Руководство для пользователя среды Scratch

Версия 0.2 Евгений Патаракин

2007-12-19

Скретч это новая среда программирования, которая позволяет детям создавать собственные анимированные и интерактивные истории, игры и другие произведения. Этими произведениями можно обмениваться внутри международной среды, которая постепенно формируется в сети Интернет.

Скретч базируется на традициях языка Лого и Лего-Лого. Может быть не всем в России знаком язык Лого и его агент-исполнитель Черепашка, но уж конструкторы Лего знают все. В среде Скретч используется метафора кирпичиков Лего, из которых даже самые маленькие дети могут собрать простейшие конструкции. Но, начав с малого, можно дальше развивать и расширять свое умение строить и программировать. Скретч создавался специально для того, чтобы подростки 10 – 16 лет использовали его самостоятельно в сети внешкольного обучения на базе Компьютерных Клубов, в городах и районах с низким уровнем доходов. Эта новая технологическая среда позволяет им выразить себя в компьютерном творчестве.

Разработчики этого направления во главе с Аланом Кеем считают, что наиболее важным в мощных и недорогих персональных компьютерах является то, что они создают новую среду чтения и письма. В этой новой среде люди могут обсуждать и играть с новыми мощными идеями совсем не так, как они это делали с книгами. С этим и связана работа, которую ведет исследовательская группа во главе с Аланом Кеем со средой со Сквик. Цель деятельности группы - помочь детей научиться думать лучше и глубже, чем это могут делать большинство взрослых. Для достижения этой цели они создали среду Squeak как новую форму электронной бумаги, на которой могут быть представлены новые пути представления новых мощных идей. Теперь в среде Сквик есть новые письменные образцы этой новой литературы и эти образцы публикуются в сети Интернет, для того, чтобы и взрослые и дети могли бы читать и играть с ними. Одним из ответвлений среды Сквик является проект Scratch, в котором вновь появляются многие идеи программирования, свойственные среде Лого и Лего-Лого. Но, теперь они воплощены на более высоком уровне. Пользователи могут собирать свои

программы-процедуры из блоков так же, как они собирали конструкции из кирпичиков Лего. Из конструкций и управляющих структур могут быть собраны различные агенты, выполняющие простые инструкции и, следовательно, обладающие поведением. Эти агенты могут взаимодействовать между собой и воссоздавать сообщество в среде Скретч.

Возможность собрать программу из отдельных кусочков и обсудить задачу, которую выполняет каждый отдельный фрагмент, существует и в постепенно входящей в мир компьютерного образования среде Squeak и ее клонах Alice и Scratch. Учебная среда **Scratch** появилась совсем недавно и делает в настоящее время свои первые шаги. Ее будущее выглядит очень многообещающим, поскольку в основании среды лежит опыт Лого, мощные возможности современного языка **Squeak** и стоят такие люди как Мич Резник и Алан Кей. Кей в предисловии к языку Squeak подчеркивает, что компьютеры создают новую форму электронной бумаги для чтения и письма. На этой новой электронной бумаге люди могут обсуждать и играть с новыми мощными идеями совсем не так, как они это делали с книгами. Мы можем получать новое визуальное представление о гипотезах и теориях, можем связывать и сравнивать разные идеи друг с другом. Если мы будем помнить о новых возможностях, которые открывают перед нами электронные формы общения и сообщного мышления, то сможем стремиться не просто к обучению чтению и письму, но и к более глубокому усвоению знаний. Новый стиль усвоения знаний даст ученикам возможность читать литературу, обдумывать идеи и представлять их в форме доступной для обсуждения и критики. Возможность обсуждения и критики является одним из самых древних оснований обучения. Опыт не просто знакомства, но и взаимодействия с глубокими идеями помогает вырастить людей с углубленным пониманием окружающей действительности. И не просто более глубоких людей, но и людей, которых будет значительно труднее ввести в заблуждение, поскольку у них будет заложен опыт критического отношения и активной работы с любыми мифами и мимами.

Скретч разрабатывался как новая учебная среда для обучения школьников программированию и позиционировался авторами как альтернатива культуре PhotoShop. В Скретче можно создавать фильмы, играть с различными объектами, видоизменять их вид, перемещать их по экрану, устанавливать формы взаимодействия между объектами. Это объектно-ориентированная среда, в которой блоки программ собираются из разноцветных кирпичиков команд точно так же как машины собираются из разноцветных кирпичиков в конструкторах Лего. Опыт и метафоры Лего-Лого в среде Скретч лежат на поверхности. Мы можем взять любой объект и осуществить над ним действия:

- видоизменить его внешний вид при помощи фиолетовых команд-кирпичиков внешнего вида;
- переместить его при помощи синих команд кирпичиков перемещения;
- поместить команды перемещения и видоизменения внутрь оранжевых блоков управления;
- добавить к управляющим блокам фиолетовые кирпичики звуков;
- и т.д.

В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Начальный уровень программирования столь прост и доступен, что Скретч рассматривается в качестве средства обучения не только старших, но и младших школьников.

Как запустить программу Скретч?

Достаточно ткнуть указателем мышки в мордочку симпатичному зверьку – Скретч и запустится

Верхняя часть:

New Создать Новый проект

Open – открыть проект

Save – сохранить текущий проект

SaveAs – сохранить проект под новым именем

Extra – дополнительные возможности

Undo – отмена последней команды.

Левая часть рабочего экрана представляет ящички, в которых лежат кирпичики для программирования. Можно не знать английских слов и ориентироваться в цветах ящичков.

Фиолетовый Looks - Изменение внешнего вида

Синий Motion - Перемещение объектов

Лиловый - Sound - звуки

Зеленый - Numbers Операции с числами, сравнение, логические операторы

Желтый - Control контролирующие операторы

Ярко-зеленый - Pen карандаш - возможность рисовать

Голубой – Sensors датчики

Красный - Variables - переменные

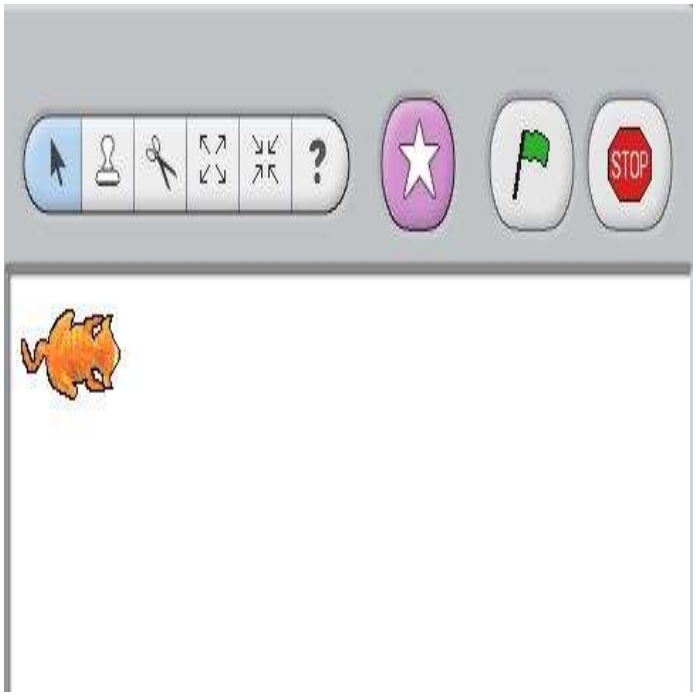
Центральная часть

Scripts - область, где из элементов собираются управляющие скрипты

Costume - костюмы, которое может надевать на себя кошка

Sounds - звуки, которые кошка может издавать.

Справа - область экрана или мира, где существует кошка и все другие существа, которых мы можем в мир призвать.



Для того чтобы вызвать новое существо – (новый объект) достаточно нажать на звездочку на фиолетовом фоне.

