

Санкт-Петербургский Государственный университет
Математико-механический факультет
Кафедра системного программирования

Анализ пользовательского интерфейса QReal:Robots

Курсовая работа студента 445 группы
Кузенковой Анастасии Сергеевны

Научный руководитель

ст. преп. Литвинов Ю.В.

Санкт-Петербург
2013

Оглавление

Введение.....	3
Постановка задачи.....	3
Глава 1. Обзор предметной области.....	5
1.1. Основная терминология.....	5
1.2. Скорость взаимодействия с системой.....	7
1.3. Количество ошибок.....	9
1.4. Скорость обучения навыкам взаимодействия.....	9
Глава 2. Метрики юзабилити.....	11
2.1. Общие определения.....	11
2.2. Метрики юзабилити интерфейса QReal:Robots.....	12
2.3. Реализация сбора показателей интерфейса QReal:Robots.....	16
Глава 4. Применение: юзабилити-тестирование QReal:Robots.....	18
4.1. Цели юзабилити-тестирования.....	18
4.2. Выбор респондентов.....	19
4.3. Составление тестовых сценариев.....	20
4.4. Выбор анкет.....	21
4.5. Результаты юзабилити-тестирования.....	22
4.5.1. Обучаемость.....	23
4.5.2. Успешность выполнения заданий.....	24
4.5.3. Субъективная удовлетворенность.....	25
4.5.4. Выявленные проблемы интерфейса.....	26
Заключение.....	28
Список литературы.....	29

Введение

В настоящее время рынок программных продуктов очень широк и разнообразен, поэтому у большинства создаваемых продуктов конкуренция достаточно высока. Для того, чтобы привлечь широкий круг пользователей и разработать конкурентоспособный продукт, недостаточно только хорошей идеи, поскольку практически любая идея уже реализована в том или ином виде. Необходима её качественная реализация на профессиональном уровне.

Человеко-компьютерное взаимодействие является важным звеном вычислительного процесса как при решении научных, так и производственных задач. Сегодня при разработке интерфейса мы имеем обширную базу принципов проектирования интерфейсов и вспомогательных программных средств, однако создавать качественные интерфейсы все так же непросто. Ожидания пользователей со временем становятся все выше. В связи с этим если окажется, что с разработанным интерфейсом неудобно работать, то пользователи, столкнувшись с ним впервые, сразу же составят отрицательное мнение о продукте и предпочтут ему другой. Порой имеет место скептический настрой программистов, в частности, в научно-исследовательских проектах, к уделению особого внимания разработке и тестированию пользовательского интерфейса, и основное внимание разработчики уделяют исключительно функциональности создаваемых программных средств. Помимо этого главной и небезосновательной причиной такого настроения является непонимание того, что такое удобный пользовательский интерфейс. Универсального ответа на вопрос «Что такое удобный пользовательский интерфейс?» не существует, поскольку он специфичен для каждого разрабатываемого программного средства. Однако для каждого конкретного проекта это понятие можно определить через набор качеств из разных концепций качества пользовательского интерфейса, с указанием их приоритетов в отдельно рассматриваемом случае.

Постановка задачи

Целью курсовой работы стало исследование вопроса об оценке качества пользовательского интерфейса и применение полученных знаний к анализу интерфейса среды визуального программирования роботов QReal:Robots — среды обучения основам программирования и кибернетики на базе роботов Lego Mindstotrms NXT 2.0.

Основными задачами курсовой работы стали следующие:

- Изучение концепций определения качества пользовательского интерфейса;
- Исследование вопроса об измерении показателей качества пользовательского интерфейса;
- Выбор метрик для показателей качества интерфейса QReal:Robots;
- Реализация автоматического сбора выбранных показателей QReal:Robots;
- Проведение юзабилити-тестирования с использованием автоматического сбора показателей интерфейса QReal:Robots;

- Анализ полученных результатов.

Глава 1. Обзор предметной области

1.1. Основная терминология

Определим основную используемую терминологию области исследования пользовательских интерфейсов.

Интерфейс — это совокупность логических и физических принципов взаимодействия компонентов технических средств вычислительной системы, т. е. совокупность правил алгоритмов и временных соглашений по обмену данными между компонентами вычислительной системы (логический интерфейс), а также совокупность физических, механических и функциональных характеристик средств подключения, реализующих такое взаимодействие (физический интерфейс) [1]. Интерфейс распространяется на все логические и физические средства взаимодействия вычислительной системы с внешней средой, например с операционной системой и пользователем.

Пользовательский интерфейс — это способ выполнения задачи с помощью каких-либо программных средств, а именно совершаемые действия и получение результатов этих действий [1].

Далее в работе будут описаны исследования, относящиеся к понятию пользовательского интерфейса.

У пользовательского интерфейса, как у человеко-машинного взаимодействия, имеются две основные составляющие: часть, связанная с аппаратно-программной реализацией интерфейса и часть, связанная с действиями со стороны пользователя. Но если машинная часть интерфейса имеет четко определенное поведение, то поведение пользователя полностью предсказать невозможно, поэтому при разработке интерфейса основная задача — максимально учитывать человеческие возможности, способность к обучению, привычки и другие физиологические и психологические особенности. Далее нас будет интересовать вопрос о том, каким образом можно оценить и измерить человеческие особенности, и в последствии применить полученные знания к разработке интерфейса.

Эргономика — это наука, изучающая действия человека в процессе работы, скорость освоения им новой техники, затраты его энергии, производительность и интенсивность при конкретных видах деятельности [2]. Эргономика включает в себя непосредственно изучение человеко-машинного взаимодействия. В связи с этим можно определить эргономические *показатели* пользовательского интерфейса — качественные и количественные характеристики человеко-машинного взаимодействия. Существует множество различных эргономических показателей, специфичных для каждого отдельно рассматриваемого проекта. Существует несколько систем из эргономических показателей. На данный момент одной из наиболее распространённых систем является система показателей Бена Шнейдермана [3]. Эта система включает в себя следующие характеристики:

- скорость работы пользователя;
- количество человеческих ошибок;
- субъективная удовлетворенность;

- скорость обучения навыкам оперирования интерфейсом;
- степень сохраняемости этих навыков при неиспользовании программного средства.

У системы показателей Шнейдермана есть свои достоинства и недостатки. Основное преимущество этой системы — ее предметность. Улучшить показания перечисленных характеристик — гораздо более конкретная задача, нежели задача «сделать интерфейс более удобным». Также показатели Шнейдермана можно в той или иной степени измерить, то есть дать им количественную оценку, а в сравнении с результатами каждой рабочей итерации по улучшению этих показателей дать качественную оценку пользовательского интерфейса. Важно, что полученные результаты могут однозначно интерпретироваться каждым участником команды разработчиков, что позволяет облегчить коммуникацию касаясь вопросов изменений в интерфейсе программного средства.

Среди недостатков системы показателей Шнейдермана можно отметить следующие. Во-первых, описанная модель не учитывает всех возможных показателей, что может вызвать искажение результатов. Во-вторых, некоторые показатели Шнейдермана в большинстве случаев конфликтуют между собой. К примеру, если основная цель разработчиков — максимально увеличить скорость работы пользователя, то зачастую произойдет снижение скорости обучаемости системе. В связи с этим высоких показателей сразу всех характеристик модели добиться практически невозможно. Считается, что, как правило, высоких значений можно добиться только по двум показателям модели Шнейдермана. Третьим недостатком системы можно отметить то, что не существует абсолютной шкалы измерений каждого из показателей. Однако модель Шнейдермана является полезным средством в определении точных характеристик интерфейса, поэтому зачастую это весьма полезный инструмент в его разработке.

Однако подход к оценке качества пользовательского интерфейса на основе модели Шнейдермана фрагментарен, поскольку концентрируется на оценке отдельных характеристик. Существует более целостный подход к вопросу об удобстве использования интерфейсов — это *юзабилити*.

Существует большое количество определений понятия «юзабилити» [4]. Рассмотрим три наиболее часто встречающихся определения.

