(https://yandex.ru/collections/share/?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&description=%D0%9F%D0%BE%D0%BE%D0%BE%D0%BE%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%2 (https://wk.com/share.php?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0% (https://www.facebook.com/sharer.php?src=sp&u=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9-(https://connect.ok.ru/offer?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0%B (https://connect.maii.ru/share?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0% (https://plus.google.com/share?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0% (https://plus.google.com/share?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0% (https://plus.google.com/share?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0% (https://plus.google.com/share?url=https%3A%2F%2Frobot-help.ru%2Flessons-2%2Flesson-12.html&title=%D0%9F%D0%BE%D0%B4%D1%80%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9+%D1%80%D0%B0%D1%8B%D0%B9+%D1%80%D0%B0%D1%8B%D0%B9+%D1%80%D0%B0%D

Урок №12 - Сумо

Содержание урока

Введение:

Урок № 12 курса "Практика" (/lessons-2.html) представляет разбор решения, наверное, самого популярного соревнования роботов - "Сумо". Главная задача робота-сумоиста состоит в том, чтобы вытолкнуть за пределы ринга своего соперника. Существует множество вариантов правил проведения этого состязания, накладывающих различные ограничения в первую очередь на размеры и конструкцию роботов, тем не менее, базовые алгоритмы поведения робота при этом очень похожи. Поэтому наш урок в значительной степени будет направлен на реализацию программы поведения, а в завершающей части урока рассмотрим некоторые советы по практической подготовке к соревнованию роботов.

12.1. Поле для проведения соревнования. Правила соревнования.

Не смотря на то, что соревновательная дисциплина "Сумо" большинством организаторов соревнований позиционируется как состязание, предназначенное для начинающих робототехников, она пользуется неизменным вниманием не только участников, но и зрителей! Соревнования проводятся на двух видах полей: либо белый круг, ограниченный черной окружностью, либо, наоборот, черный круг, ограниченный белой окружностью. Очень часто макет поля наносится на круглый подиум небольшой высоты. В этом случае побежденный робот принуждается к падению (бывает, что с подиума падают оба участника битвы).

Для нашего урока мы возьмем то же самое поле, которое использовали при разборе Урока №11 - "Кегельринг" (/lessons-2/lesson-11.html). Это поле представляет собой белый круг, ограниченный черной окружностью. С классическими правилами правидами проведения соревнования "Сумо" по версии Российской ассоциации образовательной робототехники вы можете ознакомиться под слойлером ниже:

Регламент соревнований роботов "Сумо" по версии сайта RAOR.RU (https://raor.ru/)

12.2. Конструкция робота для соревнования "Сумо".

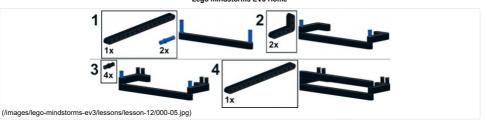
Базовое поведение робота в "Сумо" очень похоже на поведение робота в "Кегельринге". Роботу также необходимо найти внутри поля объект и вытолкать его за пределы круга. Различия, как водится, кроятся в деталях: теперь этот объект в свою очередь ищет нашего робота и тоже жаждет вытолкать его поскорее.



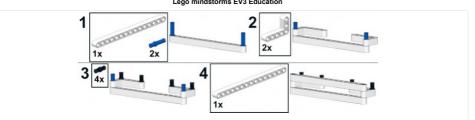
Тем не менее - сосредоточимся на своей цели: искать соперника нам по-прежнему будет помогать один из датчиков, способных определять предметы на расстоянии (инфракрасный или ультразвуковой), а своевременно определять черную границу поля будем с помощью датчика цвета. Поэтому для создания и отладки программы робота-сумоиста предлагаем вам использовать того же самого робота, которого мы подготовили для Урока Ne11 - Kerenьpuнr (/lessons-2/lesson-11.html#a112).

