**Урок №4. Механика прыжка**

**Вводный блок (10 минут)**

**Цели и задачи урока**

1. «Целью нашего занятия является знакомство с игровой механикой «прыжок».
2. Рассмотрим сильно упрощенную модель прыжка.

**Обсуждение**

Вопросы к ученикам:

1. Какие силы действуют на человека, когда он совершает прыжок?
2. Какие физические величины при этом используются?

Итак. Когда персонаж совершает прыжок, в этот момент он отталкивается от земли, тем самым воздействуя на землю и производя по отношению к ней определенную силу.

В момент, когда персонаж уже летит, на него воздействует другая сила - сила притяжения, под действием которой он начинает лететь вниз. Под действием силы притяжения персонаж с каждым мгновением начинает падать все быстрее и быстрее.

**Повторение пройденного материала и работа над проектом**

**(25 минут)**

**Часть №1. Создаем персонажа и землю под ним**

**Давайте вспомним**

1. Как создавать прямоугольник?
2. Как создавать функцию для обновления кадров?

**Ответы**

1) **context.fillRect** - отрисовка прямоугольника, где - первые два значения являются координатами левого верхнего края, а последние два ширина и высота соответственно.

2)

function move() {

x += 1;

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

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

requestAnimationFrame(move);

}

move();

**Задание**

1. Создать квадрат размерами 50х50 на координатах 200х450
2. Создать прямоугольник размерами 700х500 на координатах 50, 50
3. Создать функцию для обновления кадров move

**Проверяем задание**

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

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

let x = 200;

let y = 450;

function move() {

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

context.strokeStyle = "rgb(180,0,0)"

context.strokeRect(0, 500, 700, 50);

context.strokeStyle = "rgb(0,150,150)"

context.strokeRect(x, y, 50, 50);

requestAnimationFrame(move);

}

move()

**Познание нового (25 минут)**

**Часть №2. Создаем силы для реализации прыжка**

На следующем этапе нам нужно определиться со следующими силами:

* сила, которая будет отталкивать созданный квадрат от земли;
* сила, которая будет использоваться для притяжения к земле.

Создадим неизменяемые переменные **jumpPower** и **gravityPower**. Эти переменные будут изменять переменную **jump**, с помощью которой и будет происходить изменение координат по вертикали

**const jumpPower = 10;**

**const gravityPower = 0.3;**

**let jump = 0;**

Создадим событие, отвечающее за нажатие на кнопку прыжка, пускай это будет пробел. Внутри этого события необходимо написать обработку нажатия на пробел и задать **jump** значение переменной **jumpPower**, тем самым давая импульс для квадрата лететь вверх

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

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

**jump = jumpPower;**

**}**

})

После этого мы должны как-то воздействовать на объект с помощью переменной **jump**, поэтому нужно от переменной **y** отнимать значение переменной **jump**

function move() {

**y -= jump;**

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

context.strokeStyle = "rgb(180,0,0)"

context.strokeRect(0, 500, 700, 50);

context.strokeStyle = "rgb(0,150,150)"

context.strokeRect(x, y, 50, 50);

requestAnimationFrame(move);

}

**Вопросы для учеников:**

1. Как вы думаете, почему объект постоянно летит вверх? (Воздействует та сила, которую мы задали и она никак не изменяется)
2. Как это можно изменить? (добавить силу гравитации)

**Давайте порассуждаем (вместе с учениками)**

На земле квадрат находится в том случае, когда координата **y** квадрата равна 450. Если значение по y больше 450, то он находится в воздухе и на него воздействует сила гравитации. Соответственно, мы от переменной **jump** должны отнять силу гравитации **gravityPower.**

Если же координата **y** больше, чем 450 (значит мы находимся на земле), то необходимо переменную **jump** обнулять.

**Некомпьютерная активность (5 минут)**

* Игра «Угадай фигуру». Один ученик рисует на листе бумаги какую-либо фигуру, а другие должны угадать тип фигуры и ее параметры (цвет, размеры и угол поворота), задавая вопросы, на которые первый может отвечать только «да» или «нет». Затем ученики меняются ролями.

**Работа над проектом (20 минут)**

Тема: Создание нового проекта и написание программы, которая реализует механику прыжка.

**Часть №3. Задания**

1. Написать условие для добавления гравитации.
2. Написать условие для обнуления гравитации.

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

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

let x = 200;

let y = 450;

const jumpPower = 10;

const gravityPower = 0.3;

let jump = 0;

function move() {

y -= jump;

if (y < 450) {

jump -= gravityPower;

}

if (y > 450) {

jump = 0;

y = 450;

}

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

context.strokeStyle = "rgb(180,0,0)"

context.strokeRect(0, 500, 700, 50);

context.strokeStyle = "rgb(0,150,150)"

context.strokeRect(x, y, 50, 50);

requestAnimationFrame(move);

}

move()

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

if (event.key == " ") {

jump = jumpPower;

}

})

**Дополнительные задания**

1. Добавить перемещение объекта влево и вправо.
2. В представленном варианте можно на пробел нажимать бесконечно и тогда даже в воздухе будет реализовываться механизм прыжка. Необходимо поставить ограничитель, который позволит прыгать только тогда, когда объект находится на земле.

**Рефлексия (5 минут)**

"Что нового вы узнали сегодня?"

"Как реализуется механика прыжка? Какие переменные мы для этого использовали и для чего они нужны при реализации механики?"