Semesterprojekt Modul PE FS 2024

Authoren

Dario Brunner,
ZHAW Student School of Engineering, Computer Science BSc,
brunndar@students.zhaw.ch

Timo Sigrist,
ZHAW Student School of Engineering, Computer Science BSc,
sigritim@students.zhaw.ch

Dozent
Kurt Pernstich,
pern@zhaw.ch

Zusammenfassung

Im Kurs ”Physik Engines (PE) FS24” an der ZHAW werden mittels physikalischer Simulationen die Wechselwirkungen zwischen zurückgelegter Strecke, aktueller Geschwindigkeit, Beschleunigung und den auf Körper wirkenden Kräften untersucht. Diese Simulationen werden mithilfe von Unity durchgeführt, wobei grafische Darstellungen zur Veranschaulichung der Ergebnisse dienen.
Der Bericht ist in zwei Experimente unterteilt. Im ersten Experiment wird eine Feder von einem Würfel entfernt, woraufhin dieser einmal hin und her pendelt. Die Kraft wirkt zunächst in eine Richtung, bis die Feder zusammengedrückt ist, danach wirkt sie in entgegengesetzter Richtung. Sobald die Feder vollständig ausgedehnt ist, wird sie vom Würfel gelöst. Anschließend wird der Würfel durch Windkraft bewegt, bis er auf eine zweite Feder trifft. Diese zweite Feder befindet sich zwischen unserem bewegten Würfel und einem zweiten Würfel, der doppelt so schwer ist. Der bewegte Würfel stösst auf die zweite Feder, wodurch direkt eine Kraft auf den zweiten Würfel ausgeübt wird. Nach dem Zusammenstoss gleiten die beiden Würfel ohne Einwirkung von Windkraft auseinander. Wärend des gesamten Experiemtes erfahrt keiner der beiden Würfel eine Reibungskraft des Untergrunds.

April 23, 2024

Contents

1 Aufbau des Experiments

1.1 Teil 2: Beschleunigung und elastischer Stoss

Die nachfolgender Auflistung bezieht sich auf Figure 1. Im ganzen Experiment wird die Reibungskraft vernachlässigt.

1. Der Würfel 1 (blauer Würfel) wird wie im harmonischen Oszillator von Aufgabe 1 eine Federkraft in Bewegung gesetztz. Die Kraft der Feder 1 (rechte grüne Feder) soll genug gross sein, dass der blaue Würfel nach wenigen Sekunden eine Geschwindigkeit von 1 m/s erreicht wird. Der blaue Würfel hat eine Masse von 1 kg und eine Seitenlänge von 1 m.

2. Die Feder 1 soll ausgedehnt werden und somit den Würfel in die entgegengesetzte Richtung beschleunigen. Die Federkonstante und Länge der Feder 1 soll passend bestimmt werden.

3. Die Feder wird vom Würfel 1 entfernt. Er gleitet in Richtung des Würfel 2. Dabei wird er von einer Windkonstante angestossen, bis er mit der Feder 2 (linke grüne Feder) zusammenstosst.

4. Beim Zusammenstoss von Würfel 1 auf Feder 2 wirkt ein Impuls auf die Feder 2. Der Würfel 2 (roter Würfel) hat eine Masse von 2 kg und eine Seitenlänge von 1 m. Die Federkonstante und die Länge der Feder 2 soll so bestimmt werden, dass sich die beiden Würfel nicht berühren und der elastische Stoss ca. 1 Sekunde andauert.

5. Der Impuls der Feder 2 wirkt Energie auf die beiden Würfel aus. Somit gleiten sie in entgegengesetzer Richtung voneinander weg. Ab Beginn des Stosses wirkt keine Windkraft mehr.

