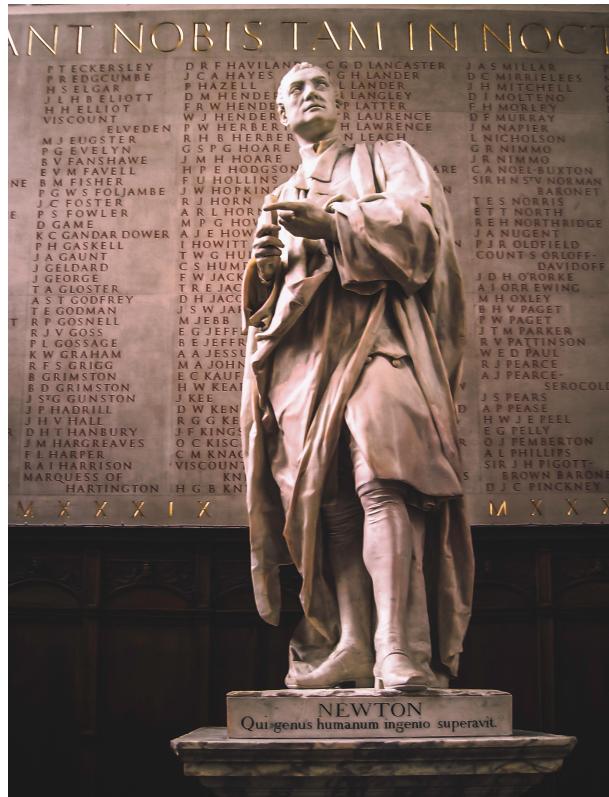


# Semesterprojekt Physik Engines

Kim Lan Vu, Michel Steiner, Asha Schwegler

30. Mai 2023



# Inhaltsverzeichnis

<b>1 Zusammenfassung</b>	<b>3</b>
<b>2 Aufbau des Experiments</b>	<b>4</b>
2.1 Aufbau des Lab 2, „Würfel 1 bewegt sich und stösst“ . . . . .	4
<b>3 Physikalische Beschreibung der einzelnen Vorgänge</b>	<b>5</b>
3.1 Lab 2: Würfel bewegt sich und stösst . . . . .	5
3.1.1 Konstante Kraft . . . . .	5
3.1.2 Elastischer Stoss . . . . .	6
3.1.3 Inelastischer Stoss . . . . .	7
3.2 Teil 3: Beide Würfel schwingen gedämpft . . . . .	7
<b>4 Beschreibung der Implementierung inklusive Screenshots aus Unity</b>	<b>8</b>
4.1 Lab 2: Würfel bewegt sich und stösst . . . . .	8
4.2 Lab 3: Beide Würfel schwingen gedämpft . . . . .	11
<b>5 Resultate mit grafischer Darstellung</b>	<b>13</b>
5.1 Lab 2 . . . . .	13
<b>6 Rückblick und Lehren aus dem Versuch</b>	<b>16</b>
<b>A Anhang</b>	<b>17</b>

# 1 Zusammenfassung

An der ZHAW wird im Physik Engine, kurz PE, physikalische Zusammenhänge von Bewegungsrichtungen und Beschleunigung, sowie potenzieller und kinetischer Energie im Raum und Zeit durch Simulationen auf Körper untersucht. Dabei werden über das Semester hinweg in Kleingruppen zwei Labs bearbeitet.

Im ersten Lab werden die inelastischen und elastischen Stöße untersucht.

Diese Untersuchungen werden mit den physikalischen Gesetzen in der Game-Engine Unity simuliert. Dabei werden laufend Daten zu den aktuellen Parameter einzelner Objekte gesammelt und mit diversen Grafiken veranschaulicht. Die aus den zwei Experimenten gewonnenen Erkenntnisse werden in diesem Bericht niedergeschrieben.

## 2 Aufbau des Experiments

### 2.1 Aufbau des Lab 2, „Würfel 1 bewegt sich und stösst“

Für den Aufbau des Experimentes sind zwei Würfel mit den Dimensionen von 1.5m Seitenlänge und dem Gewicht von 2 Kilogramm gegeben. Wie in der Abbildung 1 zu entnehmen ist, wird der linke Würfel Julia und der rechte Romeo benannt. Daneben existiert eine Feder die horizontal an einer Wand befestigt ist. Bei dem gesamten Experiment wird der Reibungswiderstand ignoriert.