Зеленый флажок служит в качестве сигнала, который воспринимают и на который реагируют все присутствующие в данном проекте объекты

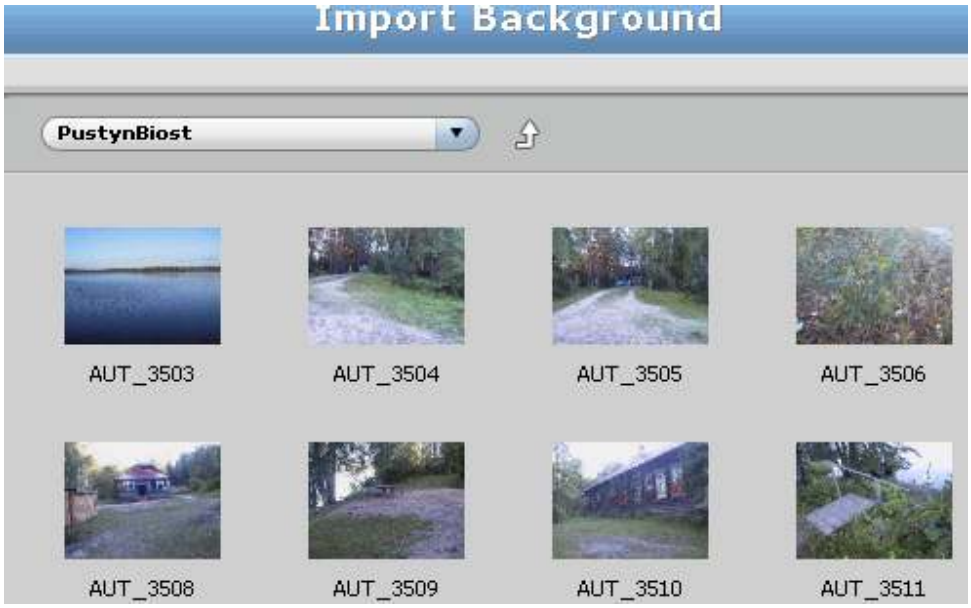
Красная кнопка Stop - останавливает движения и действия всех объектов.

Создание новых объектов

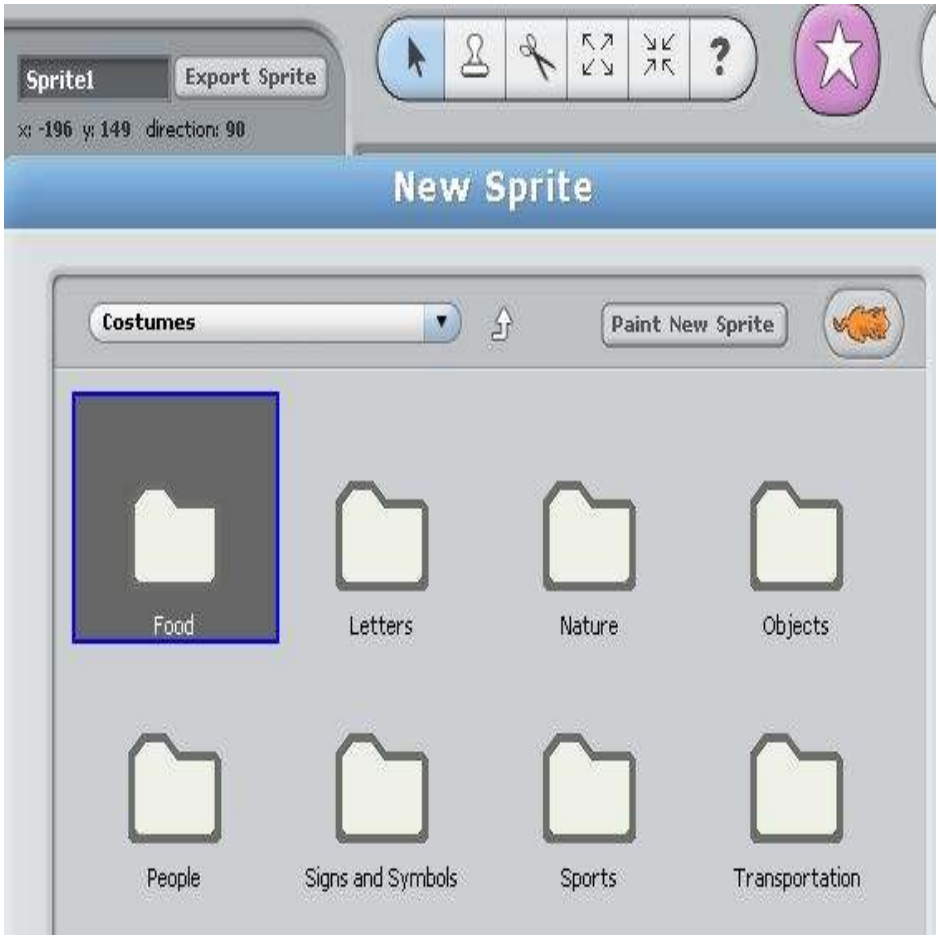
Мы можем создавать новые объекты – для этого достаточно просто нажать на звездочку на розовом фоне в правом верхнем углу экрана.

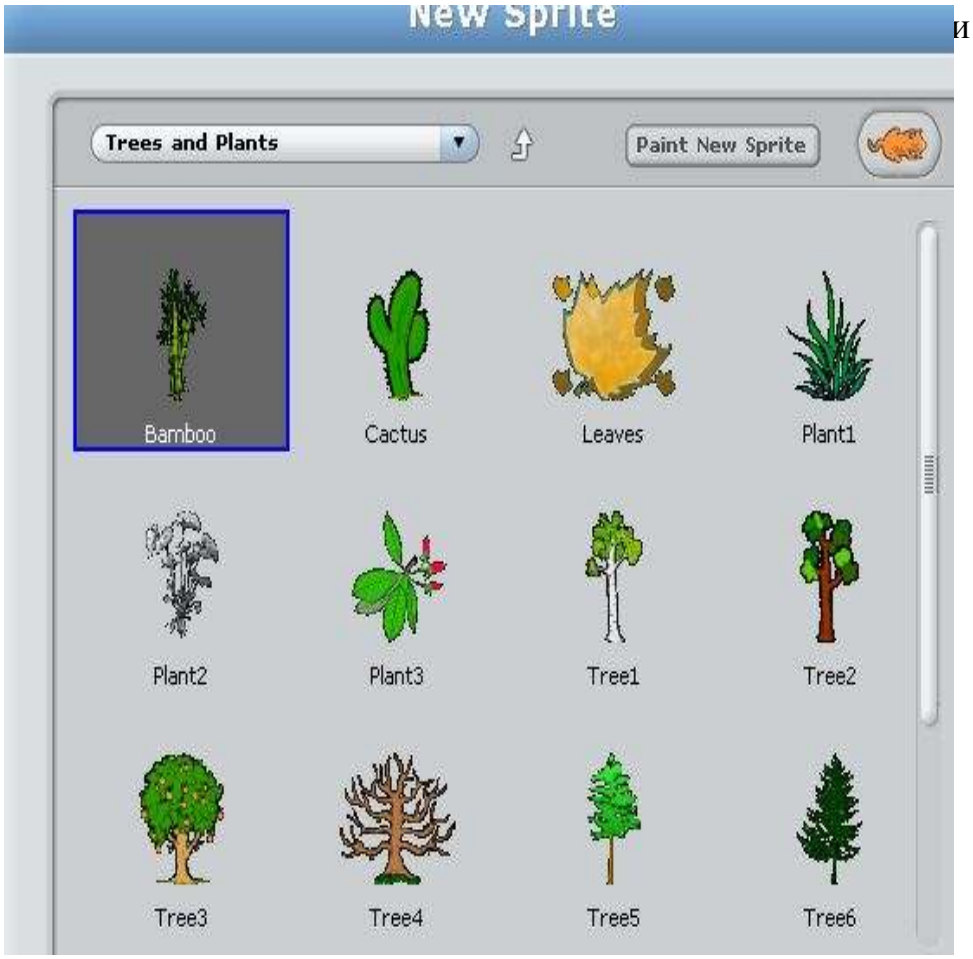
Среда, на фоне которой происходит действие, является таким же импортируемым и видоизменяемым объектом. Мы можем нарисовать фон, на котором происходит действие. Мы можем взять для этого фона картинку из коллекции, которая пришла к нам вместе со средой Скретч, и мы можем самостоятельно добавить к коллекциям фона свои рисунки или фотографии. Размер рисунков и фотографий для фона – 480 на 360

В данном примере мы добавили фотографии, которые были сделаны на биостанции в Старой Пустыни.



