**Урок №16. Подготовка к итоговому проекту**

**Вводный блок (5 минут)**

**Цели и задачи урока**

1. Рассмотреть этапы создания проекта.
2. Обсудить возможные варианты проекта.
3. Объединиться в команды и начать работу над проектом.

На протяжении этого курса мы изучали конструкции языка JavaScript, а также различные игровые механики, к ним относится: перемещение, прыжок, стрельба, коллизия. Помимо этого мы научились создавать графический интерфейс и подключать аудио. Теперь же это все нужно применить на практике самостоятельно и создать свой проект.

**Познание нового (15 минут)**

Цель итогового проекта

Целью итогового проекта является создание игры с использованием любых пройденных механик. Это может быть лабиринт из которого нужно выбраться, игра Dino где динозавриком нужно обходить препятствие, игра Flappy Bird или же змейка, все зависит только от вас.

Для того, чтобы создать проект, необходимо пройти несколько этапов:

1. Планирование проекта.

Перед тем как начать кодировать, важно спланировать ваш проект. Решите, какую игру или приложение вы хотите создать и что ваша разработка будет выполнять. Помимо этого, не забудьте определить основные функции разрабатываемого проекта.

2. Выбор игровых механик.

Помните о том, что мы рассмотрели очень много различных игровых механик и все поместить в один проект не получится. Определитесь с тем, что именно вы хотите видеть в вашем проекте.

3. Разработка прототипа.

Начните с создания прототипа. Можно даже на листе бумаги начать рисовать то, что вы хотите видеть потом в виде реализованной игры.

4. Подумайте над контентом.

Зачастую на этом этапе у многих возникают проблемы. Если вы не можете найти нужное изображение, то ничего страшного, поскольку можно воспользоваться нейросетями (YandexGPT на русском или Leonardo AI) для создания контента. Но код там генерировать точно не нужно, потому что вы можете сами запутаться в том, что там написано.

**Некомпьютерная активность (30 минут)**

**Примеры проектов**

Давайте разберем несколько примеров

1. Игра "Змейка":

Описание: Классическая игра "Змейка", где игрок управляет змейкой, которая постепенно растет, поедая "яблоки" на игровом поле.

Основные функции: Отрисовка змейки и яблок, обработка ввода пользователя (стрелки), логика движения и роста змейки, проверка коллизий, подсчет очков.

Технологии: Canvas API, JavaScript, таймеры, массивы, условные операторы, циклы.

2. Визуализация сортировки:

Описание: Визуализация различных алгоритмов сортировки (например, быстрая сортировка, сортировка слиянием, сортировка пузырьком) с помощью анимации на Canvas.

Основные функции: Генерация случайных массивов, отрисовка столбцов, представляющих элементы массива, анимация процесса сортировки, управление скоростью анимации.

Технологии: Canvas API, JavaScript, массивы, алгоритмы сортировки, таймеры.

3. Игра "Арканоид":

Описание: Классическая игра "Арканоид", где игрок управляет платформой, отбивая мяч, чтобы разбить блоки, расположенные в верхней части экрана.

Основные функции: Отрисовка игрового поля, платформы и мяча, обработка ввода пользователя (движение платформы), логика движения и отражения мяча, определение столкновений с блоками, подсчет очков.

Технологии: Canvas API, JavaScript, таймеры, условные операторы, циклы.

4. Анимированные диаграммы:

Описание: Создание анимированных диаграмм (столбчатые, круговые, линейные) с возможностью изменения данных в реальном времени.

Основные функции: Отрисовка различных типов диаграмм, обновление данных, анимация изменения диаграмм, добавление интерактивных элементов (легенда, подсказки).

Технологии: Canvas API, JavaScript, математические вычисления, обработка событий.

**Обсуждение своего проекта**

Ученики делятся на небольшие группы:

Учитель делит класс на группы по 3-4 человека и дает задание:

Обсудите в группах, какой проект вы хотели бы создать. У вас есть 5 минут, чтобы выбрать идею и определить основные функции вашего проекта.

Каждая группа представляет свои идеи классу:

Учитель вызывает каждую группу по очереди для представления своих идей.

После каждого выступления учитель обсуждает идеи с классом и приводит дополнительные примеры.

**Работа над проектом (30 минут)**

Создайте отдельную папку для работы и назовите её project. Все файлы храните в этой папке, чтобы был доступ к изображениям и аудио-файлам.

Создадите базовую структуру вашей будущей игры:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<canvas width="700px" height="700px" style="border: 5px solid black;">

</canvas>

<script>

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

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

function move() {

requestAnimationFrame(move);

}

move();

</script>

</body>

</html>

Помните о том, что старыми проектами можно и нужно пользоваться, потому что там мы сделали много того, что может помочь сейчас при реализации проекта.

Далее ученики начинают работу над своими проектами.

**Рефлексия (10 минут):**

Сегодня мы начали подготовку к вашему итоговому проекту. Кто может

рассказать, какой проект они будут создавать и какие функции он будет

включать?

Отлично, у вас много интересных идей. Помните, важно тщательно

планировать ваш проект и следовать этому плану.