

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

## **Юзабилити в проекте QReal:Robots**

Курсовая работа

Выполнила: Соковикова Н.А.

Научный руководитель: Литвинов Ю.В.

Санкт-Петербург 2012

## Оглавление

Введение .....	3
Обзор существующих решений.....	5
Оценка юзабилити.....	7
Проведение тестирования.....	10
Сделанные изменения.....	10
Повторное тестирование .....	12
Заключение .....	13
Список литературы .....	14

## Введение

Программирование – важный предмет школьной программы. Детям сложно его изучать без специальных программ и приложений, так как они не могут видеть исполнение своего кода, могут только абстрактно представить себе ход его выполнения, и получить некоторый результат, по которому часто нельзя судить о правильности работы программы. Одним из первых проблему нематериальности программирования увидел информатик-психолог из MIT Сеймур Пейперт. Для ее решения он предложил язык LOGO. Результаты программирования на нем – это движение черепашки по экрану. Но даже красиво отображающийся на экране исполнитель недостаточно нагляден.

Современный уровень развития вычислительной техники позволяет создать недорогие устройства, как исполняющие внутреннюю программу, так и управляемые по беспроводной связи с персонального компьютера, и средства программирования для них. Самые распространенные на данный момент среды визуального программирования роботов – NXT-G, входящая в стандартный комплект конструктора, и Robolab. Однако у обеих есть свои недостатки. Так, NXT-G недостаточно функциональна, поэтому многие учителя информатики в школах отдают предпочтение Robolab, но Robolab требует более серьезной подготовки.

Среда программирования роботов QReal:Robots разрабатывается на основе metaCASE-системы QReal, предназначенной для создания специализированных сред визуального программирования, на кафедре системного программирования СПбГУ. Она имеет ряд преимуществ перед NXT-G и Robolab'ом. QReal:Robots предоставляет возможность работать с визуальным языком программирования и просматривать сгенерированный по блок-схеме код на языке C в режиме текстового редактора. Робот может непосредственно управляться с компьютера через Bluetooth, или на него может быть залита готовая программа. Еще одна особенность QReal:Robots – возможность выполнить программу на аналоге традиционного исполнителя – Logo-черепашки – прямо на экране компьютера. Моделируется трёхколёсная тележка с двумя ведущими колёсами. Это позволяет отлаживать программы без доступа к реальному устройству. Реализовано моделирование окружающей реальности робота, что дает возможность отлаживать с использованием модели довольно сложные программы. Однако наряду с функциональностью следует так же уделить внимание юзабилити приложения.

Юзабилити - это понятие, обозначающее итоговый уровень удобства чего-либо для использования в заявленных целях. Речь может идти о приложении, вебсайте, книге, инструменте - о чем угодно, созданном человеком. Юзабилити включает в себя анализ потребностей пользователей и исследование принципов, по которым объект воспринимается эффективным или приятным на вид.

Сложные программные продукты находят свое место в жизни людей, и в то же время, рынок заполнен соревнующимися брендами. В условиях сильной конкуренции разных брендов с примерно совпадающей функциональностью особое значение приобретает удобный и хорошо понятный пользовательский интерфейс. Это сделало юзабилити более популярным и признаваемым направлением, поскольку компании видят преимущества разработки, ориентированной на удобство пользователя.

Для проекта QReal:Robots вопрос юзабилити еще более актуален, чем обычно, так как это среда для обучения, и удобство ее использования может формировать у детей мнение об изучаемом предмете и в дальнейшем повлиять на выбор профессии. В то же время, дети более требовательны, чем взрослые, поэтому чтобы их заинтересовать новым, пока еще им непривычным средством программирования, необходимо, чтобы его интерфейс был понятен при минимальном времени, потраченном на его изучение.

Таким образом, исследование и улучшение юзабилити увеличивает шанс программы на успех, делает ее более конкурентоспособной. Моей целью было улучшить юзабилити QReal:Robots. Для этого требовалось

- наладить сотрудничество с психологом, юзабилити-аналитиком,
- подобрать подходящий способ оценки юзабилити,
- провести замеры,
- внести в проект изменения,
- снова провести исследование юзабилити.

## Обзор существующих решений

Для обучения программированию в школах в основном используются визуальные среды NXT-G и Robolab, с роботами Lego Mindstorms в качестве исполнителей программ.