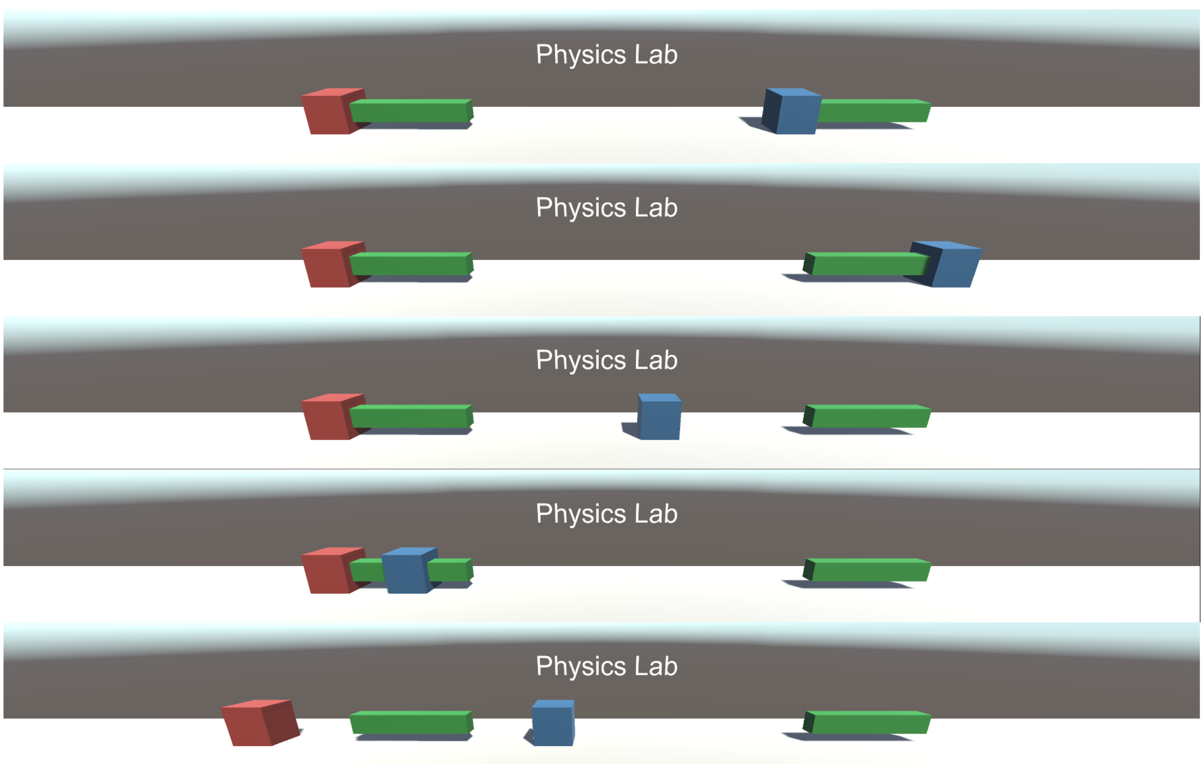

Figure 1: Verschiedene Schritte vom Experiment

Der blaue Würfel wird durch die rechte grüne Feder in Schwung gebracht. Er wird durch eine Windkonstante beschleunigt, bis er mit der zweiten linken Feder zusammenstösst. Der Zusammenstoss mit der Feder bewirkt, dass sich der rote Würfel und der blaue Würfel voneinander entfernen.

1.2 Teil 3: TBD

TBD.

2 Physikalische Beschreibung der einzelnen Vorgänge

2.1 Teil 2

Dieses Teil des Experiments untersucht die Bewegung von zwei Würfeln und deren Stösse. Es gliedert sich in fünf physikalische Teilbewegungen:

1. Würfel 1 oszilliert (Federkraft von Feder 1 wirkt auf Würfel 1)
2. Würfel 1 wird durch eine Windkonstante angestossen
3. Erste Stossphase vom Elastischen Stoss (Würfel 1 trifft auf Feder 2)
4. Zweite Sossphase vom Elastischen Stoss (Impuls der gespeicherte Energie der Knautschzone fliesst)

In den Teilbewegungen 1–4 gleiten die Würfel reibungslos auf der Oberfläche.

2.1.1 Oszillieren von Würfel 1

Der Würfel 1 (blauer Würfel) wird durche eine Feder ins oszillieren gebracht. Zu Beginn wird der Würfel durch die bereits gespannte Feder in die gegengesetzte Richtung des zweiten Würfels (roter Würfel) beschleunigt. Sobald sich die Feder in der Ruhelage befindet, wird der Würfel 1 durch die Federkraft in die Richtung des Würfel 2 beschleunigt. Nachdem der Würfel die Position der Feder wieder erreicht, wird die Feder entfernt.
Die Federkraft kann wiefolgt berechnet werden:

$$F = k_{\text{Feder}} * \Delta l \tag{1}$$

Equation (1): F Kraft in Newton $[N]$, k_{Feder} Federkonstatnte $[\frac{N}{m}]$, Δl Länge der Feder $[m]$.

Wobei die Änderung der Federlänge Δl wiefolgt definiert ist:

$$\Delta l = l - l_0 \tag{2}$$

Equation (2): Δl Änderung der Federlänge $[m]$, l aktuelle Position $[m]$, l_0 Startposition der Felder $[m]$.

2.1.2 Windkonstante

Der Würfel 1 (blauer Würfel) wird durch eine Windkraft beschleunigt, bis er die Feder des zweiten Würfels (roter Würfel) berührt. Diese Windkraft wird durch eine konstante Krafteinwirkung auf den Würfel 1 umgesetzt.
Die Beschleunigung kann durch foglende Formel berechnet werden:

$$F = ma \tag{3}$$

Equation (3): F Kraft in Newton $[N]$, m Masse $[kg]$, a Beschleunigung $[\frac{m}{s^2}]$.

2.1.3 Erste Stossphase vom Elastischen Stoss

Zu Beginn der ersten Stossphase ist die Feder 2 in der Ruhelage. Der Würfel 1 (blauer Würfel) triff auf die Feder und verformt diese bis an den Kompressionspunkt. Dabei baut sich potentielle Energie in der Knautschzone auf. Um diese Energie auszurechnen muss jedoch zuerst die kinetische Energie des Würfels berechnet werden.
Mit folgender Formel kann die kinetische Energie des Würfel 1 berechnet werden:

$$E_{kin} = \frac{1}{2}mv^2 \tag{4}$$

Equation (4): E_{kin} kinetische Energie des Würfels $[J]$, m Masse des Würfels $[kg]$, v Geschwindigkeit des Würfels $[\frac{m}{s}]$.

Anschliessend wird beim Aufprall die kinetische Energie des Würfels in potentielle Energie der Feder durch einen Impuls übertragen. Da die Geschwindigkeit des Würfel 2 0 beträgt, kann dieser Term weggelassen werden:

$$\vec{p_{in}} = m_{\text{Würfel1}} * \vec{v_{\text{Würfel1}}} + m_{\text{Würfel2}} \tag{5}$$

Equation (5): $\vec{p}$ Impuls-Vektor der in die Feder fliesst [kg $*$ m/s], m Masse des Würfels $[kg]$, $\vec{v}$ Geschwindigkeits-Vektor des Würfels $[\frac{m}{s}]$.