Ablauf des Experimentes:

1. Romeo wird mit einer konstanten Kraft (grüner Pfeil in Abbildung 1) auf 2m/s nach rechts beschleunigt.



Abbildung 1: Beschleunigung des Würfels

2. Romeo trifft nun auf die Feder. Dabei soll die Federkonstante (gelber Pfeil in Abbildung 2) so gewählt werden, dass Romeo elastisch zurückprallt ohne die Wand zu berühren.



Abbildung 2: Elastischer Zusammenstoß mit der Feder

3. Nach dem abgefederten Stoss gleitet Romeo zurück in die Richtung aus der er gekommen ist und stösst inelastisch mit Julia zusammen. Über einen FixedJoint haften die Beiden nun zusammen und gleiten mit der übertragener Energie (blaue Pfeile in Abbildung 3) weiter nach links.



Abbildung 3: Inelastischer Zusammenstoß mit dem anderen Würfel

### 3 Physikalische Beschreibung der einzelnen Vorgänge

In diesem Kapitel werden die physikalischen Vorgänge des Versuches beschrieben. Die gegebenen Massen sind:

- Gewicht[m] = 2kg
- Velocity[v] = 2m/s
- Würfelseite = 1.5m

#### 3.1 Lab 2: Würfel bewegt sich und stösst

Es werden drei Vorgänge beschrieben, die Beschleunigung durch die konstante Kraft, einen elastischen Stoß und einen inelastischen Stoß. Ein Würfel, namens Romeo, wird durch die konstante Kraft beschleunigt, bis maximal eine Geschwindigkeit von 2m/s erreicht wird. Romeo trifft auf eine Feder zu, die an einer Wand befestigt ist. Dabei geschieht ein elastischer Stoß und der Würfel gleitet wieder zurück und stößt dabei einen zweiten Würfel, Julia, diesmal passiert der Stoß inelastisch. Sämtliche Vorgänge erfolgen ohne Reibungskräfte.

##### 3.1.1 Konstante Kraft

Um die konstante Kraft zu berechnen nehmen wir die gewünschte Geschwindigkeit und berechnen damit die Beschleunigung, weil die Kraft sowohl von der Masse wie auch der Beschleunigung abhängt und gegeben ist durch die Formel[1]

$$F = m * a$$

Um dieses Anfangswertproblems zu lösen leiten wir die Geschwindigkeit ab [1]:

$$\begin{aligned} \dot{v} &= a \\ 2m * s^{-1} &\rightarrow -2m * s^{-2} \rightarrow a = [\frac{2m}{s^2}] \end{aligned}$$

Die Zeit, die gebraucht wird um den Würfel zu beschleunigen, wird durch folgende Formel beschrieben [1]:

$$v = a * t \rightarrow t = \frac{v}{a} \rightarrow \frac{2m/s}{2m/s^2} = 1s$$

Somit können wir nun die Kraft ausrechnen:

$$F = 2kg * \frac{2m}{s^2} \Rightarrow \frac{4kg*m}{s^2} = 4N$$

4N werden deshalb als konstante Kraft angewendet, damit auch die gewünschte Geschwindigkeit erreicht wird, danach wird keine Kraft mehr hinzugefügt und Romeo gleitet auf die Feder zu.

### 3.1.2 Elastischer Stoss

Beim elastischen Stoss ist die kinetische Energie vom Stosspartner vor und nach der Kollision gleich [1]. Gemäss Auftrag wird die Federlänge und Federkonstante so dimensioniert, dass der Würfel nicht auf die Wand trifft. Die kinetische Energie des Würfels wird mit folgender Formel berechnet [1]:

$$E_{kin_{Romeo}} = \frac{1}{2} * m * v^2$$

Setzt man die Massen von diesem Projekt ein erhält man:

$$\frac{1}{2} * 2kg * (\frac{2m}{s})^2 = 4J$$

Während des Stosses wird die kinetische Energie auf die Feder übetragnen. Die Feder speichert diese Energie in Form von potentieller Energie, da sie zusammengedrückt wird. Sobald sie Romeo zurück stößt, wird diese Energie in eine kinetische zurückgewandelt.

Um die Federkonstante zu berechnen, nehmen wir die Tatsache der Energieerhaltung zu Nutze und setzen die ausgerechnete kinetische Energie gleich mit der potentiellen Energie der Feder.

