



**Управление образования администрации
Старооскольского городского округа
Белгородской области**



**Муниципальное автономное общеобразовательное учреждение
«Образовательный комплекс «Лицей №3»**

Отделение дополнительного образования «Центр детского творчества «Креатив»

**Муниципальный этап регионального конкурса методических
разработок в помощь организаторам технического творчества,
посвященный Году семьи**

**Методическая разработка занятий по теме
«Применение теоремы Пифагора
на примере моделирования движения по окружности в
программе Scratch1.4. и тригонометрических функций для
применения в проектах роботов на базе робототехнических
наборов Lego Mindstorms Education EV-3»**

Номинация «Инженеры-школьяры»

**Автор:
Сабакарь Владимир Викторович,
педагог дополнительного образования**

Старый Оскол 2024

Содержание

Введение.....	3
1. Пояснительная записка.....	3
2. Цель и задачи методической разработки.....	4
3. Практическая часть.....	5
4. Заключение.....	6
5. Список используемой литературы	
6. Приложение.....	6
7. Приложение 1, занятие 1.....	7
8. Теоретическая расчётная часть.....	9
9. Ход занятия	10
10. Приступаем к написанию программы анимации	12
11. QR 1 видео рабочей программы анимации Scratch1.4.....	14
12. Приложение 1, занятие 1.....	15
13. Теоретическая расчётная часть.....	15
14. Основной этап занятия	17
15. Приступаем к написанию программы радара.....	18
16. Действующая программа и обобщающий этап	24
17. QR 2 видео рабочей программы робота сонара.....	24

Аннотация

Настоящая методическая разработка подготовлена педагогом дополнительного образования и представляет собой разработку занятий по теме «Применение теоремы Пифагора и тригонометрических функций»

В разработке занятий прописаны основные моменты: тема, вид занятия, цели, методы обучения, организация самостоятельной работы, прогнозируемый результат.

Разработка может быть полезна учащимся, педагогам дополнительного образования, работающим по направлению робототехника.

Введение

Робототехника это прикладная наука, в которой задействованы, как базовые, так и новейшие знания и технологии. Она впитала в себя предыдущий опыт физики, химии и математики. Все это заложено удачно разработчиками робототехнического набора Lego Mindstorms Education EV3. В основы управления у этого набора уже заложены: динамика движения, расчёт траектории перемещения и математические действия. По этой причине учащиеся объединения по интересам «Моделирование и робототехника» в своих программах должны применять основы, как простых школьных точных наук, а зачастую и заглядывать в область математики и физики старших классов. Сама программа Lego Mindstorms Education EV 3 не позволяет произвести симуляцию на экране монитора компьютера, потому что она предназначена для компиляции программного кода в блок управления роботом. И потому для этой цели необходимо применять такую программу, которая позволяет это произвести и увидеть визуально все действия на доступных школьнику средствах. Задача методической разработки помочь учащимся осваивать применение теоремы Пифагора в среде программирования *Mindstorms Education EV3*, Scratch1.4. и аналогичных программах.

Цель и задачи методической разработки

Цель:

- раскрытие опыта проведения занятия по изучению темы ««Применение теоремы Пифагора на примере моделирования движения по окружности в программе Scratch1.4. и тригонометрических функций для

применения в проектах роботов на базе робототехнических наборов Lego Mindstorms Education EV-3»

».

Задачи:

- обучить практическому применению теоремы Пифагора и тригонометрических функций в инженерных задачах;
- раскрыть принцип подхода к решению математических и геометрических задач применительно к своим проектам.

Пояснительная записка

Методическая разработка адресована учащимся второго года обучения (11-16 лет) и ориентирована на ряд тем дополнительной общеобразовательной программы «Мехатроника и Робототехника», где необходимо в ходе обучения объяснить учащимся необходимость применения переменных и постоянных величин при использовании теоремы Пифагора и тригонометрии. Она позволяет применять учащимися полученных знаний для разработки своих проектов на последующих занятиях в группах и на занятиях по индивидуальному учебному плану.

Актуальность методической разработки заключается в изменении подхода к обучению учащихся, а именно – практическому применению знаний геометрии и тригонометрии в программировании и робототехнике, развитие навыков учащихся, которое позволяют решать логические, эвристические и манипулятивно – конструкторские проблемы. Важно, чтобы, пройдя все этапы обучения, учащийся приобрёл практический подход к преобразованию знаний пониманию, создающий особенный тип мышления – исследовательский и творческий.

Новизна предлагаемых занятий состоит в подборе методов, приемов и средств обучения, нацеленных на формирование общеучебных компетенций; на практическое освоение учащимися теоремы Пифагора и тригонометрических функций, применения этих навыков в своих проектах.

Методическая разработка относится к серии занятий, где учащиеся осваивают программные переменные, образующиеся из изучаемых в школьной программе геометрических преобразований и закономерностей. Она помогает осваивать и практически применять учащимися знания, полученные на уроках геометрии в своих разработках программных и робототехнических проектах.

В процессе первого занятия учащиеся научатся практически применять теорему Пифагора в программе **Scratch**. На этом занятии учащиеся создают рабочую программу на геометрическую динамическую модель движения **Месяца** на экране рабочей программы. В задании на занятии учащимся не нужно строить робота или иное устройство, чтобы получить результат, потому что рабочим элементом является виртуальная программная модель, которую можно легко изменить на другую картинку, а так же изменить и перенастроить программу до необходимого, более сложного уровня.

На втором занятии учащиеся осваивают применение тригонометрических функций в процессе создавая физического устройства - Радар. Это

устройство является уже действующим измерительным механизмом для применения в проекте робота и работающего на принципах измерения расстояния специальным звуковым датчиком-сонаром и вращающим его приводом. Радар будет создан на базе конструктора **Lego EV-3** и программы **Mindstorms Education**.

Практическая часть Тематический план

№ п/п	Тема занятия	Количество часов
1.	Применение теоремы Пифагора в программе Scratch .	2
2	Создание радара на базе конструктора Lego EV-3 и программы Mindstorms Education на основе тригонометрических функций.	2

Занятие 1

Занятие 2

приложение 1

приложение 2

Заключение

В процессе выполнения первого занятия задания учащийся:

- осваивает применение теоремы Пифагора в программной задаче;
- изучает назначение операций в программе, формирует переменные для её работы;
- осваивает принцип зависимости функций от переменных и постоянных;
- осваивает методы пошагового управления графической картинкой;
- получает начальные физические представления о динамики движения картинки;
- получает начальные знания о принципах применения геометрических данных на примере теоремы Пифагора и тригонометрических функций,
- определяет необходимые величины начального и конечного положения визуального графического объекта и его размера,

В процессе выполнения второго задания учащийся:

- осваивает применение тригонометрических функций в программной задаче;
- создаёт действующую модель робота-радара на базе конструктора **Lego EV-3**;
- самостоятельно назначает операции в программе **Mindstorms Education**, формирует переменные для её работы;
- осваивает принцип зависимости тригонометрических функций от переменных и постоянных;
- назначает функции пошагового управления графической картинкой на экране управляющего блока **Lego EV-3**;

- применяет начальные физические представления о динамики движения картинки;
- настраивает необходимые величины начального и конечного положения визуального графического объекта и его размера;
- применяет масштабирование и назначает действия робота в зависимости от дистанции до объекта обнаружения.

Список используемой литературы

Руководство для начинающих:

1. Лоренс Валк «**Большая книга Lego Mindstorms EV3**» Подробное руководство для начинающих по постройке и программированию роботов Lego Mindstorms. Москва 2017.

Интернет ресурсы.

