**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 № 7**

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

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

**Тема:** Моделирование взаимодействия сил. Часть 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**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**

**Моделирование взаимодействия сил. Часть 2**

В лабораторной работе реализовано поведение объекта при столкновении с различными типами препятствий:

* При столкновении с движущимся телом используется расчёт импульса, зависящий от масс и относительной скорости:
  + relativeVelocity = v1 - v2
  + impulse = 2 \* Dot(relativeVelocity, normal) / (1/m1 + 1/m2)
  + v1' = v1 - (impulse / m1) \* normal
  + v2' = v2 + (impulse / m2) \* normal
* При столкновении с неподвижным объектом происходит отражение скорости:
  + v' = Reflect(v, normal)
  + При необходимости скорость увеличивается: v \*= 1 + k

Также сохраняются точки столкновений с помощью LineRenderer.

**Листинг кода:**

void OnCollisionEnter(Collision collision)

{

lr.positionCount++;

// Добавляем в LineRenderer точку на месте столкновения с небольшим смещением вдоль направления предыдущей скорости

lr.SetPosition(lr.positionCount - 1, transform.position + 0.5f \* prevVelocity.normalized);

if (collision.gameObject.CompareTag("Target"))

{

targetReached = true;

move = false;

}

else

{

Vector3 normal = collision.GetContact(0).normal;

normal.z = 0;

normal = normal.normalized;

if (collision.gameObject.CompareTag("Movable"))

{

Rigidbody obstacleRb = collision.collider.GetComponent<Rigidbody>();

// Относительная скорость: v\_rel = v\_object - v\_obstacle

Vector3 relativeVelocity = prevVelocity - obstacleRb.linearVelocity;

// Импульс столкновения: J = 2 \* Dot(v\_rel, n) / (1/m1 + 1/m2)

float impulse = 2 \* Vector3.Dot(relativeVelocity, normal) / (1 / rb.mass + 1 / obstacleRb.mass);

// Обновляем скорость объекта после удара: v1' = v1 - (J / m1) \* n

rb.linearVelocity = prevVelocity -= impulse / rb.mass \* normal;

// Обновляем скорость подвижного препятствия: v2' = v2 + (J / m2) \* n

obstacleRb.linearVelocity += impulse / obstacleRb.mass \* normal;

}

else

{

// Отражаем вектор скорости от поверхности: v' = Reflect(v, n)

rb.linearVelocity = Vector3.Reflect(prevVelocity, normal);

if (collision.transform.TryGetComponent(out Obstacle obstacle))

{

if (obstacle.gainSpeed)

{

// Увеличиваем скорость после столкновения, если указано: v \*= (1 + k)

rb.linearVelocity \*= 1 + speedIncrease;

}

}

// Сохраняем новую скорость

prevVelocity = rb.linearVelocity;

}

}

}

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