NXT-G входит в набор Lego Mindstorms. Это ее несомненное преимущество, так как Lego Mindstorms – самые распространенные модели стандартных роботов. Но с другой стороны, эта среда больше подходит для ознакомления с подобными программами, так как в ней имеется только небольшой функционал, который не позволяет серьезно изучать программирование. Кроме того, исследования юзабилити этого языка [1], который заявлен как «понятный всем, независимо от уровня знания программирования» показали, что даже у опытных программистов некоторые его особенности вызывают сложности. Названия некоторых блоков не вполне соответствуют выполняемым ими действиям.

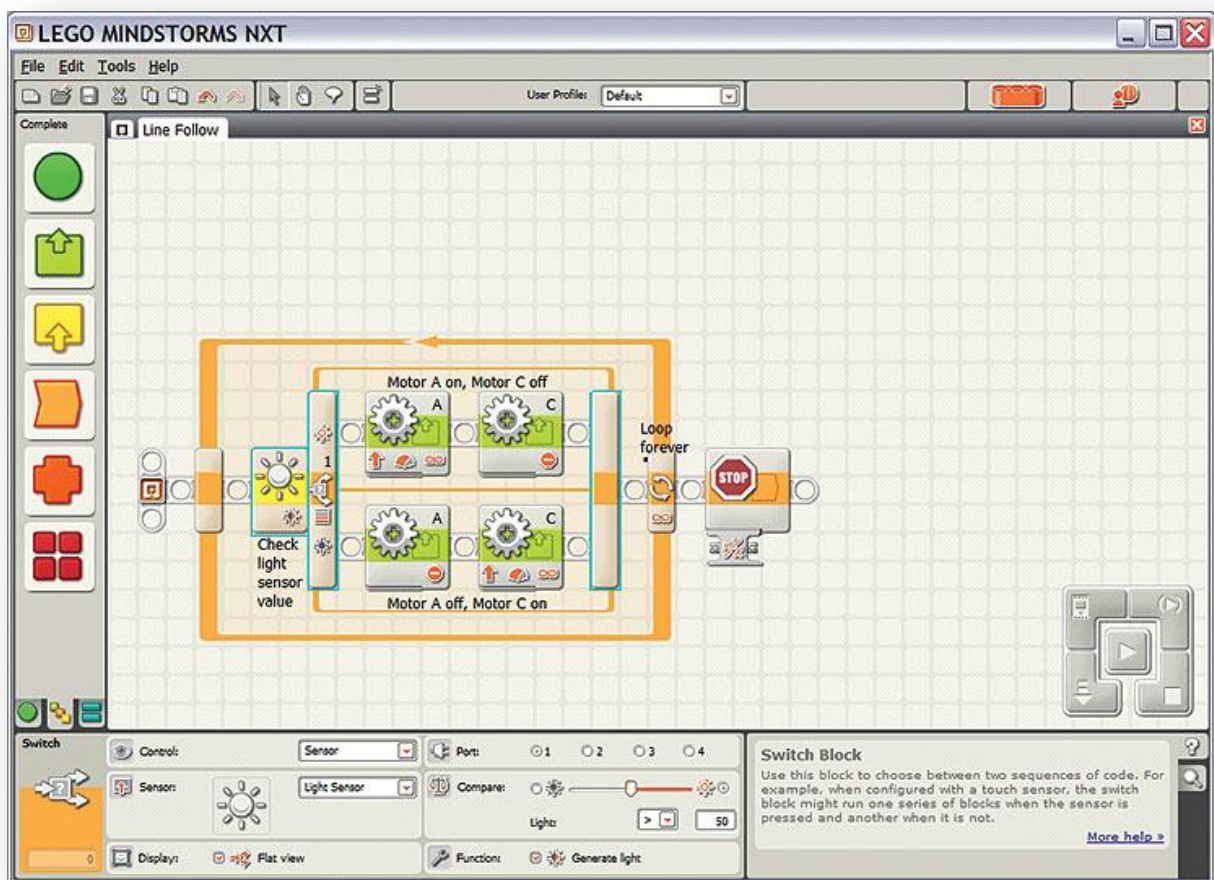


Рисунок 1 Среда визуального программирования роботов NXT-G

Среда Robolab – тоже графическая среда программирования. Она намного более функциональна, чем NXT-G. Тем не менее, и она тоже имеет ряд недостатков - в Robolab недостаточно полный и качественный перевод на русский язык, он долгое время не поддерживается и обладает устаревшим и недостаточно дружелюбным пользовательским интерфейсом. Язык в Robolab не так интуитивно понятен, как в NXT-G[2]. К тому же, Robolab является платной средой программирования.