- 1) <https://ru.pdfdrive.com/Большая-книга-lego-mindstorms-ev3-e187860366.html>

А так же используются материалы из открытых источников интернет ресурса официального сайта Lego Mindstorms:

<https://www.lego.com/ru-ru/themes/mindstorms/downloads>

2) <https://www.youtube.com/watch?v=JhbkmPSa2Rk>

https://scratch.mit.edu/scratch_1.4

Приложение 1

Занятие №1

Информационная карта занятия

Название объединения, в котором проводится занятие	Мехатроника и робототехника
Ф.И.О. педагога	Сабакарь Владимир Викторович
Продолжительность занятия	2 x 45 с 10- минутным перерывом
Возраст учащихся	11-16 лет
Тема занятия	Применение теоремы Пифагора в программе Scratch 1.4.
Задачи:	Обучить практическому применению теоремы Пифагора в графических презентациях цй2и инженерных задачах.
Тип занятия	Усвоение нового материала.
Форма проведения занятия	Традиционное учебное занятие с практической работой.
Формы организации занятия	Групповая, индивидуальная.

Методы обучения	Словесный, наглядный, аудиовизуальный, дозированная помощь педагога, инструктивно-продуктивный метод, самостоятельная работа
Средства обучения	презентация, видеоряд.
Оснащение занятия	проектор, ноутбук, программа Scratch 1.4 .
План проведения занятия	<p>1. Организационный этап</p> <p>2. Основной этап</p> <p>1) Подготовка к новому содержанию</p> <p>2) Усвоение новых знаний и способов действий</p> <p>3) Первичная проверка понимания изученного</p> <p>Закрепление новых знаний и их применение</p> <p>4) Контрольный этап</p> <p>5) Обобщение и систематизация знаний</p> <p>3. Заключительный этап</p> <p>1) Итоговый этап</p> <p>2) Рефлексивный этап</p> <p>3) Информационный этап</p>
Ожидаемые результаты	<p>В результате занятия учащийся:</p> <ul style="list-style-type: none"> - осваивает применение теоремы Пифагора в программной задаче; - изучает назначение операций в программе, формирует переменные для её работы; - осваивает принцип зависимости функций от переменных и постоянных; - осваивает методы пошагового управления графической картинкой; - получает начальные физические представления о динамики движения картинки; - получает начальные знания о принципах применения геометрических данных на примере теоремы Пифагора и тригонометрических функций, - определяет необходимые величины начального и конечного положения визуального графического объекта и его размера.
Информационные ресурсы	https://www.youtube.com/watch?v=JhbkmPSa2Rk https://scratch.mit.edu/scratch_1.4

Ход занятия

Организационный этап. Приветствие учащихся.

Здравствуйте, будущие конструкторы и программисты. Раз вы сегодня находитесь здесь, значит, вы заинтересовались данным курсом, конечно, всем вам не терпится поскорее создавать сложные устройства, но в этом стремлении имеются определённые направления, которые необходимо освоить более глубоко погрузившись в школьные предметы. До этого вы успешно осваивали прямолинейные движения объектов и применяли их в своих проектах, а теперь мы начинаем изучать движения по окружности и дополнительно, криволинейные перемещения с использованием теоремы Пифагора.

Основной этап. Подготовка к новому содержанию.

Краткая история теоремы Пифагора. Пифагор (570—490 до н. э.). - древнегреческий философ и ученый, внесший огромный вклад в развитие различных сфер античной истории. Одним из его наиболее значимых достижений считается доказательство закономерности соотношения сторон прямоугольного треугольника.

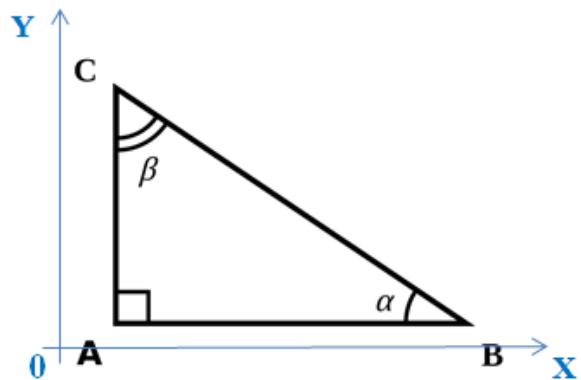


Рис 1.

Теорема Пифагора одна из основополагающих теорем евклидовской геометрии. Она устанавливает соотношение между сторонами прямоугольного треугольника: **Сумма квадратов длин катетов прямоугольного треугольника, равна квадрату длины его гипотенузы. См. Рис 1.**

$$(|CB| * |CB|) = (|AB| * |AB|) + (|AC| * |AC|);$$

Позднее эта теорема была названа в честь ученого и стала одним из столпов евклидовой геометрии. Стороны $|AB|$ и $|AC|$ это **катеты** прямоугольного треугольника, а сторона $|CB|$ называется **его гипотенузой**. Углы « α » (альфа) « β » (бета) острые углы прямоугольного треугольника, а угол при вершине А прямой называется **прямым** и равен **90** градусам. Катет **вдоль оси X** имеет размер = **«Размер X»**, а катет **вдоль оси Y** имеет размер = **«Размер Y»**. Для того чтобы месяц двигался вокруг центра обращения **«0»** нам нужно использовать в программе теорему **Пифагора**.

У нашего прямоугольного треугольника имеются **катеты «Y»; «X»** и **гипотенуза «C»**, которая располагается напротив прямого угла **«90 Градусов»**. Обозначим $X=|AB|$, $Y=|AC|$ и $C=|CB|$; тогда получим выражение $(C * C) = (X * X) + (Y * Y)$; Где $(X * X)$ читается, как **квадрат катета X=|AB|**, значит Мы два раза умножаем величину длины X саму на себя, а $(Y * Y)$ читается как **квадрат катета Y**, соответственно величину длины $Y=|AC|$ умножаем саму на себя два раза.

Усвоение новых знаний и способов действий, ознакомительная часть.

Мы напишем программу-анимацию «Восхождение Месяца на небосводе» на общеизвестной программе **Scratch 1.4**. В ней скрипт «Месяц», по программе восходит вокруг центра, который находится в точке «О», по окружности с нижней части экрана – вверх против часовой стрелки – налево.

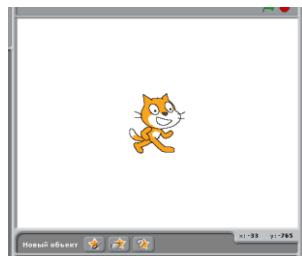
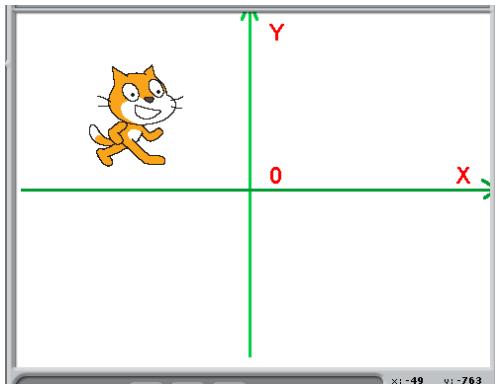


Рис 1.1

Это условно две оси «**X**» и «**Y**», которые являются координатной системой окна программы, называемой **Сценой**.

Рис 1.2



Для начала нужно условиться, что в самой программе мы должны придерживаться графических возможностей и символов, которые управляют в программе **Scratch** экранной областью. См. **Рис 1.1**

Рис 1.2. В центре Сцены находится точка с координатами «**X=0**» и «**Y=0**», положительные значения осей находятся у «**X**» справа от 0 до +240, а «**Y**» вверх от 0 до +180, отрицательные у «**X**» слева от 0 до -240, а «**Y**» вниз от 0 до -180. Это рабочее пространство Сцены (см.рис 1.1), где мы можем визуально отображать результат работы программы **Scratch 1.4**.

