

Муниципальное бюджетное образовательное учреждение дополнительного образования Центр детского технического творчества Бугульминского муниципального района Республики Татарстан

Методическая разработка занятия

«Конструкция «Всегда»»

объединение «Компьютерная графика»

Разработала:
Макаревская Светлана Алексеевна,
педагог дополнительного образования

г. Бугульма, 2023

Разработка занятия: "Конструкция «Всегда»"

Цели занятия:

Развивающие:

1. Развитие внимания, мышления.
2. Выявление закономерностей учебного материала.
3. Выявление творческих способностей.

Обучающие:

1. Формирование у учащихся понятий «алгоритмическая конструкция, ветвление, полная и неполная форма»; умений говорить на естественном и машинном языке о соответствующей алгоритмической конструкции

Воспитательные:

1. Воспитание точности, аккуратности; стремления к совершенствованию знаний; уверенности в себе, самооценке своих знаний в сравнении со знаниями одноклассников.

Задачи занятия:

- 1) обобщить представления об алгоритмической конструкции «ветвление»;
- 2) получить навыки записи составных условий;
- 3) получить навыки выполнения алгоритмов с ветвлениями для различных формальных исполнителей;
- 4) получить навыки разработки алгоритмов с ветвлениями для различных формальных исполнителей с заданной системой команд.

Планируемые результаты:

предметные – представление об алгоритмической конструкции «ветвление»; умение исполнять алгоритм с ветвлением для формального исполнителя с заданной системой команд; умение составлять простые (короткие) алгоритмы с ветвлением для формального исполнителя с заданной системой команд;

метапредметные

– умение выделять алгоритмы с ветвлением в различных процессах; понимание ограниченности возможностей линейных алгоритмов;

личностные – алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе.

Необходимые ресурсы. материалы:

1. Презентация Microsoft Power Point

Методы обучения: наглядный(по источнику получения знаний), объяснительно-иллюстративный(по степени активности познавательной деятельности учащихся)

Формы организации учебной деятельности: индивидуальная, фронтальная

Основные понятия, изучаемые на занятии:

алгоритм;
ветвление;
разветвляющийся алгоритм;
блок-схема;
операции сравнения;

простые условия;
составные условия.

Ход занятия

1. Организационный момент.

Приветствие, проверка подготовленности к учебному занятию, организация внимания детей.

2. Повторение

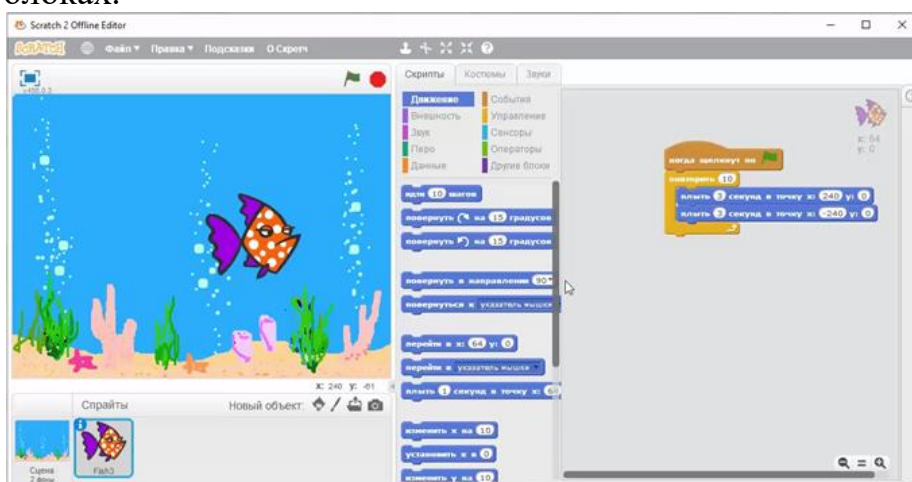
- На прошлом уроке мы с вами учились рисовать узоры и орнаменты в среде Scratch. Сегодня продолжим изучение.

3. Изучение нового материала

- Утром Царапке пришла в голову мысль: «Рыбки – это же так здорово! Можно часами смотреть на то, как они плавают. Почему бы нам не завести свой аквариум?» Сделаем в Scratch свой аквариум, в котором будет плавать рыбка.

-Для начала удалим со сцены спрайт Царапки. Котам в аквариуме не место! Подберём подходящий фон для сцены из библиотеки фонов Scratch. Выберем тему «Подводный мир». Как видим, в ней всего три подводных фона. Первые два фона немного тёмные, поэтому выберем третий. Теперь наша сцена совсем как аквариум. Добавим спрайт рыбки. Его также можно найти в библиотеке Scratch в теме «Подводный мир». Выберем в ней одну из рыбок. -
-Теперь нужно составить скрипт для спрайта рыбки. Для того, чтобы скрипт можно было без проблем запустить, начнём его с блока «Когда щёлкнут по зелёному флажку» из категории «События».

