Project Report

**Topic Name:** Physics of Platformer Game Mechanics: Sonic-style Springs with Variable Mass

**Student Name:** Dylan Parks

**Date:** April 25, 2025



### **Table of Contents**

1. Introduction
2. Physics Principles
   * Hooke’s Law and Springs
   * Newton’s Second Law and Variable Mass
3. Implementation
   * Collision Detection
   * Force and Velocity Calculations
4. Sample Scenarios
   * Light, Medium, and Heavy Character Trajectories
5. Conclusion
6. References

### **Table of Figures**

1. Figure 1: Spring force diagram
2. Figure 2: Screenshot of the prototype
3. Figure 3: Velocity vectors for different masses

### **Introduction:**

This project explores the physics behind platformer game mechanics, specifically focusing on Sonic-style spring bumpers with variable-mass characters. The rationale for choosing this topic stems from the need to demonstrate how fundamental physics principles like Hooke’s Law (F = -kx) and Newton’s Second Law (F = ma) govern dynamic interactions in games. By simulating springs with characters of differing masses, we can visualize how mass affects acceleration, bounce height, and trajectory—key concepts for realistic game physics.  
  
The scope includes developing a 2D prototype in MonoGame, featuring:

* + Three characters with distinct masses (light, medium, heavy).
  + Vertical and horizontal springs with force indicators.
  + Real-time velocity and collision feedback.
  + The deliverables are a functional executable, commented source code, and a demonstration video.

### **Physics Principles:**

Hooke’s Law and Springs

Hooke’s Law states that the force exerted by a spring is proportional to its displacement (\*x\*) from equilibrium, with a spring constant (\*k\*) determining stiffness:

F= −*kx*

In the prototype:

* + \*k\* is fixed per spring (e.g., 500 N/m for vertical springs).
  + Displacement (\*x\*) depends on collision depth (e.g., how far the character compresses the spring).

Newton’s Second Law and Variable Mass:

Newton’s Second Law relates force, mass (\*m\*), and acceleration (\*a\*):

For the three characters:

1. **Light (m = 30 kg):** High acceleration, lower momentum.
2. **Medium (m = 50 kg):** Balanced movement.
3. **Heavy (m = 80 kg):** Slower acceleration but greater momentum.

## 

### **Implementation:**

### Collision Detection

The prototype uses axis-aligned bounding boxes (AABB) for spring-character collisions. Key steps:

1. Check the intersection between the character and spring bounds.
2. Calculate compression depth (\*x\*) during collision.
3. Apply Hooke’s Law to determine force magnitude.

### Force and Velocity Calculations

* **Vertical Springs:**

Resulting velocity:

* **Horizontal Springs:**

Velocity is clamped to a maximum boost (e.g., 300 units/sec).

### **Sample Scenarios:**

### Scenario 1: Light Character (30 kg)

* **Observation:** High bounce height (~600 units) due to low mass.
* **Physics:** Greater acceleration from the same spring force.

### Scenario 2: Heavy Character (80 kg)

* **Observation:** Shorter bounce (~300 units) but longer horizontal travel.
* **Physics:** Lower acceleration but higher momentum.

**Figure 3** compares trajectories.

### **Conclusion**

This project successfully demonstrates how mass and spring physics create varied gameplay mechanics. Key takeaways:

1. Hooke’s Law and Newton’s Second Law are foundational for realistic interactions.
2. Variable mass adds depth to game design (e.g., character abilities).
3. The prototype could be extended with dampening or multi-spring systems.

For PHYS1521, this topic bridges theory (forces, motion) and practical game development, encouraging students to experiment with physics parameters.

### **References:**

1. Serway, R. A., & Jewett, J. W. (2018). *Physics for Scientists and Engineers*. Cengage.
2. Gregory, J. (2018). *Game Engine Architecture*. CRC Press.
3. MonoGame Documentation. (2023). <https://docs.monogame.net/>