Die Formel für die potentielle Energie der Feder lautet[1]:

$$E_{potFeder} = \frac{1}{2} * k * x^2$$

Die Gleichsetzung der Energien, sieht folgendermassen aus[1]:

$$E_{kin_{Romeo}} = E_{potFeder}$$

$$\frac{1}{2} * m * v^2 = \frac{1}{2} * k * x^2$$

Diese Gleichung stellen wir um und lösen nach der Federkonstante k auf:

$$k = \frac{m*v^2}{x^2}$$

Mit den eingesetzten Massen und die gewählte maximale Auslenkung erhalten wir:

$$\frac{2kg*(2m/s)^2}{(1.7m)^2} = 2.77N/m$$

Jetzt wo wir die Federkonstante und Länge haben, können wir einen langsam Stoss gewährleisten.

### 3.1.3 Inelastischer Stoss

Beim vollständigen inelastischen Stoss, werden beide Stosspartner nach der Kollision verbunden sein und dieselbe Geschwindigkeit haben[1]. Die Formeln, die wir für diesen Vorgang brauchen sind, die der Impulse der beiden Körper:

$$Impuls_{Romeo} = m_{Romeo} * v_{Romeo}$$

$$Impuls_{Julia} = m_{Julia} * v_{Julia}$$

Bei diesem Vorgang wird ein Teil des Impulses von Romeo auf Julia übertragen. Der Gesamtimpuls bleibt erhalten vor und nach dem Stoss und wird durch den Impulserhaltungssatz beschrieben[1]:

$$m_{Romeo} * v_{Romeo} + m_{Julia} * v_{Julia} = (m_{Romeo} + m_{Julia}) * v_{Ende}$$

Die Endgeschwindigkeit ist die Geschwindigkeit, die beide Körper gemeinsam haben nach dem Stoss. Die Relation zwischen der kinetischen Energie und des Impulses, können wir folgendermassen herleiten[1]:

$$E_{kin} = \frac{1}{2} * m * v^2 = \frac{(mv)^2}{2m} = \frac{p^2}{2m}$$

Wenden wir dies nach dem Stoss an, sehen wir, dass die kinetische Energie geringer wird:

$$E_{kin_{Ende}} = \frac{p^2}{2(m_{Romeo} + m_{Julia})}$$

## 3.2 Teil 3: Beide Würfel schwingen gedämpft

## 4 Beschreibung der Implementierung inklusive Screenshots aus Unity

Der Programcode für Unity wird in C# geschrieben und im folgenden Kapitel wird auf die Implementation in Unity eingegangen.

### 4.1 Lab 2: Würfel bewegt sich und stösst

Alle Kräfte und Berechnungen befinden sich im Code CubeController.cs, welches im Anhang ersichtlich ist. Zur Kontrolle der Werte und Grafik Erstellung werden zwei verschiedene CSV Dateien erstellt, eine für den elastischen Stoss relevanten Werte und eine für den inelastischen Stoss.

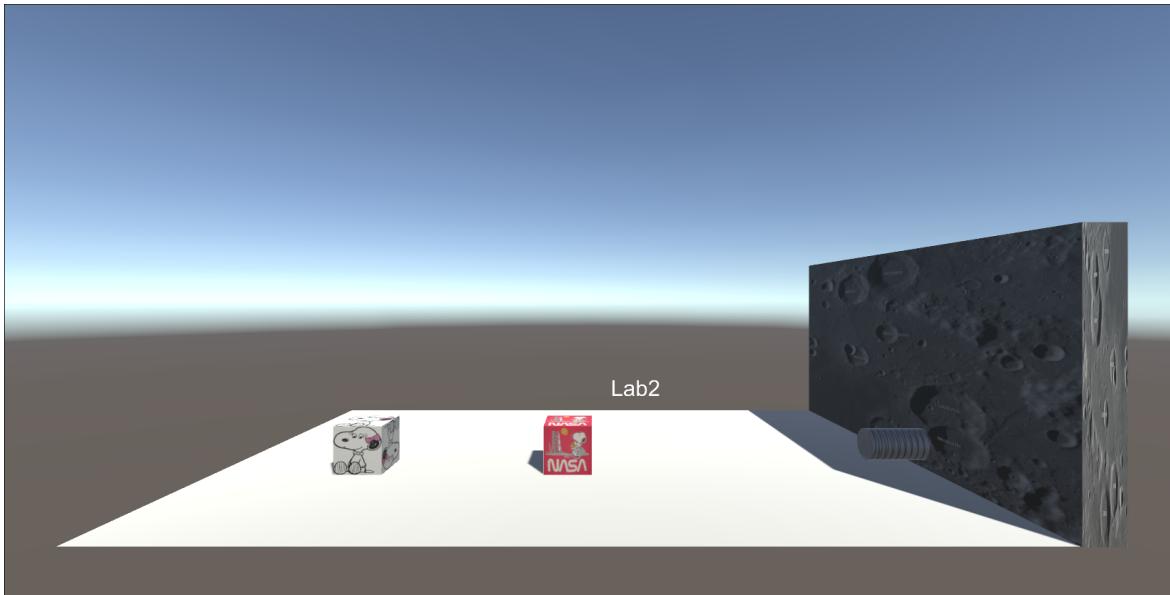


Abbildung 4: Experiment Übersicht

Folgend sind einige wichtige Eigenschaften der Lab relevanten Objekte in Unity aufgelistet:

- Julia:

Für die Seitenlänge ist die Scale auf 1.5 angepasst, da in Unity beim Würfel eine Seitenlänge von 1 gilt. Wichtig ist nicht zu vergessen, das Material auf reibungslos zu ändern, sonst gleiten die Würfel nicht korrekt.

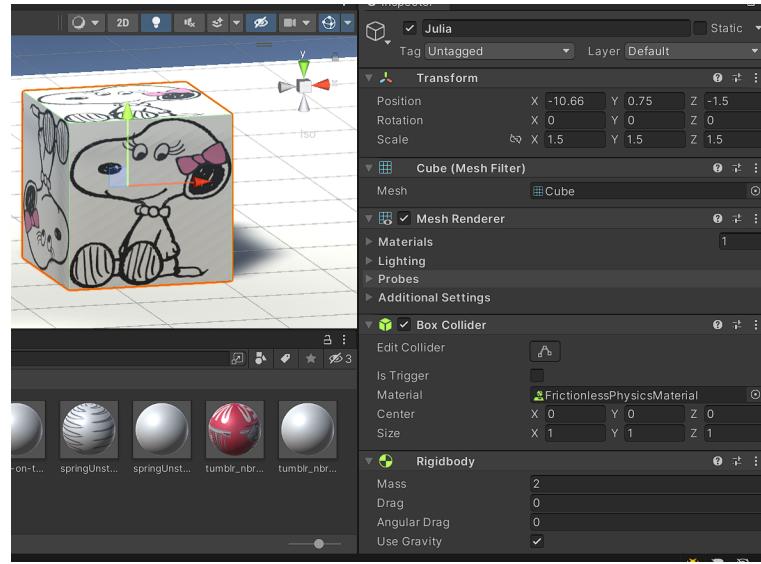


Abbildung 5: Einstellung Julia

- Romeo:

Die Eigenschaften sind ausser der Position und Farbe gleich wie bei Julia. Romeo besitzt zudem Variabel, über welcher gewisse Parameter an den ihm angehängten Code übergeben werden kann.

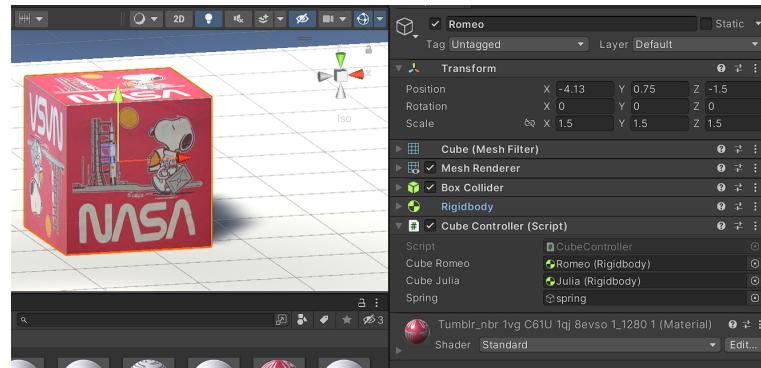


Abbildung 6: Einstellung Romeo

- Spring:

Wie in Abbildung 6 zu sehen ist die Feder nur als GameObject und nicht als Rigidbody im Code angegeben. Die Ausrichtung wurde entlang der Y-Achse belassen und um 90 Grad rotiert damit der Zylinder liegend erscheint.