Für Objekte, die elastisch verformbar sind (wie Federn), kann die potentielle Energie durch Hooke'sches Gesetz beschrieben werden:

$$E_{pot} = \frac{1}{2} * k_{\text{Feder}} * x^2 \tag{6}$$

Equation (6): E_{pot} potentielle Energie in Feder $[J]$, k_{Feder} Federkonstante $[\frac{N}{m}]$, x Auslenkung der Feder [m].

2.1.4 Zweite Stossphase vom Elastischen Stoss

Zu Beginn der zweiten Stossphase befindet sich die Feder im Kompressionspunkt. Der Impulserhaltungssatz besagt, dass der Gesamtimpuls eines abgeschlossenen Systems vor und nach einem Zusammenstoss gleich bleibt, daher stosst die Feder die beiden Objekte, mit der in der ersten Stossphase aufgebauten potentiellen Energie, durch einen Impuls auseinander:

$$E_{pot} = m_{\text{Würfel1}} * (v_{\text{Würfel1}})^2 + m_{\text{Würfel2}} * (v_{\text{Würfel2}})^2 \tag{7}$$

Equation (7): E_{pot} potentielle Energie in Feder $[J]$, m Masse des Würfels $[kg]$, v Geschwindigkeit des Würfels $[\frac{m}{s}]$.

Bei der Berechnung der jeweiligen kinetischen Energie des Würfels muss die Ungleichverteilung der Massen der Würfel berücksichtigt werden. Der Würfel 2 (roter Würfel) hat die doppelte Masse des Würfel 1 (blauer Würfel). Dazu kann folgende Formel benutzt werden:

$$E_{\text{kin Würfel1}} = \frac{m_{\text{Würfel1}}}{m_{\text{Würfel1}} + m_{\text{Würfel2}}} * E_{\text{pot}} \tag{8}$$

und

$$E_{\text{kin Würfel2}} = \frac{m_{\text{Würfel2}}}{m_{\text{Würfel1}} + m_{\text{Würfel2}}} * E_{\text{pot}} \tag{9}$$

Equation (8) und (9): $E_{\text{kin Würfel}}$ Kinetische Energie der Würfel $[J]$, m Masse des Würfels $[m]$, E_{pot} potentielle Energie in Feder $[J]$.

Dadruch lässt sich das Engergieerhaltungsgesetz darstellen:

$$E_{pot} = E_{\text{kin Würfel 1}} + E_{\text{kin Würfel 2}} \tag{10}$$

Equation (10): E_{pot} Potentielle Energie in Feder $[J]$, Kinetische Enerige in Würfel $[J]$.

Da es sich um einen elatsischen Stoss handelt, werden die Würfel voneinandern weggestossen.
Die jeweilige Krafteinwirkung auf den Würfel kann mitfolgender Formel berechnet werden:

$$F = \frac{m * v}{\Delta t} \tag{11}$$

Equation (11): F Kraft in Newton $[N]$, m Masse $[kg]$, v Geschwindigkeit des Würfels $[\frac{m}{s}]$, Δt Zeit in Sekunden $[s]$.

2.2 Teil 3

2.2.1 TBD

3 Implementierung

3.1 Teil 2

In diesem Abschnitt zeigen wir die Implementierung mit Screenshots der Simulation aus Unity.

3.1.1 Oszillieren von Würfel 1

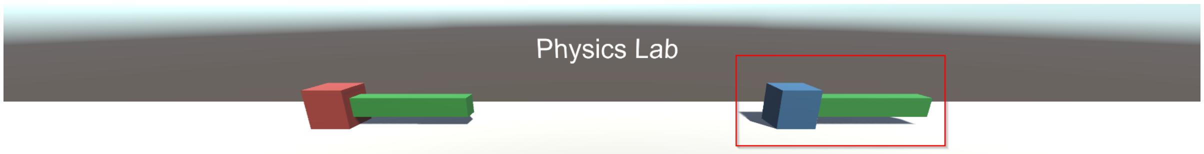

Figure 2: Oszillieren von Würfel 1

Die Anfangsposition vom Würfel 1 (blauer Würfel) wurde auf x = 5 festgelegt, die Masse beträgt m = 1 kg. Die Feder 1 hat ihren Mittelpunkt bei x = 7 und eine Federkonstante von 1 N. Die erforderliche Federkraft kann wie folgt berechnet werden:

$$F = k_{\text{Feder}} * \Delta l \tag{12a}$$

Equation (12): F Kraft in Newton $[N]$, k_{Feder} Federkonstatnte $[\frac{N}{m}]$, Δl Länge der Feder $[m]$.

Die berechnete Kraft mit k_{Feder} = 1 N und Δl = -2 m ist somit F = -2.

