

Кружок робототехники «Программирование роботов»-программирование роботов в среде VEXcode VR и LEGO NXT (стартовый уровень)
 Руководитель: Васильева Саргылана Ивановна

**План учебного занятия по теме
 «Блоки управления, блоки переменных, блоки датчиков, блоки вида, магнит»**

Программирование робота на платформе VEXcode VR

Тема занятия: Блоки управления, блоки переменных, блоки датчиков, блок вида, магнит.

Тип занятия: комбинированный.

Цель занятия: ознакомить учащихся с новой группой блоков управления VR-роботами возможностями программирования с их помощью.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход занятия

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

Рассмотрим панель блоков управления с указанием их функционала (таблица 1).

Таблица 1

Блоки управления и их функционал

 Трансмиссия	 Магнит	 Вид	 Управление	 Показания устройств	 Операторы	 Переменные	 Мои блоки
Блоки, управляющие движением и робота	Блоки, управляющие активацией магнита	Блоки, управляющие печатью в консоли пером	Блоки условных переходов и циклов	Блоки, управляющие датчиками	Блоки математических операторов	Блоки создания и управления переменными	Создание собственных блоков

Основное достоинство этих блоков — возможность создать условный переход (цикл) с их помощью. Важной группой блоков является группа **Управление**, с помощью которых организуются циклы и переходы по условиям. Виды блоков условных переходов и их функции приведены в таблице 2.

Таблица 2

Блоки условных переходов

Блок условного перехода	Функция
Повторять N раз	Устанавливает количество повторений фрагмента программы
Всегда	Организует бесконечное повторение фрагмента программы
Если...тогда	Переход по условию
Если...тогда...иначе...	Переход по условиям

Ждать пока...	Ожидание по выполнению условий
Повторять пока...	Повторение до выполнения условий

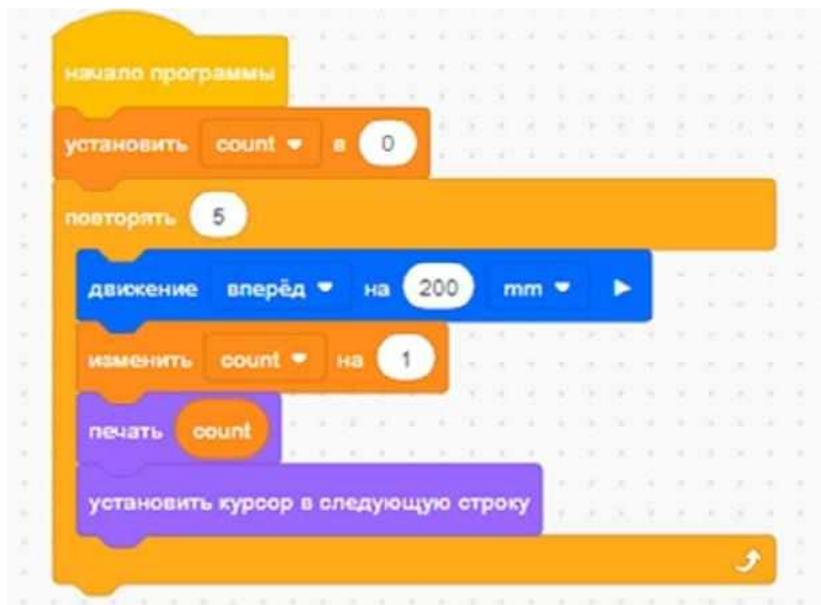


Рис. 17. Скрипт с использованием цикла **Повторять** и выводом на консоль **Экрана** значения переменной **COUNT** (счётчика итераций)

Учитель приводит примеры работы каждого блока. Например, блок, приведённый на рисунке 17, реализует следующий алгоритм: созданной обучающимся переменной **COUNT** присваивается значение 0 (чтобы реализовать счётчик итераций) → выставляем блок **Повторять 5 раз** и внутри этого блока помещаем другие блоки: **Движение вперед на 200 мм** → **Изменить COUNT на 1** → вывести текущее значение переменной на **Экран** (рис. 18).



Рис. 18. Выведенные на **Экран** значения переменной **COUNT**

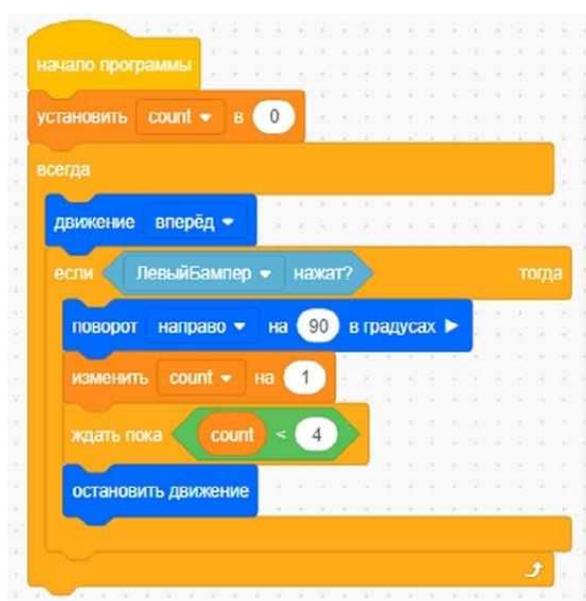
Ещё один из важных блоков — это блок **Показания устройств (Датчики)**. Описание функционала датчиков приведено в таблице 3.

Датчики. Описание функционала

Блок	Функция
Трансмиссия	4 датчика движения, которые помогают сформулировать некоторые условия для управления роботом
Бампер	При касании одним из датчиков (левым или правым) препятствия возвращают значение ИСТИНА или ЛОЖЬ. Рекомендуется разобрать с учащимися принцип действия датчика касания
Датчик расстояния	Возвращает расстояние до объекта (препятствия) или сигнализирует об обнаружении объекта. Работает как сонар (ультразвуковой излучатель / приёмник). Рекомендуется разобрать с учащимися принцип действия датчика расстояния
Датчики цвета	Определяют цвет (синий, зелёный, красный, никакой) внизу робота и перед роботом. Рекомендуется разобрать с учащимися принцип действия датчиков цвета
Положение на поле	Определяет координаты робота (X, Y) на игровом поле, а также текущий угол в градусах. Рекомендуется разобрать с учащимися принцип действия датчика положения на поле

Простейшая программа, иллюстрирующая работу управляющих блоков и датчиков VR-робота, — это уже знакомый нам обход игрового поля по периметру (рис. 19).

Рис. 19. Обход VR-роботом периметра игрового поля с использованием датчика касания



Алгоритм работает следующим образом: присвоим переменной **COUNT** (счётчик) значение 0 - используем управляющий блок **Всегда** (бесконечный цикл повторения), внутри которого разместим основной фрагмент программы — двигаемся вперёд, пока датчик не коснётся стены, — сработал датчик — поворачиваем вправо и снова двигаемся вперёд, при этом увеличивая значение переменной **COUNT** на единицу - повторяем цикл до тех пор, пока выполняется условие **COUNT < 4** (робот обходит 4 стены периметра), как только условие перестаёт выполняться, программа останавливает робота.

Более сложный алгоритм обхода VR-роботом периметра реализован на рисунке 20. В нём используется датчик расстояния и остановки проекта при его завершении. Также здесь приведён блок **Если...тогда...иначе** для организации условного перехода.

Рис. 20. Обход игрового поля по периметру с использованием датчика расстояния



Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.