- Mesh: Cylinder
- Collider direction: Y-Axis

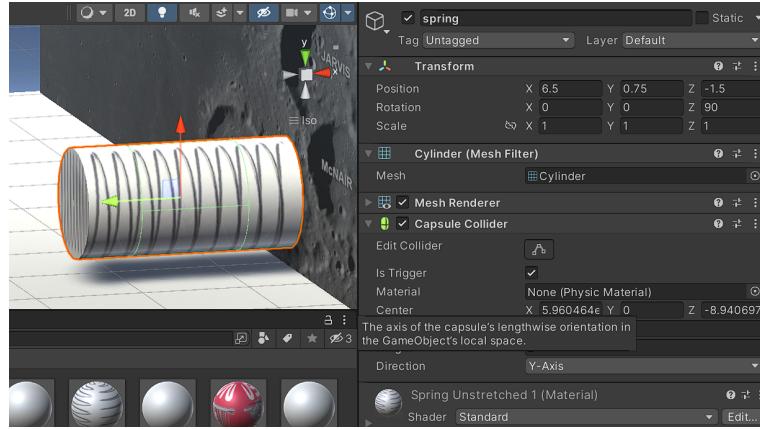


Abbildung 7: Einstellung Feder

- Plane
  - Collider Material: FrictionlessPhysicsMaterial

Da es sich beim CubeController um ein Unity Code handelt, wird von der Klasse Monobeviour geerbt, damit Methoden wie FixedUpdate oder OnCollisionEnter benutzt werden kann. Die im vorhinein berechnete konstante Kraft 4, sowie Beschleunigungszeit 1 wird im Code als Konstanten am Anfang deklariert. Für die Federauslenkung wird 1.7 gewählt, da die Feder eine Länge von 2 hat und vorher gestoppt werden muss bevor Romeo auf die Wand auftrefft. In der Start Methode wird aus den gegebenen Werten die Federkonstante berechnet.

```

1 //Maximale Auslenkung gerechnet anhand der linken seite des Feders
2 springMaxDeviation = spring.transform.position.x - springLength / 2;
```

Es wird auch die Position ermittelt, welche Romeo zum ersten Mal besitzt, wenn er auf die Feder auftrefft (springMaxDeviation) wie in Abbildung 8 rot markiert ist.

```

1 // Energieerhaltungsgesetz kinEnergie = PotEnergie : 1/2*m*v^2 = 1/2k * x^2
2 springConstant = (float)((Romeo.mass * Math.Pow(2.0, 2)) /
    (Math.Pow(springContraction, 2.0)));
```

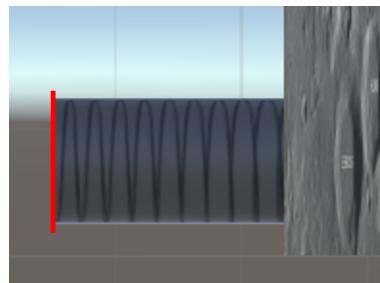


Abbildung 8: Feder mit markierter Berührungs punkt

Danach werden die Timeseries Listen deklariert, für die CSV Dateien. Das Hinzufügen der Werte passiert kontinuierlich in der Methode FixedUpdate.

In FixedUpdate gibt es zwei If-Bedingungen. Die erste ist zum Hinzufügen der konstanten Kraft mit der Methode .AddForce bis die Beschleunigungszeit vorüber ist. Die zweite If-Bedingung ist zur

Überprüfung, ob Romeo die Feder berührt. Dafür muss die Position der rechten Kante von Romeo berechnet werden und mit der springMaxDeviation verglichen werden. Für den inelastischen Stoss wird die Komponente FixedJoint in der Methode OnCollision implementiert. So bleiben Romeo und Julia nach ihrer Kollision zusammen und gleiten gemeinsam in den Sonnenuntergang.

## 4.2 Lab 3: Beide Würfel schwingen gedämpft

Für die dritte Aufgabe sind neben dem CubeController.cs noch weitere Scripte dazugekommen. Diese sind:

- LineCrontroller.cs
- LineJulia.cs
- SwingJulia.cs
- SwingRomeo.cs

Zur Kontrolle der Werte und Grafik Erstellung werden zwei verschiedene CSV Dateien erstellt, eine für den Schwung von Julia und eine für den Schwung von Romeo. Der gesamte Labaufbau in Unity ist in der [9](#) zur Laufzeit visuell dargestellt.