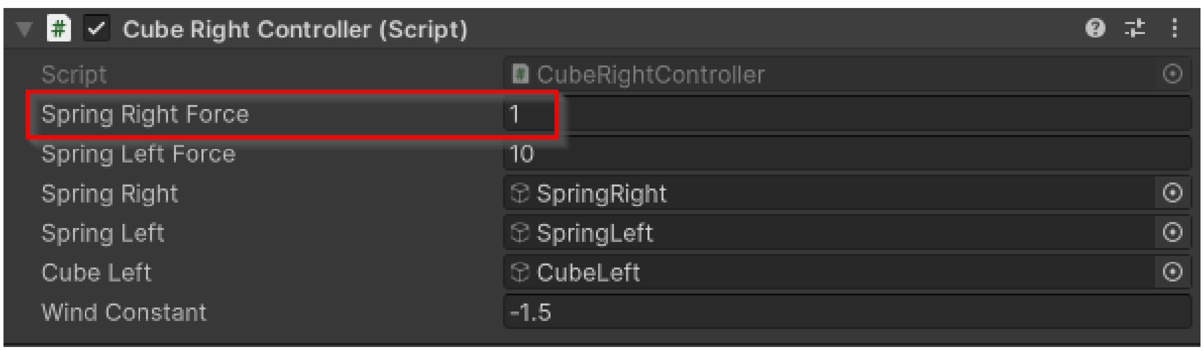

Figure 3: Federkonstante Feder 1

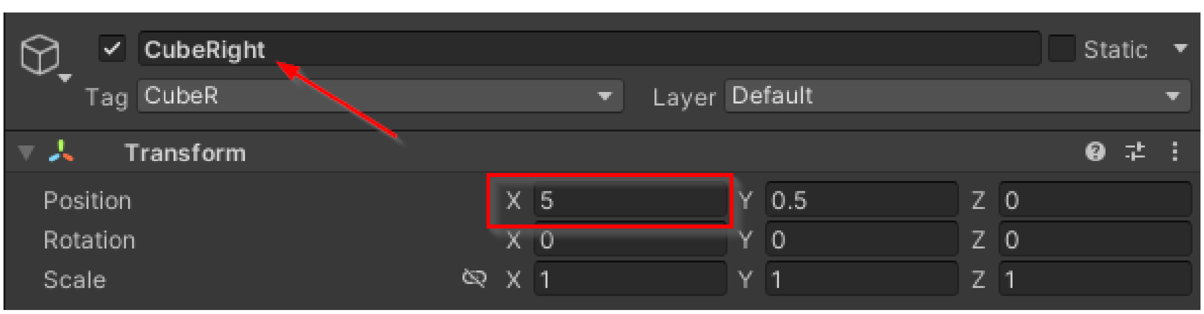

Figure 4: Anfangsposition Würfel 1

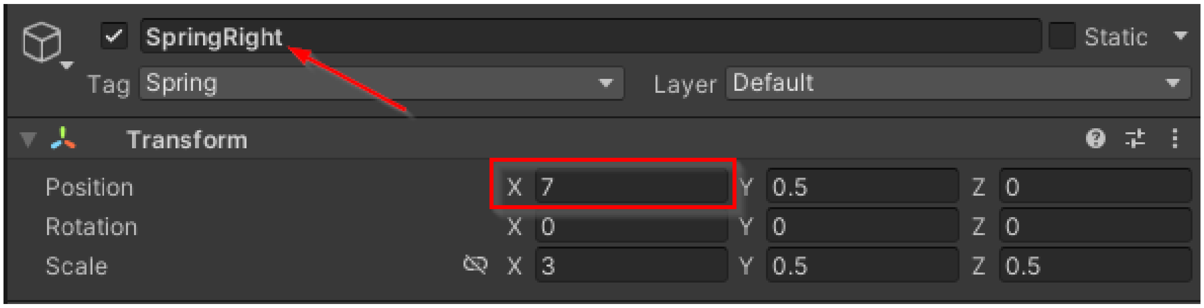

Figure 5: Anfangsposition Feder 1

3.1.2 Windkonstante

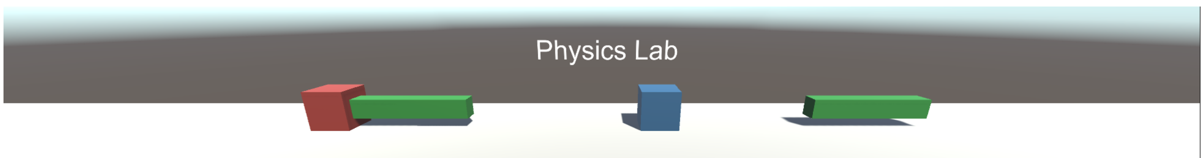

Figure 6: Windkonstante wird auf Würfel 1 angewendet

Wir entfernen die Feder 1 vom Würfel, sodass dieser nicht mehr oszilliert. Gleichzeitig setzt ein Wind ein, der den Würfel 1 beschleunigt. Der Würfel 1 hat eine Masse m = 1 kg und die Windkonstante beträgt F = -1.5 N. Der Würfel gleitet reibungsfrei. Die Beschleunigung kann durch foglende Formel berechnet werden:

$$F = ma \tag{13}$$

Equation (13): F Kraft in Newton $[N]$, m Masse $[kg]$, a Beschleunigung $[\frac{m}{s^2}]$.

Somit ist die Maximalbeschleunigung a = -1.5 m/s^2.