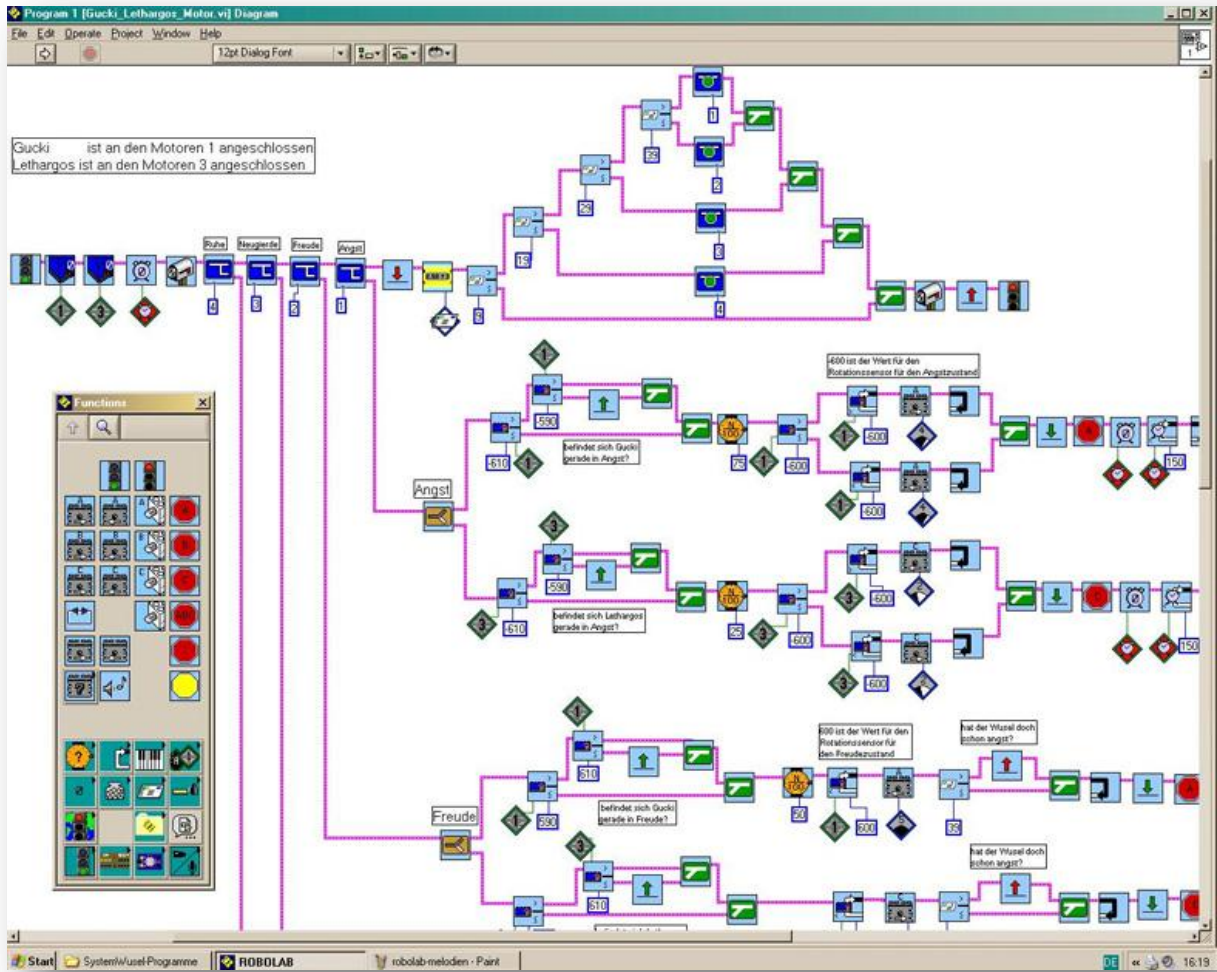


Рисунок 2 Среда визуального программирования роботов RoboLab

QReal:Robots является свободно распространяемым средством программирования роботов. В нем есть такие интересные особенности, как пошаговая отладка программы и переход от диаграммы поведения робота к текстовому представлению, что актуально для тех, кто больше интересуется программированием, нежели кибернетикой, или собирается перейти с визуальных языков на текстовые.