Для того, чтобы защитить впереди расположенный датчик от взаимодействия с соперником, соорудим бампер и закрепим его на нашем роботе. Ниже приведены подробные инструкции для сборки, как из домашней, так и из образовательной версии конструктора Lego mindstorms EV3. Можете поэкспериментировать и придумать собственный вариант конструкции.

Lego mindstorms EV3 Home



Lego mindstorms EV3 Education

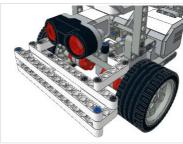


Получившийся элемент закрепим на передней балке нашего робота.

Lego mindstorms EV3 Home



Lego mindstorms EV3 Education



Наш учебный робот готов. Приступим к созданию программы робота-сумоиста. Замечательно, если у вас есть возможность отлаживать программу, используя ещё одного робота! Если же нет, то ничего страшного: можно задействовать в качестве соперника, например, радиоуправляемую модель автомобиля или те же кегли от "Кегельринга".

12.3. Создание программы для соревнования "Сумо".

Первая мысль, которая приходит в голову: использовать программу для "Кегельринга", внеся в неё косметические изменения. Действительно, алгоритмы поведения робота в "Кегельринге" и в "Сумо" очень похожи. Они реализуют поиск объекта и выталкивание его за пределы поля. Можно загрузить в робота-сумоиста программу для "Кегельринга", но работать такой сумоист будет не очень эффективно. Тем не менее, знания, полученные на предыдущем уроке, пригодятся нам сейчас.

Настало время загрузить в среду программирования наш проект "lessons-2", создать в нём новую программу "lesson-12" и подключить робота к среде программирования.

Поведенческую модель робота-сумоиста можно условно разделить на две части: **поиск соперника** и **атака соперника**. Сначала займемся реализацией первой части - **поиска соперника**.

Подробно пропишем последовательность действий нашего робота при обнаружении соперника на поле:

- 1. вращаться вокруг своей оси, пока впереди расположенный датчик не обнаружит соперника;
- 2. остановиться напротив соперника.

Эта последовательность действий полностью повторяет алгоритм поиска роботом кегли в **"Кегельринге"**, но, так как, расстояние между роботами в **"Сумо"** может превышать расстояние от робота до кегли, то нам необходимо выбрать другое пороговое значение для используемого датчика.

Установим соперников на поле напротив друг друга, как показано на рисунке ниже.



Такое положение практически соответствует максимальному удалению роботов друг от друга во время состязания, поэтому текущее показание датчика, измеряющего расстояние до соперника можно взять за пороговое. Важно: так как пороговое значение будет достаточно большим - необходимо чтобы за пределами поля на расстоянии около 1 м. во время работы робота также отсутствовали посторонние предметы, способные помещать поиску

На "Странице аппаратных средств", находящейся в правом нижнем углу среды программирования, выберем вкладку "Представление порта" (Рис. 1, 2 поз. 1) и снимем показание датчика, определяющего расстояние до соперника, установив соответствующий режим отображения поставления от пределяющего расстояние до соперника, установив соответствующий режим отображения поставления по соперника, установив соответствующий режим отображения поставления по соперника, установив соответствующий режим отображения по составления по соответствующий режим отображения по соответствующий режим от соответствующим от соотве

В нашем случае ультразвуковой датчик в режиме "Расстояние в сантиметрах" показывает значение - 56,1 (Рис. 1 поз. 2). За пороговое значение примем число - 57.

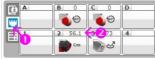


Рис. 1

Инфракрасный датчик в режиме "Приближение" показывает значение - 68 (Рис. 2 поз. 2). За пороговое значение примем число - 70.