Figure 7: Masse vom Würfel 1

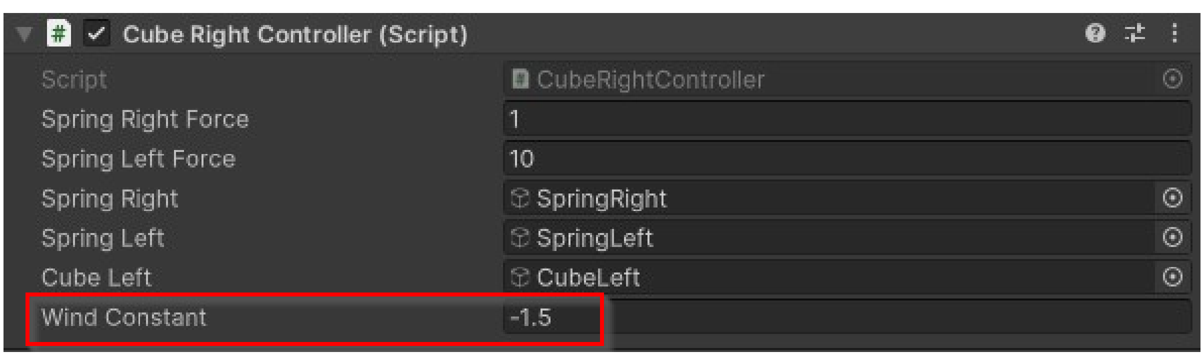

Figure 8: Windkonstante

3.1.3 Erste Stossphase vom Elastischen Stoss

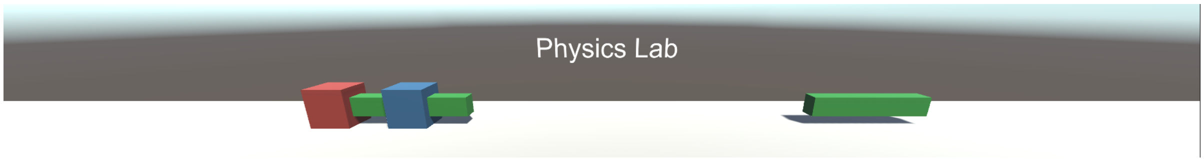

Figure 9: Würfel 1 stosst auf Feder 2

In dieser Phase wird Energie in der Knautschzone gespeichert. Durch integrieren der Beschleunigung vom Würfel 1 stosst dieser elastisch mit einer Maximalgeschwindigkeit von -5.929 m/s auf die Feder 2. Die Feder 2 hat eine Startposition von x = -7. Der Wind stoppt, sobald der Stoss beginnt. Somit kann die kinetische Energie des Würfel 1 berechnet werden:

$$E_{kin} = \frac{1}{2}mv^2 \tag{14}$$

Equation (14): E_{kin} kinetische Energie des Würfels $[J]$, m Masse des Würfels $[kg]$, v Geschwindigkeit des Würfels $[\frac{m}{s}]$.

Somit haben wir eine kinetische Energie vom Würfel 1 von $E_{kin} = 17.578\ J$ beim Zusammenstoss mit der Feder.

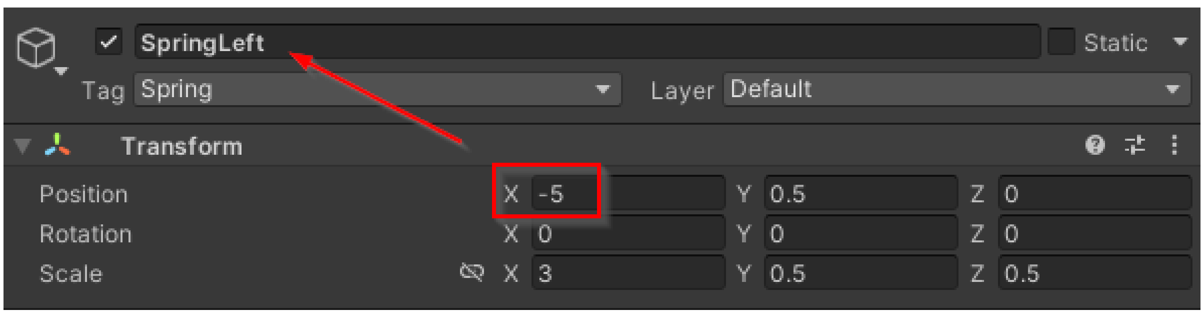

Figure 10: Position Feder 2

Die Federkonstante beträgt 10 N/m und die ausgedehnte Auslennkung beträgt x = 3 m. Für die Feder 1 kann die potentielle Energie durch Hooke'sches Gesetz berechnet werden:

$$E_{pot} = \frac{1}{2} * k_{\text{Feder}} * x^2 \tag{15}$$

Equation (15): E_{pot} potentielle Energie in Feder $[J]$, k_{Feder} Federkonstante $[\frac{N}{m}]$, x Auslenkung der Feder $[m]$.

Damit können wir die maximale Energie berechnen, welche die Feder 2 aufnehmen kann. Wir können mit der kinetischen Energie vom Würfel 1 auch die Auslenkung der Feder berechnen.Angenommen die Feder 2 würdekomplett zusammengedrückt, so ergäb das eine maximale potentielle Energie von $Epot = 45\ J$. Mit der vorherigen Berechnung stösst der Würfel 1 mit $E_{kin} = 17.578\ J$ auf die Feder, somit wird die Feder um a = 1.874 m gestaucht.