## Оценка юзабилити

Для того, чтобы можно было отслеживать прогресс, было необходимо вначале оценить юзабилити текущего проекта – то есть выбрать подходящий метод оценки интерфейса и по нему сделать определенные выводы. Далее следовало выявить недочеты интерфейса и исправить их. Затем нужно проконтролировать юзабилити в надежде, что изменения интерфейса помогли ее улучшить.

Юзабилити помогает сделать продукт более понятным и естественным для использования. То есть у человека, открывшего приложение даже впервые, не должно быть больших проблем с нахождением необходимых элементов интерфейса, их поиск не должен утомлять. Важно так же, как много времени понадобится пользователю, чтобы научиться с помощью приложения решать какие-то реальные проблемы. В интерфейсе все должно быть расположено так, чтобы основные функции всегда находились под рукой, были хорошо заметны. В то же время, если слишком много функционала будет вынесено на основную рабочую панель, лишние кнопки и значки будут сбивать и отвлекать внимание, что в конечном счете приведет к возникновению разнообразных ошибок, потере времени и снижению производительности. К тому же, по данным Forrester Research [3] 40% пользователей не возвращаются к приложениям, с которым имели негативный опыт работы, поэтому желательно, чтобы интерфейс приложения сразу заинтересовывал. Таким образом, оценить юзабилити, фактически, значит оценить, насколько программа соответствует следующим критериям:

- обучаемость - насколько просто пользователям освоить интерфейс;
- эффективность - как быстро они смогут выполнять различные задания, после знакомства с интерфейсом;
- запоминаемость - как легко пользователи могут восстановить свои навыки работы с приложением, не используя его некоторое время;
- наличие и серьезность ошибок - как много делают ошибок, почему они возникают, насколько серьезны и насколько сложно их исправить;
- удовлетворенность - насколько приятно использовать данный интерфейс;

Существует много различных методов исследования юзабилити.

### 1. Методы, не предполагающие участия пользователей – экспертная оценка.

- GOMS-модель – чисто количественный показатель. При такой модели оценки специалист просчитывает максимальное время, необходимое пользователю для выполнения стандартных задач. В расчет принимается, например, время, требуемое для нажатия кнопки на клавиатуре или клика мышкой. Эта модель не дает объективной оценки, так как интерфейс с довольно подробными инструкциями может быть лучше, но по GOMS-модели время получится довольно большим.
- Оценка качества пользовательского интерфейса на основе опыта проводящего экспертизу специалиста.

### 2. Методы с участием пользователей.

- методы, основанные на наблюдении за зрачками пользователя. По их результатам составляются карты экрана - туманные или тепловые, или графики взглядов, показывающие, на каких участках взгляд задерживается дольше всего. На туманных картах весь экран виден как будто сквозь туман, чем плотнее туман, тем меньше внимания пользователей привлекает эта область экрана. На тепловых картах части, на которых концентрируется внимание выделяются самыми “горячими” цветами, а те, мимо которых проскальзывает взгляд –

- холодными;
- “mouse-tracking” - запоминается траектория движения курсора мыши по экрану;
- ценить интерфейс можно количественно. Нужно померить скорость выполнения пользователями стандартных задач, учитывая общую скорость и количество ошибок, а так же успешность решения задачи;
- опросник удовлетворенности после работы с продуктом – пользователи отвечают на вопросы анкеты. Проведя опрос, можно получить более правдивое и объективное мнение об интерфейсе, так как опрос снижает влияние внешних факторов (таких как симпатия/антипатия к человеку, который тестирует, усталость и пр.).

По рекомендации специалиста, было решено совместить последние 2 метода. То есть вначале пользователям предлагалось выполнить несложное задание, а затем ответить на вопросы.