Согласно международному стандарту (ISO 9241-11) *юзабилити* — это степень эффективности, трудоемкости и удовлетворенности, с которыми продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей.

Определение, предложенное UPA (Usability Professionals Association) концентрируется больше на понятии юзабилити в контексте процесса разработки программного продукта: «*Юзабилити* — это подход к разработке продукта, который вовлекает обратную связь с пользователем на всех этапах разработки с целью создать продукт, отвечающий нуждам пользователя.»

Стив Круг [5] в своей книге дает следующее простое определение: «В действительности юзабилити — просто убеждение в том, что что-то работает хорошо — будь то веб-сайт, пульт управления или вращающаяся дверь — по прямому назначению и не оставляет пользователя безнадежно разочарованным.»

Все эти три определения, как и другие определения юзабилити, используют следующие общие тезисы:

- *Пользователь* вовлечен в процесс.
- Пользователь производит некоторые *действия*.
- Пользователь производит некоторые действия с *системой, продуктом* или *предметом*.

Таким образом, понятие «юзабилити» предоставляет полноценный контекст для работы с пользовательским интерфейсом, в отличие от модели Шнейдермана: среда, в которой пользователи взаимодействуют с продуктом — условия для анализа пользовательского интерфейса.

Но нас интересует вопрос о том, какие именно показатели пользовательского интерфейса характеризуют юзабилити. В связи с этим остановимся подробнее на первом определении ISO 9241-11. В определении используются такие базовые показатели качества интерфейса, как эффективность, трудоемкость и удовлетворенность. С одной стороны, на первый взгляд эти показатели в достаточной мере отображают характеристики, которые могут описать интерфейс пользователя. Но выбранные показатели вызывают множество споров среди юзабилити-специалистов [6] по следующим причинам. Во-первых, невозможно провести четкую границу между эффективностью и трудоемкостью интерфейса. К примеру, скорость работы пользователя даже в рамках одной и той же системы в разных случаях можно отнести и к эффективности, и к трудоемкости. Таким образом в случае, когда мы оперируем показателем интерфейса, который можно измерить непосредственно, сложно однозначно определить, за какой из *базовых* показателей качества интерфейса он отвечает.

Во-вторых, в стандарте ISO 9241-11 в описание понятий эффективности и трудоемкости попала одна из непосредственных характеристик продукта — мощность. Увеличение мощности системы в общем случае не имеет прямую пропорциональную зависимость с улучшением качества пользовательского интерфейса: некоторые продукты являются маломощными с целью сохранения упрощенного интерфейса взаимодействия и невысокой стоимости [7].

В связи с перечисленными выше причинами появилась распространенная переформулировка определения из международного стандарта с использованием показателей, определенных в модели Шнейдермана: «*Юзабилити* — показатель скорости взаимодействия с системой, количества ошибок, скорости обучения навыкам взаимодействия и субъективной удовлетворенности определенных пользователей продукта, достигающих определенных целей в определенном контексте использования.» Рассмотрим подробнее каждый из показателей юзабилити интерфейса.

1.2. Скорость взаимодействия с системой

Скорость взаимодействия с системой является важным показателем качества интерфейса. Непосредственно задача увеличения скорости работы пользователя с системой встречается редко, однако улучшение этого показателя качества всегда воспринимается положительно.

Длительность выполнения работы пользователем состоит из следующих этапов [1]:

- длительность восприятия исходной информации;

- длительность ментальной деятельности пользователя;
- длительность физических действий пользователя;
- длительность реакции системы на действия пользователя.

Длительность восприятия исходной информации заключается в том, что пользователь должен определить, какая информация о выполняемой задаче у него имеется и в каком состоянии сейчас находятся инструменты для решения этой задачи. Основное время на этом шаге отводится на считывание показаний системы.

Длительность ментальной деятельности пользователя состоит из 7 шагов:

1. Формирование цели действий
2. Определение направления действий
3. Определение, какие действия необходимо выполнить
4. Выполнение действий
5. Восприятие нового состояния системы
6. Интерпретация состояния системы
7. Оценка результата

Можно заметить, что шесть из семи этапов представляют собой исключительно мыслительную деятельность пользователя. Следовательно, повышение скорости мыслительных действий значительно увеличивает скорость взаимодействия с системой в целом. Но, к сожалению, повысить скорость собственного мышления пользователя невозможно. Тем не менее, существуют методы, позволяющие уменьшить влияние факторов, замедляющих процесс мышления и отвлекающих пользователя от поставленной цели, тем самым повысить скорость работы с интерфейсом. Рассмотрение этих методов выходит за рамки данной работы.

Длительность физических действий пользователя зависит от степени автоматизации выполнения требуемых действий и степени необходимой точности действий. В общем случае обсуждать степень автоматизации затруднительно, поскольку она сильно зависит от автоматизируемого процесса. Рассмотрим подробнее степень точности действий.

Любое действие пользователя, совершаемое при помощи мускулатуры, является либо быстрым, либо точным, и одновременно быстрым и точным оно может быть только при достаточно высокой выработанной степени автоматизма. Этот факт объясняется физиологическими особенностями человека. Таким образом, если действие человека должно быть быстрым, то оно не должно быть точным. Взаимодействие человека с компьютером происходит преимущественно двумя способами: с помощью клавиатуры и с помощью мыши.

При использовании клавиатуры особенной точности при нажатии на клавиши не требуется. Благодаря этому клавиатура дает преимущество для быстрых действий пользователя. Манипуляции при помощи клавиатуры позволяют довольно быстро добиться определенной степени автоматизма в действиях. С этим связано активное использование в интерфейсах горячих клавиш.

Мышь же, напротив, чувствительна к манипуляциям пользователя — действия зависят от скорости перемещения мыши. Поэтому именно оптимизация использования

мыши в системе позволяет значительно повысить скорость работы пользователя.

1.3. Количество ошибок

Сначала определим, что понимается под ошибкой пользователя. В общем случае ошибка определяется как любое действие, которое препятствует выполнению определенной задачи наиболее эффективным способом. Часто в области человеко-машинного взаимодействия можно встретить определение ошибки пользователя как действие пользователя в системе, которое приводит к не ожидаемой им реакции системы, то есть более узкое понимание — считается, что достаточно, чтобы пользователь умел решить поставленную задачу в системе, совершая действия адекватные тем, которые он хотел совершить для достижения своих целей. Это определение, на наш взгляд, более точно описывает ошибку пользователя как показатель качества интерфейса, поскольку зачастую одну и ту же задачу можно решить в системе различными способами, из которых невозможно однозначно выделить наиболее эффективный.

Универсальное определение действий, которые составляют ошибки пользователя, дать затруднительно. Ошибки могут основываться на различных действиях пользователя, например:

- Ввод некорректных данных в форму для заполнения (к примеру, неверный логин/пароль);
- Неверный выбор элемента в выпадающем списке (к примеру, выбор «Удалить» вместо «Изменить» в пункте меню);
- Выполнение некорректной последовательности действий (форматирование диска вместо записи на него информации)

Диапазон возможных ошибок зависит от конкретной исследуемой системы. Для того, чтобы определить, что потенциально может являться ошибкой пользователя в системе, необходимо составить список всех действий, которые пользователь может производить в системе, а уже после классифицировать всевозможные ошибки, которые могут возникнуть при работе пользователя.

1.4. Скорость обучения навыкам взаимодействия

Большинство систем, в особенности не имеющих аналогов, требуют времени и усилий для обучения работы с ними. Но, как правило, целенаправленного обучения системе не происходит, а навыки работы приобретаются через некоторое время. Оценить уровень владения системой можно по времени, в течение которого человек использует систему и по задачам, которые он умеет решать с ее помощью. Обучение навыкам взаимодействия может занимать короткий или длительный период времени. Когда обучение занимает короткий период времени, то пользователь может исследовать вопрос о различных способах решения одной задачи. Короткий период времени в данном случае — это минуты, часы или дни. За время использования продукта пользователь быстро разрабатывает для себя ментальную модель того, как работает система. То есть, обучаемость в данном случае больше связана с адаптивными стратегиями повышения эффективности выполнения задач, и меньше связана со способностью пользователя запоминать информацию. Если же обучение занимает длительный период времени — недели, месяцы или годы — то в этой ситуации часто имеет место значимый перерыв в использовании системой. Здесь уже важную роль играет память.