Рис. 2

По аналогии с "Кегельрингом" мы можем запрограммировать нахождение роботом соперника, только, чтобы немного дистанцироваться от предыдущего урока, изменим направление вращения робота на противоположное:

Ультразвуковой датчик

- 1. Для того, чтобы заставить робота вращаться вокруг своей оси, воспользуемся программным блоком "Независимое управление моторами" "Зеленой палитры", Режим работы блока установим "Включить", значение мощности для порта "В" установим равным -30, значение мощности для порта "С" установим равным 30 (Рис. 3 поз.1).
- Для поиска соперника используем программный блок "Ожидание" "Оранжевой палитры" в режиме "Ультразвуковой датчик Сравнение
 Расстояние в сантиметрах" с пороговым значением срабатывания датчика, равным 57 (Рис. 3 поз. 2).

3. После того, как робот окажется напротив соперника, используя программный блок "Независимое управление моторами" "Зеленой палитры" выключим моторы (Рис. 3 поз. 3).



Инфракрасный датчик

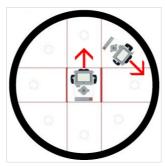
- 1. Для того, чтобы заставить робота вращаться вокруг своей оси, воспользуемся программным блоком "Независимое управление моторами" "Зеленой палитры", Режим работы блока установим "Включить", значение мощности для порта "В" установим равным -30, значение мощности для порта "С" установим равным 30 (Рис. 4 поз.1).
- 2. Для поиска соперника воспользуемся программным блоком "Ожидание" "Оранжевой палитры" в режиме "Инфракрасный датчик -Сравнение - Приближение", с пороговым значением срабатывания датчика, равным 70 (Рис. 4 поз. 2).

 3. После того, как робот окажется напротив соперника, используя программный блок "Независимое управление моторами" "Зеленой
- палитры" выключим моторы (Рис. 4 поз. 3).



На этапе отладки этого алгоритма вам придется, подбирая значения "Мощность" моторов "В" и "С" а также пороговое значение датчика, добиться от вашего робота точного обнаружения и остановки строго напротив соперника. Только после этого можно будет переходить к

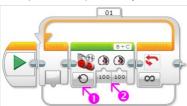
Если поиск соперника в "Сумо" очень похож на поиск кегли в "Кегельринге", то выталкивание соперника имеет важное отличие! Начиная атаку, первое, что необходимо сделать, это прямолинейно устремиться на максимальной мощности моторов в сторону обнаруженного соперника. проверяя датчиком цвета обнаружение границы ринга. Но ведь наш соперник тоже может двигаться! Поэтому вполне возможна ситуация, когда соперник выйдет в сторону из-под направления нашей атаки. В этом случае, наш робот, промахнувшись, будет двигаться в сторону границы ринга, теряя соперника и драгоценное время



Следовательно, нам необходимо во время прямолинейного движения вперед анализировать оба датчика и прекращать атаку в случае, если робот потеряет соперника ИЛИ робот достигнет границы ринга. Поэтому нам необходимо отказаться от использования программного блока "Оранжевой палитры" и самостоятельно в цикле получать и обрабатывать показания двух датчиков.

Приступим к поэтапной реализации алгоритма атаки соперника: для этого создадим в проекте временную программу "lesson-12-1" и начнем её наполнение программными блоками

- 1. Возьмем программный блок "Цикл" "Оранжевой палитры".
 2. Внутрь блока "Цикл" поместим программный блок "Независимое управление моторами" "Зеленой палитры". Режим работы блока установим в значение "Включить" (Рис. 5 поз. 1), мощности моторов "В" и "С" установим в максимальное значение 100 (Рис. 5 поз. 2).



3. Следом за блоком "Независимое управление моторами" поместим программ ый блок **"Датчик цвета" "Желтой палитры"**. Режим работы блока установим в значение "Сравнение - Яркость отраженного света" (Рис. 6)



Рис. 6

В этом режиме программный блок "Датчик цвета" "Желтой палитры" визуально очень похож на программный блок "Ожидание" "Оранжевой палитры" в режиме "Датчик цвета - Сравнение - Яркость отраженного света". Но, в отличие от блока "Ожидание", этот программный блок не ждет выполнения условия, указанного параметрами "Тип сравнения" (Рис. 7 поз. 1) и "Пороговое значение" (Рис. 7 поз. 2), а сразу выдает логическое значение ("Истина" или "Ложь") в выходном параметре "Результат сравнения" (Рис. 7 поз. 3) и измеренное значение - в выходном параметре "Освещение" (Рис. 7 поз. 4).