- Мы знаем, что для плавного движения спрайта можно использовать блок «Плыть в точку». Пусть рыбка плавает между двумя точками по краям аквариума. Для этого возьмём уже знакомый нам блок «Повторить» и зададим в нём 10 повторений. Добавим в него два блока «Плыть в точку». Зададим в них точки с координатами двести сорок, ноль и минус двести сорок, ноль. Укажем время перемещения между точками: 3 секунды. Так скорость плавания рыбки можно регулировать, меняя время плавания в блоках.

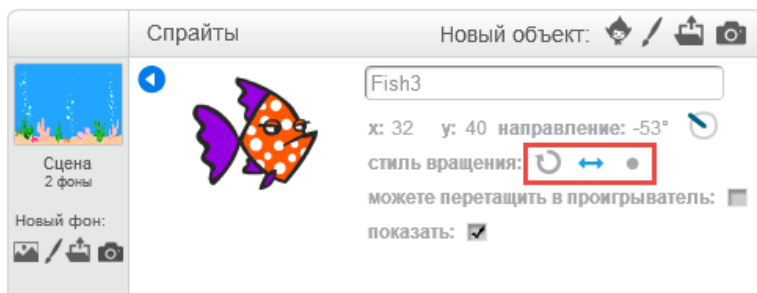


- Попробуем запустить скрипт на выполнение. Рыбка плавает из одного края сцены в другой, но нормально она плывёт только вправо, а вот влево она плывёт хвостом вперёд. Так быть не должно. Чтобы это исправить, сделаем так, чтобы рыбка поворачивалась по краям сцены. Добавим для этого после каждого блока «Плыть в точку» блок «Повернуть в направлении». Оказавшись на правом краю сцены, рыбка должна повернуться влево, то есть в направлении -90 градусов. Соответственно, в левом краю сцены рыбка должна повернуться в направлении 90 градусов.

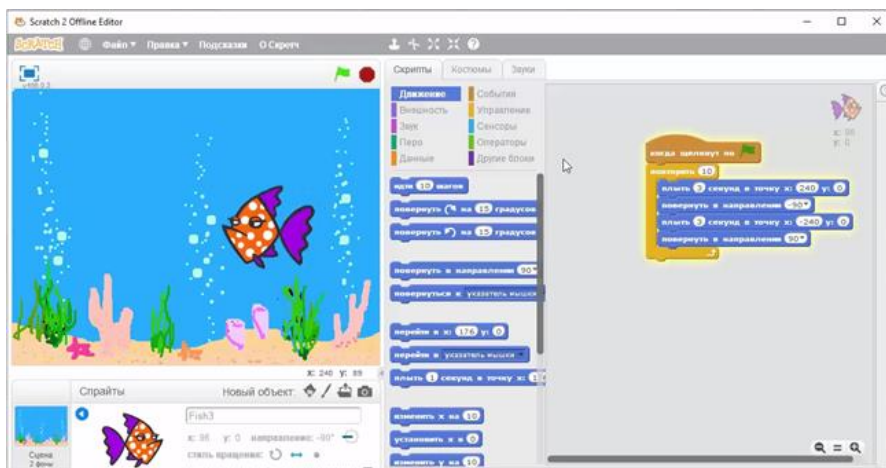
- Снова запустим скрипт на выполнение. Рыбка поворачивается, достигнув края сцены. Только вот теперь влево она плывёт вверх тормашками. Почему так происходит? Дело в том, что при выполнении команды «Повернуть в направлении» спрайт рыбки поворачивается относительно своего центра так, как это показано на экране. Как же это изменить?



- Спрайтам Scratch можно задавать разные стили вращения. Всего их три: относительно центра, относительно центральной оси и без вращения. Они влияют только на то, что происходит с костюмом спрайта на сцене, и никак не влияют на рисование. Стил вращения можно установить в окне информации о спрайте. Рассмотрим стили вращения подробнее. **Вращение относительно центра** мы знаем. При установке этого стилиа костюм спрайта свободно вращается относительно своего центра. **Вращение относительно центральной оси** работает немного иначе. Если выставить этот стил, то у костюма будет всего два положения. Какое из них будет показано, зависит от значения направления, в которое повёрнут спрайт. Если оно положительное или нулевое, то костюм будет иметь изначальный вид. Если направление отрицательно, то костюм будет отражён по горизонтали. Название стилиа «**Без вращения**» говорит само за себя. Если установить такой стил для спрайта, то его костюм не будет меняться с изменением направления.