Теоритическая, расчётная часть.

Для правильного отображения результатов работы нашей программы в рабочей области принимаем центр координат за центр обращения **Месяца**. Гипотенуза на **Рис. 2** имеет размер радиуса $C = R$ равный радиусу восхождения по которому в программе перемещается **Месяц**. «**Радиус R**» окружности с центром «**0**», является гипотенузой выбранного на схеме прямоугольного треугольника.

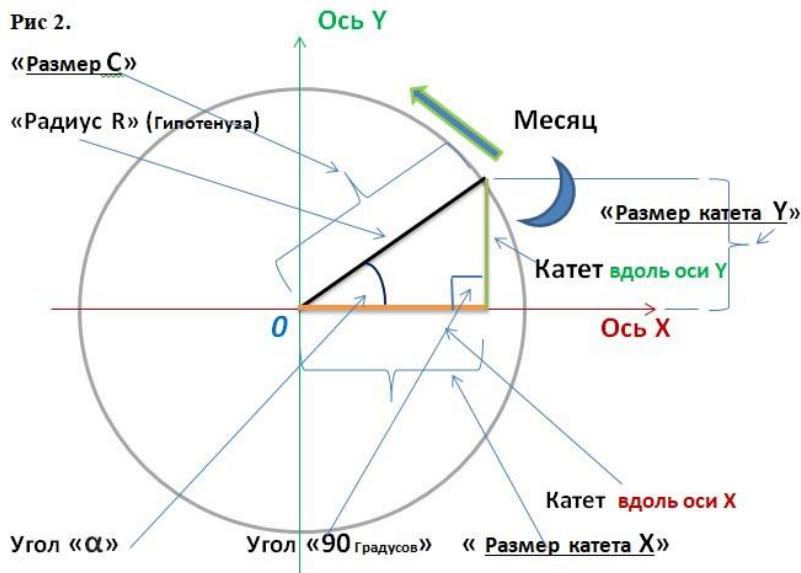


Рис 2.

Для того чтобы наш **Месяц** в программе стал обращаться по кругу с центром **0** нужно, чтобы программа рассчитывала точки для расположения его на окружности с заданным радиусом. Для того, чтобы программа все точки по координатам **X** и **Y**, рассчитывала сама в неё нужно встроить теорему Пифагора и мы тем самым отразим в программе формулу расчёта точек восхода Месяца по необходимым координатам.

Первоначально **См. Рис 3.** мы указываем центр обращения «**0**» и радиус $R = \langle C \rangle$, по которому будет двигаться Месяц. По Закону Пифагора: гипотенуза **C** равна **Корню квадратному из суммы квадратов её катетов** $((X^*X) + (Y^*Y))$, где Корень квадратный это математическое преобразование обратное умножению числа на это же число.

Для экрана монитора не характерны размеры в миллиметрах, потому что в этом случае более правильно считать размеры в пикселях рабочего окна программы. В нашем случае это по $Y=180$ пикселей, а по $X=240$ пикселей. Мы можем выбрать любые координаты и величины катета **X** и **Y**, но в нашей программе мы выбрали **X = 60**, а **Y = -80** с центром вращения **0** в центре рабочего окна программы. В этом случае радиус орбиты Месяца предварительно примем **100** пикселей.

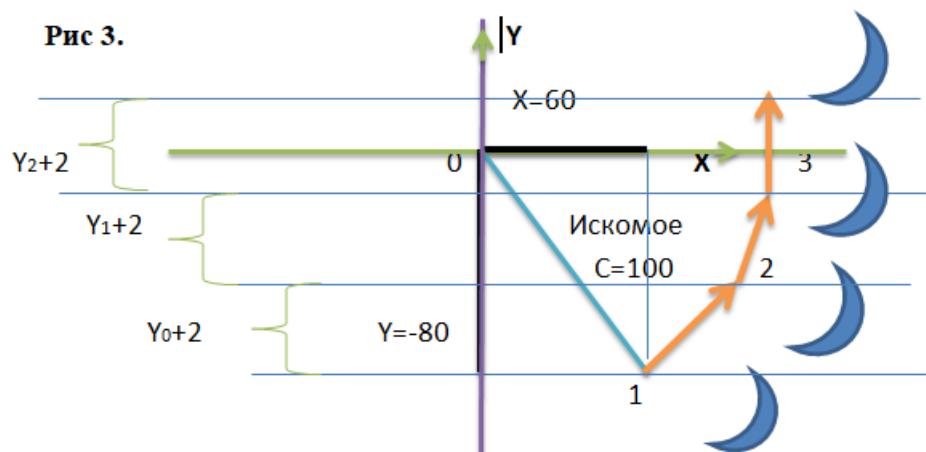
Для примера рассмотрим прямоугольный треугольник со сторонами кратными

$$X = 6, \text{ и } Y = -8. \text{ Тогда гипотенуза: } C = ((X^*X) + (Y^*Y))^{1/2} = \\ = ((6^*6) + (-8^*8))^{1/2} = (36+64)^{1/2} = 100^{1/2} = 10;$$

Мы назначаем начальные положения месяца согласно параметров экранной области **Scratch** в десять раз большее: **X0 = 60**, а **Y0 = -80**. Таким образом $C = ((60^*60) + (80^*80))^{1/2} = (3600+6400)^{1/2} = 10000^{1/2} = 100$;

эти параметры очень хорошо впишутся в размер рабочего окна программы для сцены с восхождением месяца в программе **Scratch 1.4.**, где от начальной точки $Y0 = -80$ и мы каждый раз к новому его положению прибавляем «2» пикселя до значения $Y = 100$. **См. Рис 3.**

Рис 3.

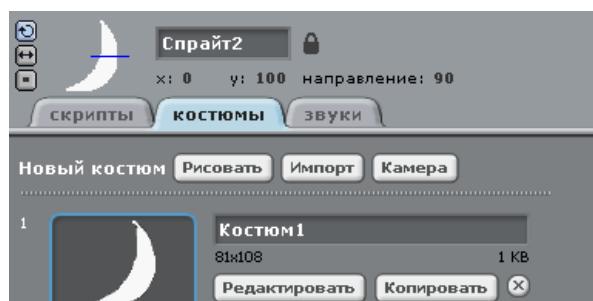


Таким образом по орбите **Радиус С**, который равен **C=100**, вокруг центра «**0**», вокруг которого будет обращаться наш **Месяц – Спрайт2** по точкам **1;2;3;...** При этом к начальному значению **Y0** будет каждый шаг программы прибавляться **2** пикселя.

Первичная проверка понимания изученного

Приступаем к написанию программы. Радиуса вращения «**C**» в процессе программы будет **100** пикселей, но в начале программы мы должны назначить переменную «**C**» и присвоить ей значение «**0**».