Чем длительнее перерывы — тем большая ответственность ложится на память пользователей, что нужно учитывать в интерфейсе системы.

Итак, мы определили, какие показатели юзабилити являются базовыми и рассмотрели их детально: скорость взаимодействия с системой, количество ошибок, скорость обучения навыкам взаимодействия и субъективная удовлетворенность. Тем самым мы определили ответ на вопрос «Что измерять?». Теперь перейдем к исследованию вопроса «Как измерять?».

Глава 2. Метрики юзабилити

2.1. Общие определения

Метрика — это способ измерения или оценки определенного явления или предмета. Метрика представляет соглашение о том, как измерять и оценивать те или иные характеристики и предоставляет целостный и надежный подход к процессу измерения. Метрики существуют в разных областях исследований, и юзабилити не является исключением. Как и все метрики, метрики юзабилити основываются на достоверной системе измерений: каждый раз использование одного и того же набора измерений с целью сравнения полученных результатов. Все метрики юзабилити должны быть обозримыми, явно или неявно. К примеру, определение успешности выполнения задания пользователем или время, которое ему потребовалось для выполнения задачи являются наблюдаемыми показателями. Все метрики юзабилити должны быть измеряемыми, то есть они должны иметь количественную интерпретацию. И наконец, все метрики юзабилити должны представлять тот или иной аспект базовых показателей юзабилити в числовом виде: например, юзабилити-метрики могут давать знание о том, что 65% участников успешно справились с заданием менее, чем за минуту.

Отличительные особенности метрик юзабилити связаны с тем, что эти метрики описывают действия *пользователя* в системе, то есть здесь мы сталкиваемся с личными особенностями конкретного человека. Поэтому метрики юзабилити не могут дать полноценный и полностью достоверный ответ на вопрос о том, как повысить качество интерфейса пользователя, но эти метрики позволяют выявить критические проблемы интерфейса и получить рекомендации о том, на какие элементы интерфейса стоит обратить внимание при дальнейшем исследовании. Важно также, что метрики юзабилити позволяют проводить сравнительный анализ различных версий продукта.

Метрики юзабилити специфичны для каждого конкретного проекта и зависят от целей проводимого исследования. Перейдем к рассмотрению исследования и определению метрик применительно к среде визуального программирования роботов QReal:Robots.

2.2. Метрики юзабилити интерфейса QReal:Robots

QReal:Robots¹ - система визуального программирования роботов, разрабатываемая на кафедре системного программирования Санкт-Петербургского государственного университета. Среда позволяет создавать графические программы для роботов Lego® Mindstorms® NXT 2.0 и исполнять эти программы прямо на компьютере, посылая команды роботу через Bluetooth или USB-интерфейс.

В базовые возможности среды программирования QReal:Robots входит:

- управление роботом Lego Mindstorms NXT 2.0 с компьютера через Bluetooth и USB-интерфейс;
- возможность исполнять программу на двумерной модели робота на экране без использования настоящего робота;
- отображение хода выполнения программы с помощью подсветки исполняемого блока;
- поддержка использования математических выражений, переменных и показаний датчиков в вычислениях.

Ниже приведен общий вид интерфейса QReal:Robots.

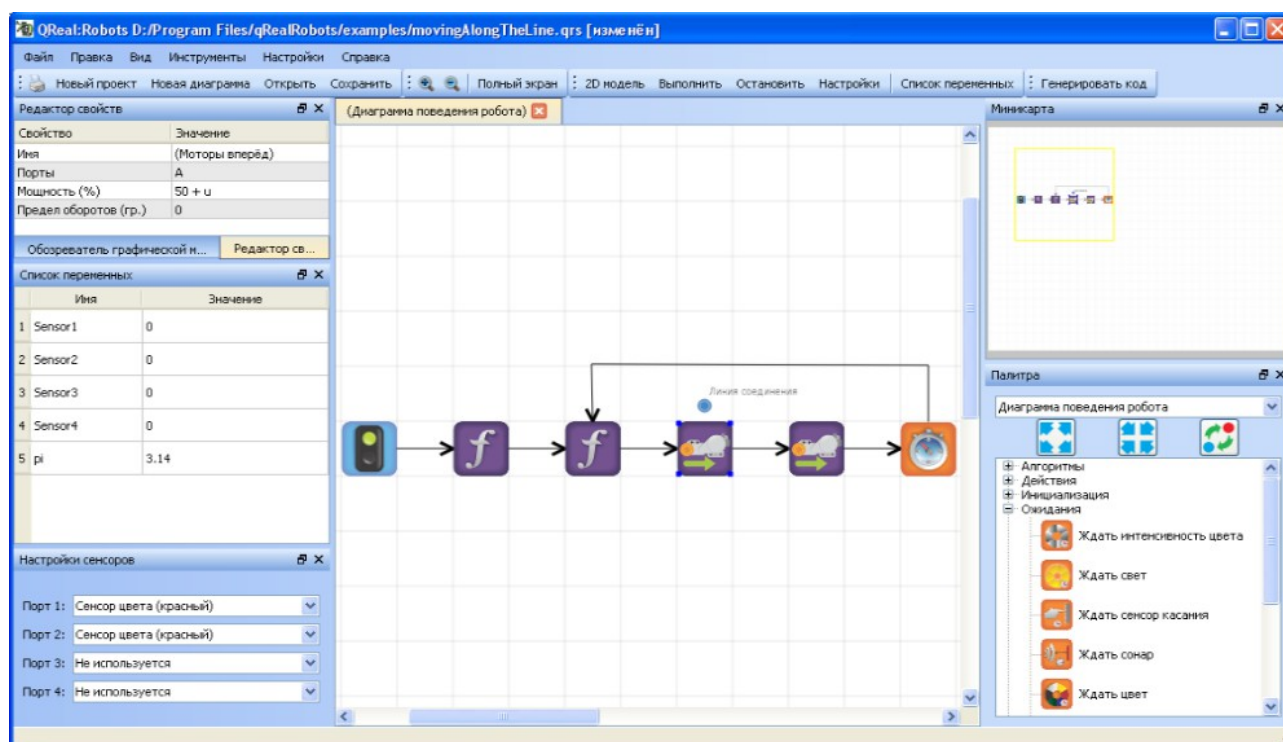


Рисунок 1. Интерфейс QReal:Robots

Пользовательский интерфейс QReal:Robots представляет собой реализацию шаблона интерфейса Canvas Plus Palette [8]. Шаблон представляет собой несколько панелей с настройками, палитру, при помощи которой создаются объекты, и холст, на который они помещаются. Типичный вариант использования включает создание новых объектов и их размещение в виртуальном пространстве. Существуют различные механизмы, используемые

1 <http://robots.qreal.ru>

для создания объектов в рамках шаблона. Наиболее распространено использование drag-and-drop механизма, который и используется в среде QReal:Robots.

Дадим краткое описание элементов интерфейса QReal:Robots.

- *Главное меню* содержит базовый набор операций и настроек среды;
- *Сцена* — главная рабочая область, расположенная в центре, отображает диаграмму, над которой сейчас ведется работа, и позволяет ее редактировать. При исполнении диаграмм текущий интерпретируемый блок подсвечивается на сцене.
- *Миникарта* — отображение сцены в уменьшенном масштабе. Миникарта располагается в правом верхнем углу окна системы.
- *Палитра элементов* располагается справа под миникартой и содержит набор доступных элементов для создания диаграмм.
- *Редактор свойств* расположен в левом верхнем углу и используется для редактирования значений свойств выделенного на сцене элемента.
- *Список переменных* располагается слева под редактором свойств и отображает название и значение всех переменных, используемых во время исполнения диаграммы, а также всех зарезервированных переменных среды.
- *Настройки сенсоров* находятся слева под списком переменных и отображают и предоставляют возможность настроить расположение сенсоров на портах робота.