Der

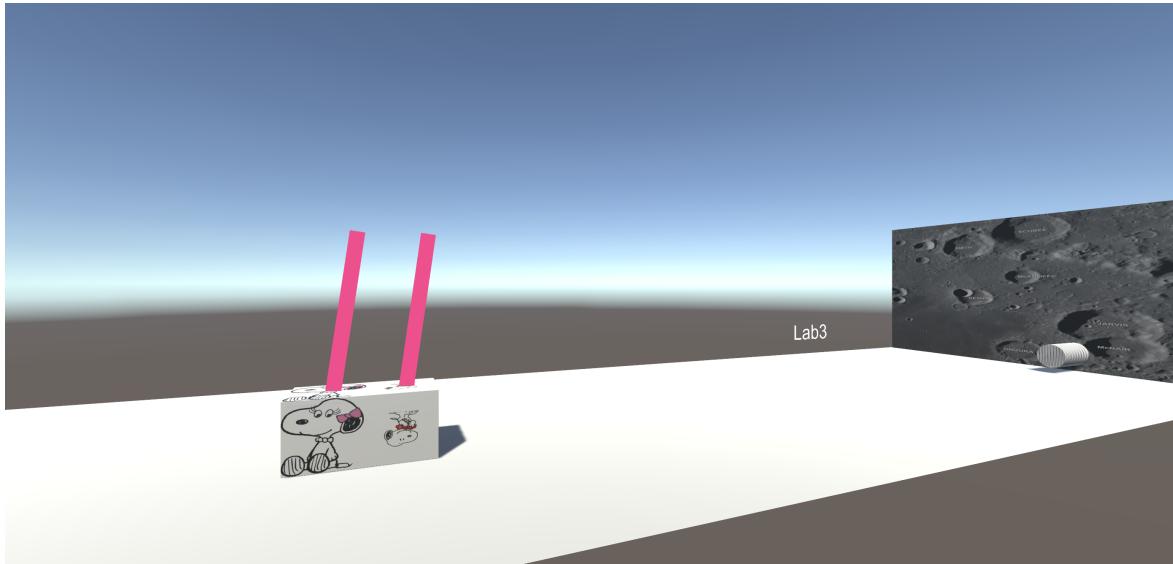


Abbildung 9: Experiment Übersicht

Folgend sind die neuen wichtigen Eigenschaften des Lab relevanten Objekte in Unity aufgelistet:

- Seile: Die Seile die in der Abbildung [9](#) in Pink ersichtlich sind, werden von den Würfel zu den vordefinierten Punkten wie in Abbildung [10](#) ersichtlich sind, gezeichnet.

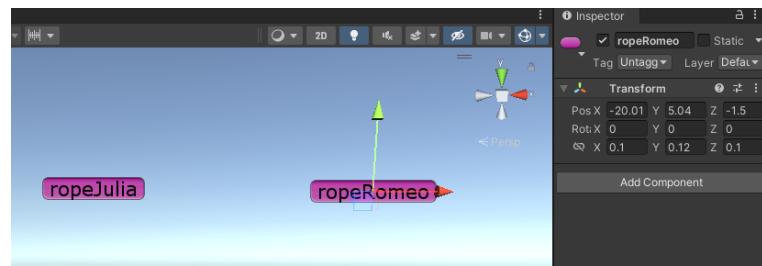


Abbildung 10: Experiment Übersicht

## 5 Resultate mit grafischer Darstellung

### 5.1 Lab 2

Während dem Durchlauf des Experimentes des Lab 2, werden diverse Daten physikalische Vorgänge gesammelt. Diese Daten umfassen Ort und Geschwindigkeit, sowie kinetische und potentielle Energie. Nachfolgend werden alle Daten als Funktion der Zeit in der Abbildung 11 bis Abbildung 15 aufgegliedert.

In Abbildung 11 ist deutlich zu erkennen, wie Romeo während der Beschleunigungsphase in den ersten Sekunden an Impuls gewinnt. Nach fünf Sekunden Gleitphase stösst Romeo auf die Feder, wodurch er abgebremst wird und Impuls verliert, bis er schliesslich bei null ankommt. Der Impuls wird jedoch in der gespannten Feder gespeichert und beim Entspannen der Feder wieder auf den Würfel übertragen. Dadurch hat der Würfel nach der Beschleunigungsphase wieder den gleichen Impuls wie zuvor. Da der Gesamtimpuls erhalten bleibt, haben Romeo und Julia nun beide ein Impuls von 2 Ns. In Abbildung 12 sieht man die addierten Impulse und wie dies nach der Kollision in der Tat gleich bleibt.