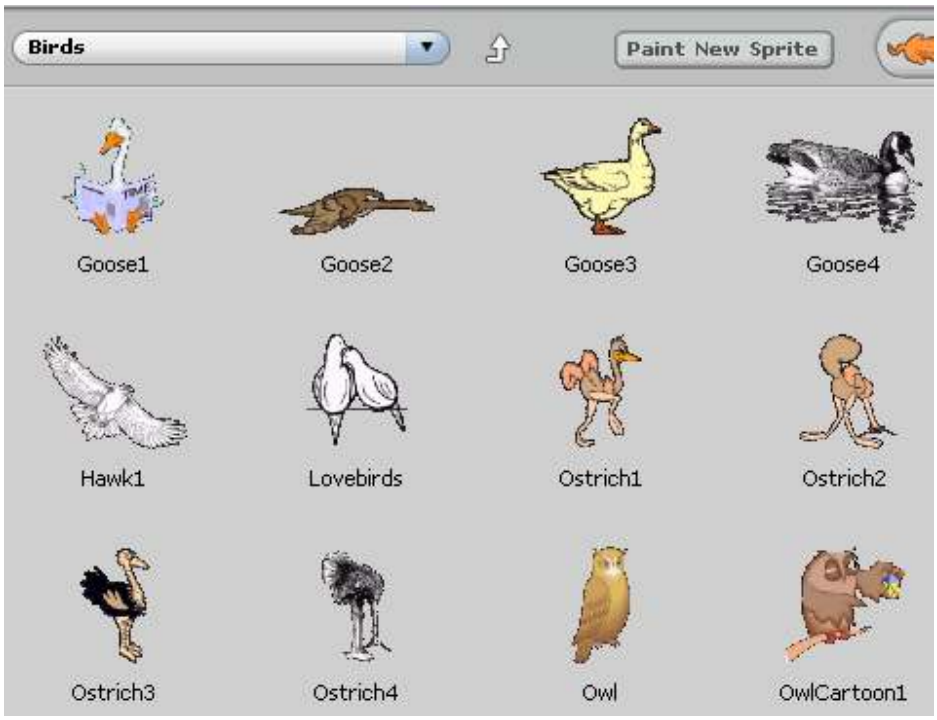
Когда мы создаем новое существо, то мы можем выбрать его внешний вид из библиотеки костюмов - тут люди, звери, растения, продукты, машины, сказочные существа. Мы и сами можем добавить любое изображение и даже нарисовать и отредактировать это изображения прямо внутри программной среды.





и выбрать там папку, в

Другая папка содержит изображения птиц, которые мы и будем использовать в ближайших примерах:



Добавим два объекта к фону пустынского пейзажа:

Управляющие программы - скрипты

Перейдем к ящичкам, в которых сложены кирпичики. Если мы посмотрим на содержимое ящичков повнимательнее, то увидим, что содержимое отличается не только цветом. Есть ящички, где большей частью лежат строительные блоки, которые можно объединять друг с другом. Это кирпичики для изменения внешнего вида объектов, управления звуками, перемещения объектов и рисования. Эти строительные кирпичики делятся на закрытые и самодостаточные, к которым ничего добавить нельзя и внутрь которых ничего нельзя запихать. Например: beep, show, done, if on edge bounce. Есть кирпичики, в которых можно изменять значения или вставлять в них значения. Например: move(20), set size ()%

Есть ящички, в которых лежат блоки, которые нельзя соединять с другими, но которые можно вставлять внутрь других строительных блоков. Это ящики, где лежат цифры и датчики.

Отдельный ящик (Control) с управляющими конструкциями, которые позволяют организовывать повторения команд, ставить условия, при которых блоки команд начинают выполняться и условия, когда действие команд прекращается

Внешний вид объекта Looks

set costume to () – выбрать костюм для объекта. По умолчанию объект одет в костюм кошки, но мы можем выбрать и другие костюмы из библиотеки, пользуясь командой import

change costume by() – изменить значение костюма на одну или несколько единиц. Имеет смысл только тогда, когда у нас несколько «импортных» костюмов.

say() – сказать фразу, которую мы можем записать в окошечке команды. К сожалению только латинские буквы.

say nothing – буквально промолчать. Команда убирает с экрана все предварительно сказанные слова.
Например: say(Hello, world); wait(2); say nothing

clear graphic effects – очистить все графические эффекты. Если мы производили над объектом видоизменение, то в результате этой команды все они отменяются

change (color, fisheye, whirl, pixalate, mosaic, brightness) effect by() – видоизменить объект по одному из перечисленных параметров на указанную величину. Ниже приводится перечень параметров, по которым можно видоизменять внешний вид

объекта. Обратите внимание, что команды все больше напоминают обычные короткие предложения

change fisheye effects by(25) – объект округляется

change fisheye effects by(-25) – объект становится более тощим

Whirl+50

Whirl-50

Transparency

pixelate

mosaic

change size by ()

set size to ()%

show – показать объект – он становится видимым

hide – спрятать объект – он становится невидимым

go to front – объект перемещается в первый слой

go back () layers – объект уходит на несколько слоев внутрь изображения.

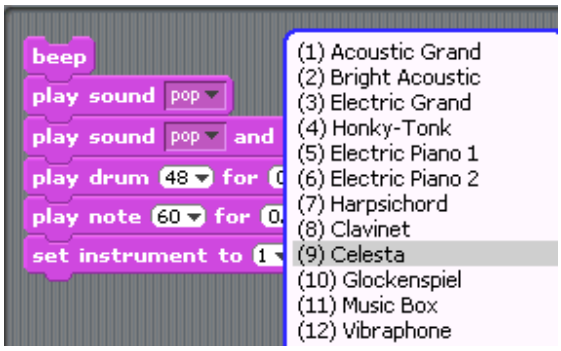
Вот такой полезный фиолетовый ящик.

Команды приобретают смысл, когда Вы объединяете их в последовательность

Например: change fisheye effects by(25) wait 2 change fisheye effects by(-25) – и птица начинает махать крыльями

Все команды применимы не только к подвижным объектам, но и фону! Т.е. мало того, что можно добавлять разные фотографии в качестве фона. Над этими фоновыми фотографиями можно совершать любые операции при помощи команд из фиолетового ящика.

