

**Николаенкова Мария Николаевна,**  
магистрант 1-го курса направление «Теория и методика обучения и воспитания  
(по областям и уровням образования)», профиль «Образовательная робототехника»,  
кафедра информационных технологий в образовании,  
физико-математический факультет, БГПУ  
E-mail: masha.nikolaenkova@mail.ru

**Научный руководитель: Климович Анна Фёдоровна,**  
кандидат педагогических наук, доцент,  
декан физико-математического факультета, БГПУ

## **ВОЗМОЖНОСТИ ВИЗУАЛЬНОЙ СРЕДЫ SCRATCH 3.0 И КОНСТРУКТОРА LEGO WEDO 2.0 ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ**

**Nikolaenkova Mariya,**  
1st year undergraduate specialization in “Theory and Methods of Teaching and Education  
(in areas and levels of education)”, profile “Educational Robotics”,  
Department of Information Technologies in Education,  
Faculty of Physics and Mathematics, BSPU  
E-mail: masha.nikolaenkova@mail.ru

**Scientific adviser: Klimovich Anna,**  
Candidate of Pedagogical Sciences, Associate Professor,  
Dean of the Faculty of Physics and Mathematics, BSPU

## **FEATURES OF THE SCRATCH 3.0 VISUAL ENVIRONMENT AND LEGO WEDO 2.0 FOR TEACHING PROGRAMMING**

### **Аннотация:**

В статье рассматриваются вопросы обучения написанию программного обеспечения роботов, которые могут применяться для обучения школьников началам программирования и робототехники. Представлен опыт использования среды Scratch 3.0 и конструктора Lego Wedo 2.0 на занятиях.

### **Abstract:**

The article discusses the issues of teaching robots to write software that can be used to teach students the basics of programming and robotics. The experience of using Scratch 3.0 and Lego Wedo 2.0 in the classroom is presented.

**Ключевые слова:** визуальное программирование, робототехника, Scratch, LEGO

**Keywords:** visual programming; robotics, Scratch, LEGO

Информатика как учебная дисциплина появилась во второй половине XX века. В процессе развития вычислительной техники, постепенно возникла необходимость в подготовке специалистов, умеющих работать с разной вычислительной техникой.

В настоящее время программа школьного курса информатики претерпела некоторые изменения. Вопрос «чему учить современного школьника?» имеет множество ответов. Некоторое время назад обучение алгоритмизации и программированию отводилось не большое количество

часов, однако постепенно возникла необходимость построить школьное обучение информатики именно программированию. Новая программа и новые учебники предполагают обучение нескольким языкам программирования.

При обучении школьников программированию особое значение приобретает использование игровых методов и технологий, а также разработка учащимися самостоятельно компьютерных игр [1]. Познавательную активность формирует и тот вопрос, что всем хочется узнать, как же устроена игра. Погружая учащихся в структуру игры мы показываем им, что все построено на тех же алгоритмах и задачах. Продемонстрировать нам это поможет среда программирования Scratch в работе с конструкторами Lego.

Создатели языка программирования Scratch в 2019 году изменили внешнюю оболочку среды и сделали её работу доступной в сети Интернет, а так же дополнили её возможностью подключения робототехнических конструкторов. (рис.1)

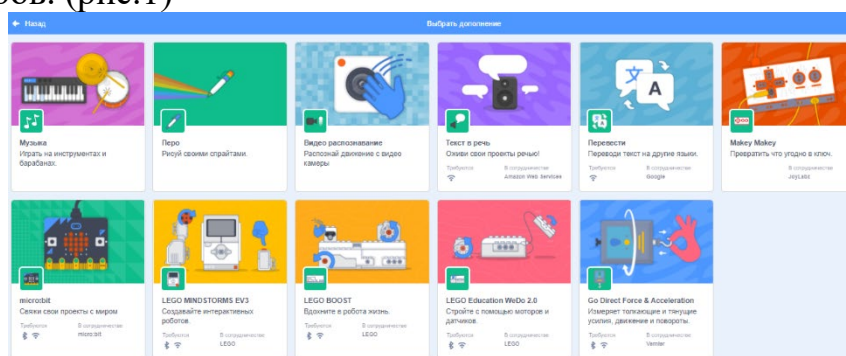


Рисунок 1 Перечень дополнительных блоков среды программирования Scratch

Разработчики робототехнических наборов безусловно предоставляют собственную среду программирования для каждого из своих наборов. Стандартная среда программирования для набора Lego Education Wedo 2.0 хороша для работы на первых этапах обучения, когда мы пытаемся сформировать конструкторский навык, знакомим с основами конструирования и не углубляемся в программирование. Навык сбора моделей разрабатывается достаточно быстро и возрастает запрос учащихся на выполнение более сложных задач, им становится недостаточным однократное действие робота.

Параллельно с обучением программированию на стандартном языке Wedo вводим соответствующие понятия и блоки в среде Scratch. (рис.2)

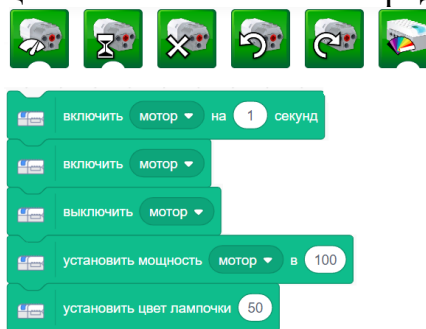


Рисунок 2. Группы блоков среды Wedo и Scratch

В качестве примера игры, сочетающей возможности Lego Education Wedo 2.0 и Scratch уместно предлагать игры, которые широко известны данной возрастной группе. На примере игры «Порхающая птичка» можно оценить степень владения разными навыками, приобретенными в процессе обучения.