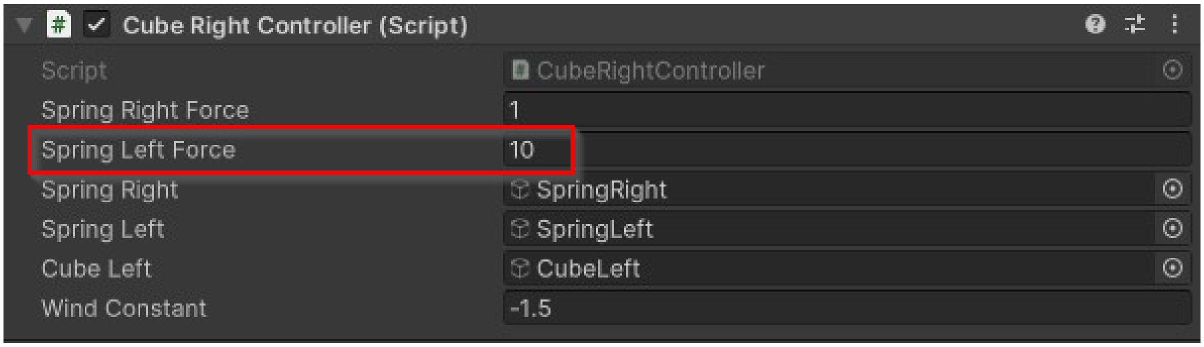

Figure 11: Federkonstante Feder 2

3.1.4 Zweite Stossphase vom Elastischen Stoss

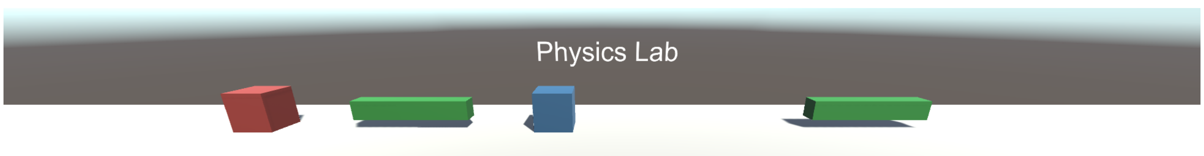

Figure 12: Energie fliesst als Impuls auf beide Würfel

In der zweiten Stossphase sorgt die gespeicherte potentielle Energie dafür, dass noch einmal so viel Impuls fliesst, wie in der ersten Stossphase geflossen ist. Der Würfel 1 wirkt mit einer Energie von $E_{kin} = 17.578\ J$ und der Würfel 2 mit $E_{kin} = 0\ J$, da dieser sich nicht bewegt. Die aufzuteilende potentielle Energie ergibt sich aus folgender Gleichung:

$$E_{pot} = m_{\text{Würfel1}} * (v_{\text{Würfel1}})^2 + m_{\text{Würfel2}} * (v_{\text{Würfel2}})^2 \tag{16}$$

Equation (16): E_{pot} potentielle Energie in Feder $[J]$, m Masse des Würfels $[kg]$, v Geschwindigkeit des Würfels $[\frac{m}{s}]$.

Also erhalten wir eine potentielle Energie von $E_{kin} = 17.578\ J$, welche wir nun auf die beiden Würfel aufteilen müssen. Dafür verwenden wir folgende Formel:

$$E_{\text{kin Würfel1}} = \frac{m_{\text{Würfel1}}}{m_{\text{Würfel1}} + m_{\text{Würfel2}}} * E_{\text{pot}} \tag{17}$$

und

$$E_{\text{kin Würfel2}} = \frac{m_{\text{Würfel2}}}{m_{\text{Würfel1}} + m_{\text{Würfel2}}} * E_{\text{pot}} \tag{18}$$

Equation (17) und (18): $E_{\text{kin Würfel}}$ Kinetische Energie des Würfels $[J]$, m Masse des Würfels $[m]$, E_{pot} potentielle Energie Feder $[J]$.

Somit erhaltet Würfel 1 $E_{kin} = 5.859\ J$ und Würfel 2 $E_{kin} = 11.718\ J$ als Impuls in der zweiten Stossphase.

3.2 Teil 3

3.2.1 TBD

4 Resultat

Die Figure 13 veranschaulicht die Geschwindigkeit der Würfel in den drei Phasen von Teil 2. Der Würfel 1 oszilliert zwischen x = 2 und x = -2 bis wir die Feder entfernen. Danach wirkt die Windkonstante. Ersichtlich im Zeitintervall von 3 s bis ca. 7.5 s. Die dritte Phase beeinträchtigt beide Würfel. Sie werden mit einer konstanten Geschiwindigkeit voneinander weggestosst. Die beiden Würfel gleiten danach ohne Reibung.

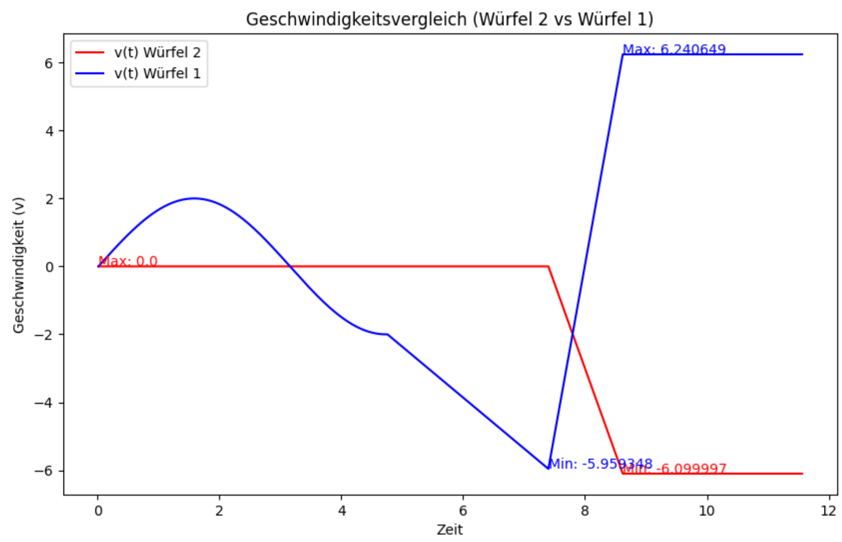

Figure 13: Geschwindigkeitsdiagramm der beiden Würfel

Die Position der Würfel wird in der Figure 14 veranschaulicht. Der Würfel 1 oszilliert von der Feder 1 bis wir die Feder entfernen und die Windkonstante greifft. Danach wird bis zum Zusammenstoss die Position vom Würfel 1 immer kleiner. Beim Zusammenstoss gehen die beiden Würfel auseinander.