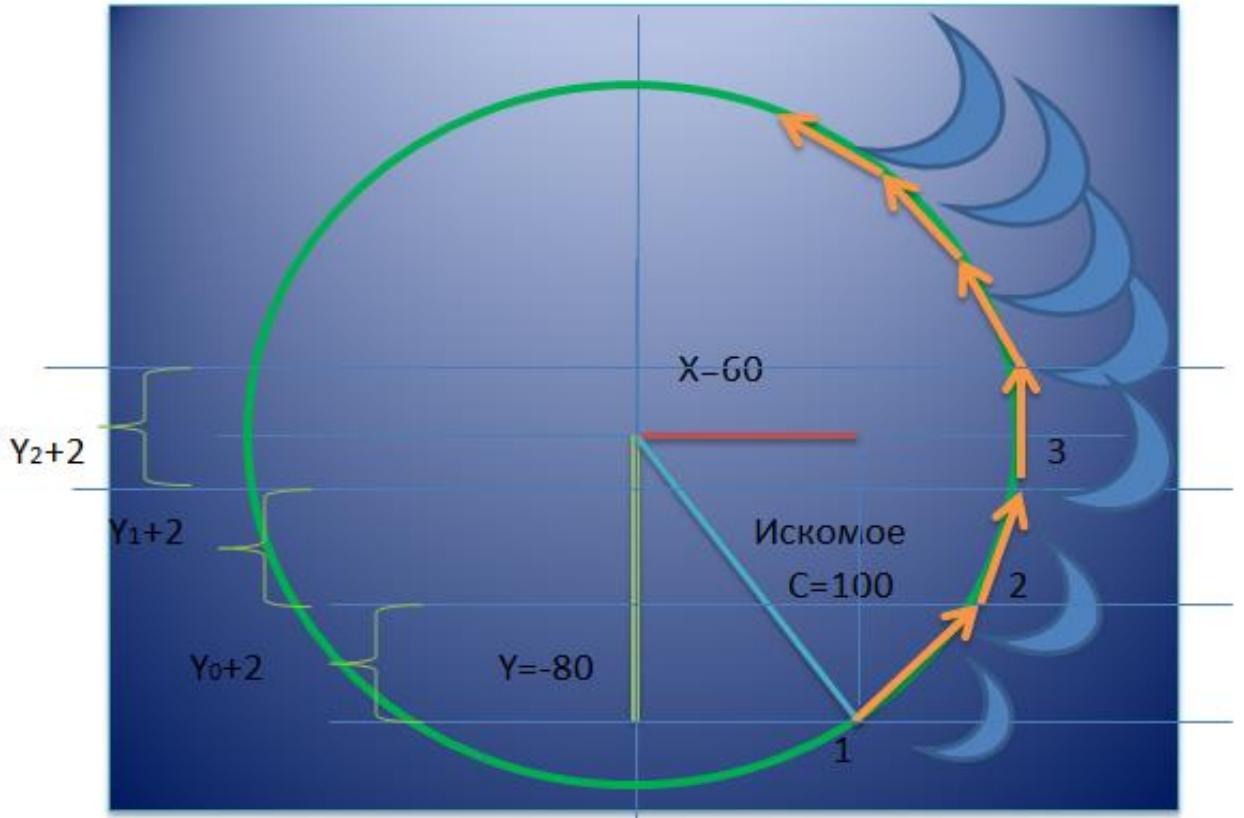
Блок нашей части программы **Спрайт2 – Месяц** желательно установим дальним слоем позади всех фигурок-спрайтов, чтобы «**Костюм 1**» = **Месяц** перемещался за всеми объектами и казался дальше них. «**Месяц**» рисуем сами



Назначаем в программе переменную (**0C**) = «**C**» равной радиусу окружности орбиты Месяца. Зная радиус окружности, и изменяя одну из координат на постоянную величину, как в нашем варианте координату **Y**, мы сможем определять через теорему Пифагора координату **X**. Запустим программу **Scratch**, создадим «**сцену**»-небосвод и «**костюм1**» - **месяц** и напишем следующую программу.

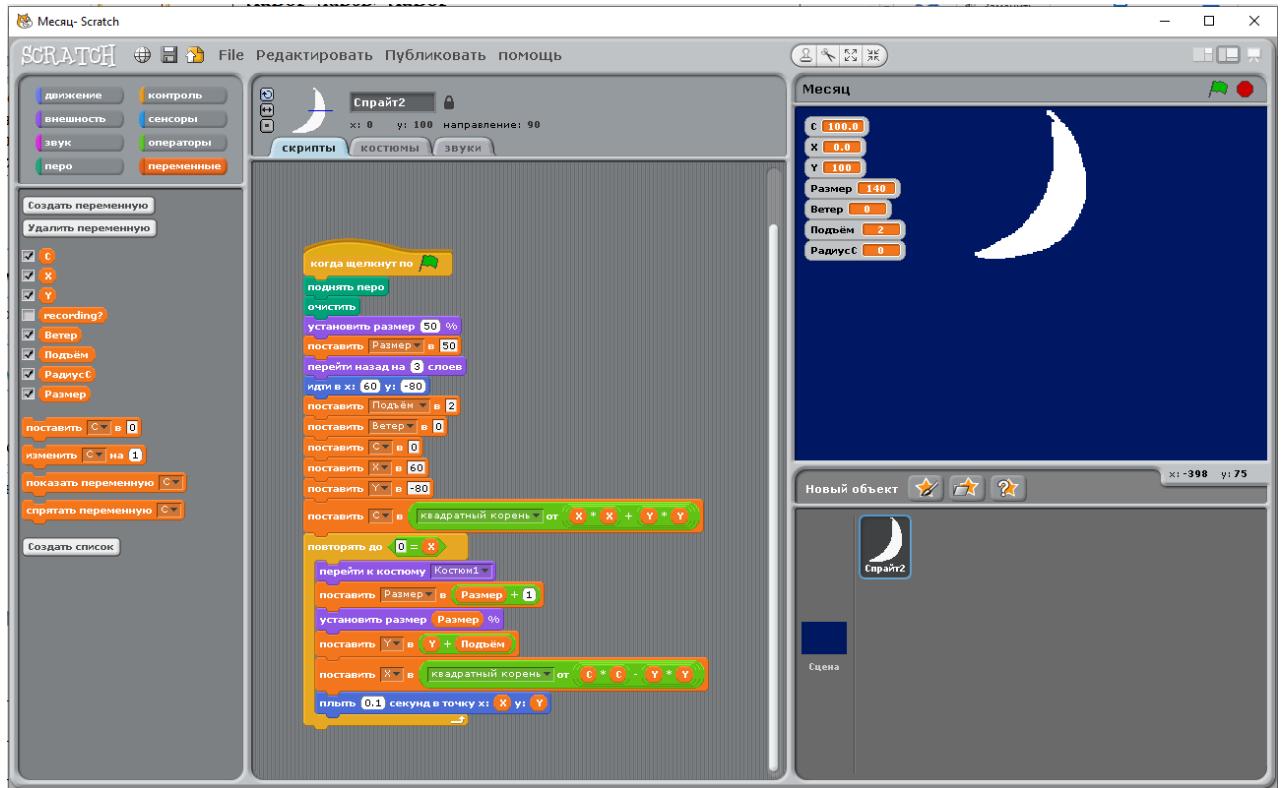
	Старт (начало)
	Если Перо опустить то будет рисоваться линия перемещения
	Эта команда устанавливает размер Месяца на экране.

	Назначаем этот размер
	Месяц располагаем на 3 слое
	Начальная точка расположения Месяца
	Величина или шаг перемещения Месяца.
	Начальные задания для точек на осях X иY .
	Назначение переменной «C»
	Расчёт переменной «C»
	Будем поднимать Месяц пока координата «X» не примет «0» значение
	Это наш Месяц.
*	
	Увеличим размер Месяца на +1%
	Теперь размер Месяца (M)%+1%=(M)%
	$Y_0+2=Y_1$: $Y_0+2=Y_1$: $Y_1+2=Y_2$ $Y_n+2=Y_{n+1}$:
	Расчёт катера «Х» согласно теореме Пифагора
	Новое значение «Х»
	Перемещает Месяц в новую координату.
	Вернуться в начало цикла.



