**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 № 5**

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

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

**Тема:** Моделирование движения под углом к горизонту\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**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 движения под углом к горизонту**

В лабораторной работе реализовано движение тела под углом к горизонту, с учётом двух видов:

* Свободный полёт с начальной скоростью и ускорением свободного падения;
* Движение с заданным ускорением по осям.

Использованные формулы:

1. **x(t) = vₓ·t** и **y(t) = h + vᵧ·t – 0,5·g·t²** — для движения под углом без дополнительного ускорения;
2. **x(t) = vₓ·t + 0,5·aₓ·t²**, **y(t) = h + 0,5·aᵧ·t²** — для движения с постоянным ускорением;
3. **v(t) = √(vₓ² + vᵧ²)** или **v(t) = √((aₓ·t)² + (aᵧ·t)²)** — скорость как модуль вектора;
4. **s += Δs**, где **Δs = расстояние между точками** — путь;
5. **vₐᵥg = s / t** — средняя скорость.

Модель отражает как баллистическое движение тела, так и движение с постоянным ускорением в любом направлении.

**Листинг кода:**

void Update()

{

if (move)

{

// Увеличиваем общее время симуляции

time += Time.deltaTime;

if (acceleration == 0)

{

// Положение по X при равномерном движении: x = vₓ \* t

pos.x = speedVector.x \* time;

// Положение по Y при вертикальном ускорении (падении): y = h + vᵧ \* t - 0.5 \* g \* t²

pos.y = height + speedVector.y \* time - 0.5f \* g \* time \* time;

// Модуль скорости: √(vₓ² + (vᵧ - g·t)²)

speed = Mathf.Sqrt(Mathf.Pow(speedVector.x, 2) + Mathf.Pow(speedVector.y - g \* time, 2));

}

else

{

// Положение по X с ускорением: x = vₓ \* t + 0.5 \* aₓ \* t²

pos.x = speedVector.x \* time + 0.5f \* accelerationVector.x \* time \* time;

// Положение по Y с ускорением: y = h + 0.5 \* aᵧ \* t² (начальная скорость по y считается нулевой)

pos.y = height + 0.5f \* accelerationVector.y \* time \* time;

// Модуль скорости: √((aₓ \* t)² + (aᵧ \* t)²)

speed = Mathf.Sqrt(Mathf.Pow(accelerationVector.x \* time, 2) + Mathf.Pow(accelerationVector.y \* time, 2));

}

if (pos.y <= 0)

{

pos.y = 0;

move = false;

ButtonChange.ChangeSprite(false);

}

// Пройденное расстояние: накапливаем расстояние между кадрами

distance += Vector2.Distance(lastPos, pos);

// Средняя скорость: s / t

avgSpeed = distance / time;

lastPos = pos;

transform.position = pos;

output.UpdateOutput(pos.x, speed, avgSpeed, distance, time);

}

}

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