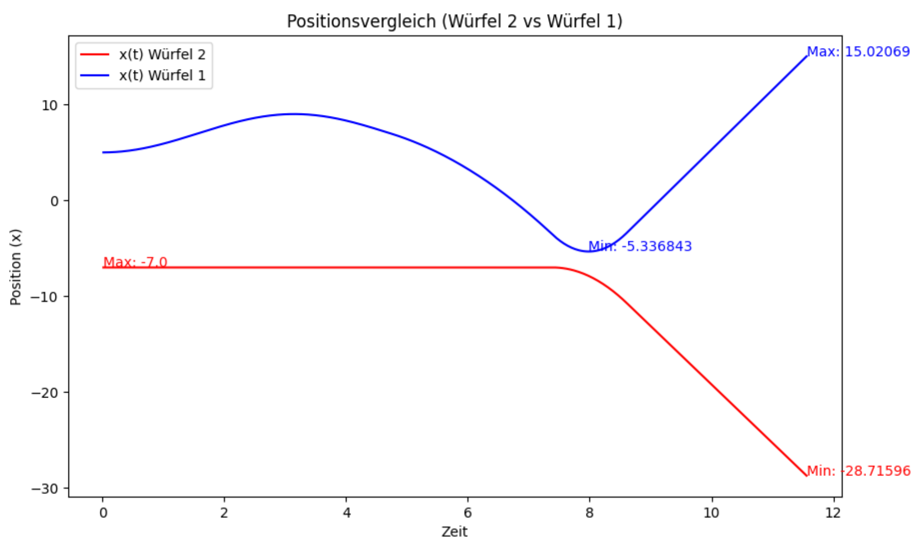

Figure 14: Positionsvergleich der beiden Würfel

Die Figure 15 veranschaulicht die Kraft der Würfel in den drei Phasen der Aufgabe. Die Kraft vom Würfel 1 oszilliert bis wir die Feder entfernen. Die Windkraft wirkt direkt und konstant auf den Würfel 1. Bis da hin wirkt keine Kraft auf den Würfel 2. Beim Zusammenstoss wirkt eine Kraft auf beide Würfel.

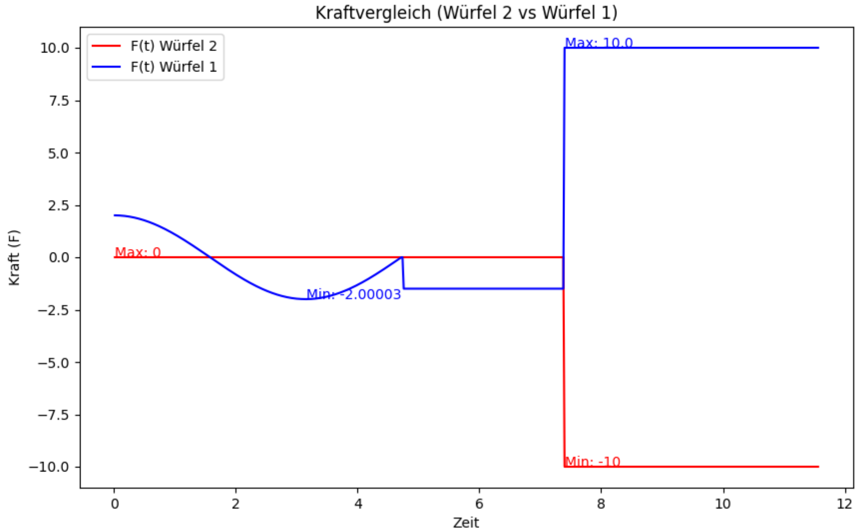

Figure 15: Kräftevergleich der beiden Würfel

5 Rückblick

TBD

List of Figures

6 Appendix

List of Listings

A Source codes

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

using System.IO;
using UnityEditor;
using UnityEngine.SceneManagement;
using System;
using UnityEngine.Assertions.Must;

/*
    Accelerates the cube to which it is attached, modelling an harmonic oscillator.
    Writes the position, velocity and acceleration of the cube to a CSV file.

    Remark: For use in "Physics Engines" module at ZHAW, part of physics lab
    Author: kemf
    Version: 1.0
*/
public class CubeRightController : MonoBehaviour
{
    private Rigidbody rigidBody;

    public int springRightForce;
    public int springLeftForce;
    public GameObject springRight;
    public GameObject springLeft;
    public GameObject cubeLeft;
    public float windConstant;
    public bool oscillate;


    private float currentTimeStep;
    private List<List<float>> timeSeries;
    private Phase phase = Phase.oscillator;
    private float cubeRightForceX;
    private float cubeLeftForceX;
    private float springLeftFurthestPointRight;

    enum Phase
    { oscillator, wind, collision, none }

    // Start is called before the first frame update
    void Start()
    {
        rigidBody = GetComponent<Rigidbody>();
        timeSeries = new List<List<float>>();
        timeSeries = new List<List<float>>();
        springLeftFurthestPointRight = springLeft.transform.position.x +
            GetSpringLeftWidth() / 2f;
    }

    // FixedUpdate can be called multiple times per frame
    void FixedUpdate()
    {
```

```
        currentTimeStep += Time.deltaTime;
        switch(phase)
        {
            case Phase.oscillator:
                var deltaL = rigidBody.position.x - springRight.transform.position.x;
                cubeRightForceX = -deltaL * springRightForce;

                AddForceToRigidBody(cubeRightForceX);