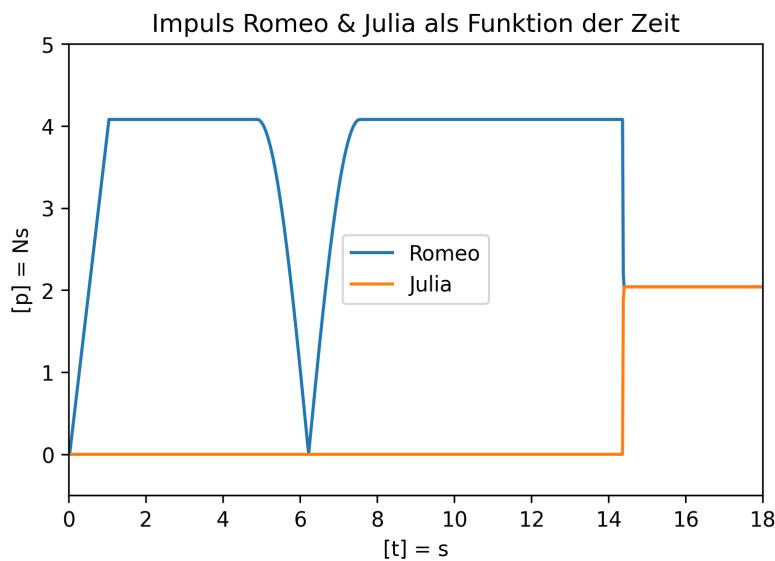


Abbildung 11: Impuls Romeo & Julia als Funktion der Zeit

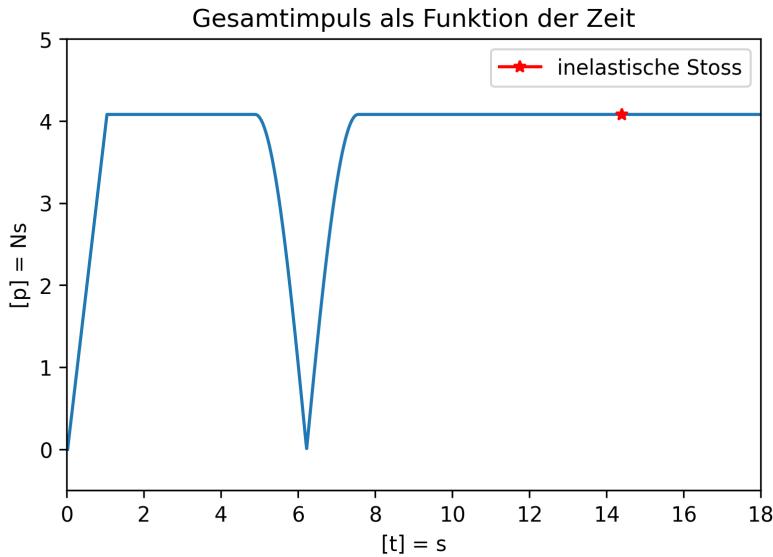


Abbildung 12: Gesamtmpuls als Funktion der Zeit

Im Ort-Zeit Diagramm in der Abbildung 13 ist ersichtlich, dass Romeo bis Sekunde 5 sich bewegt und danach auf die Feder auftritt. Romeo bewegt sich dann gleichmäig in die entgegengesetzte Richtung und nach 14 Sekunde ist zu sehen wie beide Würfel dann zusammen sich bewegen.

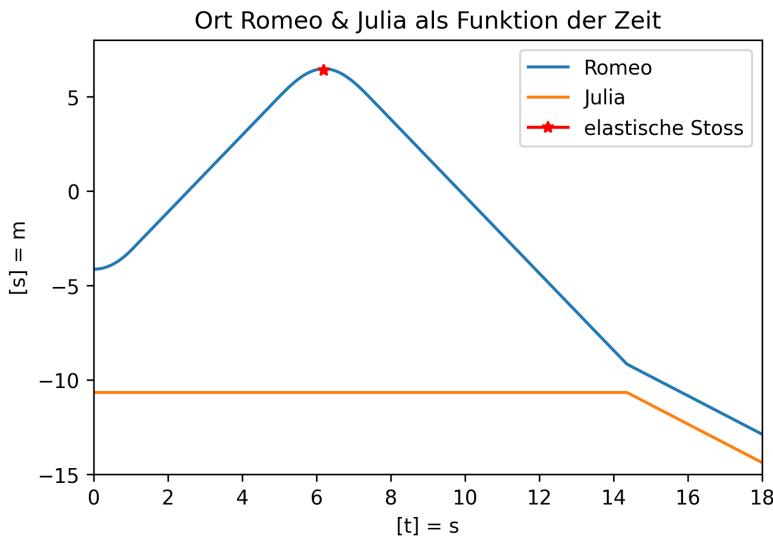


Abbildung 13: Ort Romeo & Julia als Funktion der Zeit

Gemäss Berechnung erreicht Romeo nach einer Sekunde die maximale Geschwindigkeit von 2 m/s und bewegt sich weiter mit dieser Geschwindigkeit bis es kurz vor Sekunde 6 auf die Feder trifft. Dann verändert sich wegen den Rückschlag die Richtung der Geschwindigkeit. Beim inelastischen Stoß kleben Romeo und Julia zusammen und fahren mit der Schwerpunktgeschwindigkeit vEnde von 1 m/s, wie im Kapitel Physikalisch Beschreibung berechnet, weiter. In der Abbildung 15 sieht man, dass dies in unseren Versuch auch übereinstimmt.

