

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

## **Методическая разработка занятия по компьютерной графике**

**«Датчик случайных чисел»**

разработала педагог ДО:  
Макаревская С.А.

Бугульма, 2024

## Разработка занятия: Датчик случайных чисел

### Цель занятия:

Развитие понимания датчика случайных чисел на языке Pascal.

### Задачи занятия:

#### Образовательные:

Дать понятие случайных чисел;

Показать работу датчика чисел на Паскале;

Разработать алгоритм вычисления поиска чисел в массиве.

#### Развивающие:

Развить пространственное представление ;

Развить логическое мышление;

Развить творческие способности.

#### Воспитательные:

Воспитать терпения в работе;

Воспитать четкую организацию учебного процесса;

Воспитать информационную культуру и культуру общения.

**Форма занятия:** комбинированное занятие.

## Ход занятия

### 1. Организационный момент

-Здравствуйте, ребята, присаживайтесь.

### 2. Актуализация знаний

-Какую тему вы изучали на прошлом уроке?

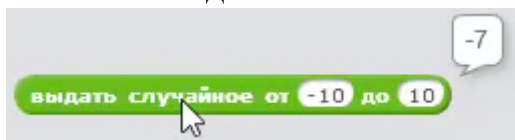
- Что такое составные условия?

- Что такое датчик случайных чисел?

### 3. Изучение нового материала.

- Датчик случайных чисел в Scratch представляет собой очередной блок, которых мы знаем уже не мало. Он находится на вкладке «Скрипты» в категории «Операторы». Надпись на блоке говорит о его назначении «Выдать случайное». С помощью чисел в ячейках блока пользователь может указать, в каких пределах нужно случайное число. Пока ничего менять не будем.

- Щёлкнем по блоку левой кнопкой мыши. Над блоком появилось число в рамке. Попробуем щёлкнуть по блоку ещё несколько раз. И каждый раз блок выдаёт случайное целое число от 1 до 10 включительно. Изменим минимальное значение в блоке. Укажем его равным -10. Ещё несколько раз щёлкнем по нему левой кнопкой мыши. Теперь среди результатов появились и случайные числа до -10. Итак, датчик случайных чисел генерирует целые случайные числа в заданных пределах. Есть ещё один нюанс использования этого блока. Попробуем поменять числа в ячейках блока местами и щёлкнем по нему ещё несколько раз. Блок работает всё также. Он выдаёт случайные числа от -10 до 10.



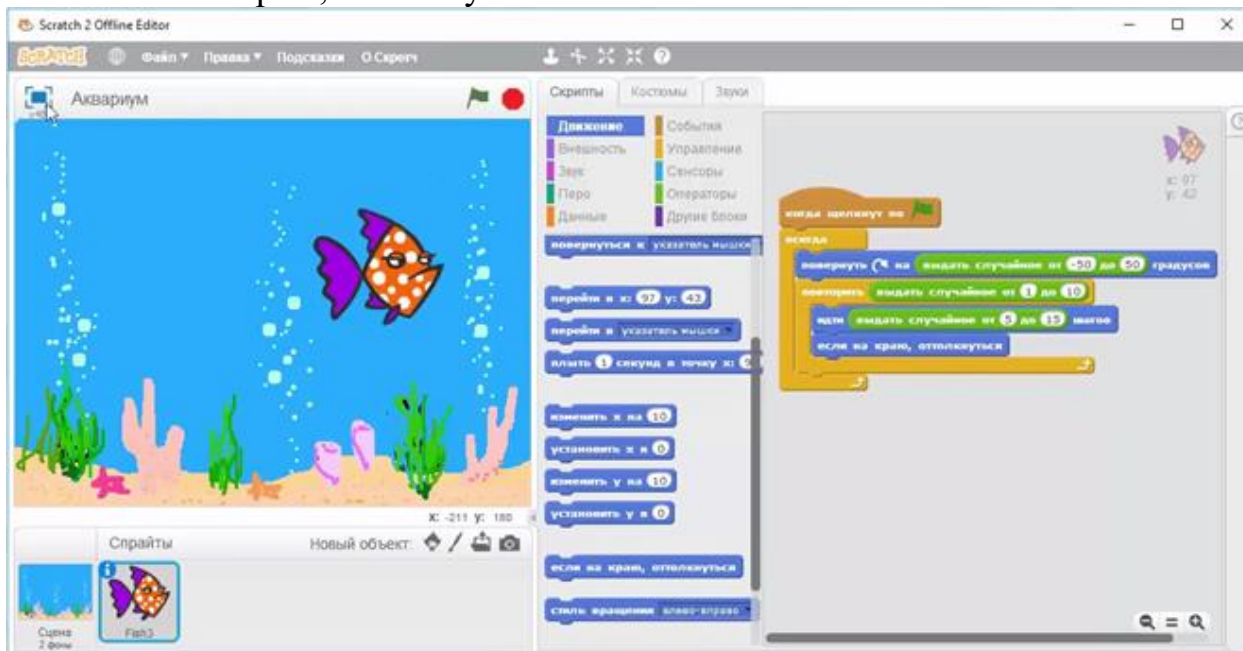
-Как же использовать датчик? Наверняка, многие из вас уже по форме блока поняли, что он вставляется в другие блоки. Его можно установить на место любого числового или текстового параметра в скрипте. Из категории «Внешность» возьмём блок «Изменить размер на...». Установим внутрь него наш датчик случайных чисел. Несколько раз щёлкнем по блоку левой кнопкой мыши. Спрайт Царапки то уменьшается, то увеличивается и всегда по-разному. Это значит, что датчик случайных чисел работает нормально.