Параметры "Тип сравнения" и "Пороговое значение" на Рис. 7 поз. 1, 2 установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 7 поз. 3) выдавал логическое значение "Истина" при пересечении датчиком цвета черной границы ринга.



4. В случае использования ультразвукового датчика за блоком "Датчик цвета" установим программный блок "Ультразвуковой датчик" "Желтой палитры". Режим работы блока установим в значение "Сравнение - Расстояние в сантиметрах" (Рис. 8 поз. 1). Параметр "Тип сравнения" (Рис. 8 поз. 2), параметр "Пороговое значение" (Рис. 8 поз. 3) установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 8 поз. 4) выдавал логическое значение "Истина" в случае потери из виду роботом соперника.

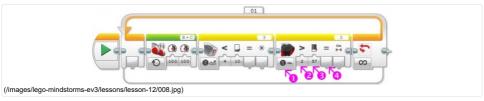


Рис 8

В случае использования инфракрасного датчика за блоком "Датчик цвета" установим программный блок "Инфракрасный датчик" "Желтой палитры". Режим работы блока установим в значение "Сравнение - Приближение" (Рис. 9 поз. 1). Параметр "Тип сравнения" (Рис. 9 поз. 2), параметр "Пороговое значение" (Рис. 9 поз. 3) установим таким образом, чтобы выходной параметр "Результат сравнения" (Рис. 9 поз. 4) выдавал логическое значение "Истина" в случае потери из виду роботом соперника.



Рис. 9

Давайте ещё раз проанализируем промежуточный код нашего алгоритма атаки: мы включили моторы на максимальную мощность и движемся вперед, постоянно в цикле опрашивая датчики. Если наш робот пересечет черную линию границы ринга, то значение выходного параметра "Результат сравнения" "Датчика цвета" примет значение "Истина". Если наш робот потеряет соперника, то значение выходного параметра "Результат сравнения" датчика, следящего за соперником, также примет значение "Истина". В любом из этих случаев нам следует прекратить атаку, завершив наш цикл. В этом нам поможет программный блок "Логические операции" "Красной палитры". Познакомимся с этим блоком подробнее: программный блок "Логические операций над погическими данными (Рис. 10).



Рис. 10

Выбранный режим программного блока "Логические операции" "Красной палитры" определяет одну из четырех операций над логическими данными: "И (AND)", "ИЛИ (OR)", "Исключаеше ИЛИ" и "Исключение (NOT)". Два входных параметра "а" и "b" (для операции "Исключение (NOT)" - один входной параметр "а") передают в программный блок входные значения, а результирующее значение выдается выходным параметром "Результат". Если вы ранее не сталкивались с логическими операциями, то можете ознакомиться с базовыми знаниями в поилагаемой справке под спойлером.

Логические операции

- 5. За программным блоком "Ультразвуковой датчик" или "Инфракрасный датчик" поместим программный блок "Логические операции" "Красной палитры".
- Выходной параметр "Результат сравнения" программного блока "Датчик цвета" (Рис. 11, 12 поз. 1) соединим с входным параметром "а" программного блока "Логические операции" (Рис. 11, 12 поз. 4).
- Выходной параметр "Результат сравнения" программного блока "Ультразвуковой (инфракрасный) датчик" (Рис. 11, 12 поз. 2) соединим с входным параметром "b" программного блока "Логические операции" (Рис. 11, 12 поз. 5).
- Режим работы программного блока "Логические операции" установим в "ИЛИ (ОR)" (Рис. 11, 12 noз. 3). В этом случае результат выполнения логической операции будет принимать значение "Истина", только если будет выполнено одно из условий: датчик цвета пересёк черную линию, робот потерял сопераника.
- «Установив режим программного блока "Цикл" в значение "Логическое значение" (Рис. 11, 12 поз. 7), выходной параметр "Результат" программного блока "Погические операции" (Рис. 11, 12 поз. 6) соединим с входным параметром "Пока не будет истина" программного блока "Цикл" (Рис. 11, 12 поз. 8). Данные настройки завершат выполнение цикла при "Истинном" результате выполнения логической операции.

