системы. Было необходимо перевести все подсистемы: анимации, трассировки, графиков, редактор РДО. Наибольшая сложность возникла в связи с заменой текстового редактора. Редактор основан на использовании сторонней библиотеки Scintilla. Старая версия РДО работала на версии сайнтиллы 1.6.2.

Современная версия Scintilla 3.3.0. Прежде, чем перепроектировать редактор, было необходимо обновить Scintilla. После этого был произведен перевод классов редактора на Qt, которые можно увидеть на диаграмме классов (показать на диаграмму классов редактора).

- Подойти к листу 4

Редактор тесно связан со стилями, которые отвечают за внешний вид системы, механизмы табуляции, автозавершения и некоторые другие. В старой версии системы за данную функциональность отвечало два класса, класс стилей и класс тем. Они были перепроектированы, и теперь все стили, относящиеся к определенной части системы, хранятся в одном классе, что позволило избавиться от лишних сущностей.

- Подойти к листу 8

Был разработан конвертор настроек из старой системы в новую. Его работа представлена на диаграмме последовательностей.

- Подойти к листу 9

Для того чтобы пользователь мог работать со стилями и менять их, необходим диалог настроек. Он был разработан на Qt. Его структура изображена в виде диаграммы компонентов на данном листе. Диалог имеет 4 вкладки, каждая из которых отвечает за свой набор настроек.

- Подойти к листу 8 (диаграмма состояний)

Возможные состояния диалога настроек представлены на данной диаграмме. При изменении пользователем настроек происходит сравнение новых настроек со старыми и в зависимости от этого активируется/деактивируется кнопка применения настроек.

- Подойти к листу 7 (алгоритм сравнения настроек)

Более подробно алгоритм сравнения настроек представлен на данном листе.

- При закрытии RAO-studio происходит сохранение настроек в операционную систему. Последовательность сохранения представлена на данном листе.

- В качестве исследовательской части была поставлена задача разработки нового алгоритма поиска ближайшего слова при автозавершении. Автозавершение слов – автоматическое дополнение системой текста, введенного пользователем. (на листе исследований на примере \$Parameters показать) В старой версии RAO-studio использовался стандартный механизм, встроенный в библиотеку сантиллы. Данный алгоритм был регистр-зависимым и не учитывал возможность ввода пользователем текста, определяющего не начало слова, а его часть. Поэтому были проведены исследования и разработан новый алгоритм, показанный на листе 7. Вначале происходит

инициализация списка ключевых слов, далее каждое слово из списка сравнивается с текстом, введенным пользователем, и рассчитывается приоритет этого слова. Приоритет рассчитывался двумя способами – в первом случае учитывалась только позиция вхождения паттерна пользователя в ключевое слово, во втором случае учитывается так же и отличие между паттерном пользователя и ключевым словом. Был выбран второй вариант, так как он более быстро находит искомое слово и показывает наиболее близкие слова. Результаты сравнения алгоритмов представлены на листе 10 (исследовательская часть). Здесь видно, что новый алгоритм почти во всех случаях работает лучше. Алгоритм, используемый в версии рдо 1.0, работает хуже всего, особенно в случаях, когда требуется регистр-независимый поиск или поиск по части слова.

Таким образом, в результате работы над дипломным проектом были решены следующие задачи:

1. Разработана кроссплатформенная версия RAO-studio, которая может работать под ОС Windows и Linux.
2. Перепроектирован GUI (лист 12). Новые экранные формы представлены на листе 12
3. Реализована поддержка Unicode
4. Обновлено документация
5. Подготовлена база для дальнейшего развития системы
6. Проведено функциональное тестирование системы