Во время работы программы Месяц начнёт подниматься от начального положения в точке №1 до вертикального значения где $X=0$, каждый раз на 2 пикселя экранной области **Scratch**, а его размер будет увеличиваться с каждым шагом на 1% и это создаст эффект космического движения.

Это и есть элемент **Scratch** программы где применена теорема Пифагора для визуализации движения Месяца по расчётной окружности - орбите. В этом блоке находится цикл расчёта перемещения Месяца.



Обобщение и систематизация знаний

Вы написали программу, основы которой можно успешно применять в презентациях, проектах, для исследования движения по окружности визуальных графических объектов. В начале занятия мы упоминали о криволинейном перемещении графической картинки. Предлагается вам самостоятельно подумать, как можно отклонить движение Месяца от движения по окружности и получить криволинейное движение (ответы учащихся). Спасибо за ответы. Предложу свой вариант ответа: нужно пошагово изменять размер радиуса на каждой итерации (шаге программы).

Заключительный этап. Итоговый этап.

Программа написана, можно начинать проверку и наладку программы в среде Scratch.

Для сравнения с заданием наведите смартфон на QR 1 - для просмотра фрагмента видео.



QR 1

Рефлексивный этап.

Предлагаю учащимся определиться с их пониманием пройдённой темы и выбрать смайлики:

	<i>Я всё понял и у меня всё получилось!</i>
	<i>Мне многое было понятно!</i>
	<i>Мне нужна помощь! Я многое не понял!</i>

Приложение 2

Занятие 2

Информационная карта занятия

Название объединения, в котором проводится занятие	Мехатроника и робототехника
Ф.И.О. педагога	Сабакарь Владимир Викторович
Продолжительность занятия	2 x 45 с 10- минутным перерывом
Возраст учащихся	10-16 лет
Тема занятия	Создание радара на базе конструктора Lego EV-3 и программы Mindstorms Education на основе тригонометрических функций.
Задачи:	Обучить практическому применению тригонометрических функций в инженерных задачах.
Тип занятия	Усвоение нового материала
Форма проведения занятия	Традиционное учебное занятие с практической работой
Формы организации занятия	Групповая
Методы обучения	словесный, наглядный, аудиовизуальный, дозированная помощь педагога,

	инструктивно-продуктивный метод, самостоятельная работа
Средства обучения	презентация, мультфильм, модель робота, конструктор
Оснащение занятия	проектор, ноутбук, комплекты конструкторов LEGO Mindstorms Education. EV3.
План проведения занятия	<p>1. Организационный этап</p> <p>2. Основной этап</p> <p>1) Подготовка к новому содержанию</p> <p>2) Усвоение новых знаний и способов действий</p> <p>3) Первичная проверка понимания изученного Закрепление новых знаний и их применение</p> <p>4) Контрольный этап</p> <p>5) Обобщение и систематизация знаний</p> <p>3. Заключительный этап</p> <p>1) Итоговый этап</p> <p>2) Рефлексивный этап</p> <p>3) Информационный этап</p>
Ожидаемые результаты	<p>В процессе выполнения задания учащийся:</p> <ul style="list-style-type: none"> - осваивает применение тригонометрических функций в программной задаче; - в программе изучает назначение операций с данными и работу их; - осваивает принцип зависимости функций от переменных и постоянных; - осваивает методы управления графической информацией отображаемой на экране управляющего блока; - получает начальные физические представления о динамики обработки, расчёта и отображения информацией; - получает начальные знания о принципах обработки и применения в программах тригонометрических функций, - определяет необходимые величины начального и конечного положения визуального графического объекта,
Информационные ресурсы	https://www.lego.com/ru-ru/themes/mindstorms/downloads https://ru.pdfdrive.com/Большая-книга-lego-mindstorms-ev3-e187860366.html

Ход занятия.

Организационный этап. Приветствие учащихся.

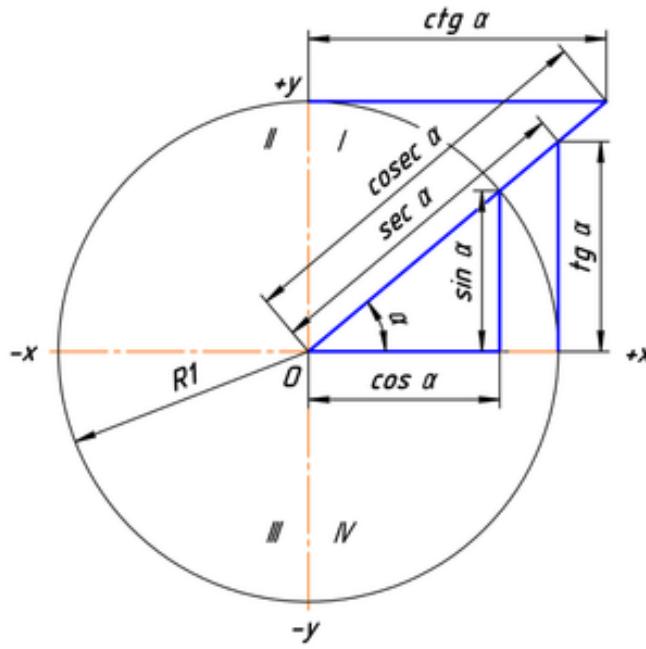
Здравствуйте, будущие робототехники и ИТ специалисты. Планы у вас поскорее создавать сложные устройства и проекты, но мы с вами сегодняшнее занятие вновь посвятим школьной теме, и это будут тригонометрические функции и уравнения. А значит, мы снова окунаемся в часть геометрии, которую нам ни как не обойти на сегодняшнем занятии. До этого вы успешно осваивали прямолинейные движения объектов и применяли их в своих проектах. На предыдущем занятии мы использовали теорему Пифагора и изучали, как её применять в программе для перемещения графических объектов на экране программы. А сегодня мы применим уже и тригонометрические функции в разработке программы для радара созданного на базе робототехнического набора Lego EV-3.

Основной этап. Подготовка к новому содержанию.

Краткая история. Тригонометрия – слово греческое и в буквальном переводе означает измерение треугольников (trigwnon - треугольник, a metrew- измеряю). Возникновение тригонометрии связано с землемерением, астрономией и строительным делом. Впервые способы решения треугольников, основанные на зависимостях между сторонами и углами треугольника, были найдены древнегреческими астрономами **Гиппархом** (2 в. до н. э.) и **Клавдием Птолемеем** (2 в. н. э.). Позднее зависимости между отношениями сторон треугольника и его углами начали называть тригонометрическими функциями.

Усвоение новых знаний и способов действий. Мы должны внести в программирование наших проектов новое понятие: тригонометрические функции: *Функцией называют такую зависимость переменной « f » от переменной « x », у которой каждому значению « x » соответствует единственное значение « f ».* $f(\alpha) = \cos(\alpha) = \sin(90 - \alpha)$

Обычно тригонометрические функции определяются геометрически. В **декартовой системе координат** на плоскости построим **окружность единичного радиуса «R1»** с центром в начале координат «0». Угол образованный положением точки расположенной на окружности и осью абсцисс-«x», при этом направление поворота против часовой стрелки считаем положительным, а по часовой стрелки – отрицательным (см. Рис 2). **Абсциссу** точки обозначим **$\cos(\alpha)$** , а **ординату** – **$\sin(\alpha)$** .

