**Урок №3. Условия. Перемещение объектов**

**Вводный блок (10 минут)**

**Цели и задачи урока**

1. «Целью нашего занятия является знакомство с условными операторами и их применение для перемещения объектов на холсте».

**Обсуждение**

Вопросы к ученикам:

1. Играли ли вы в игры? На какие кнопки там двигаются персонажи?
2. Как вы думаете сложно ли создать алгоритм для перемещения объектов? Как бы вы это сделали?

**Познание нового (15 минут)**

**Знакомство с условиями**

**Условия**

Для начала разберемся как работать с условиями и их синтаксисом. Предположим, мы в магазине хотим купить футболку, которая стоит 2000 рублей. И здесь встает вопрос о том, а есть ли у нас такая сумма. Если есть такая сумма, то мы можем ее купить.

С точки зрения программирования это будет записано следующим образом

<script>

const canvas = document.querySelector('canvas');

const context = canvas.getContext('2d');

let ourMoney = 2500;

const cost = 2000;

**if (cost < ourMoney) {**

**context.font = "32px serif";**

**context.fillStyle = "rgb(0,100,200)";**

**context.fillText("Мы можем купить эту футболку", 100, 100);**

**}**

</script>

В данном случае созданы две переменные: cost - стоимость футболки и outMoney - сумма, которая у нас есть. В случае, если значение в переменной ourMoney больше, то будет выводиться надпись, сигнализирующая о том, что мы можем купить эту футболку.

Для условия используется условный оператор if, после которого в скобках записывается само условие, которое должно сработать. Если условие сработало (правда), то будет выполняться все то, что написано в фигурных скобках.

**Вопрос к ученикам**

1. Что будет, если у нас всего 1500?
2. А 2000?
3. Как сделать так, чтобы при 2000 условие срабатывало?

**Операторы сравнения**

Есть и другие операторы, которые могут быть прописаны в скобках:

< - меньше

> - больше

<= - меньше или равно

>= - больше или равно

== - равно

!= - не равно

С помощью условий мы сможем руководить перемещением объекта на странице.

Для этого нам нужно воспользоваться событиями.

**Некомпьютерная активность (5 минут)**

* Игра «Верю — не верю». Один участник загадывает условие (например, «если сегодня вторник, то завтра среда»), а другой должен ответить, верно ли это утверждение. Задача второго участника — определить, является ли условие истинным или ложным, задавая уточняющие вопросы, на которые первый может отвечать только «да» или «нет».

**Работа над проектом**

Тема: Создание нового проекта и написание программы, которая управляет движением персонажа в разные стороны.

**Часть №1. Создание анимации (20 минут)**

**Событие** - интерактивная часть работы со страницей (нажатие на любой элемент страницы, загрузка страницы и т.д.)

В нашем случае необходимо отслеживать нажатие на клавиши на клавиатуре.

Для этого воспользуемся стандартным управлением:

W - вверх

S - вниз

A - влево

D - вправо

Создадим квадрат на холсте размерами 100х100

Для этого, в самом начале скрипта, создадим две переменные, отвечающие за местоположение объекта (x,y) и переменную, отвечающую за размеры (size)

<script>

const canvas = document.querySelector('canvas');

const context = canvas.getContext('2d');

let x = 100;

let y = 100;

let size = 100;

context.fillRect(x, y, size, size)

</script>

Первым делом необходимо создать алгоритм, который позволит анимировать. Для обновления кадров используется функция move (о синтаксисе функций мы поговорим на следующих занятиях), в которой переменная x меняется каждый кадр на 1, затем очистка кадра и отрисовка всех компонентов заново. Для того, чтобы происходила анимация, вызывается функция requestAnimationFrame в которой указывается move для обновления кадра

function move() {

x += 1;

context.clearRect(0, 0, canvas.width, canvas.height);

context.fillRect(x, y, size, size);

requestAnimationFrame(move);

}

move();

**Задание для учеников**

Заставить квадрат двигаться по диагонали в левую нижнюю сторону.

**Часть №2 Перемещаем объект по нажатию на кнопки (35 минут)**

Для начала создадим две переменные (dx, dy) которые будут хранить в себе направление для перемещения, например, если движение будет вверх, то в переменной dx хранится 0, а в dy хранится -10 и т.д.

Значения переменных dx и dy добавляются к переменным x и y соответственно

let x = 100;

let y = 100;

let size = 100;

**let dx = 0;**

**let dy = 0;**

function move() {

**x += dx;**

**y += dy;**

context.clearRect(0, 0, canvas.width, canvas.height);

context.fillRect(x, y, size, size);

requestAnimationFrame(move);

}

Для отслеживания нажатия и перемещения объекта используют команду AddEventListener

**window.addEventListener('keydown', (event)=>{**

**})**

Внутри скобок указываются два параметра - отслеживаемое событие и функция, которая должна выполниться при нажатии на клавиши (записывается внутри **(event)=>{ })**

На примере нажатия на клавишу W рассмотрим перемещение объекта

window.addEventListener('keydown', (event) => {

**if (event.key.toLowerCase() == "w") {**

**dy = -10;**

**dx = 0;**

**}**

})

Для более плавного перемещения объекта нужно создать еще одно событие keyup - отжатие кнопки, нужно будет прописать следующее:

window.addEventListener('keyup', (event) => {

if (event.key.toLowerCase() == "w") {

dy = 0;

dx = 0;

}

})

**Задание для учеников**

Дописать условия перемещения для остальных кнопок (влево, вправо, вниз и вверх).

**Дополнительное задание**

При выходе за пределы окна отображать объект с противоположной стороны.

**Рефлексия (5 минут)**

"Что нового вы узнали сегодня?"

"Что такое условия? Какие конструкции используются на JavaScript для составления условий?"