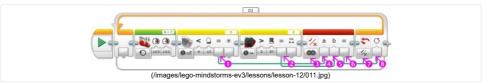


Рис. 11

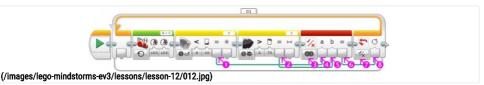


Рис. 12

Давайте протестируем получившийся алгоритм атаки! Для этого поместим нашего робота внутрь ринга, напротив установим неподвижного соперника и запустим программу атаки на выполнение. Наш робот должен уверенно вытолкать соперника за пределы ринга и остановиться над черной границей поля. Получилось? Значит наш сумоист верно контролирует границу ринга.

Проведем второй эксперимент: снова установим напротив робота неподвижного соперника и запустим программу атаки. Когда наш робот устремится к сопернику и приблизится достаточно близко, резко уберём соперника в сторону. Наш робот должен, потеряв соперника, остановиться.

Подведем итог: мы реализовали алгоритм поиска соперника и успешно его протестировали, также прошел проверку алгоритм атаки.

Законченная программа сумоиста должна в бесконечном цикле выполнять последовательно поиск соперника, а затем - атаку соперника. Можно был бы уже объединить обе части нашей программы, если бы не одно маленькое дополнение. Если наш робот остановился над границей ринга, то перед тем, как начать поиск, роботу следует, отъехав немного назад, вернуться внутрь ринга. Дополним нашу программу атаки следующим кодом: за пределами цикла атаки, воспользуемся программным блоком "Переключатель" "Оражжевой палитры". Режим работы блока "Переключатель" установим в "Датчик цвета - Сравнение - Яркость отраженного света". Параметры "Тип сравнения" и "Пороговое значение" установим аналогично ранее используемым в программном блоке "Датчик цвета" "Желтой палитры". Следовательно, если наш робот остановился над черной линией, то выполнение будет передано верхнему контейнеру программного блока "Переключатель". Именно в верхний контейнер поместим программный блок "Рулевое управление" "Зеленой палитры", с настройками параметров, заставляющими робота отъехать назад на один оборот моторов. В нижний контейнер программного блока "Переключатель" поместим программный блок, выключающий моторы (Рис. 13). Повторно протестировав алгоритм атаки, убедимся, что после того, как робот-сумоист вытолкал соперника за пределы ринга, он вернулся немного назад.

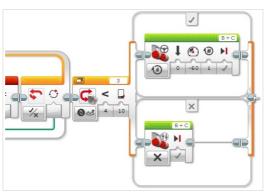


Рис. 13

Вот теперь можно завершить разработку программы для робота-сумоиста. Внутрь бесконечного цикла последовательно вложим программу поиска соперника, а затем программу атаки соперника. Попробуйте выполнить эту работу самостоятельно, не подглядывая в решение.