Лиловый ящик - Звуки



beep – самая простая команда просто бибикнуть.

play sound() – команда издать звук. При этом звук можно выбрать в библиотеке – там этих звуков достаточно много. Так же как и библиотека картинок, библиотека звуков расширяется и к ней можно добавлять свои местные звуки в формате wav

set instrument() – выбрать инструмент, который будет играть. Инструментов в Скретче множество в несколько прокруток экрана.

play drum () for () sec

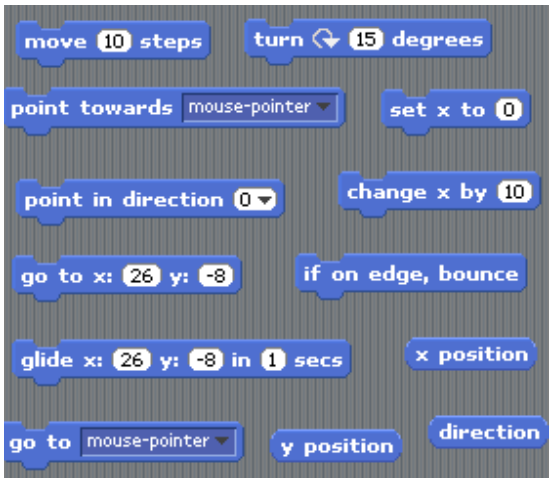
play note () for () sec - играть определенную ноту указанное количество времени (в секундах). Ноты записаны в цифрах, но против каждой цифры стоит ее звучание.

Нота или инструмент могут быть только выбраны из списка существующих нот и инструментов. Просто что попало внутрь звуковых кирпичиков не запихнешь.

К разным объектам в Скретче можно добавлять свои музыкальные партии. Таким образом можно собирать музыкальные дуэты, три, квартеты, квинтеты и т.д.

Синий ящик - Движение

В синем ящике сложены строительные блоки, управляющие перемещением объектов по экрану.



Очевидно, что кошка является прямым потомком черепашки Лого и наследовала многие из команд передвижения. При этом ошибку в команде сделать все сложнее, поскольку количество шагов или градусов интегрированы с самой командой. Поверхность экрана, по которой перемещается Кошка, напоминает по своим свойствам экран, по которому перемещалась Черепашка. Центральная точка экрана соответствует координатам X=0, Y=0. Вправо от нулевой точки X возрастает и это область положительных значений. Влево – область отрицательных значений.

move() - пройти вперед на указанное количество шагов.
Например: **move(-10)** – сделать десять шагов назад

turn() - повернуться - стрелочка указывает по часовой или против часов стрелки выполняется поворот
Например: **turn(45)** повернуться на 45 градусов (стрелочка указывает по или против часовой стрелки)

point in direction() - повернуться в указанном направлении

Например: **point in direction(180)**

point towards - повернуться в направлении другого существа или координат мышки. После команды всегда существует перечень объектов, которые в данный момент присутствуют в системе и на которые можно реагировать. В самом простом случае, когда других объектов нет, предлагается повернуться в сторону, где находится указатель мышки.
Например:
change x by () изменить положение по оси X на указанное число шагов.
Например: **change x by (-10) –** объект смещается влево на 10 шагов

set x to () установить объект положение по оси x
Например: **set x to (50)**

go to x() y() переместиться в точку с указанными координатами
Например: **go to x(70) y(-50)**

glide x() y() in () secs плавно переместиться в точку с указанными координатами за указанное время. На перемещение затратиться время указанное в секундах
Например: **glide x(90) y(90) in (20) secs –** объект медленно (в течение 20 секунд) будет переползать в указанную точку.

go to ()- перейти в точку, где расположен указатель мыши или другая фигура
Например: **go to** (mousepointer)

if on edge, bounce - если попадаешь в край экрана, то отразись от него

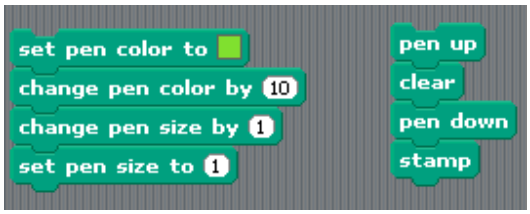
Следующие несколько блоков нельзя использовать в качестве отдельных строительных блоков, но они могут быть вставлены внутрь строительных кирпичиков в качестве входных параметров

x position - возвращает значение по оси X
Например: set y (xposition)

y position - возвращает значение по оси Y
Например: set x (yposition)

direction - возвращает направление
Например: **point in direction** (-**direction**)

Рисование



clear – очистить экран от всех следов, которые на нем оставили объекты

pen down – опустить карандаш. После этой команды за движущимся объектом будет оставаться след.

pen up – поднять карандаш.

set pen color() – выбрать цвет, которым мы собираемся рисовать

change pen color by() – изменить цвет

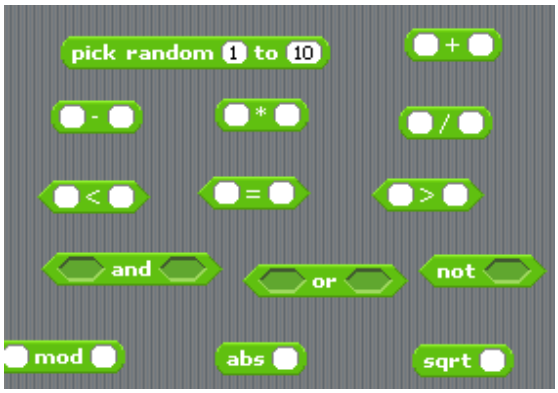
change pen shade() – установить размер тени, который оставляет объект

stamp – отпечатать объект на экране.

Кирпичики рисования естественно соединяются с кирпичиками движения:

Clear; pen down ; move(80); stamp

Числа Numbers



Все это кирпичики, которые можно использовать только внутри строительных блоков. Сами по себе как строительный материал эти кирпичики использовать нельзя. Они возвращают результат

Операции сложения, вычитания, умножения и деления

() + () сложение

() – () вычитание

() * () умножение

() / () деление

pick random () to () случайное число в интервале от и до

Например: **pick random (0) to (10)** возвратит 1 или 3, или 9 -

() < () Сравнение больше, равно, меньше

Например: ((5) * (5)) < ((4) * (10)) – вполне разумное выражение, результат которого должен выдавать отрицание (false)

логические операторы **And, Or, Not**

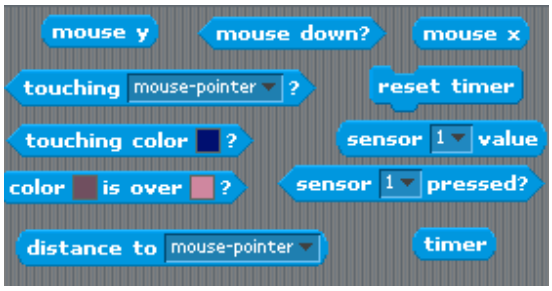
Эти блоки содержат входные окошки, куда вставляются кирпичики сравнений

Например: AND () ()

Как правило, все арифметические и логические блоки используются вместе с блоками управления.

Mod, abs, sqrt

Датчики Sensing



mouse x – возвращает значение указателя мыши по оси x

mouse x - возвращает значение указателя мыши по оси x

Ряд проверочных блоков, которые возвращают «да» или «нет» в зависимости от выполнения условий. Как правило, такие блоки используются внутри конструкции if

touching color () ? – касается ли наш объект указанного цвета

Например: **touching color () ? –**

Color (1) is over (2) ? – цвет 1 накладывается на цвет 2

Mouse down? – нажата ли управляющая клавиша мышки?

Touching mouspointer ? - касается ли наш объект мышки или другого существа

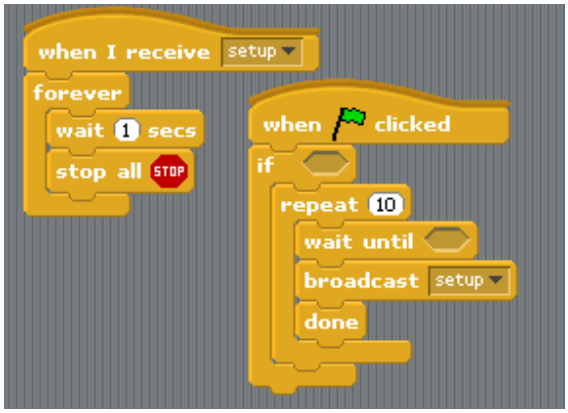
касается ли наш объект цвета()

соприкасается ли цвет() с цветом()

расстояние от другого объекта

включение и выключение таймера

Управление



when (F) clicked – запускает выполнение блока команд в ответ на нажатие зеленого флажка.

when () key pressed - запускает выполнение блока команд в ответ на нажатие выбранной клавиши. Позволяет передать управление на клавиатуру.
Например: when (g) key pressed move(10)

wait() – команда ожидания. Параметр указывает сколько секунд следует ждать.
Например: wait(0.2) – подождать 2 десятых секунды.