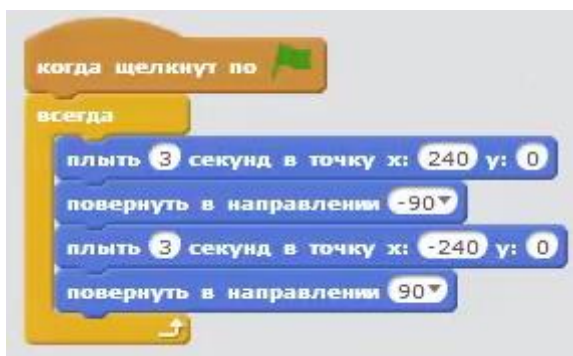
- Изменим стиль вращения спрайта рыбки. Для этого откроем окно информации о спрайте. Рассмотрим кнопки стилей вращения. Выделение первой кнопки показывает, что выбрано вращение относительно центра. Установим стиль «Без вращения», нажав на третью кнопку, и запустим скрипт на выполнение. Теперь рыбка перестала поворачиваться. Остановим выполнение скрипта, нажав на красный восьмиугольник над сценой. Укажем стиль вращения относительно центральной оси, нажав на вторую кнопку, и снова запустим скрипт на выполнение. Ура! Теперь рыбка поворачивается прямо как настоящая!



- Но нашу задачу всё ещё нельзя считать решённой. У составленного нами скрипта всё ещё есть несколько недостатков. Первый недостаток довольно очевидный. Нигде не написано, сколько времени рыбка должна плавать в аквариуме. Логично предположить, что рыбка должна плавать до тех пор, пока её не остановит пользователь. В нашем же скрипте количество повторений ограничено десятью. Кто-то справедливо заметит, что можно увеличить количество повторений цикла до очень большого числа. И верно: если мы укажем 6 миллионов повторений цикла, то такой скрипт будет выполняться больше года (если, конечно, пользователь не остановит его выполнение раньше). Но какое бы число мы ни указали, оно будет конечно. **Если нужно, чтобы цикл выполнялся неограниченное количество раз до остановки пользователем, в Scratch используется**

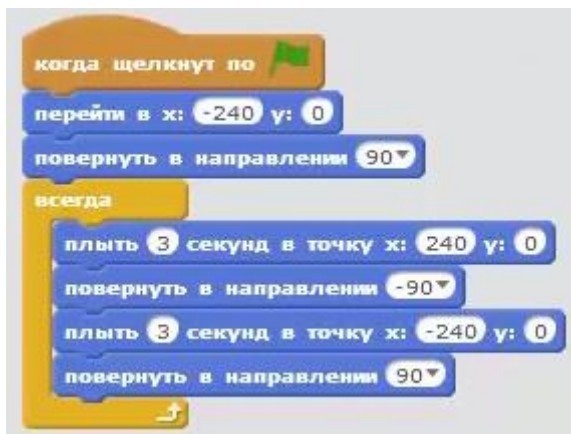
бесконечный цикл «Всегда». Его блок, как и блок «Повторить», находится в категории «Управление».

- Чтобы изменить цикл, который используется в нашем скрипте, отделим тело цикла от остальной части скрипта. Удалим блок цикла «Повторить» и перетащим на его место блок «Всегда». Перетащим тело цикла на место. Снова запустим скрипт на выполнение. Он выполняется точно так же, как и раньше. Правда, теперь он может выполняться неограниченно долго.

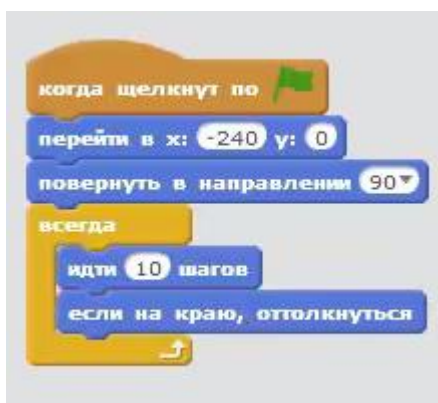


- Однако у нашего скрипта есть ещё несколько недостатков. Они становятся видны при его запуске. Остановим выполнение скрипта в тот момент, когда рыбка будет двигаться влево. И снова запустим его на выполнение. Теперь рыбка двигается задом наперёд, пока не достигнет правого края сцены. Мы хорошо знаем, как решить эту проблему. Достаточно задать рыбке верное начальное направление. Для этого добавим перед циклом блок «Повернуть в направлении...» из категории «Движение». Рыбка в начале всегда движется вправо, это соответствует направлению 90 градусов, укажем его в блоке.