Для проведения подобных исследований чаще всего используются три вида анкет [4]:

1. QUIS (Questionnaire for User Interface Satisfaction) - анкета, разработанная в университете Мэриленда и содержащая 27 вопросов (например, “Ваша реакция на приложение в целом”). На каждый вопрос нужно ответить числом от 1 (“Ужасно!”) до 10 (“Великолепно!”);
2. Words - это список из 118 слов, напротив каждого из которых оставлено место для метки. Пользователя просят отметить слова, наиболее точно описывающие его впечатление от работы с приложением. Можно выбрать любое количество слов;
3. SUS (System Usability Scale) - этот опросник состоит из десяти утверждений, каждое из которых нужно оценить от 1 до 5, где 5 означает полное согласие с утверждением, а 1 - его совершенное отрицание.

1. Я бы хотел(а) еще поработать с этой программой.
2. Программа слишком сложная.
3. Этой программой легко пользоваться.
4. Мне понадобится помощь, чтобы научиться пользоваться этой программой
5. Разные функции в этом приложении правильно сгруппированы.
6. В приложении слишком много несоответствий.
7. Большая часть людей очень быстро научится пользоваться этой программой.
8. Это приложение очень трудно использовать.
9. Я уверенно себя чувствовал(а), используя это приложение.
10. Мне пришлось многому научиться, прежде чем я смог(ла) работать с приложением.

Рисунок 3. Анкета SUS

Для первого тестирования были выбраны школьники. Чтобы их не утомить длинным списком вопросов и по рекомендации эксперта, была выбрана третья анкета.

Результаты этой анкеты рассчитываются для каждого пользователя по следующим формулам [5]:

1. для вопросов с нечетными номерами из ответа пользователя вычитается 1;



2. для вопросов с четными номерами ответы пользователя вычитаются из 5;
3. теперь все оценки лежат в интервале от 0 до 4, где 4 - самый положительный ответ;
4. полученные цифры складываются и умножаются на 2.5;

Полученные оценки - от 0 до 100, но не стоит путать их с процентами. Примерный график перевода набранных баллов в проценты представлен на Рисунке 4. Средний считается интерфейс, набравший 68 баллов (соответствует 50%), а отличным - набравший 85 баллов и более.