После изучения структуры интерфейса QReal:Robots следующим этапом стал выбор характеристик интерфейса, информация о которых будет фиксироваться в процессе взаимодействия пользователя со средой. Для выбора таких показателей необходимо изначально определить цели для дальнейших исследований интерфейса QReal:Robots.

Основные действия в среде QReal:Robots связаны с созданием диаграмм, следовательно, одним из важных показателей интерфейса является удобное расположение доступных элементов для создания диаграмм в палитре. В связи с этим выбранным фиксируемым показателем стало количество использований блоков визуального языка. В данном случае использование определяется как появление элемента на сцене любым из доступных способов. Это позволяет понять, какие элементы используются часто, какие элементы не используются и в зависимости от этого определить, каким образом следует располагать элементы в палитре.

Как говорилось ранее, оптимизация манипулирования в среде при помощи мыши может значительно повысить скорость работы пользователя. Основываясь на этом, мы выбрали в качестве фиксируемых показателей координаты кликов мышью. Эту информацию можно впоследствии использовать для построения тепловой карты кликов [ссылка или сноска]. Карта кликов помогает выявлять наиболее часто используемые для выбора элементы, определять, какие элементы интерфейса ошибочно видятся пользователю кнопкой, ссылкой или изменяемым полем ввода. Клики на карте подсвечиваются разными цветами в зависимости от их частоты. На тепловой карте кликов теплые цвета соответствуют частым кликам, а холодные — редким. Также информация о количестве кликов может быть использована для сравнительного анализа выполнения одной и той же задачи в системе до и после внесения отдельных изменений.

Параметры среды QReal:Robots задается в специальном окне настроек.

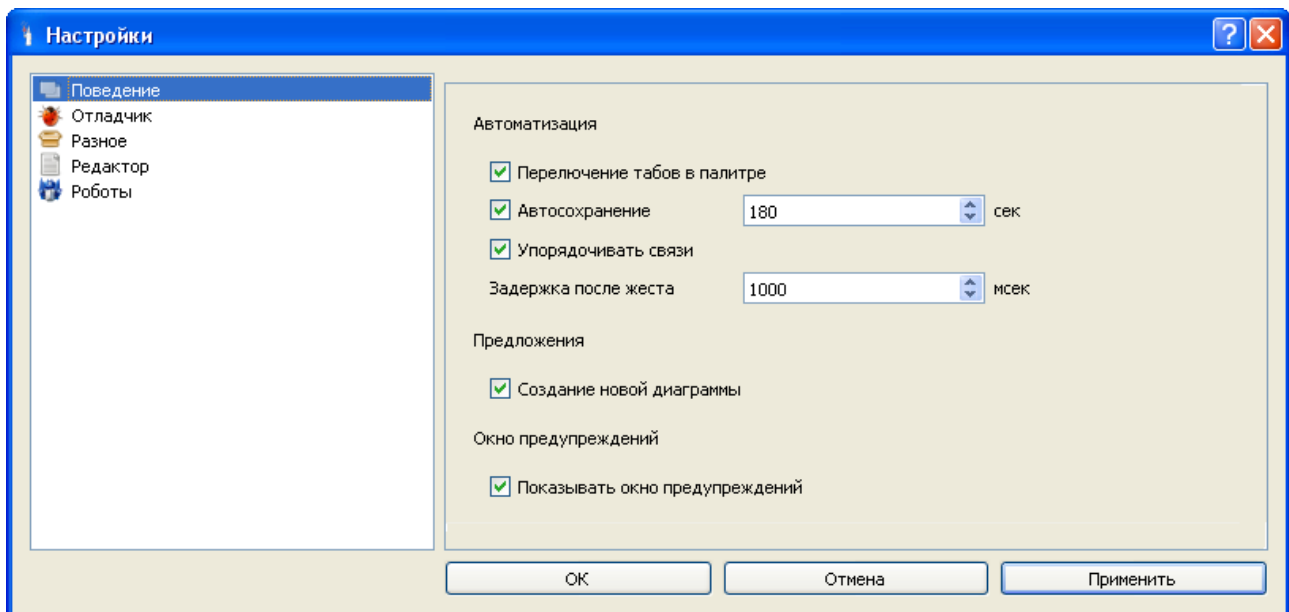


Рисунок 2. Окно настроек среды QReal:Robots

Следить за изменением этих параметров также стало важной задачей, поскольку это позволяет выявить актуальность имеющихся возможностей настроек среды, выявить, какие настройки и как часто изменяются, и определить, насколько оптимальны значения настроек среды по умолчанию. Поэтому мы решили, что необходимо фиксировать информацию об изменении настроек среды пользователем.

Самый очевидный из возможных фиксируемых параметров — длительность одной сессии работы с системой. В частности, эта информация позволяет понять, насколько важно учитывать утомляемость пользователя при работе с системой и выявить типичную длительность сессии.

Для определения наиболее часто используемых пунктов меню полезно собирать информацию о их выборе пользователем. Это позволит определить, каким образом имеет смысл улучшить структуру меню и какие пункты вынести в отдельную панель инструментов.

И последним фиксируемым параметром стали сообщения об ошибках. Сообщение об ошибках сигнализирует пользователю о некорректных действиях с его стороны по отношению к системе. Сообщения об ошибке являются проблемным местом многих интерфейсов по следующим причинам. Во-первых, появление сообщений об ошибках при работе системы всегда говорит о недостатках в ее проектировании. Система создается в первую очередь для решения пользовательских задач, а следовательно, возникновение ошибок связано с тем, что интерфейс недостаточно интуитивно понятен пользователю. Во-вторых, текст сообщения об ошибке как правило не дает пользователю полноценное понимание проблемы, а главное — как ее исправить. Часто это связано с тем, что для полного описания проблемы требуется относительно большой текст. Грамотное сообщение об ошибке должно отвечать на 3 вопроса [1]:

- В чем заключается возникшая проблема?
- Как ее исправить *сейчас*?
- Как сделать так, чтобы она больше *не повторялась*?

Сообщения об ошибке позволяют выявить наиболее проблемные места в интерфейсе

взаимодействия, а также определить, какие сообщения об ошибке повторяются систематически, то есть не дают пользователю понимание в решении проблемы.

2.3. Реализация сбора показателей интерфейса QReal:Robots

На основе выбранных показателей в рамках курсовой работы был реализован автоматизированный сбор информации об этих характеристиках интерфейса. Информация по умолчанию фиксируется в течение каждой сессии работы с системой, записывается в отдельный файл для каждого показателя и сохраняется на диске по завершении сессии с указанием даты и времени сессии.

- Информация об использовании элементов для создания диаграмм записывается в следующем формате: [порядковый номер]-[название элемента]-[дата и время создания элемента].
- Информация о кликах мышью фиксируется в формате [порядковый номер клика]-[координата x]-[координата y]-[дата и время клика].
- Информация о смене настроек среды записывается в формате [порядковый номер]-[название параметра настройки]-[прежнее значение]-[новое значение]-[дата и время изменения].
- Информация о длительности сессии в миллисекундах записывается также в отдельный файл.
- Информация об использовании пунктов меню имеет следующий формат: [порядковый номер]-[название пункта меню]-[дата и время использования].
- Информация о сообщениях об ошибках фиксируются в виде [порядковый номер]-[элемент интерфейса, инициировавший сообщение об ошибке]-[тип ошибки]-[текст сообщения об ошибке]-[дата и время возникновения сообщения об ошибке].

Показатели интерфейса имеет возможность собирать в двух режимах: фоновом и тестовом. Выше описан фоновый режим — информация сохраняется в течение работы системы в описанном формате.