- Также с помощью датчика случайных чисел можно сделать случайными движения спрайта. Загрузим из рабочей папки уже знакомый нам проект «Аквариум». Сейчас рыбка в нём двигается не случайно. Пока рыбка изменяет направление движения, лишь достигнув края сцены. Удалим тело цикла в скрипте рыбки и составим его снова. Сперва добавим в цикл блок, который случайным образом будет менять направление движения спрайта.

- Для этого из категории «Движение» возьмём блок «Повернуть по часовой стрелке на...». Внутри него установим датчик случайных чисел. Сделаем так, чтобы этот датчик генерировал случайное число от -50 до 50. Так блок поворота будет поворачивать спрайт на случайный угол до пятидесяти градусов по часовой стрелке или против часовой стрелки.

- Далее рыбка должна двигаться со случайной скоростью на случайное расстояние. Для этого сделаем случайное количество шагов на случайное расстояние. Установим в цикл блок «Повторить», в котором с помощью датчика случайных чисел зададим случайное число от 1 до 10. Во вложенном цикле поместим блок «Идти». В нём зададим случайное количество шагов, например от 5 до 15. После блока «Идти» из категории «Движение» добавим блок «Если на краю, оттолкнуться».



- В режиме презентации запустим скрипт на выполнение. Теперь рыбка двигается совсем как живая. Она в любой момент может изменить

направление движения и его скорость. Мы сделали рыбку в аквариуме более живой. Сохраним проект.

- Можно поселить в наш аквариум ещё одного жителя и сделать из этого игру. Сделаем игру, в которой акула, которая управляется пользователем, должна поймать рыбку.

- Сперва добавим на сцену спрайт акулы. Он находится в библиотеке Scratch в категории «Подводный мир» под именем «Shark». Установим нужные размеры акулы и рыбки с помощью кнопок. Посмотрим, какие костюмы есть у нашей акулы. Для этого перейдём на вкладку «Костюмы». Посмотрим на второй костюм. Оказывается, что наша акула умеет открывать рот. Это пригодится для того, чтобы схватить рыбку. Третий костюм имеет вид грустной акулы. Такой акула будет, если схватить рыбку не удастся. Установим для акулы такой же стиль вращения, что и у спрайта рыбки, то есть относительно центральной оси.



- Начнём составление скрипта для управления акулой. Он будет начинать работу при нажатии на зелёный флажок. Чтобы закрыть рыбку собой, акула должна быть в верхнем слое. Чтобы так и было, установим в скрипт из категории «Внешность» блок «Перейти в верхний слой». Так как наша акула будет менять костюмы, установим для неё нужный костюм в начале скрипта.

- Для этого из категории «Внешность» возьмём блок «Сменить костюм на...». В блоке укажем первый костюм. Он называется «shark-a».