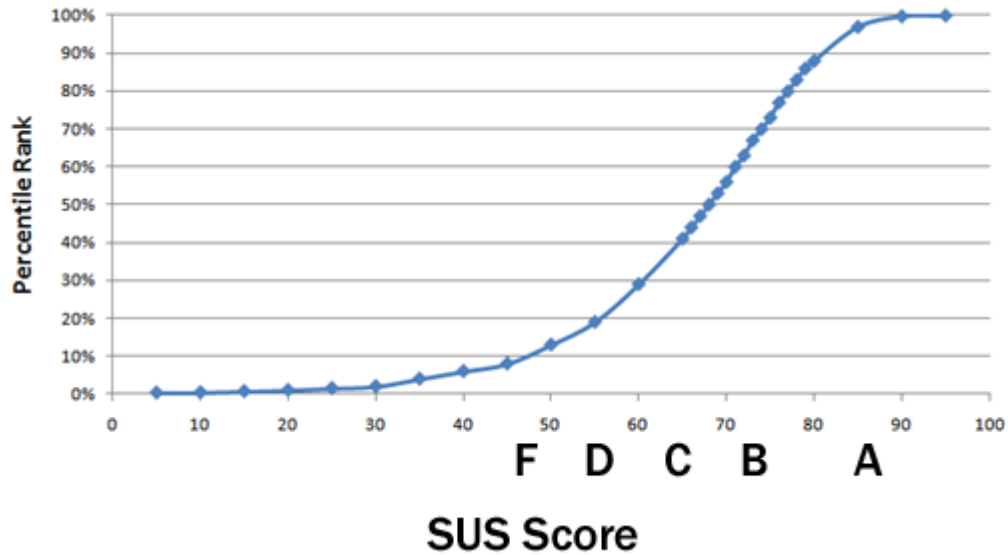


Рисунок 4 Перевод баллов в проценты

## Проведение тестирования

1. При проведении тестирования важно правильно выбрать людей – это должны быть наиболее вероятные пользователи программы. В данном случае, подходящие пользователи – ученики школ, студенты вузов и их учителя и преподаватели (так как QReal:Robots предполагается использовать для обучения школьников и студентов программированию и кибернетике).
2. Ни в коем случае нельзя допускать, чтобы тестируемые думали, что на самом деле оценивают их знания/способности, а не приложение. Это заставит их нервничать и, возможно, не правдиво отвечать на вопросы анкеты.
3. Нужно просить пользователей решить базовые, стандартные задачи. Цель этих задач – лишь определить, как быстро пользователь сможет использовать основные функции программы.
4. Следует индивидуально работать с каждым пользователем, но нельзя подсказывать, акцентировать на чем-либо внимание или задавать наводящие вопросы.
5. После выполнения/не выполнения задач, нужно расспросить человека о возникших ошибках, причинах их возникновения, как их удалось исправить. Попросить пользователя рассказать, что ему понравилось и не понравилось в программе.
6. Не стоит сразу же исправлять все замечания пользователя – они могут быть субъективными, однако их все же следует учитывать.

Первое тестирование проводилось на открытых соревнованиях Санкт-Петербурга по робототехнике. По проекту был сделан стендовый доклад и некоторых заинтересовавшихся им участников и гостей соревнований просили решить несложную задачу с помощью приложения. Она заключалась в том, чтобы составить программу, чтобы нарисованный робот доехал до стенки и остановился, затем изменить алгоритм так, чтобы он доехал до стенки и прогудел, потом добавить действия и подправить модель, чтобы робот после столкновения поехал обратно до стенки, находящейся изначально за ним. После чего им было предложено пройти анкетирование и рассказать о своих впечатлениях от QReal:Robots. Всего было опрошено 8 человек. Все они смогли справиться с задачей, не всегда без проблем и небольших ошибок... 6 из 8 посещают кружок по робототехнике ФМЛ №239, их опыт работы с Robolab очень помог им в составлении диаграммы поведения робота. Двое других ребят – тоже участники городских соревнований по робототехнике и также имеют большой опыт в работе с подобными приложениями.

Средние результаты анкеты - 76 баллов (наименьший результат - 22, наибольший - 96).

## Сделанные изменения

Специалисты утверждают, что вместо одного крупного исследования лучше провести несколько мелких тестов и вносить изменения в дизайн после каждого из них: так можно сразу исправлять обнаруженные ошибки. При этом для достоверных результатов достаточно, чтобы в нем принимали участие 5-10 человек [6].

В ходе первого тестирования участники столкнулись со сложностями, после анализа которых удалось найти некоторые недостатки текущего интерфейса. Многие

замечания относились к двумерной модели робота. Привыкшие к Robolab'у школьники, не только те, кто принял участие в исследовании, заинтересовались возможностью отлаживать программу на нарисованном роботе.