Тестовый режим позволяет фиксировать показатели во время выполнения определенного задания и записать информацию об этом в файл. При включении в системе тестового режима появляются кнопки «Начать тест» и «Завершить тест». В случае, если требуется измерить показатели для выполнения конкретного тестового задания, то при использовании этих кнопок пометки о начале, завершении и порядковом номере теста заносятся во все имеющиеся файлы с информацией. Именно этот режим был использован для проведения анализа интерфейса с рамках процедуры юзабилити-тестирования.

Для перехвата кликов мышью по элементам управления QReal:Robots в класс приложения, наследующийся от *QApplication*, была добавлена реализация метода *notify()*, который по умолчанию перехватывает все события, поступающие от операционной системы в данное приложение. Метод был реализован таким образом, что все если событие оказывалось кликом мыши, происходит определение координат клика в системе координат окна приложения и данные передаются в специальный класс *UsabilityStatistics*, обеспечивающий обработку и фиксирование этих данных. Далее в *notify()* происходит вызов реализации метода по умолчанию, таким образом работа с событиями в приложении остается неизменной.

Сбор всех остальных эргономических показателей среды QReal:Robots происходит на

более высоком уровне, нежели события операционной системы. Например, для фиксации нажатия на определенные пункты меню происходит добавление обработчика на срабатывание сигнала *trigger()* у соответствующих QAction-объектов. Для этого в нужные места кода были добавлены необходимые вызовы *connect()*. В роли подобных обработчиков выступали методы класса *UsabilityStatistics*, что позволило обрабатывать всю информацию централизованно. Для этого класс *UsabilityStatistics* был реализован с помощью шаблона «Одиночка» (Singleton). Работа с настройками в QReal:Robots реализована в модуле *qrkernel*, в который помещается код, используемый всеми остальными модулями системы. Во избежание круговых зависимостей (*UsabilityStatistics*, при этом *qrkernel* тоже должен использовать *UsabilityStatistics* для фиксации изменения настроек) был применен шаблон проектирования Inversion of Control. У *UsabilityStatistics* был выделен необходимый интерфейс, по которому происходила работа с информацией об изменении настроек в *qrkernel*. Это позволило разорвать зависимость.

Глава 4. Применение: юзабилити-тестирование QReal:Robots

Юзабилити-тестированием называется эксперимент, направленный на измерение показателей качества пользовательского интерфейса и выявление проблем в нем. Юзабилити-тестирование — широко применяемая практика в оценке пользовательского интерфейса, однако часто для этой процедуры используется дорогостоящее оборудование, привлечение тестируемых также требует денежных и временных затрат. В данной работе исследовался вопрос о том, как проводить юзабилити-тестирование с минимальными затратами и как получить при этом достаточное количество информации о качестве интерфейса.

Основные этапы юзабилити-тестирования следующие:

- Определение целей юзабилити-тестирования;
- Выбор респондентов;
- Составление тестовых сценариев;
- Выбор анкет;
- Проведение юзабилити-тестирования;
- Анализ полученных результатов.

4.1. Цели юзабилити-тестирования

Определение целей юзабилити-тестирования - задача, на которой будут базироваться все последующие действия. От целей юзабилити-тестирования напрямую будет зависеть, какую информацию необходимо собирать и анализировать, какие тестовые задания составлять и каких людей привлекать в процессе тестирования. Важной особенностью формулирования целей является то, что в ней должен необходимо присутствовать критерий успешности, то есть, если имеется проблема, то как это определить.

В исследовании интерфейса QReal:Robots основными целями стали следующие:

- Определить, насколько быстро пользователь обучается работе с 2D-моделью робота (при выполнении однотипных заданий длительность выполнения уменьшается, после использования примеров пользователь понимает значения блоков и может создать собственную программу без ошибок);
- Определить, насколько понятно пользователю переключение с режима работы 2D-моделью робота на режим работы с реальным роботом (пользователь с первого раза переключается с одного режима на другой);

В целях тестирования указываются основные направления исследования. Помимо этого, как правило, в процессе тестирования выявляются основные проблемы, имеющиеся в интерфейсе.

4.2. Выбор респондентов

Определим основные критерии, по которым будут выбираться респонденты — участники юзабилити-тестирования.

Первый критерий — уровень владения тестируемой системой. В общем случае, если система уже использовалась ранее, то желательно, чтобы половина участников имела опыт работы с системой, а половина не имела такого опыта. Если у системы имеются конкурирующие с ней, то наиболее предпочтительно другое соотношение: треть имеет опыт работы с исследуемой системой, треть имеет опыт работы с конкурирующей системой, а треть не имеет опыта работы с подобными системами [источник]. В нашем случае, поскольку первоначальной целью стала проверка обучаемости работы с системой, то половина респондентов не имела опыта работы с QReal:Robots, а половина знакома с ранней версией среды. Такой выбор был обоснован тем, что проверить обучаемость среде после внесения значительных изменений также стало одной из задач тестирования.

Вторым критерием является возраст респондентов. В этой ситуации наиболее подходящим соотношением является следующее: три четверти респондентов имеет возраст целевой аудитории, а одна четверть имеет возраст старше возраста целевой аудитории, поскольку на этой аудитории можно определить больше проблем интерфейса [источник]. Целевая аудитория среды программирования QReal:Robots — школьники средних и старших классов, а также студенты, интересующиеся основами кибернетики. Мы остановились на целевой аудитории в виде студентов кафедры системного программирования в силу ограниченности во времени и возможностях процесса привлечения респондентов, а также в силу того, что при однородной по возрасту аудитории выявление схожих проблем прослеживается более явно.

Меньшее влияние имеет следующий критерий — пол респондентов. Однако желательно, чтобы среди респондентов присутствовали женщины — на женщинах лучше выявляются проблемы при внедрении, поскольку согласно исследованиям, женщины дольше обучаются, но после обучения выполняют работу более качественно [источник].

Вопрос о том, сколько респондентов требуется для проведения юзабилити-тестирования, является спорным.

Одним из устоявшихся мнений стало мнение о том, что для проведения юзабилити-тестирования достаточно 5-8 респондентов [9, 10]. Данное умозаключение основывается на следующем: чтобы найти 70% проблем, достаточно 5 респондентов, еще 3 респондента нужны для того, чтобы определить еще 15% проблем. При дальнейшем увеличении количества респондентов рост количества обнаруженных проблем происходит незначительно.

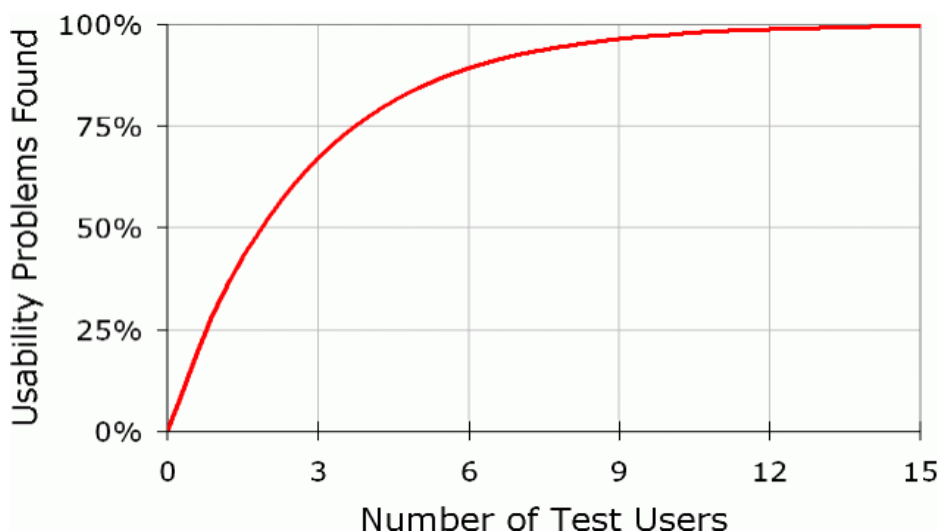


Рисунок 3. Зависимость доли найденных проблем от количества респондентов

Но здесь есть несколько важных особенностей. Во-первых, такие выводы были сделаны на основе экспериментов над небольшими системами. Во-вторых, при таком количестве участников сложно говорить о хорошей точности измерений показателей интерфейса. В-третьих, при таком количестве испытуемых сложно добиться необходимых соотношений, удовлетворяющих критериям, в выборе испытуемых. Тем не менее, первые 5 участников тестирования в большинстве случаев, как показывает практика, помогают определить наибольшую часть проблем в интерфейсе системы [9].