Forever – блок команд, заключенных внутрь конструкции будет выполняться постоянно.
Например: forever (move(20) ; rt (45)) объект будет постоянно двигаться и поворачиваться, пока мы не нажмем кнопку Stop

If () – условие, при выполнении которого должны выполняться команды, заключенные внутри конструкции if

repeat() – повторение. Параметр указывает, сколько раз нужно повторить блоки команд, заключенные внутри блока repeat()

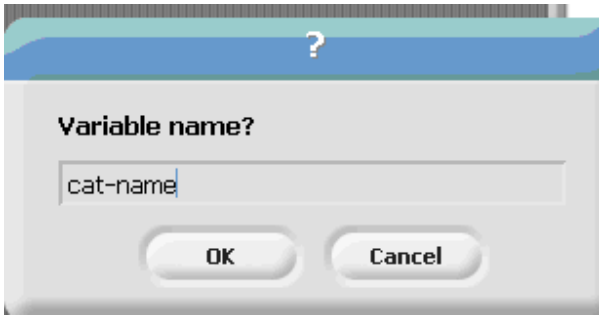
broadcast()

broadcast and wait

Переменные

Создание переменных. Глобальные переменные для всех объектов внутри данного проекта.

Локальная переменная действует только для данного объекта.



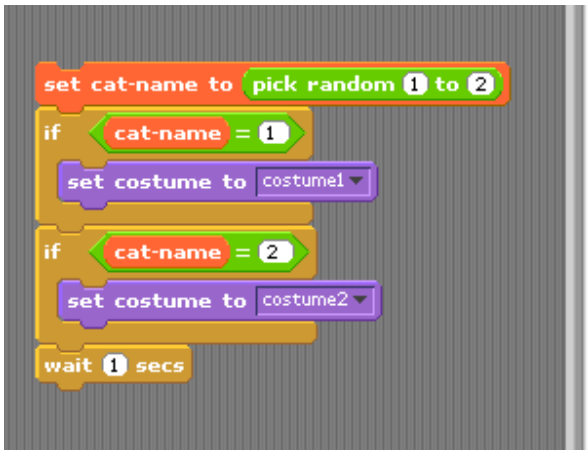
Мы создадим новую переменную с именем cat-name. После того как переменная создана мы можем со значением этой переменной совершать действия. Мы можем присваивать переменной любые числовые значения и менять ее значения на числовые значения (увеличивать или уменьшать)

Воспользуемся новыми возможностями и покажем, как мы можем управлять внешним видом кошки. Мы создадим две версии костюма кошки, и в одной версии кошка будет с голубыми линзами, а в другой с фиолетовыми. Теперь присвоим переменной значение, которое выдаст датчик случайных чисел от одного до двух. Иногда этот датчик будет выдавать число 1, иногда 2.

Дальше мы поставим два блока проверки.

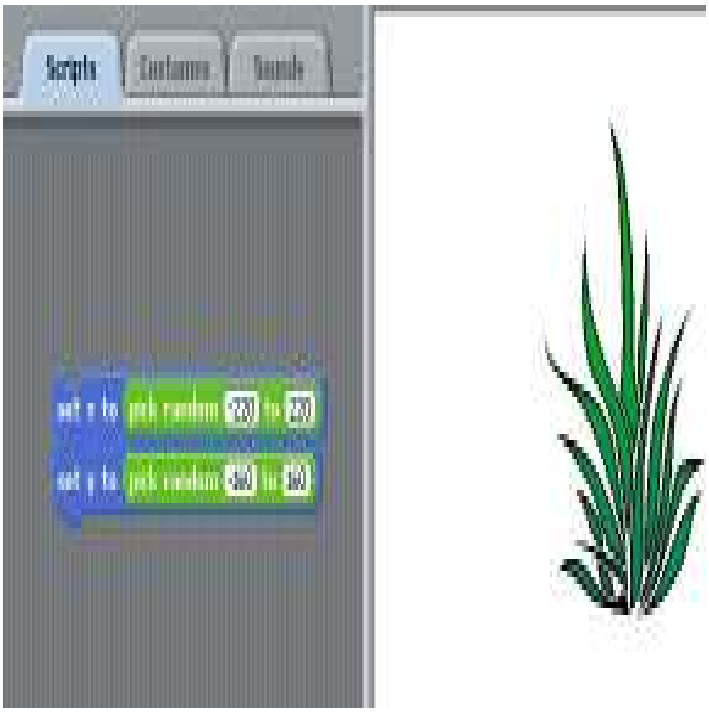
Если значение переменной = 1, то кошка надевает голубые линзы.

Если значение переменной = 2, то кошка надевает фиолетовые линзы.

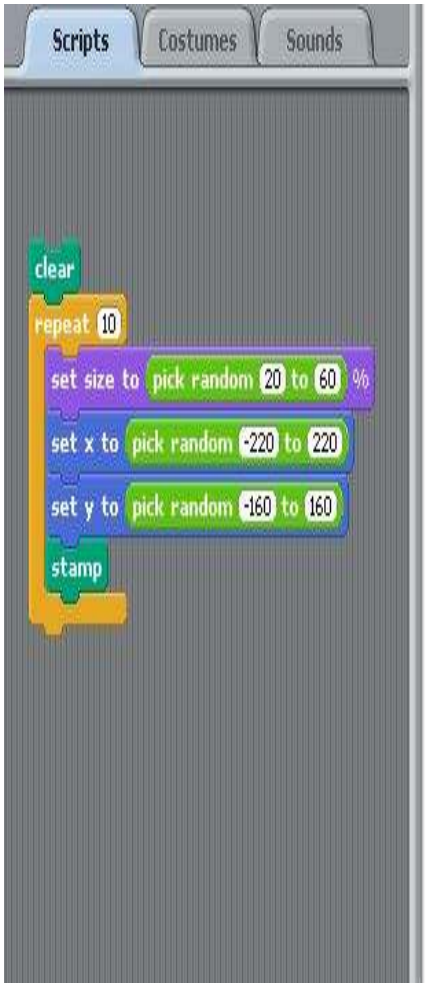

set cat-name to pick random 1 to 2
if cat-name = 1
set costume to costume1
if cat-name = 2
set costume to costume2
wait 1 secs

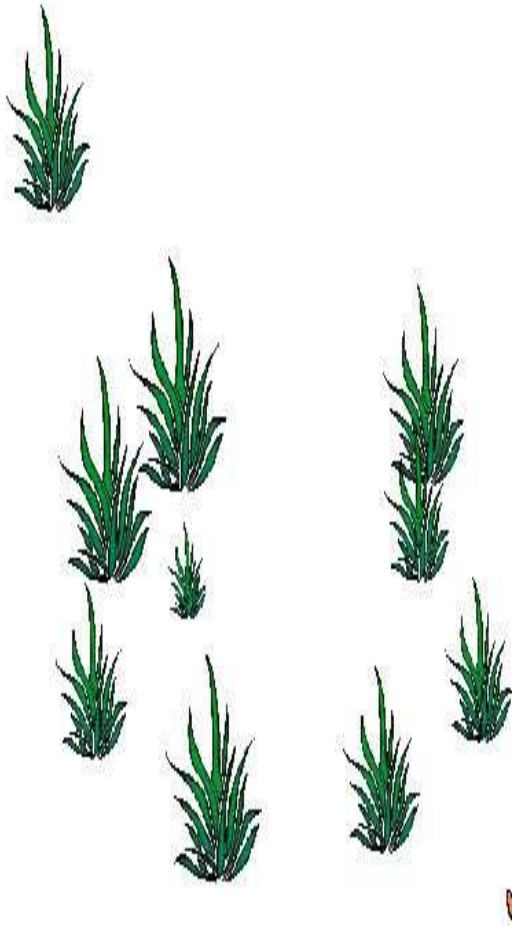
Примеры

Установить объект в случайную точку на экране:

