**Урок №6. Массивы. Спрайты**

**Вводный блок (10 минут)**

**Цели и задачи урока**

1. Познакомиться с синтаксисом создания массива и работы с ним.
2. На примере загрузки спрайтов рассмотреть методы и свойства работы с массивом.
3. Создать анимированного персонажа с использованием массивов.

**Вспомним прошлые занятия**

Вопрос к ученикам:

С помощью какой функции отображать изображения на холсте? (drawImage)

**Обсуждение**

1. Как вы думаете, как работает анимация персонажа?
2. Слышали ли вы когда-нибудь слово спрайт?

Спрайт - это двухмерные картинки в играх. Из них состоят объекты, которые будут двигаться на экране. Спрайты впервые появились в 80-х годах. Они до сих пор используется при создании различных игр, включая браузерные. Современное развлекательное программное обеспечение тоже иногда имеет ее в своем составе.

**Познание нового (10 минут)**

Для того, чтобы создать анимированного персонажа нужно загрузить несколько изображений. Сами изображения нужно где-то хранить. Для этого можно воспользоваться массивами. Разберемся что это такое и как они устроены.

Массив - это переменная, которая может хранить в себе сразу несколько значений, где сами значения записываются в квадратных скобках через запятую.

Пример создания массива

const array = [1, 2, 3, 4, 5, 6, 7, 8, 9]

Но тогда возникает вопрос, а как обращаться к элементам массива?

Для этого используется следующая конструкция: сначала записывается имя массива, потом квадратные скобки и внутри квадратных скобок записывается номер элемента, к которому мы обращаемся.

При этом, индексация (порядковый номер) массива начинается с нуля, то есть, если мы хотим обратиться к 3 в массиве array, то индекс ее будет 2, соответственно

array[2] // 3

**Некомпьютерная активность (5 минут)**

**Задание**

Осмотреть аудиторию и сказать, какие однотипные элементы там есть (столы, стулья и тд); какие бы имена вы использовали при создании массивов (например, стулья - chairs, столы - tables и т.д.)?

**Работа над проектом (10 минут)**

Тема: Создание нового проекта и написание программы, в которой загружаются спрайты для персонажа и создается эффект параллакса.

**Задание**

1. Создать неизменяемый (const) массив из трех элементов и назвать его RED. В созданный массив поместить значения 180, 10, 10. С помощью этого массива задать задний цвет фона для холста.
2. Создать неизменяемый (const) массив с именем PURPLE и задать в него значения 150, 30, 150. С помощью созданного массива PURPLE задать цвет для прямоугольника размерами 200х200 и координатами 200х200.

**Познание нового (25 минут)**

**Загружаем спрайты**

Загрузим все спрайты в отдельный массив runSprites

<script>

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

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

let sprite = new Image();

const runSprites = [

"run\_animation/run-1.png",

"run\_animation/run-2.png",

"run\_animation/run-3.png",

"run\_animation/run-4.png",

"run\_animation/run-5.png",

"run\_animation/run-6.png",

"run\_animation/run-7.png",

"run\_animation/run-8.png"

];

</script>

В качестве элементов массива выступают строки в которых записаны пути к изображениям.

Для смены спрайта создадим переменную **index** (0, 1, 2 и т.д.) и будем изменять ее в функции move до того момента, пока не выйдем за размеры массива (последний индекс - 7). Для отслеживания выхода за размеры массива воспользуемся свойством length, в котором хранится размер массива.

let index = 0;

function move() {

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

index += 1;

if (index > runSprites.length) {

index = 0;

}

sprite.src = runSprites[index];

context.drawImage(sprite, 250, 250);

requestAnimationFrame(move);

}

**Вопросы к ученикам**

1. Как вы думаете, почему персонаж движется очень быстро?
2. Как это можно изменить?

**Ответ**

Кадры сменяются быстро, нужно замедлить этот процесс.

Для избежания быстрой смены кадров, можно добавлять не целое число, а дробное.

Для этого можно использовать функцию **Math.floor()**, которая округляет число вниз, например 5.4 преобразуется в 5 и т.д.

function move() {

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

**index += 0.3;**

if **(index > runSprites.length + 1)** {

index = 0;

}

sprite.src = runSprites[**Math.floor(index)**];

context.drawImage(sprite, 250, 250);

requestAnimationFrame(move);

}

move()

Таким образом происходит смена кадров, ее можно подстраивать по своему усмотрению, изменяя значения добавления к переменной index. При увеличении значения **index** кадры будут сменяться быстрее, при уменьшении - медленнее.

**Работа над проектом (25 минут)**

**Добавляем задний фон и землю**

**Задание**

1. Добавить задний фон и землю (6.4)
2. Заставить смещаться задний фон каждый кадр (6.5)

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

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

let background = new Image();

background.src = "../background.png";

let sprite = new Image();

const runSprites = [

"../run\_animation/run-1.png",

"../run\_animation/run-2.png",

"../run\_animation/run-3.png",

"../run\_animation/run-4.png",

"../run\_animation/run-5.png",

"../run\_animation/run-6.png",

"../run\_animation/run-7.png",

"../run\_animation/run-8.png"

];

let index = 0;

function move() {

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

index += 0.3;

if (index > runSprites.length + 1) {

index = 0;

}

sprite.src = runSprites[Math.floor(index)];

context.drawImage(background, 0, 0, canvas.width, canvas.height);

context.fillRect(0, 600, canvas.width, 100)

context.drawImage(sprite, 250, 550);

requestAnimationFrame(move);

}

move()

**Рефлексия (5 минут)**

"Что нового вы узнали сегодня?"

"Что такое массивы? Как обратиться к элементу массива?"

"Как создать анимацию на холсте используя массивы? Как осуществляется переход от одного спрайта к другому?"