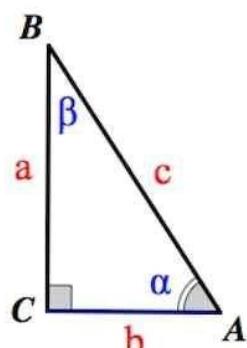


Вновь мы рассматриваем прямоугольный треугольник с двумя катетами и гипотенузой. Катеты тут отображены, как функции $\cos(\alpha)$ и $\sin(\alpha)$, от угла α при вершине прямоугольного треугольника. На прошлом занятии мы расчитывали неизвестный катет по радиусу-гипотенузе и известному катету по теореме Пифагора, а теперь мы можем расчитывать те же катеты по углу между гипотенузой и катетом расположенным на оси абсцисс-« x ».

Если условится, что гипотенуза равна «1», то можно представить эти функции, как теорему Пифагора:

$$\sin^2(\alpha) + \cos^2(\alpha) = 1$$

Рис. 2.



$\sin\alpha = \frac{a}{c}$	$\cos\alpha = \frac{b}{c}$
$\tg\alpha = \frac{a}{b}$	$\ctg\alpha = \frac{b}{a}$

Кроме того, 1) $\sin\alpha = \cos\beta$ 2) $\sin^2\alpha + \cos^2\alpha = 1$
 $\cos\alpha = \sin\beta$

3) $\tg\alpha = \frac{\sin\alpha}{\cos\alpha}$, $\ctg\alpha = \frac{\cos\alpha}{\sin\alpha}$, $\tg\alpha \cdot \ctg\alpha = 1$

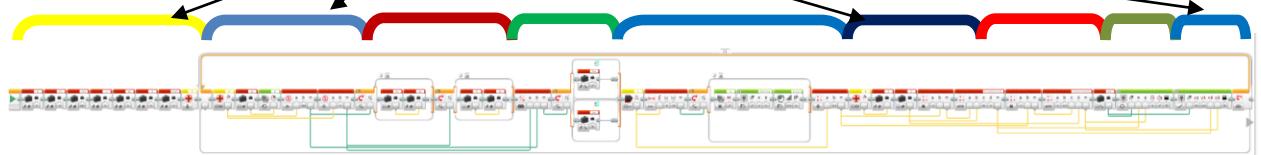
В этой таблице находятся и другие тригонометрические функции, которые для проектов индивидуального учебного плана можно изучить дополнительно.

Начнём применять тригонометрические функции в нашем проекте

Первичная проверка понимания изученного

Приступаем к написанию программы.

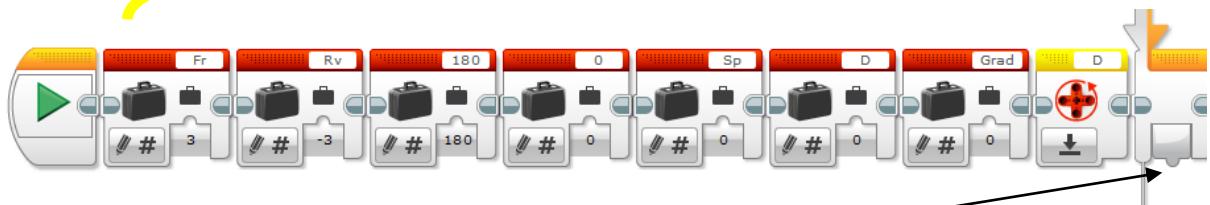
В таком относительно уменьшенном виде представлена вся структура программы в среде программирования LEGO MINDSTORMS Education EV3. Дополнительно все сегменты помечены цветными скобками. Это необходимо, чтобы описать функциональность отдельных участков программы.



Теперь мы в увеличенном виде рассмотрим описание сегментов программы. Рис 1. это блок назначения переменных для программы, так называемый «**setting-variables**» - установка переменных, или «**Stand by**» - подготовиться (быть готовым), которое сокращённо часто применяется как «**Stb**».

Знак цветная скобка обозначает описываемый сегмент программы и является местомстыковки графических элементов программы.

Рис 1.



где: Сброс датчика мотора **D**

Fr – мощность движения в прямом направлении для мотора **D**,

Rv – мощность движения в обратном направлении для мотора **D**,

180 – значение максимального поворота радара в градусах,

Sp – итоговая мощность мотора вращения D **Spid** (скорость) ,

D – переменная для датчика угла поворота мотора D.

Фрагмент программы для управления углом поворота мотора **D**

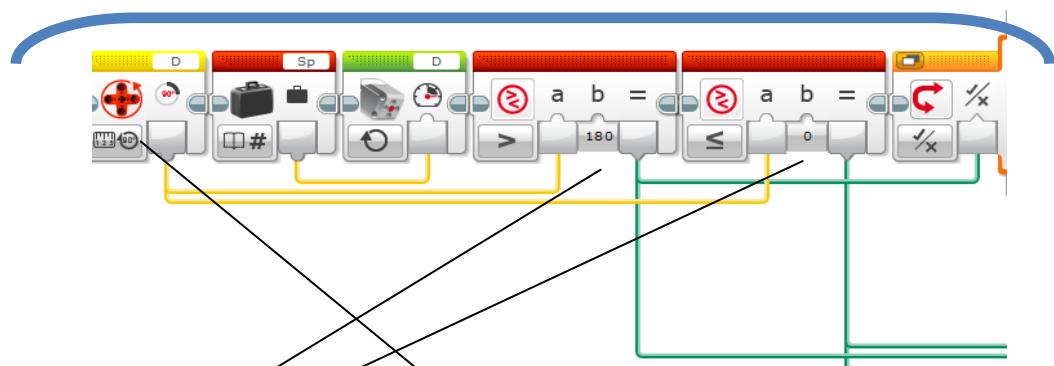


Рис 2.

В этой части программы направление вращения мотора **D** установлены границы положения его углового датчика. Если угловой датчик ещё не

достиг угла 180 градусов, то он вращается в прямом направлении, а если достиг 180 градусов и более, то его скорость изменяется на противоположную. Эта скорость действует до тех пор пока его датчик не достигнет значения «0» градусов и менее. При этом значение скорости **Fr** и **Rv** соответствующее прописываются в **фрагменте** для переменной **Sp**.

См Рис.3

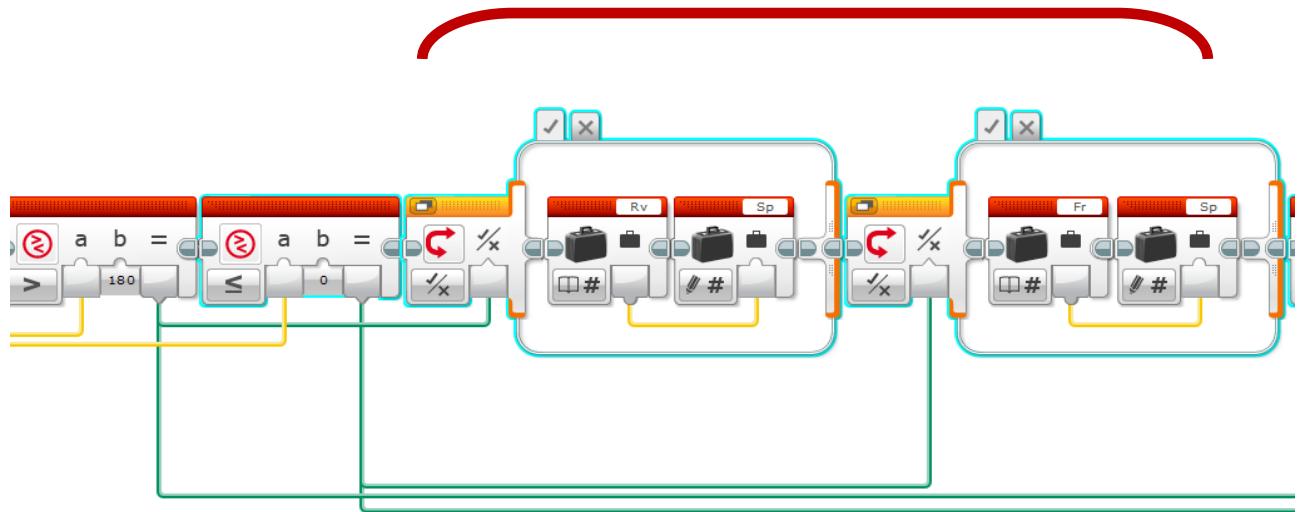


Рис 3.