- Дальше будет часть для управления спрайтом. Она будет работать, пока игра не будет закончена, для этого поместим её в бесконечный цикл «Всегда». Управлять акулой пользователь будет с помощью стрелок. Пусть при нажатии боковых стрелок акула поворачивается в соответствующую сторону и плывёт туда, а при нажатии стрелок вверх и вниз просто перемещается в соответствующих направлениях. Установим в скрипт блок для сокращённой формы ветвления. В качестве условия установим в нём сенсор нажатия на клавишу «Стрелка влево». В ветвление из категории «Движение» добавим блоки «Повернуть в направлении» и «Идти». При

нажатии на стрелку влево акула должна повернуться влево, то есть в направлении  $-90$  градусов. А перемещение установим равным 3 шагам. Для стрелки вправо продублируем этот блок ветвления и кое-что в нём изменим. - В сенсоре установим стрелку вправо, а в блоке поворота укажем направление  $90$  градусов.

- Теперь добавим блок ветвления, который будет срабатывать при нажатии стрелки вверх. В него в качестве условия поместим сенсор, срабатывающий при нажатии стрелки вверх. В этом случае акула не должна никуда поворачиваться, а должна просто смещаться вверх. Для этого добавим внутрь ветвления из категории «Движение» блок «Изменить игрек на». Установим перемещение так же, как и у боковых стрелок, равным 3. Скопируем этот блок ветвления и изменим его для стрелки вниз. Для этого в сенсоре изменим клавишу на стрелку вниз, а в блоке изменения Игрек установим число  $-3$ .



- Часть скрипта для управления перемещением акулы закончена.

Протестируем её. Для этого запустим скрипты в режиме презентации, нажав на зелёный флажок. Наша старая рыбка двигается абсолютно случайно, как и до этого. А перемещением акулы мы теперь без проблем можем управлять с помощью стрелок. Теперь нужно добавить в скрипты команды для хватания рыбки.

- Сделаем, чтобы акула попыталась схватить рыбку, когда пользователь нажмёт на клавишу «Пробел». Для этого добавим в цикл блок для сокращённой формы ветвления, в котором установим сенсор нажатия на клавишу «Пробел». Для того чтобы схватить рыбку, акула должна открыть рот. Для этого установим в блок ветвления из категории «Внешность» блок «Сменить костюм на», в котором укажем костюм «shark-b». Дальше акула

должна быстро переместиться в направлении своего движения. Для этого добавим цикл «Повторить». В цикл добавим блок «Идти». Укажем 3 повторения по 10 шагов.

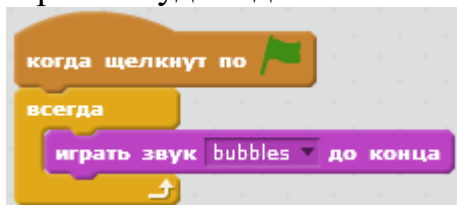
- Далее нужно проверить, попала ли рыбка. Для этого добавим блок для полной формы ветвления. В нём в качестве условия установим сенсор касания рыбки. Для этого в сенсоре укажем спрайт рыбки, то есть «Fish3». Если акула касается рыбки, она должна её поглотить. Для этого достаточно просто установить акулу поверх рыбки. Добавим в ветвление блок «Перейти в...», в выпадающем списке которого укажем рыбку, то есть спрайт «Фиш три». После выполнения блока центры спрайтов акулы и рыбки будут совпадать. Далее акула должна закрыть рот. Для этого снова сменим костюм акулы на «shark-a». Раз у акулы получилось поймать рыбку, то игра на этом закончилась и нужно завершить все скрипты. Добавим блок «Стоп», в выпадающем списке которого укажем все скрипты.

- Теперь рассмотрим случай, когда акуле не удалось поймать рыбку. Мы условились, что акула должна стать грустной. Для этого сменим костюм акулы на «shark-c». После этого, через какое-то время, акула должна вернуться в нормальное состояние, и игра продолжится. Добавим блоки «Ждать 3 секунды» и «Сменить костюм на “ shark-a ”».

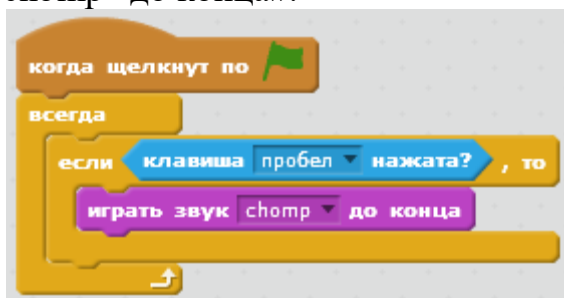


- По сути наша игра закончена. Попробуем в неё сыграть. Для этого в режиме презентации нажмём на зелёный флажок. Рыбка маленькая и движется значительно быстрее акулы, поэтому поймать её не так просто. Если рыбку не получается поймать, то акула грустит. Если же получится поймать рыбку, то акула поглотит её, а после этого игра закончится. Всё работает так, как задумано.