На аналитическом этапе демонстрируем готовый проект, просим учащихся предположить какое количество спрайтов задействовано в проекте, какие алгоритмические конструкции возможны для конкретного спрайта(рис.3).

Следующим этапом рассматриваем готовый код программы для каждого спрайта, определяем какие алгоритмы использовались и почему. Учащиеся, которые уже изучали данную среду владеют терминологией и структурой алгоритмического языка, соотносят вид и тип алгоритма с принципом его работы.

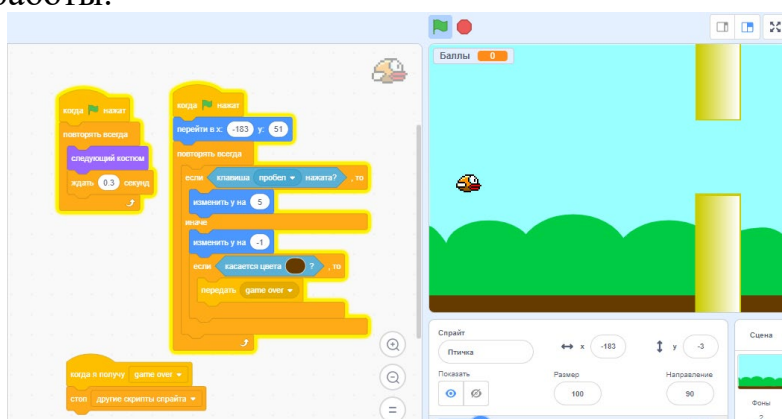


Рисунок 3. Пример готового проекта игры "Порхающая птичка"

Завершающим этапом работы предлагается учащимся разработать робототехническую модель, которая позволит им выполнить все те же действия для успешного прохождения уровней игры. Уместным, при возникновении трудностей, будет натолкнуть учащихся на решение подобной задачи с использованием джойстика, который может быть сконструирован за небольшое время. Прежде чем приступаем к программированию работы устройства, оговариваем какие изменения должны произойти в программном коде игры, какие блоки будем брать и какое устройство будет основным в работе джойстика. Для развития и оценки уровня сформированности алгоритмического и программного мышления в качестве примера изменений программы даём два варианта составления программного кода, обсуждаем ответы, проводим эксперимент с работающим джойстиком.(рис.4)

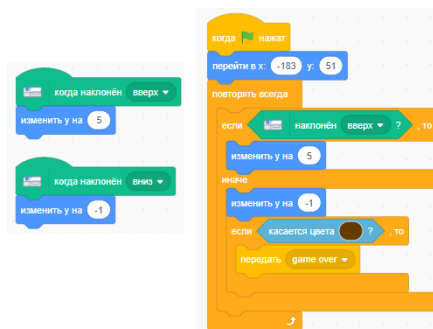


Рисунок 4. Пример изменения готового кода программы

Успешность подобной работы безусловно зависит от того, как давно идет обучение, знакомы ли учащиеся со средой программирования ранее или только начали изучать. Визуализация работы алгоритмов способствует лучшему усвоению принципов их работы, помогает структурировать материал и отработать все виды алгоритмов. Данный принцип даёт возможность учащимся, которые воспринимают информацию по-разному (визуально, на слух или тактильно) усвоить материал удобным для себя способом.

Сочетание робототехники и визуального программирования дает возможность педагогу определить среди учащихся тех, кто больше склонен к конструированию и тех, кто более заинтересован в программировании. Сложность программ и робототехнических конструкций будет так же зависеть от того, какое время у вас отводится на занятие. Если ваше занятие длится не более академического часа (45 минут), то не всегда сложные робототехнические конструкции могут быть собраны быстро, что чаще всего будет требовать от вас один и тот же проект отрабатывать на двух занятиях. Учащимся же чаще всего хочется видеть результат сразу и ожидание может их оттолкнуть от обучения.

Выстраивая работу по обучению визуальному программированию и робототехнике, следует разделять и структурировать учебный материал таким образом, чтобы он мог постепенно усложняться и сочетать оба аспекта. Изучив среду программирования возможно дополнить ее стандартной робототехнической моделью, которая сможет претерпевать незначительные конструкторские изменения (установка датчиков) и одновременно отрабатывать разные типы алгоритмов. На последующих этапах сочетание проекта и робота может строиться на главенстве либо проекта, либо модели. Работая по такому принципу будет возможно сохранить интерес у детей к изучению материала, а так же позволит отработать все необходимые навыки.

#### Список литературы

1. Данильчук Е.В., Куликова Н.Ю., Гермашев И.В. Методические особенности формирования готовности будущего учителя информатики к разработке и использованию компьютерных игр в обучении алгоритмизации и программированию // Известия Волгоградского государственного педагогического университета. – 2018. - № 5 (128). – С. 42-49.

2. . Проект игры «Порхающая птичка» [Электронный ресурс].  
Режим доступа: <https://scratch.mit.edu/projects/263884586>. – Дата доступа:  
20.11.2020.