В нашем случае, поскольку юзабилити-тестирование охватывает лишь часть функциональности системы, мы остановились на 10 участниках тестирования.

В результате к участию в тестировании интерфейса QReal:Robots было привлечено 10 респондентов.

4.3. Составление тестовых сценариев

Тестовый сценарий представляет собой тестируемые аспекты системы. Тестовые сценарии состоят из заданий для респондента, метрик юзабилити, которые будут фиксироваться во время выполнения задания, а также критерия успешности выполнения задачи.

Определим ключевые моменты, которые необходимо учесть при создании тестовых сценариев.

- Задания тестовых сценариев должны быть типичны для работы пользователя с тестируемой системой. То есть задание должно иметь ценность для пользователя и при этом не быть неоправданно сложным.
- Задания должны быть адекватны поставленным целям. Именно для грамотного составления тестовых сценариев необходимо формулировать изначально цели тестирования, поскольку от этого будет зависеть не только сама структура задания, но и такие параметры, как влияние тестовых заданий друг на друга: к примеру, в случае, когда мы хотим измерить обучаемость системе, задания следует составлять так, чтобы они были связаны между собой.

- Не следует составлять задания, которые охватывают полную функциональность системы. Во-первых, это редко требуется для целей тестирования. Во-вторых, тестирование обычно длится не более, чем полтора часа, для достижения адекватных результатов по выполнению заданий от респондента, чтобы не вызвать излишней утомляемости.
- Выбирать фиксируемые метрики необходимо также исходя из целей тестирования. Но в случае, когда метрики не собираются автоматически, а наблюдаются, их не должно быть много. При фиксировании наблюдений вручную следует выбирать наиболее важные, и не более 2-3 метрик.
- Наконец, необходимо сформулировать критерий успешности выполнения задания — путь решения задачи, который является успешным. Вариантов решения заданий может быть несколько, но любые отклонения в этом пути считаются ошибкой пользователя.

В рамках данной работы было составлено 7 тестовых заданий: 5 заданий направлено на работу с двумерной моделью робота, 2 задания направлены на выполнение задач с настоящим роботом.

В качестве фиксируемых метрик юзабилити были выбраны следующие:

- время выполнения тестового задания (будет использоваться для выявления обучаемости системе);
- наличие ошибок пользователя (ошибками пользователя считаются создание на диаграмме блоков, выбор пунктов меню, изменение настроек среды, не требуемое для решения поставленной задачи, используется для определения успешности выполнения задания);
- субъективная удовлетворенность пользователя будет оцениваться с помощью анкеты, о которой речь пойдет далее.

Отметим, что для первых двух метрик будет использоваться автоматический сбор показателей интерфейса QReal:Robots. Заметим, что здесь собираемая информация в рамках тестирования имеет уже иную интерпретацию, нежели при сборе той же информации в фоновом режиме.

Скорость работы пользователя — одна из базовых метрик юзабилити в данном случае, на наш взгляд, учитывать в рамках одной сессии юзабилити-тестирования не имеет смысла, поскольку данная характеристика может быть оценена только в сравнительном тестировании (с аналогами тестируемой системы или с несколькими версиями рассматриваемой системы).

4.4. Выбор анкет

Если такие базовые характеристики, как скорость работы, количество ошибок и обучаемость системе можно в той или иной мере измерить явным образом, поскольку требуемые показатели доступны для наблюдений непосредственно, то субъективная удовлетворенность пользователя большей частью доступна только самому пользователю. Чтобы получить требуемую для оценки качества интерфейса информацию, необходимо задать пользователю заранее определенные вопросы. Для оценки субъективной удовлетворенности пользователя и служит анкетирование.

Существуют анкеты, которые помогают классифицировать функциональность тестируемого интерфейса согласно степени удовлетворенности пользователя и выявить основные проблемы в нем. Примером такой анкеты может служить анкета для измерения ожиданий пользователя от системы, разработанная в 2003 году [11]. Состоит данная методика в следующем: пользователю предлагают после прочтения условия задания и перед его выполнением оценить от 1 до 7, насколько сложным на его взгляд является задание (1 - очень сложное, 7 — очень простое). Далее после выполнения задания пользователь должен указать, насколько сложным оказалось для него задание по той же оценочной системе. После проведения юзабилити-тестирования полученные соответствующие результаты по каждому заданию усредняются (берется среднее арифметическое по каждому заданию). Далее строится следующий график из набора отдельных точек: по горизонтальной оси откладывается усредненное значение, которое соответствует ожидаемому пользователем уровню сложности задачи, а по вертикальной оси — уровень сложности, оцененный пользователем уже после выполнения задачи. Так мы получаем 4 потенциальные группы задач:

1. Респонденты ожидали, что задание будет несложным (5-7 баллов), но после выполнения оценили задание как трудное (1-3 балла). Группа таких заданий называется «Fix it fast». Такого рода задания вызывают у пользователей наибольшее разочарование, поэтому требуют немедленного исправления со стороны разработчиков системы.
2. Респонденты посчитали задание легким (5-7 баллов), и оно таковым и оказалось. То есть эти задания характеризуют функциональность, которая работает соответствующим представлением пользователя образом и не вызывает затруднений. Не рекомендуется вносить изменения в часть интерфейса, связанной с этой функциональностью, поэтому группа заданий носит название «Don't touch it».
3. Респонденты оценили задание как сложное, а оно оказалось для них нетрудным. Такие задания становятся приятной неожиданностью как для разработчиков, так и для пользователей. Они демонстрируют особенности системы, ее преимущество перед другими аналогичными средами программирования. Группа таких заданий называется «Promote it».
4. Респонденты определили задание как сложное, и оно оказалось сложным. Эта группа заданий вызывает интерес в том смысле, что имеется возможность сделать их проще для пользователя, и тем самым сделать их преимуществом тестируемой системы. Эта группа заданий носит название «Big opportunity».

Эту метрику мы будем использовать в процессе проведения юзабилити-тестирования для оценки субъективной удовлетворенности пользователей.

4.5. Результаты юзабилити-тестирования

В рамках курсовой работы было проведено юзабилити-тестирование с участием 10 респондентов. Участникам тестирования предлагалось выполнить последовательно 7 заданий. Проанализируем полученные результаты тестирования исходя из целей тестирования, а также опишем выявленные проблемы в интерфейсе QReal:Robots.

4.5.1. Обучаемость

Тестовые задания были построены следующим образом:

Пять заданий были направлены на работу с двумерной моделью робота. Первое задание было вводным, при его выполнении использовались базовая функциональность среды программирования QReal:Robots, а именно запуск программы на 2D-модели робота и изменение параметров отдельных блоков для необходимого изменения поведения 2D-модели. Второе, третье и четвертое задание затрагивало эмуляцию сенсоров в 2D-модели робота и эмуляцию модели окружающего мира. Задания были однотипными, третье и четвертое задание были непосредственно связаны между собой. При выполнении этих четырех заданий требовалось использовать заранее подготовленные примеры диаграмм и вносить в них при необходимости небольшие изменения (заменить один блок на другой, изменить значение одного из свойств). При выполнении пятого задания необходимо было создать диаграмму, задающую поведение робота, самостоятельно.

Оставшиеся два задания были связаны с работой пользователя с настоящим роботом. Шестое задание было связано с предыдущим. Седьмое задание, аналогично пятому, требовало создать самостоятельно программу и исполнить ее уже на реально устройстве.