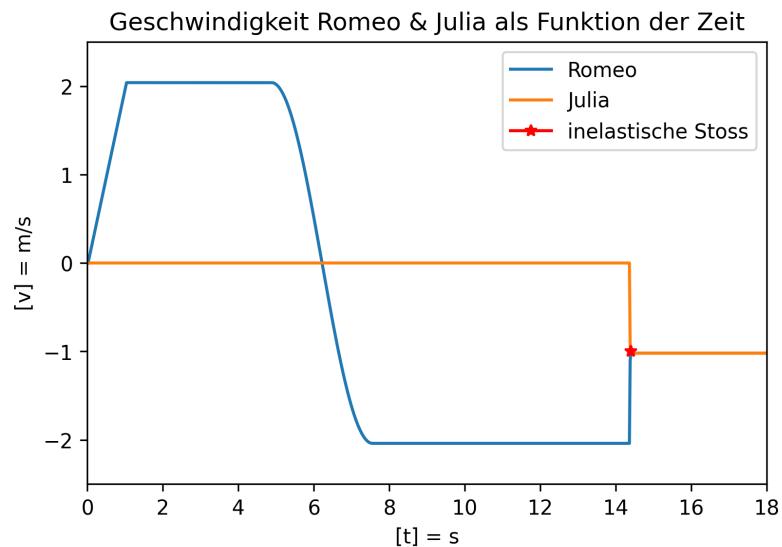


Abbildung 14: Geschwindigkeit Romeo & Julia als Funktion der Zeit

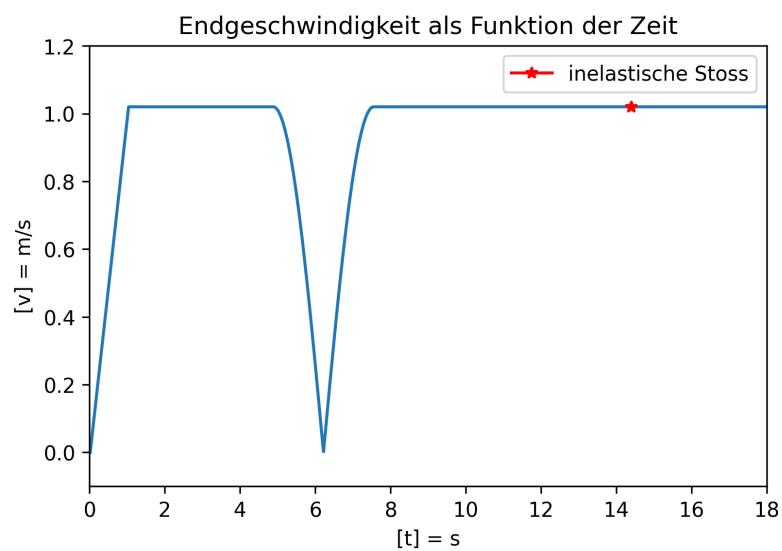


Abbildung 15: Endgeschwindigkeit als Funktion der Zeit

## **6 Rückblick und Lehren aus dem Versuch**

## **A Anhang**

## Abbildungsverzeichnis

1	Beschleunigung des Würfels . . . . .	4
2	Elastischer Zusammenstoss mit der Feder . . . . .	4
3	Inelastischer zusammenstoss mit dem anderen Würfel . . . . .	4
4	Experiment Übersicht . . . . .	8
5	Einstellung Julia . . . . .	9
6	Einstellung Romeo . . . . .	9
7	Einstellung Feder . . . . .	10
8	Feder mit markierter Berührungs punkt . . . . .	10
9	Experiment Übersicht . . . . .	11
10	Experiment Übersicht . . . . .	12
11	Impuls Romeo & Julia als Funktion der Zeit . . . . .	13
12	GesamtImpluls als Funktion der Zeit . . . . .	14
13	Ort Romeo & Julia als Funktion der Zeit . . . . .	14
14	Geschwindigkeit Romeo & Julia als Funktion der Zeit . . . . .	15
15	Endgeschwindigkeit als Funktion der Zeit . . . . .	15

## Literatur

- [1] Tipler, Paul A. u.a. *Physik Für Studierende Der Naturwissenschaften Und Technik*. Springer Spektrum. ISBN: 978-3-662-58280-0.