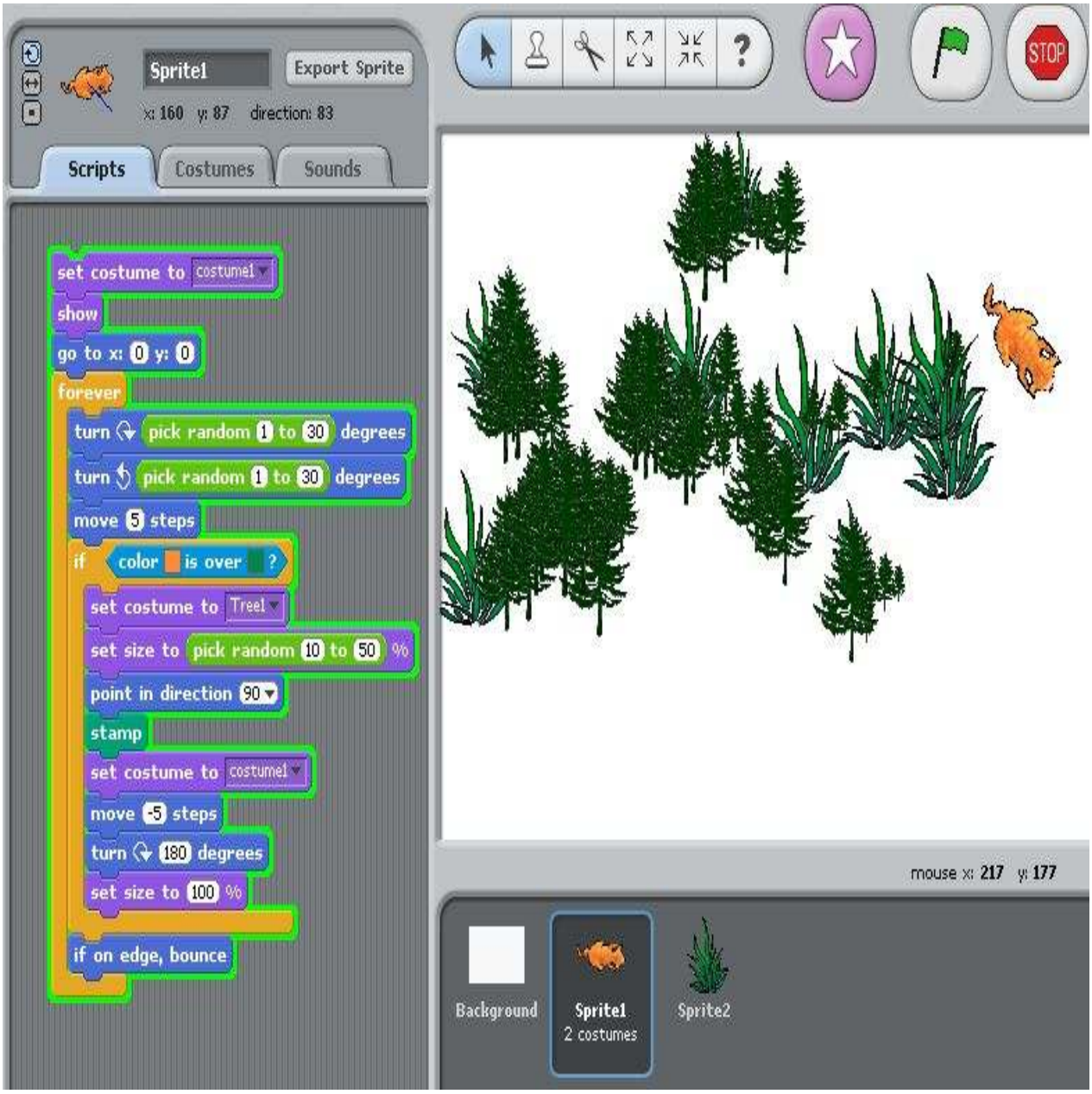


Посадить на экране несколько разных размеров растений в случайно выбранных местах.



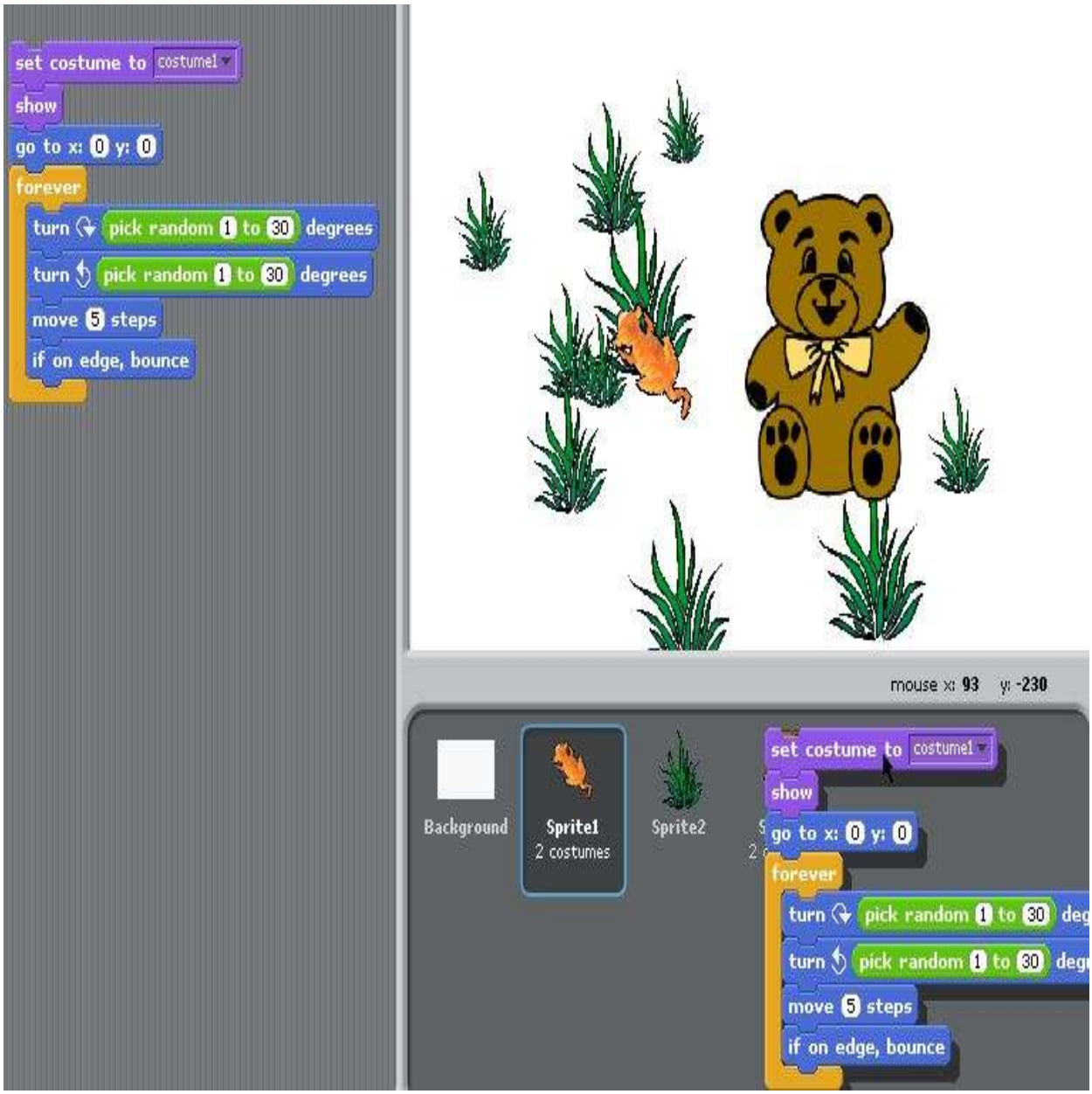


Теперь наша кошка будет бродить среди трав. Если она будет натыкаться не траву, то она будет оставлять в этом месте дерево. Такая кошка - лесничий.