Рассмотрим полученные результаты. Ниже представлен график, где для каждой задачи указано среднее время ее выполнения в секундах и таблица с указанием времени выполнения каждого задания респондентом.

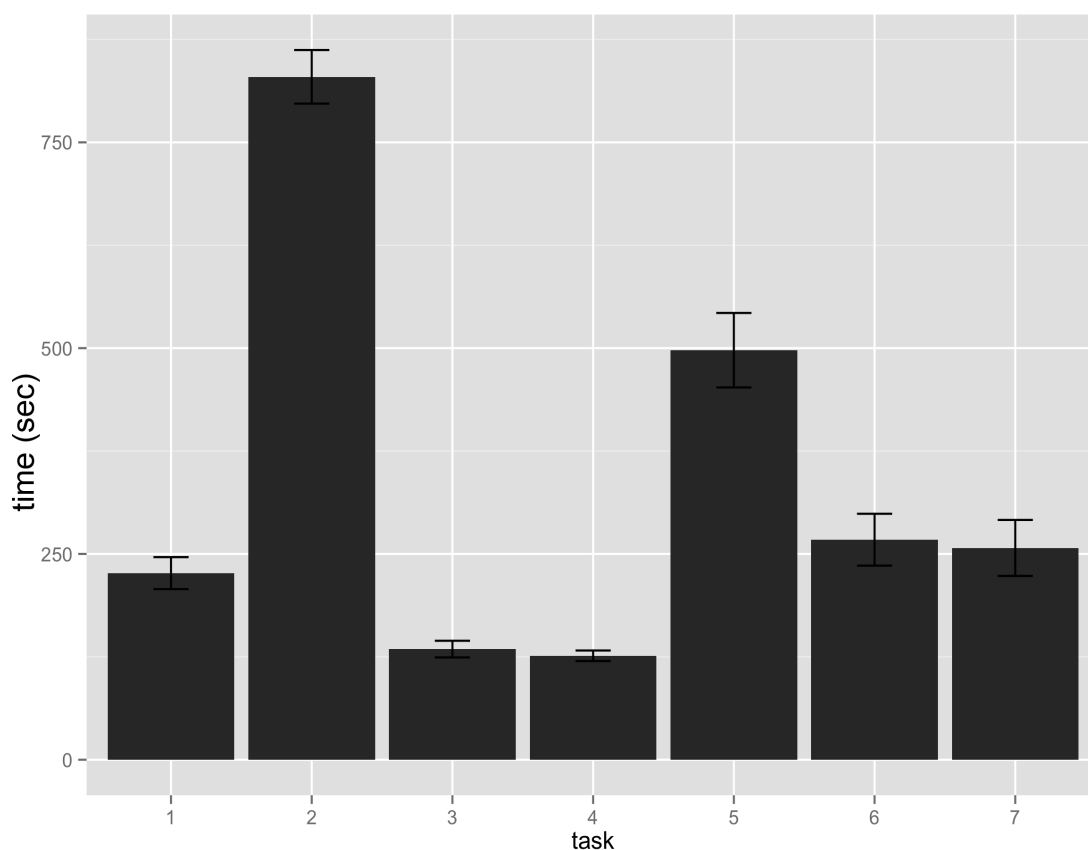


Рисунок 4. Среднее время выполнения тестовых заданий

	Task #1	Task #2	Task #3	Task #4	Task #5	Task #6	Task #7
Participant #1	267	706	81	130	544	368	230
Participant #2	230	840	201	140	408	308	544
Participant #3	213	1007	131	125	659	345	234
Participant #4	320	765	108	154	454	407	304
Participant #5	187	980	132	143	706	342	201
Participant #6	240	849	120	130	264	240	176
Participant #7	312	760	123	109	670	210	180
Participant #8	140	870	139	103	430	179	230
Participant #9	213	810	149	89	390	120	260
Participant #10	145	709	160	140	450	154	214
Mean	226,7	829,6	134,4	126,3	497,5	267,3	257,3
Count	10	10	10	10	10	10	10
Standard Deviation	61,4058448177	102,9727040423	32,0215205413	20,2322624648	143,0875800185	99,6048303157	107,5216257318
95% Confidence Level	38,0590375725	63,8219704284	19,8467793583	12,539855768	88,68497128	61,7345790232	66,6413694939

Таблица 1. Результаты тестирования: время выполнения заданий

На рисунке 4 мы можем видеть, что выполнение второго задания заняло у респондентов значительно больше времени, чем выполнение всех остальных заданий. Однако заметим, что после выполнения второго задания, выполнение третьего и четвертого потребовало наименьшее количество времени, причем время выполнения третьего и четвертого задания отличаются незначительно. Эта информация наглядно демонстрирует возможность наличия высокого уровня обучаемости системы — пользователю достаточно выполнить одно задание, чтобы в дальнейшем легко справляться с выполнением подобных заданий.

Также заметим, что выполнение пятого задания, которое по типу отличалось от предыдущих, потребовало большее количество временных затрат от участников тестирования.

Отсюда можно предположить, что уровень обучаемости системе довольно высокий, поскольку респондентам было достаточно выполнить одно задание, чтобы выполнять задания подобного типа за сравнительно меньшее время.

4.5.2. Успешность выполнения заданий

У каждой тестовой задачи имелись пути ее решения, характеризующие безошибочность действий пользователя. Мы классифицировали успешность выполнения задания следующим образом:

- Пользователь полностью выполнил задание;
- Пользователь выполнил задание, но во время выполнения ему требовалась помощь или он совершал ошибочные действия;
- Пользователь не выполнил задание или отказался от его выполнения.

Полученные результаты представлены на рисунке 6.

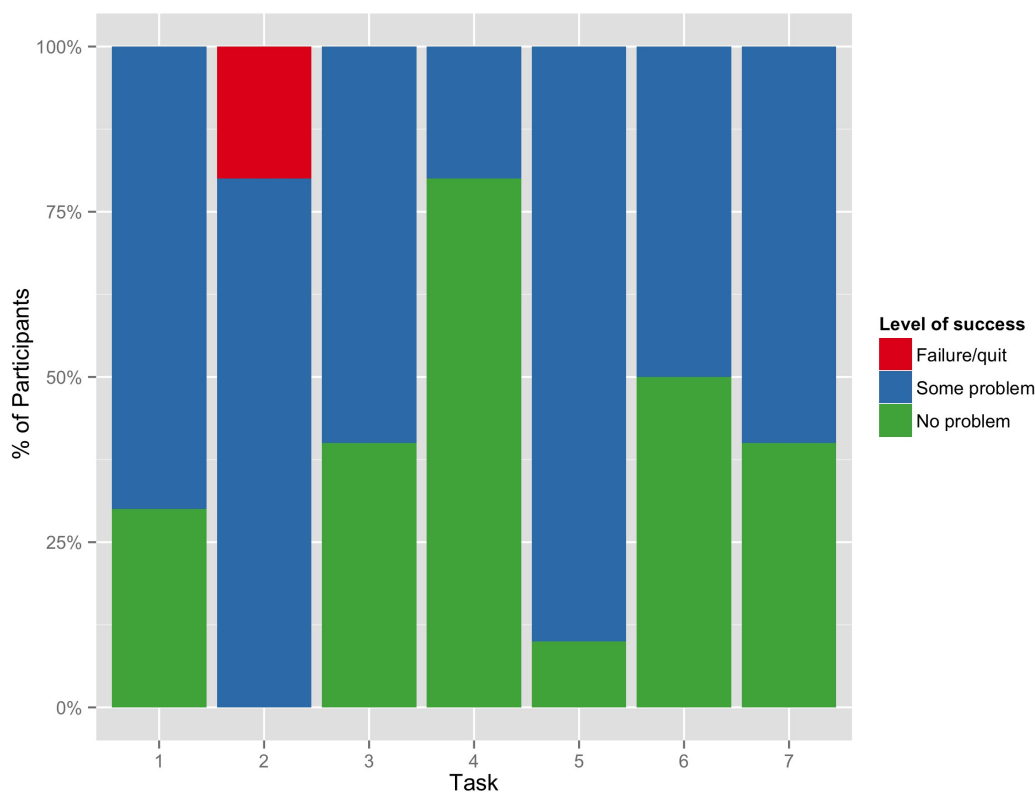


Рисунок 6. Успешность выполнения заданий

Как видно из рисунка 6, наиболее сложным для участников оказалось второе задание (о возникших трудностях будет сказано далее), а четвертое задание оказалось самым простым из предложенных для респондентов. Основной вывод, который можно сделать из этого графика — что процент участников, имеющих трудности с выполнением заданий, существенный. Далее будут рассмотрены основные проблемы интерфейса, выявленные в результате тестирования.