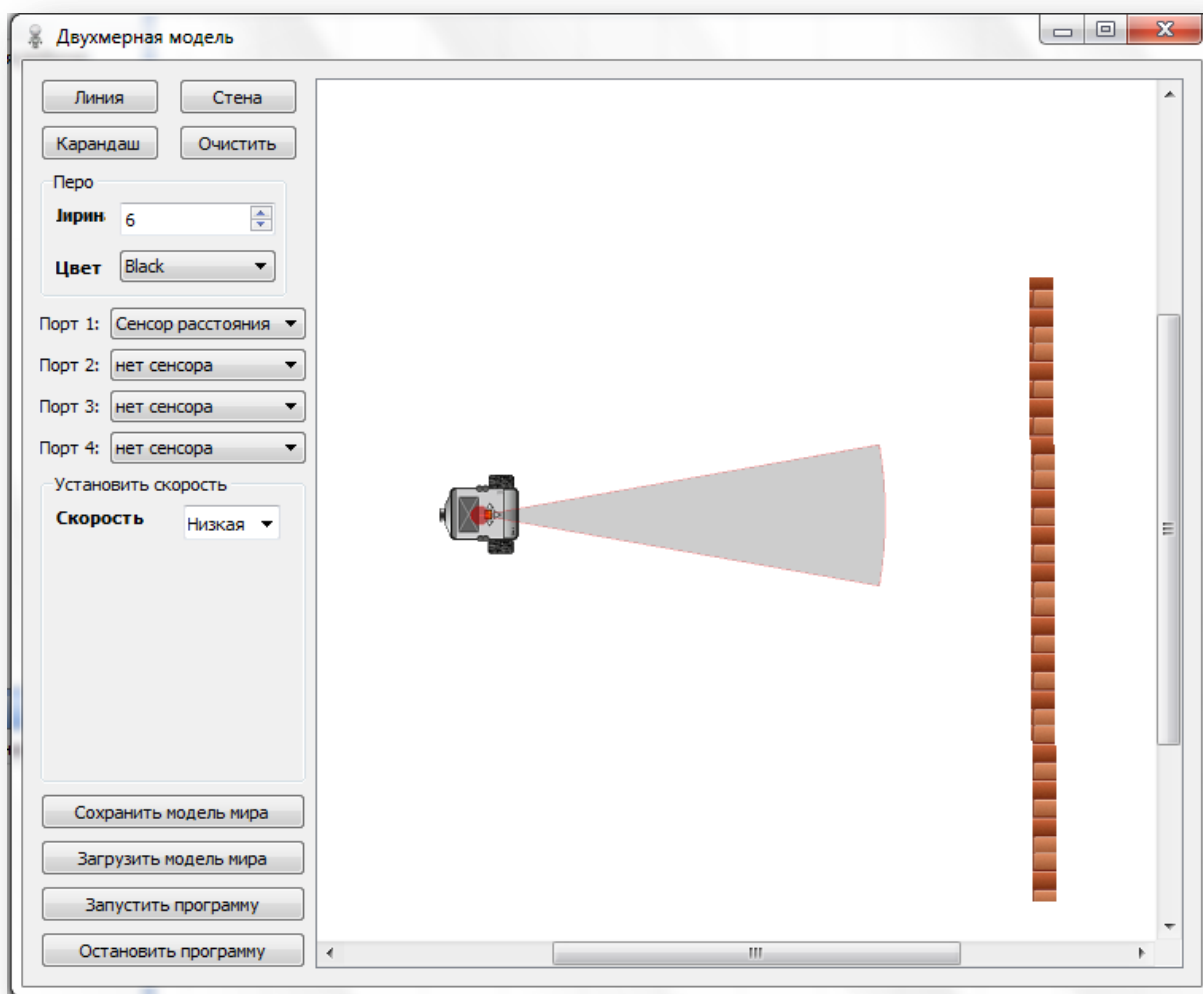


Рисунок 5 QReal:Robots, двумерная модель робота

Несмотря на то, что для настоящего исполнителя намного труднее подобрать в программах правильные коэффициенты (так как нужно учитывать различные характеристики покрытия и прочие внешние условия), двумерный робот может существенно облегчить написание программы. При работе с настоящим исполнителем, для отслеживания любых изменений в программе, ее приходится снова переносить на робота. Таким образом, с помощью нарисованной модели можно составить структуру программы, а затем найти правильные коэффициенты, подбирая их по настоящему роботу. Важность двумерной модели подчеркнул также руководитель кружка по робототехнике ФМЛ №239 Филлипов С. А., считая ее одной из наиболее значимых особенностей QReal:Robots. Поэтому нужно внимательно относиться к небольшим недочетам, имеющимся в ней. У всех без исключения участников тестирования возникли сложности с добавлением сенсоров на двумерную модель робота. Нужно было выбрать тип сенсора, нажать на кнопку "добавить", кликнуть мышью на место на экране, куда должен быть установлен сенсор, и лишь потом его становилось видно. Теперь достаточно выбрать тип сенсора, и он появится в середине изображения робота. Кроме того, скорость двумерной модели была очень низкой. Теперь ее можно менять, и

робот будет выполнять программу быстрее. Было сделано также множество других изменений - изменена палитра блоков (блоки сгруппированы по тематике), добавлена новая функциональность (блоки для балансировки сегвея).

## Повторное тестирование

Как уже было отмечено, рекомендуется проводить тестирования как можно чаще, поэтому после всех внесенных в проект изменений, было необходимо снова оценить юзабилити.

Основными предполагаемыми пользователями QReal:Robots являются школьники, студенты и их преподаватели информатики. Для проведения этого тестирования были выбраны студенты 1-3 курса Математико-Механического факультета. Как оказалось, они практически не знакомы со средами визуального программирования роботов, в отличие от участвовавших в первом тестировании школьников. Снова было выбрано 8 добровольцев и тестирование проходило по той же схеме, что и раньше, использовалась та же анкета (SUS).