В этом **фрагменте** формируется логическая переменная, которая стирает информацию с экрана, когда сонар проходит крайние точки « ≤ 0 » градусов и « > 180 » градусов. Рис 4.

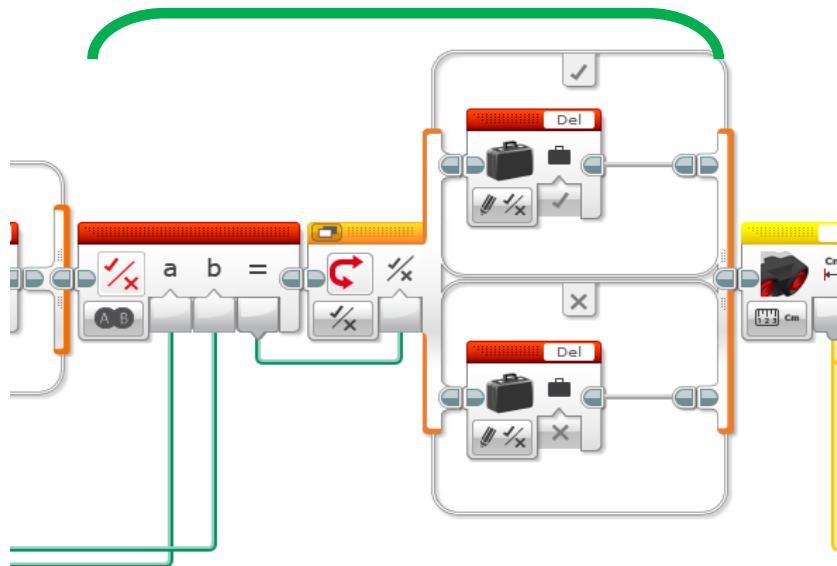


Рис 4.

Нижний блок программы отвечает за настройку тревожной зоны для сонара и формирует длину луча радара на экране. Тревожная зона расположена в интервале **от 5 см до 30 см**. Блок математического деления измеренной дистанции сонара в 2.7 раза необходим для ограничения на экране максимальной дальности его работы. Это коэффициент масштабирования. В нашем случае датчик расстояния сонар работает на максимальной дистанции 255 см, а для размещения этой дистанции на экране управляющего блока эту величину нужно уменьшить приблизительно в 2.7 раза. Коэффициент масштабирования можно изменять программно или просто установить любой необходимый. Любой предмет, который сонар определит в интервале от 5 см до 30 см вызовет остановку привода сонара в секторе нахождения этого предмета, последует звуковой сигнал ***Sonar*** и изменится цвет индикатора на панели блока управления на красный. См **Рис. 5.**

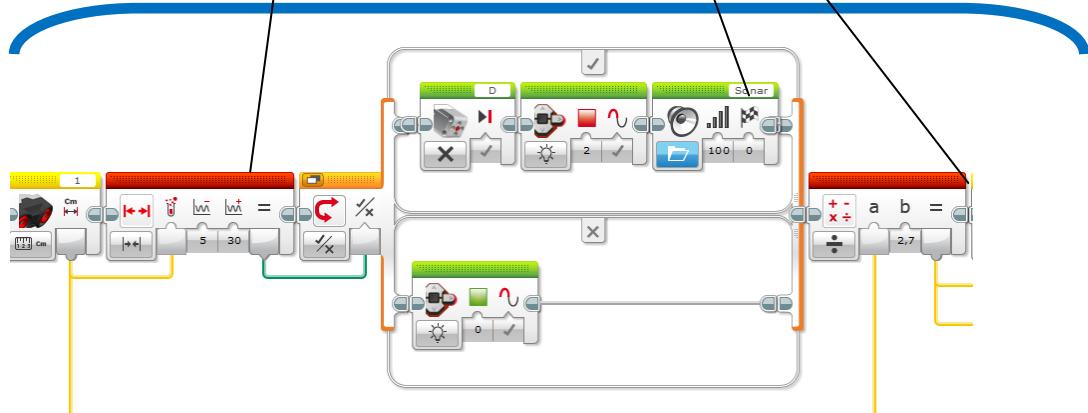


Рис 5.

Следующий **участок** см **Рис 6.** программы отвечает за построение на экране блока управления EV-3 исследуемой зоны. Она выглядит, как половинка окружности в виде сектора с углом 180 градусов. Для эргономического отображения информации на экране радара необходимо начало его координат на экране блока управления перенести с координаты «**0-0**», в координаты **X=90** и **Y=110**. Тем самым мы расположим начальную точку отображения графической информации сонара в точке координат по центру и в нижней части экрана. См. **Рис 7.**

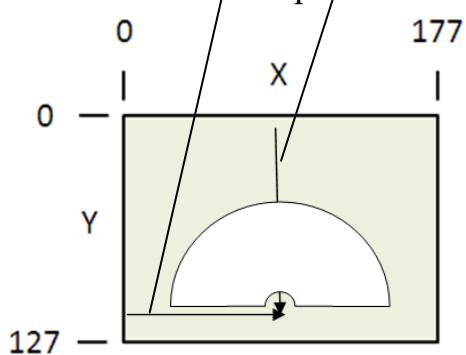
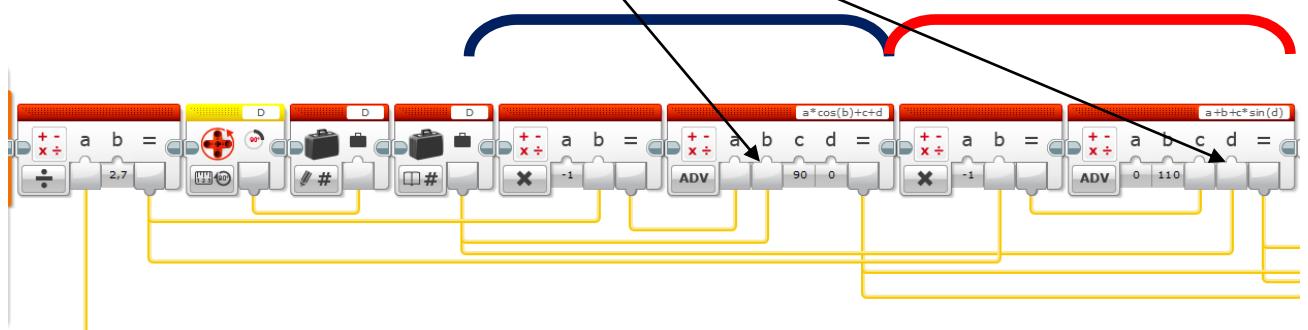


Рис 6.