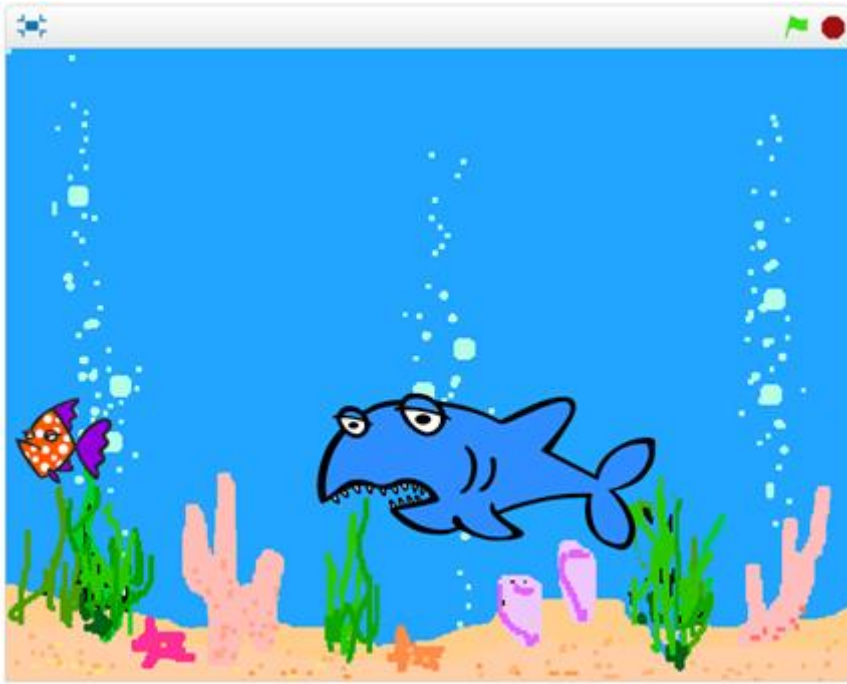
- Мы можем немного оживить нашу игру, добавив в неё звуки. Сперва добавим подходящий звук окружения. Для этого выделим сцену и перейдём на вкладку «Звуки». Удалим уже имеющийся там звук. Из библиотеки звуков Scratch, из категории «Эффекты», прослушаем звук «bubbles». Его название с английского языка можно перевести как «Пузыри», и для звука окружения он нам замечательно подходит. Выберем его. Теперь составим скрипт для звука сцены. Он будет запускаться, как и остальные скрипты, при нажатии на зелёный флажок и длиться до конца игры. Внутри цикла «Всегда» в этом скрипте будет единственный блок «Играть звук “ bubbles ” до конца».



Теперь выделим спрайт акулы. Удалим уже имеющийся у него звук и добавим новый. В библиотеке Scratch, в категории «Животное», прослушаем звук с названием «chomp», что в переводе означает «Чавканье». Этот звук вполне подойдёт для хватания акулой добычи, поэтому добавим его к спрайту акулы. Для звука акулы также составим отдельный скрипт. Он будет запускаться при нажатии на зелёный флажок и длиться до конца игры. Звук должен воспроизводиться, когда акула попытается схватить рыбку, то есть при нажатии на клавишу «Пробел». Поэтому в цикл «Всегда» добавим блок сокращённой формы ветвления. В нём в качестве условия поместим сенсор нажатия на клавишу «Пробел». В ветвлении разместим блок «Играть звук “ chomp ” до конца».



Мы добавили нужные звуки. Снова запустим игру в режиме презентации и посмотрим как они работают. Всё работает правильно и звуки воспроизводятся как надо. Наша игра закончена. Сохраним её в рабочей папке под именем «Рыбка и акула».



#### **4. Рефлексия**

Как вы понимаете датчик случайных чисел?

Как в scratch организовать случайный порядок чисел?

Приведите примеры случайного порядка чисел.

#### **5. Итог занятия**

- Что было не понятно на занятии?

- Что запомнилось больше всего?