4.5.3. Субъективная удовлетворенность

Результаты опроса респондентов об ожидаемом и реальном уровне сложности выполнения заданий представлены на рисунке 7. Из графика видно, что большинство, а именно, пять заданий попало в группу «Don't touch it». Можно сделать вывод, что большинство тестируемой функциональности с точки зрения интерфейса устроены приемлемым для пользователя образом, и выполнения большей части заданий не вызывает дополнительных трудностей. Однако, одно из заданий, связанное с работой с конфигурацией сенсоров в 2D-модели робота, оказалось предельно близко к группе заданий «Fix it fast», а следовательно, требует внесения исправлений в ближайшее время. Также одно задание оказалось близко к группе «Promote it». Это задание было связано с подключением к реальному устройству робота. Многие участники тестирования также устно отмечали, что подключение к роботу оказалось довольно несложным.

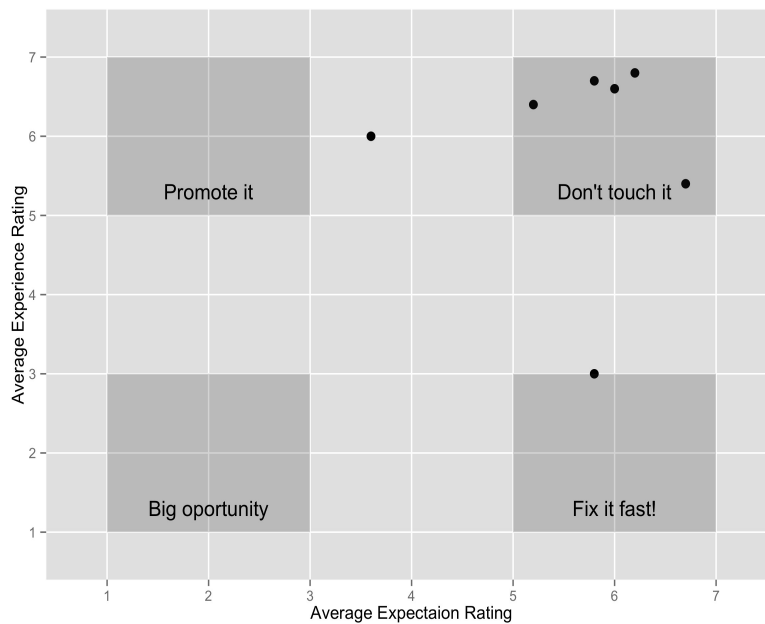


Рисунок 7. Результаты анкетирования

4.5.4. Выявленные проблемы интерфейса

Одна из важных особенностей юзабилити-тестирования - это выявление основных проблем интерфейса в результате наблюдения за выполнением заданий респондентами. Перечислим наиболее часто встречающиеся во время тестирования проблемы при работе с интерфейсом QReal:Robots:

1. **Структура главного меню.** Участники тестирования большинство функциональности пытались найти именно в главном меню, и здесь возникало несколько сложностей. Во-первых, респонденты не могли найти с первого раза нужный пункт меню для выполнения своих задач, что говорит о том, что в структуре меню, в названиях пунктов меню имеются проблемы. Во-вторых, меню «Файл» не имеет унифицированную структуру подобных программных средств. Например, не содержит пункт меню «Создать». Стоит отметить, что создание диаграмм доступно из панели инструментов, но большая часть респондентов не воспользовались ей, а предпочли переделать текущую диаграмму, а не создавать новую.
2. **Системное и реальное время.** В тестовых заданиях было указано конкретное время, которое робот должен совершать определенное действие. Это время необходимо было указывать в специальном блоке. Но когда происходила проверка программы — ее исполнение на 2D-модели робота, то участники не могли понять, нужно ли время робот выполнял действие, так как оно не соответствовало реальному времени, которое было указано в блоке. Дело в том, что в среде 2D-модели робота используются другие промежутки времени, но пользователь об этом не знает. Одним из возможных решений этой проблемы служит добавление таймера в среду 2D-модели робота, который будет демонстрировать системное время выполнения действий.
3. **Подключение к роботу.** Несмотря на то, что пользователи оценили простоту подключения к реальному устройству робота, требовалось дважды предпринять попытку подключиться к роботу, чтобы подключение произошло. Это вызвало

затруднение у респондентов — не было понятно, что необходимо сделать повторное подключение к роботу. Это скорее относится к функциональным недостаткам системы.

4. **Единицы измерения.** В заданиях были указаны конкретные значения параметров, которые необходимо использовать для выполнения заданий, время — в секундах, расстояние — в сантиметрах. Но в редакторе свойств не определено, в каких единицах необходимо указывать эти параметры, что вызывало вопросы у всех участников тестирования.
5. **Конфигурация сенсоров.** Установка конфигурации сенсоров происходит как в основном рабочем окне, так и в 2D-модели робота, причем при установке конфигурации в настройках среды автоматически устанавливается та же конфигурация в 2D-модели робота. Но если изменить конфигурацию сенсоров уже в 2D-модели робота, то в настройках среды она останется прежней, что вызывало у пользователей затруднение — при удалении элементов сенсоров в 2D-модели настройки сбрасывались, и уже при запуске программа не исполнялась должным образом. Связь этих настроек не является очевидной для участников тестирования и также вызывала ряд вопросов.

Заключение

В рамках курсовой работы было проведено исследование области оценки качества пользовательского интерфейса и юзабилити. В результате работы были выделены основные эргономические показатели и реализован автоматический сбор информации, связанной с выбранными показателями качества интерфейса. Применение сбора характеристик среды QReal:Robots было проведено в рамках процедуры юзабилити-тестирования, в результате которой были выделены основные проблемы интерфейса и получены результаты об уровне обучаемости системе QReal:Robots, а также субъективной удовлетворенности пользователя.

Список литературы

- [1] Джеф Раскин «Интерфейс: новые направления в проектировании компьютерных систем», 2010.
- [2] URL: <http://www.ergonomikapc.ru/> // дата обращения: 26.05.2013.
- [3] Vitaly Friedman, 30 Usability Issues To Be Aware Of // URL: <http://www.smashingmagazine.com/>, 2007.
- [4] Tom Tullis, Bill Albert “Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics”, 2008.
- [5] Стив Круг «Не заставляйте меня думать!», 2002.
- [6] Jakob Nielsen, Usability Metrics: Tracking Interface Improvements // IEEE Software, Volume 13 Issue 6, 1996, p.12-13.
- [7] Влад Головач «Дизайн пользовательского интерфейса: искусство мыть слона», 2010.
- [8] Дженнифер Твидел «Разработка пользовательских интерфейсов», 2010
- [9] Jakob Nielsen, Thomas K. Landauer, A mathematical model of the finding of usability problems // CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, 1993, p.206-213.
- [10] Robert A. Virzi, Refining the test phase of usability evaluation: how many subjects is enough? // Human Factors - Special issue: measurement in human factors, 1992, p. 457-468
- [11] William Albert, and E. Dixon, Is This What You Expected? The Use of Expectation Measures in Usability Testing // Proceedings of Usability Professionals Association 2003 Conference, 2003.