- Снова запустим наш скрипт на выполнение. Рыбка плывёт правильно. Сделаем это несколько раз и убедимся, что рыбка теперь всегда повернута в нужном направлении. Ещё один недостаток нашего скрипта состоит в том, что у рыбки при каждом запуске разная начальная скорость, иногда она движется очень медленно. Давайте подумаем, почему. Дело в том, что независимо от того, где рыбка находится изначально, она движется в точку двести сорок, ноль ровно три секунды. Вот и получается, что, когда рыбка далеко от этой точки, она плывёт к ней быстро, а когда близко – медленно. Эту проблему можно решить, установив рыбку на левом краю сцены, в точке (-240; 0). Тогда она всегда будет проходить сцену за 3 секунды. Добавим блок для перехода в эту точку в начало скрипта. И запустим скрипт на выполнение.



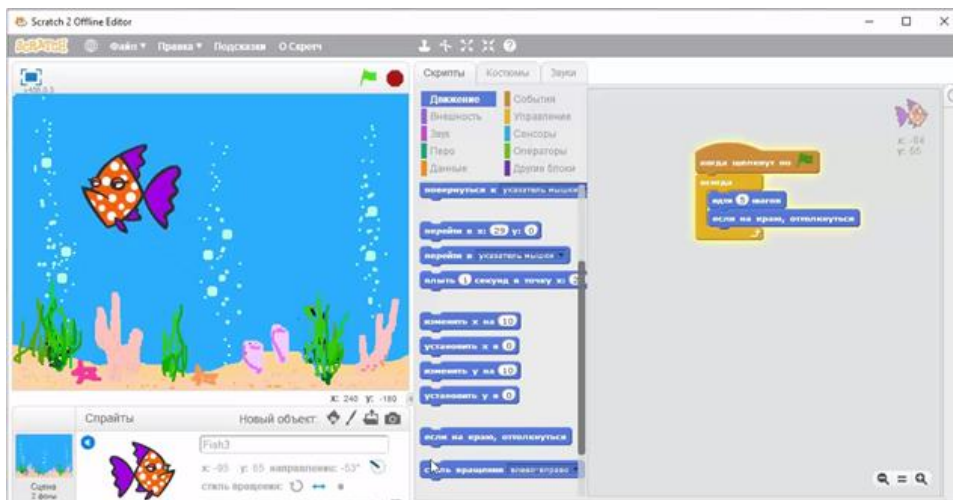
- Теперь рыбка всегда плавает с одинаковой скоростью, но всё же она выглядит как-то странно. Да она же всегда плавает по одному и тому же простому маршруту! Это можно исправить, усложнив маршрут, но так всё равно будет заметно, и к тому же это сильно увеличит размер нашего скрипта. Поэтому мы поступим иначе. Мы полностью изменим тело цикла в скрипте. Для этого остановим выполнение скрипта и удалим старое тело цикла. В качестве нового тела установим всего два блока из категории «Движение»: «Идти 10 шагов» и «Если на краю, оттолкнуться».



- Снова запустим скрипт на выполнение. Скрипт работает почти так же, как и до этого, только рыбка стала плавать быстрее. Регулировать скорость рыбки можно, меняя количество шагов в блоке «Идти», установим его равным 5. Теперь рыбка значительно замедлилась. Да, теперь скрипт немного короче, но как заставить рыбку плавать по-разному? Это можно сделать очень просто. Остановим выполнение скрипта. Отделим цикл от остальной части скрипта и удалим блоки, которые устанавливают начальное состояние рыбки.

- Вернём цикл на место. Снова запустим скрипт на выполнение. Он стал намного короче, но пока работает точно так же, как и до этого. Немного терпения... Это легко изменить. Просто установим любое случайное направление спрайта, не параллельное границам сцены, и понаблюдаем за рыбкой. Теперь она двигается по всему аквариуму и меняет своё

направление, касаясь его краёв. Наш аквариум с рыбкой совсем как настоящий! Сохраним наш проект в рабочей папке под именем «Аквариум».



4. Подведение итогов.

Ветвление - алгоритмическая конструкция, в которой в зависимости от результата проверки условия (да или нет) предусмотрен выбор одной из двух последовательностей действий (ветвей). Алгоритмы, в основе которых лежит структура «ветвление», называют разветвляющимися.

5. Рефлексия.

- Что понравилось на занятии?
- Почерпнули ли Вы что-то новое для себя?
- Что не понравилось?
- Что вызвало трудности? Почему?