Поскольку опрашиваемые не имели опыта работы с подобными программами, некоторые их замечания были вызваны тем, что они пробовали совершенно новую для себя среду программирования, но были и более конструктивные. Они обращали внимание на другие аспекты приложения, такие как

- необходимость более подробной справки;
- некоторые особенности работы со стрелками (они не зацепляются за блоки, если сделать излом, не удаляются автоматически вместе со связанным с ними блоком);
- некоторые обратили внимание, что им было бы удобнее перемещаться по экрану, указывая на нужную область в мини-карте,
- удобнее было бы с «горячими клавишами»;
- сложно найти двумерного робота, его лучше сделать в основном окне;
- по сообщениям об ошибках сложно понять, что не так;
- на двумерной модели плохо видно кнопки «запустить» и «остановить» программу;

Результаты анкетирования удивительны. В этот раз среднее набранное количество баллов оказалось 63.75 (самый маленький результат - 32.5, самый большой - 85). Возможно, это объясняется участием новых людей, которые раньше никогда не работали с подобными приложениями. Участники этого тестирования предложили намного больше идей по улучшению интерфейса, чем их предшественники, так что, вероятно, они просто отнеслись к оцениванию серьезнее и более осознанно отвечали на вопросы анкеты.

Все замечания пользователей следует теперь тщательно проанализировать. Понять, какие изменения стоит внести по результатам этого тестирования, сделать их и снова оценить юзабилити.

## Заключение

Среда визуального программирования роботов QReal:Robots, разрабатываемая на основе metaCASE средства QReal, имеет ряд преимуществ перед своими ранее появившимися конкурентами, такими как NXT-G, Robolab. Однако, функциональных преимуществ недостаточно для новой среды программирования, чтобы соперничать с уже давно привычными продуктами. Любой недочет может заставить заинтересовавшихся новым приложением пользователей вернуться к ранее изученным программам. Поэтому, чтобы QReal:Robots занял свое место в мире робототехники, необходимо сделать его одновременно эффективным и удобным в использовании, создать интерфейс, понятный практически интуитивно. Работу над всеми этими критериями включает в себя юзабилити. Чтобы улучшить уровень юзабилити проекта, нужно для начала выявить недостатки существующего решения. Выбрав подходящую методику оценки и анкету, мы провели юзабилити-тестирование на городских соревнованиях по робототехнике. С помощью тестирования удалось не только оценить юзабилити текущего интерфейса, но и выявить некоторые замечания, работа над которыми должна улучшить ее. После внесения в проект небольших изменений было проведено еще одно тестирование. Оно отличалось от первого выбором участников.

Юзабилити в первую очередь качественный показатель, в то время как анкета – количественный. И несмотря на результаты анкет, стоит отметить, что в этом тестировании даже менее опытные участники не упоминали проблем, которые возникали в первом тестировании. То есть выбранная методика (проведение тестирования, анализ проблем и ошибок, возникших у участников, работа над ними и повторение того же цикла) приносит результаты, поэтому планируется продолжать работать таким же образом

## Список литературы

1. Khuong A. Nguyen: A case study on the usability of NXT-G programming language, [http://www.scribd.com/fullscreen/63036716?access\\_key=key-cxsmmoddekdocaot6jq](http://www.scribd.com/fullscreen/63036716?access_key=key-cxsmmoddekdocaot6jq) [дата просмотра: 23.05.2012]
2. Dick Swan: Programming Solutions for the LEGO Mindstorms NXT, <http://find.botmag.com/100701>, [дата просмотра: 23.05.2012 ]
3. Forrester Research, <http://www.forrester.com/home>, [дата просмотра 21.05.2012 ]
4. Thomas S. Tullis and Jacqueline N. Stetson: A Comparison of Questionnaires for Assessing Website Usability, <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf>
5. Jeff Sauro, Measuring Usability With The System Usability Scale (SUS), <http://www.measuringusability.com/sus.php>
6. Якоб Нильсен (перевод - Александр Качанов), Элементарные основы юзабилити, <http://www.webmascon.com/topics/testing/14a.asp>
7. Брыксин Т.А., Литвинов Ю.В., Среда визуального программирования роботов QReal:Robots // Материалы международной конференции "Информационные технологии в образовании и науке". Самара. 2011. С. 332-334