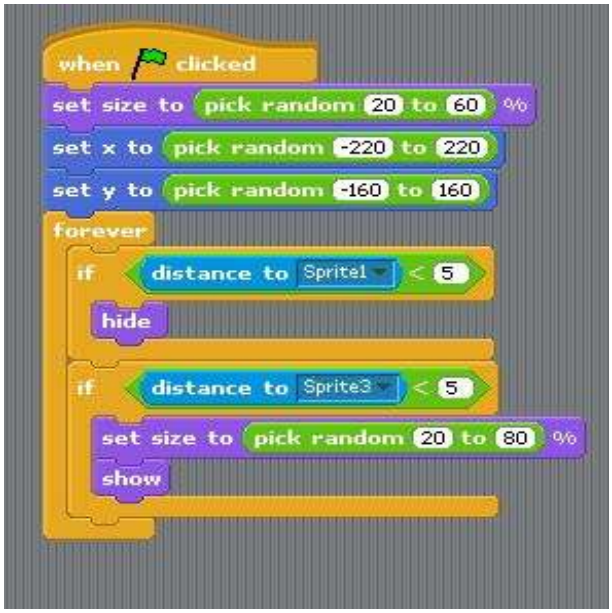


А теперь мы добавим к кошке и травке еще и плюшевого мишку. Здесь очень интересная особенность - мы можем копировать скрипт программы, управляющим поведением одного объекта и добавлять его в скрипт поведения другого объекта.

Например, мы взяли скрипт, управляющий поведением кошки и скопировали его в скрипт, управляющий поведением мишки. Просто взяли кирпичики программы и перетащили их от кошки к мишке.



Мы размножили траву - создали несколько объектов с ее программой и костюмами травы и деревьев. Каждое растение всегда выполняет следующие действия - если приближается кошка, то оно прячется, а если приближается мишка, то оно появляется.



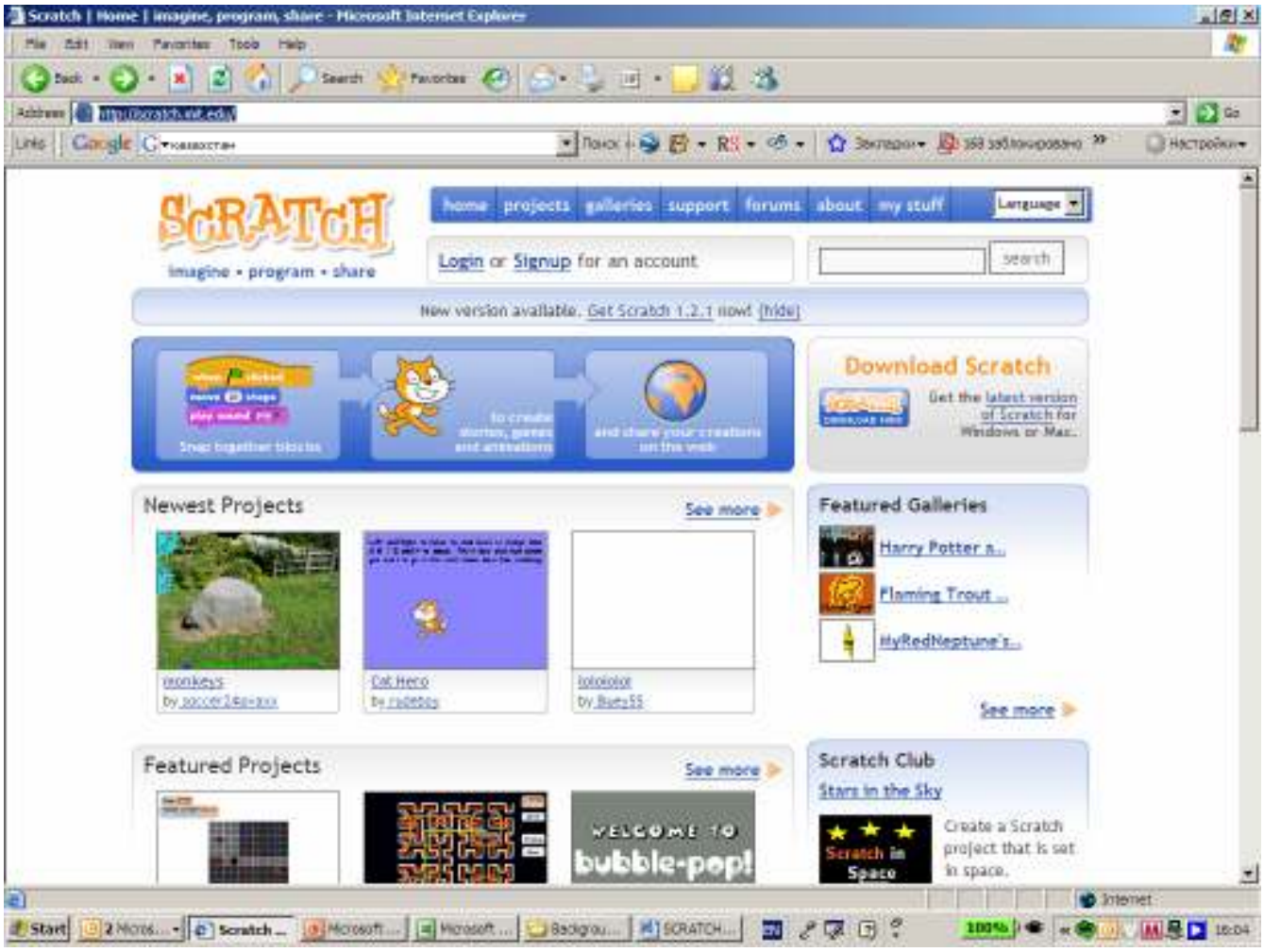
Блок, управляющий поведением каждого растения -

подошел мишка к дереву. Дерево чувствует приближение мишки и становится видимым.

