**федеральное государственное автономное образовательное учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 8**

**Дисциплина:** Физическое моделирование компьютерных игр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Тема:** Сила трения\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Выполнил(а): студент(ка) группы \_221-3710\_\_**

\_\_\_\_\_\_Пельт Сергей Александрович\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва2025**

**Сила трения**

В лабораторной работе продемонстрировано действие силы трения на наклонной плоскости и определены условия, при которых объект начинает движение. Два сценария:

1. **Если объект связан с грузом массой m2**:
   * Сила, приложенная от груза: F = m \* g
   * Сила трения: F\_friction = m1 \* g \* (μ \* cos(α) + sin(α))
   * Ускорение системы: a = (F - F\_friction) / (m1 + m)
2. **Если груз отсутствует, но действует внешняя сила F**:
   * Сила трения: F\_friction = μ \* m1 \* g \* cos(α)
   * Проекция веса: F\_gravity = m1 \* g \* sin(α)
   * Необходимое условие для начала движения: F > F\_gravity + F\_friction

Также демонстрируется, как рассчитать минимальную массу груза или необходимую силу, чтобы преодолеть трение и сдвинуть объект.

**Листинг кода:**

void Update()

{

if (move)

{

time += Time.deltaTime;

if (!startedMoving)

{

// Сила, которая прикладывается к телу: F = F0 + A \* t

force = F0 + A \* time;

// Масса груза со временем увеличивается: m = m2 + mx \* t

mass = m2 + mx \* time;

}

if (m2 > 0)

{

// Сила трения по наклонной плоскости:

// F\_friction = m1 \* g \* (μ \* cos(α) + sin(α))

float frictionForce = m1 \* g \* (friction \* Mathf.Cos(angleRad) + Mathf.Sin(angleRad));

// Тяговая сила от груза: F\_drive = m \* g

float drivingForce = mass \* g;

if (drivingForce > frictionForce)

{

obj1.isKinematic = false;

obj2.isKinematic = false;

// Ускорение системы: a = (F\_drive - F\_friction) / (m1 + m)

float acceleration = (drivingForce - frictionForce) / (m1 + mass);

// Направление силы вдоль плоскости

Vector3 forceDirection = Quaternion.Euler(0, 0, angle) \* Vector3.right;

// Приложение ускорения к телу 1 и телу 2

obj1.AddForce(forceDirection \* acceleration, ForceMode.Acceleration);

obj2.AddForce(Vector3.down \* acceleration, ForceMode.Acceleration);

}

else

{

// Масса, необходимая для преодоления трения: m\_required = F\_friction / g

float requiredMass = frictionForce / g;

Debug.Log($"Необходимая масса для старта: {requiredMass:0.0} кг");

}

}

else

{

// Сила трения без груза: F\_friction = μ \* m1 \* g \* cos(α)

float frictionForce = friction \* m1 \* g \* Mathf.Cos(angleRad);

// Сила тяжести вдоль плоскости: F\_gravity = m1 \* g \* sin(α)

float gravityForce = m1 \* g \* Mathf.Sin(angleRad);

// Если внешняя сила превышает сумму силы тяжести и трения — объект начнёт движение

if (force - gravityForce > frictionForce)

{

obj1.isKinematic = false;

Vector3 forceDirection = Quaternion.Euler(0, 0, angle) \* Vector3.right;

// Сила, с которой действуем: F\_result = F - (F\_gravity + F\_friction)

obj1.AddForce(forceDirection \* (force - (gravityForce + frictionForce)));

}

else

{

// Минимально необходимая сила: F\_required = F\_gravity + F\_friction

float requiredForce = frictionForce + gravityForce;

Debug.Log($"Необходимая сила для старта: {requiredForce:0.0} Н");

}

}

// Фиксируем момент начала движения

if (obj1.linearVelocity.magnitude > 0f && !startedMoving)

{

Debug.Log(force);

startedMoving = true;

}

output.UpdateOutput(obj1.linearVelocity.magnitude, force, mass);

}

}

Ссылка на билд: <https://github.com/Sergey-Pelt/Physics>