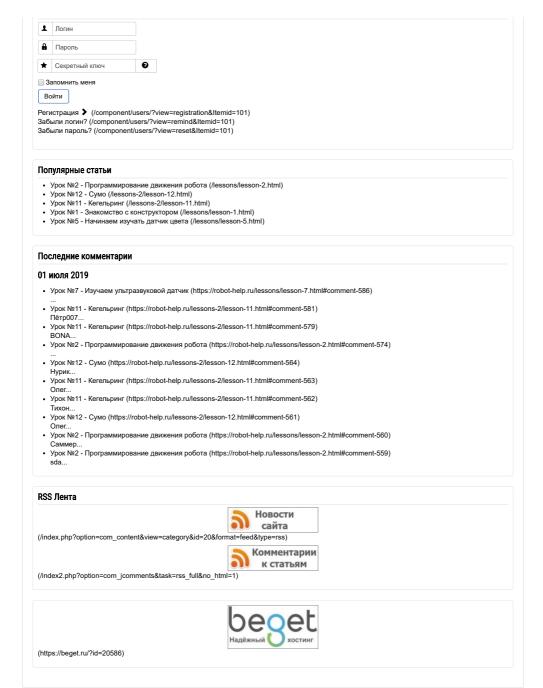
Программа робота-сумоиста

Заключение:

Программа, которую мы разобрали с вами на этом уроке, реализует только один прямой силовой алгоритм поведения робота-сумоиста. Она подразумевает, что в прямом силовом противостоянии робот должен непременно одолеть своего соперника. Но наш учебный робот, конечно же, своем не покож на мускулистого борца-сумо. Для того, чтобы уверенно выступить в выступить в ном состязании, необходимо уделить самое пристальное внимание в первую очередь конструкции робота, создать прочную, защищенную платформу, с помощью дополнительных ведущих колес или гусениц повысить сцепление с поверхностью ринга. На популярном видеохостинге Youtube.com (https://www.youtube.com/) по запросу "сумо lego роботов" можно найти множество видеороликов с реальных соревнований роботов, из которых вы непременно почерпнёте для себя интересные идеи для реализации в собственных конструкциях.

Главная же цель этого урока - на практическом примере показать вам метод непрерывной обработки показаний от пары датчиков. Можно ли усовершенствовать нашу программу? Безусловно! Например, используя программный блок "Случайное значение" "Красной палитры", изменить алгоритм поиска соперника таким образом, чтобы задавать случайное вращение робота влево или вправо, тем самым, дезориентируя соперника. Попробуйте самостоятельно встроить в нашу программу этот дополнительный код. Подумайте так же над тем, какие изменения нужно внести в прорамму, в случае проведения соревнования на черном ринге с белой границей. Возможно, что у вас появятся собственные идеи упучниения: полевитесь ими в комменталиях к уроку!

ментарии (/component/jcomments/feed/com_content/12.html)	
1 2 3 4 5 6	
#51 (/lessons-2/lesson-12.html#comment-537) Олег Олейник 28.03.2019 20:17	+3
Цитирую Житлухин Евгений:	
здравствуйте я смотрел битвы сумо и роботы в начале они ждали пять секунд	
Так поставьте в начале программы блок ожидания по времени на 5 секунд 🖰	
Tak neorable of the late specificant of the consequents to be perfectly the consequence of the consequence o	Цитироват
#52 (/lessons-2/lesson-12.html#comment-558) Житлухин Евгений 13.04.2019 13:55 у меня он ждёт 10 секунд	+1
	Цитироваті
#53 (/lessons-2/lesson-12.html#comment-561) Олег Олейник 19.04.2019 16:30	
Цитирую Житлухин Евгений:	
у меня он ждёт 10 секунд	
Не видя программы ничего не могу подсказать. Увы.	
	Цитироват
#54 (/lessons-2/lesson-12.html#comment-564) Нурик 22.04.2019 11:05	-2
Цитирую Житлухин Евгений:	
у меня он ждёт 10 секунд	
извените сылка можете дат спс	
	Цитироват
RSS лента комментариев этой записи (/component/jcomments/feed/com_content/12.html) Saвить комментарий Внимание!	
Возрастная категория посетителей сайта - (10+). Все комментарии перед публикацией проходят модерацию.	
Имя (обязательное)	
E-Mail (обязательное) ⇒ ⇔ ⇔ ⇔ ⇔ ⇔ ⇔ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕	
Осталось: 1000 символов Подписаться на уведомления о новых комментариях	
Я не робот recapticha	
Конфиденциальность - Усповия использования Отправить	
	JComments (http://www.joomlatune.



© 2019 robot-help.ru Hasepx