                //Cube comes back from spring and now only the wind carries the cube
                if (rigidBody.position.x < springRight.transform.position.x && rigidBody
                    .velocity.x < 0 && !oscillate) {
                    cubeRightForceX = 0;
                    phase = Phase.wind;
                }

                break;

            case Phase.wind:
                cubeRightForceX = windConstant;
                AddForceToRigidBody(cubeRightForceX);

                if(rigidBody.position.x <= springLeftFurthestPointRight)
                {
                    phase = Phase.collision;
                }

                break;

            case Phase.collision:
                cubeRightForceX = springLeftForce;
                AddForceToRigidBody(cubeRightForceX);

                cubeLeftForceX = -cubeRightForceX;
                cubeLeft.GetComponent<Rigidbody>().AddForce(new Vector3(cubeLeftForceX,
                    0f, 0f));

                if (rigidBody.position.x >= springLeftFurthestPointRight)
                {
                    phase = Phase.none;
                }
                break;

            case Phase.none:
                if (rigidBody.position.x >= 15) {
                    UnityEditor.EditorApplication.isPlaying = false;
                }
                break;

        }
        timeSeries.Add(new List<float>() { currentTimeStep, cubeLeft.GetComponent<
            Rigidbody>().position.x, cubeLeft.GetComponent<Rigidbody>().velocity.x,
            cubeLeftForceX, rigidBody.position.x, rigidBody.velocity.x, cubeRightForceX
            });
    }
```

```
    private void AddForceToRigidBody(float cubeRightForceX) {
        rigidBody.AddForce(new Vector3(cubeRightForceX, 0f, 0f));
    }

    // Helper function to get the width of springLeft
    private float GetSpringLeftWidth()
    {
        Renderer renderer = springLeft.GetComponent<Renderer>();
        if (renderer != null)
        {
            return renderer.bounds.size.x;
        }
        else
        {
            Debug.LogError("Renderer component not found on the object.");
            return 0f;
        }
    }

    void OnApplicationQuit() {
        WriteTimeSeriesToCSV();
    }

    void WriteTimeSeriesToCSV() {
        using (var streamWriter = new StreamWriter("data.csv")) {
            streamWriter.WriteLine("t,x(t) cubeL,v(t) cubeL,F(t) cubeL,x(t) cubeR,v(t)
                cubeR,F(t) cubeR");

            foreach (List<float> timeStep in timeSeries) {
                streamWriter.WriteLine(string.Join(",", timeStep));
                streamWriter.Flush();
            }
        }
    }
}
```

Listing 1: CubeRightController.cs

```python
import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file
df = pd.read_csv('../data.csv')

# Extract data from columns
t = df['t']
x_left = df['x(t) cubeL']
v_left = df['v(t) cubeL']
F_left = df['F(t) cubeL']
x_right = df['x(t) cubeR']
v_right = df['v(t) cubeR']
F_right = df['F(t) cubeR']

# Calculate min and max points
x_left_min, x_left_max = min(x_left), max(x_left)
v_left_min, v_left_max = min(v_left), max(v_left)
F_left_min, F_left_max = min(F_left), max(F_left)
x_right_min, x_right_max = min(x_right), max(x_right)
v_right_min, v_right_max = min(v_right), max(v_right)
F_right_min, F_right_max = min(F_right), max(F_right)

# Plotting x(t)
plt.figure(figsize=(10, 6))

plt.plot(t, x_left, label='x(t) W rfel 2', color='red')
plt.plot(t, x_right, label='x(t) W rfel 1', color='blue')

plt.xlabel('Zeit')
plt.ylabel('Position (x)')
plt.title('Positionsvergleich (W rfel 2 vs W rfel 1)')
plt.legend()

plt.text(t.iloc[x_left.idxmin()], x_left_min, f'Min: {x_left_min}', color='red')
plt.text(t.iloc[x_left.idxmax()], x_left_max, f'Max: {x_left_max}', color='red')
plt.text(t.iloc[x_right.idxmin()], x_right_min, f'Min: {x_right_min}', color='blue')
plt.text(t.iloc[x_right.idxmax()], x_right_max, f'Max: {x_right_max}', color='blue')

plt.savefig('position_comparison.png')
plt.close()

# Plotting v(t)
plt.figure(figsize=(10, 6))

plt.plot(t, v_left, label='v(t) W rfel 2', color='red')
plt.plot(t, v_right, label='v(t) W rfel 1', color='blue')

plt.xlabel('Zeit')
plt.ylabel('Geschwindigkeit (v)')
plt.title('Geschwindigkeitsvergleich (W rfel 2 vs W rfel 1)')
plt.legend()

plt.text(t.iloc[v_left.idxmin()], v_left_min, f'Min: {v_left_min}', color='red')
plt.text(t.iloc[v_left.idxmax()], v_left_max, f'Max: {v_left_max}', color='red')
plt.text(t.iloc[v_right.idxmin()], v_right_min, f'Min: {v_right_min}', color='blue')
```

```python
plt.text(t.iloc[v_right.idxmax()], v_right_max, f'Max: {v_right_max}', color='blue')

plt.savefig('velocity_comparison.png')
plt.close()

# Plotting F(t)
plt.figure(figsize=(10, 6))

plt.plot(t, F_left, label='F(t) W rfel 2', color='red')
plt.plot(t, F_right, label='F(t) W rfel 1', color='blue')

plt.xlabel('Zeit')
plt.ylabel('Kraft (F)')
plt.title('Kraftvergleich (W rfel 2 vs W rfel 1)')
plt.legend()

plt.text(t.iloc[F_left.idxmin()], F_left_min, f'Min: {F_left_min}', color='red')
plt.text(t.iloc[F_left.idxmax()], F_left_max, f'Max: {F_left_max}', color='red')
plt.text(t.iloc[F_right.idxmin()], F_right_min, f'Min: {F_right_min}', color='blue')
plt.text(t.iloc[F_right.idxmax()], F_right_max, f'Max: {F_right_max}', color='blue')

plt.savefig('force_comparison.png')
plt.close()
```

Listing 2: Plotter.py