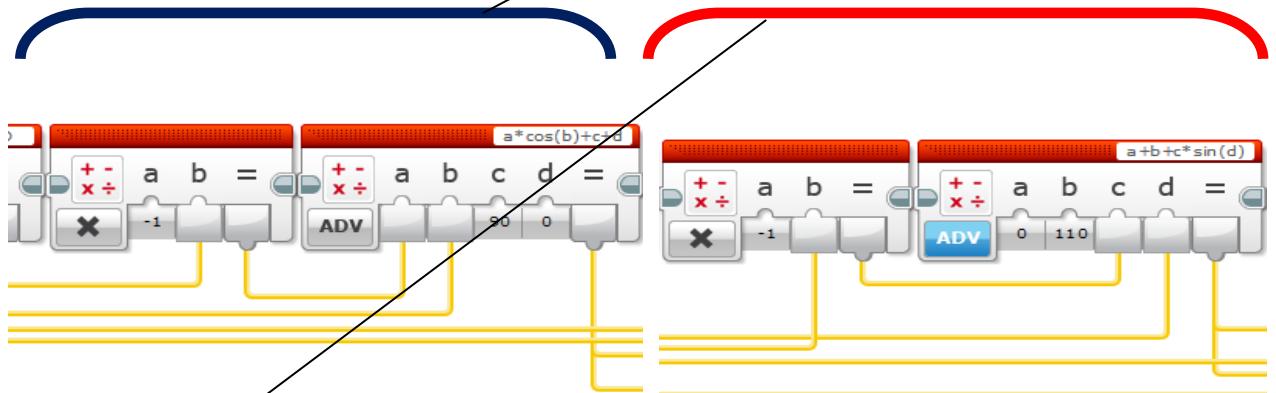
После уменьшения данных с сонара в 2.7 раза (масштабирования) для естественного отображения на экране, так как положительное направление осей **X** и **Y** на экране относительно кнопок расположено сверху вниз, то необходимо инвертировать данные датчика умножив их по двум координатам на «**-1**». Угловая **величина** на которую повёрнут сонар направляется в математические формулы, а величина расстояния до объекта (гипотенузы) задаётся значением **расстояния от сонара до объекта**. **Рис 7.**



За центр отображения дуги радара отвечают два математических блока **Рис. 8** и **9**. Они же расчитывают и тригонометрические построения в программе на экране. Длину гипотенузы, или радиуса дуги устанавливают показания сонара конструктора **Lego EV-3** как расстояние до объекта, а точку её расположения непосредственно на дуге программа получает из тригонометрического расчёта. В этом расчёте **сегмент** сдвигает по координате «**X**» центр дуги на 90 пикселей экрана, на **c=X=90** и вертикально делит экран на две равные части . См. **Рис 8.**,

Рис .

Рис 9.

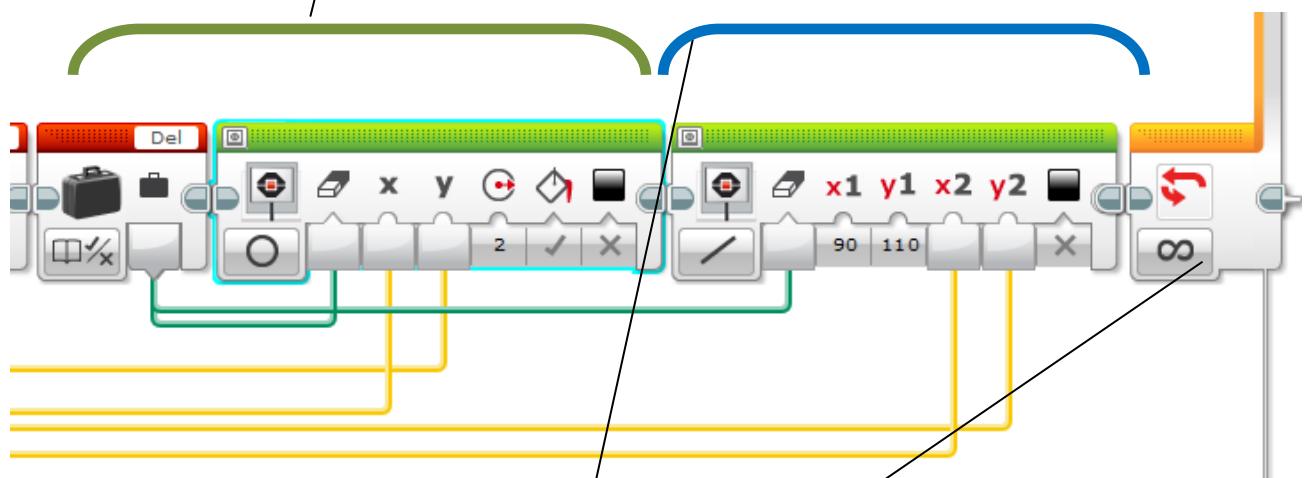


а **сегмент** отвечает за определение координаты «**Y**» и устанавливает центр сектора отображения исследуемой зоны в нижней части экрана и отвечает за нижний уровень отображаемой радаром зоны **b=Y=110**, равный 110 пикселям экрана от верхней кромки вниз. См. **Рис 9.** Этими настройками мы

располагаем центр сектора окружности в необходимом месте для экранной области радара. В нижней части экрана по оси **Y** от 110 пикселей до 127 пикселей зарезервировано место для отображения цифровой или текстовой информации, которое вы можете отобразить по желанию сами. Опускать сцентр сектора наблюдения ниже 110 пикселей визуально менее наглядно.

Программный блок отображения работы радара на экране осуществляется блоками на **Рис 10**. Само графическое построение на экране блока управления Lego EV-3 осуществляется двумя программными блоками **Рис 10**. Логическая переменная **«Del»**, как было описано выше, производит полное стирание всей графической информации после достижения крайних точек сектора, которые находятся в интервале менее «0» угловых градусов и более-равно 180 градусов. То есть эта переменная освобождает экран от старой информации для отображения новой. После логической переменной **«Del»** первый экранный **сегмент** предназначен для назначения размеров точек которыми на экране отображается фигура в выбранном нами секторе, с расчётыми координатами **X** и **Y**. Толщина линии задана диаметром **2** пикселя.

Рис 10.



Следующий экранный **сегмент** строит отрезок по двум точкам непосредственно на экране. Начиная с начальной координатой в нулевой точке (**x1;Y1**) расположенной в координатах **X=90** и **Y=110**, и второй координаты (**x2;Y2**) расчётной, которую рассчитывает программа при помощи тригонометрических функций. Эту координату мы получили в предыдущем блоке **Рис. 8-9**. Вся программа кроме установочных переменных **Блок «Stb»** заключается в **цикл**. После чего она готова к загрузке в блок Lego EV-3 и проводятся испытания радара.

Обобщение и систематизация знаний

Вы написали программу, основы которой можно успешно применять для сканирования предметов, исследования окружающего робота пространства, создания охранных систем и других проектов. В процессе занятия мы упоминали о криволинейном масштабировании дальности настройки радара. Предлагается вам предложить усовершенствовать программу, чтобы можно было изменять масштабирование в зависимости от необходимой дистанции (ответы учащихся). Спасибо за ответы. Предложу свой вариант ответа: например нужно пошагово изменять коэффициент масштабирования в зависимости от скорости движения робота, чем больше скорость тем большее значение масштабирования необходимо применять. Это даст возможность в обратном порядке уменьшать скорость если расстояния до объектов сокращаются.

Заключительный этап. Итоговый этап.

Программа написана, можно начинать проверку устройства - Радар на базе конструктора **Lego EV-3** и наладку программы **Mindstorms Education** работающую на основе тригонометрических функций.

QR 2



Наведите смартфон на **QR 2** - для просмотра фрагмента видео настроенного радара.

Рефлексивный этап.

Предлагаю учащимся определиться с их пониманием пройдённой темы и выбрать смайлики:

	<i>Я всё понял и у меня всё получилось!</i>
	<i>Мне многое было понятно!</i>
	<i>Мне нужна помощь! Я многое не понял!</i>