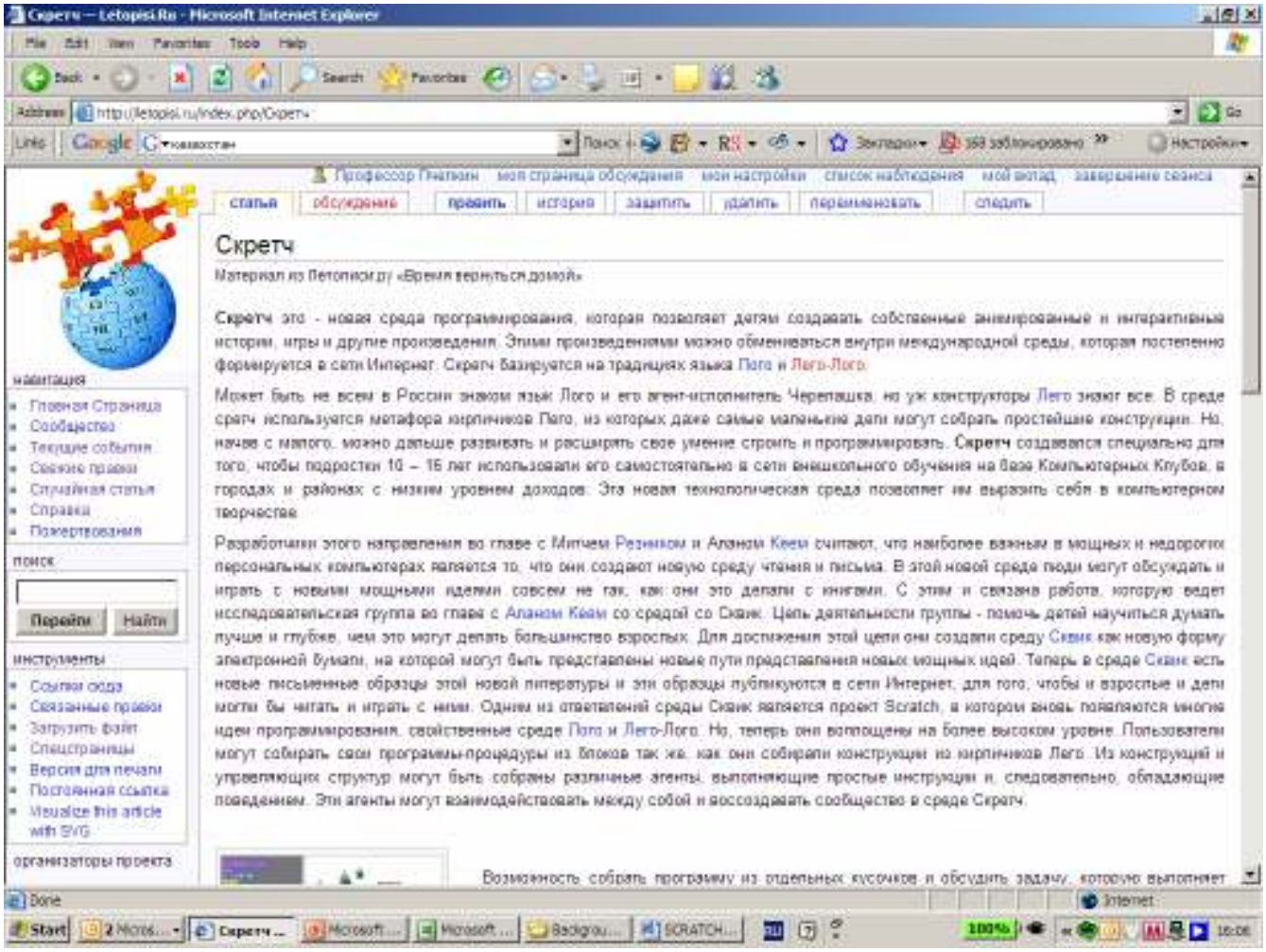
Задания – варианты проектов

Литература:

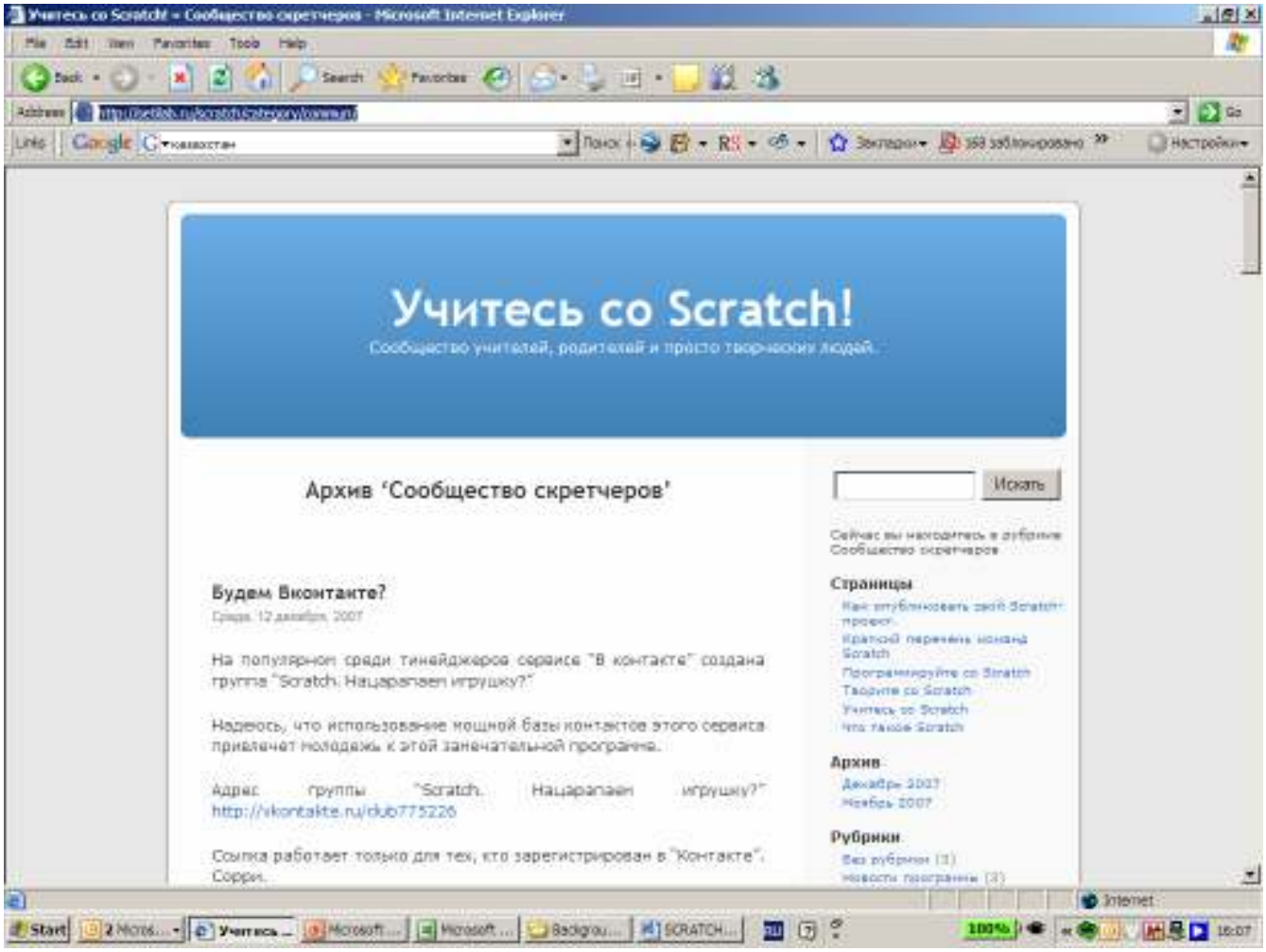
http://scratch.mit.edu – официальный сайт Scratch



http://letopisi.ru/index.php/Скретч - Скретч в Летописи.ру



http://setilab.ru/scratch/category/commun - Учитесь со Scratch!


Учитесь со Scratch! - Сообщество скретчеров - Microsoft Internet Explorer
Учитесь со Scratch!
Сообщество учителей, родителей и просто творческих людей.
Архив 'Сообщество скретчеров'
Искать
Будем Вконтакте?
На популярном среди тинейджеров сервисе "В контакте" создана группа "Scratch. Нацарапаем игрушку?"
Надеюсь, что использование мощной базы контактов этого сервиса привлечет молодежь к этой замечательной программе.
Адрес группы "Scratch. Нацарапаем игрушку?" http://vkontakte.ru/club775226
Ссылка работает только для тех, кто зарегистрирован в "Контакте". Сорри.
Страницы
Архив
Рубрики