

DER SIMULATIONSBAUKASTEN ALGODOO
BESCHREIBUNG UND BEISPIELE

**Schriftliche Hausarbeit
für die erste Staatsprüfung
für das Lehramt am Gymnasiums**

von

Markus Zang

eingereicht am: 02. April 2012

Betreuer: Prof. Dr. Thomas Wilhelm

**JULIUS-MAXIMILIANS-UNIVERSITÄT WÜRZBURG
FAKULTÄT FÜR PHYSIK UND ASTRONOMIE
LEHRSTUHL FÜR PHYSIK UND IHRE DIDAKTIK**

Inhaltsverzeichnis

1. Einleitung	1
2. Was sind Simulationsprogramme?	3
2.1. Modellbildung im Physikunterricht	3
2.2. Animationen und Simulationen	4
2.2.1. Definitionen	4
2.2.2. Umsetzung von Modellen in Simulationsprogrammen	5
2.2.3. Probleme bei Simulationen	6
2.2.4. Simulationsarten	6
2.2.5. Multimodalität, Multicodierung und Interaktivität	7
2.3. Algodoo	8
2.3.1. Die Entwicklung von Algodoo	9
2.3.2. Ausblick	10
2.3.3. Kosten	10
3. Einführung in Algodoo	11
3.1. Die Benutzeroberfläche	11
3.2. Grundlagen	14
3.2.1. Die Werkzeuge von Algodoo	14
3.2.2. Das Material-Menü	18
3.2.3. Das “Aussehen-Menü”	20
3.2.4. Gravitation und Luftreibung	21
3.3. Weiterführendes	23
3.3.1. Kollisionsgruppen	23
3.3.2. Graphen	23
3.3.3. Geometrieaktionen	24
3.3.4. Die CSG-Optionen	25
3.3.5. Komponenten	25
3.4. Die Skriptsprache Thyme	26
3.5. Algobox	30
3.6. Lektionen	31

4. Beispielszenen	33
4.1. Mechanik	33
4.1.1. Freier Fall	34
4.1.2. Kräfte	36
4.1.3. Der waagrechte Wurf	38
4.1.4. Beschleunigte Bewegungen	41
4.1.5. Kreisbewegung: Karussell	44
4.1.6. Kreisbewegung: Erde - Mond	45
4.1.7. Impuls - Zentraler Stoß zweier Massen	47
4.1.8. Teilchenmodell des Gases	49
4.2. Optik	52
4.2.1. Schatten und Ausbreitung von Licht	52
4.2.2. Sonnen- und Mondfinsternis	55
4.2.3. Prisma und Linsen	57
4.2.4. Ein Brillant	58
4.2.5. Das menschliche Auge	60
4.2.6. Teleskop	63
4.3. Elektronik	65
4.3.1. Elektrische Felder	65
4.3.2. Bewegung im elektrischen Feld	67
4.3.3. Leitermodell	69
4.4. Atomphysik	74
4.4.1. Atommodelle	74
4.4.2. Rasterkraftmikroskop	77
5. Fazit	81
A. Literatur	83
B. Anhang	84
Danksagung	85
Selbstständigkeitserklärung	86

1. Einleitung

Der Einsatz von Computern und mittlerweile auch interaktiven Whiteboards im Unterricht verbreitet sich zunehmend. Der Vorteil dieser neuen Medien liegt auf der Hand. Nicht nur, dass Bilder, Videos und Audiodateien leichter und schneller in den Unterricht integriert werden können, sondern es können auch diverse Computerprogramme verwendet werden. Gerade im Physikunterricht lassen sich eine Vielzahl von Programmen nutzen um zum Beispiel Messwerte zu erfassen oder physikalische Sachverhalte visuell darzustellen. In einer Erhebung von WILHELM und TREFZGER haben 83 % der befragten Lehrern bereits Simulationen im Unterricht eingesetzt.¹

Gerade auch wegen der zunehmenden Verbreitung von interaktiven Whiteboards an deutschen Schulen wird der Simulationsbaukasten *Algodoo* für den Physikunterricht interessant. Dieses Programm erlaubt dem Nutzer, in einer vorgefertigten Umgebung, in der zunächst die gewohnten physikalischen Gesetze gelten, sehr frei zu experimentieren und diese Gesetze nachträglich auch zu ändern. Algodoo ist wegen seiner Benutzeroberfläche sehr gut für interaktive Whiteboards geeignet. Speziell mit dem Entwickler *Smart Technologies* arbeitet *Algoryx*, der Hersteller von Algodoo, eng zusammen.²

Nach WILHELM und TREFZGER allerdings ließen lediglich 12 % der befragten Lehrer ihre Schüler selbst mit Simulationen arbeiten. 40 % nutzten sie lediglich als Demonstration. Da sich Programme wie Algodoo gut dazu eignen, von Schülern selbst genutzt zu werden, wird in dieser Arbeit in erster Linie darauf eingegangen, ob die erstellten Beispiele eher dazu gedacht sind, vom Lehrer präsentiert zu werden, oder ob Schüler damit selbstständig experimentieren können.

In Kapitel 2 wird zunächst kurz auf die Begrifflichkeiten eingegangen. Es wird geklärt, was der Unterschied zwischen Simulationen und Animationen ist und was bei deren Nutzung im Physikunterricht zu beachten ist.

In Kapitel 3 wird eine Einführung in Algodoo gegeben. Die Benutzeroberfläche wird erklärt und auf die verschiedenen Werkzeuge eingegangen, die zur Erstellung einer Szene wichtig sind. Es wird erklärt, wie man die physikalischen Eigenschaften erstellter Objekte verändert und Einfluss auf die im Hintergrund berechnete Physik, wie zum Beispiel Gravitationskraft oder Luftwiderstand, nimmt. Anschließend wird noch etwas tiefer in das Programm eingetaucht und komplexere aber wichtige Funktionen erklärt. Auch die Skriptsprache *Thyme*, mit der man noch komplexere, Szenen gestalten kann, wird kurz erläutert. Diese Skriptsprache ist ausufernd, aber leider auch sehr unzugänglich, da von

¹Wilhelm, Thomas/Trefzger, Thomas: Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern. Didaktik der Physik - Hannover 2010, Frühjahrstagung der DPG 2010.

²Algoryx: Algodoo teacher edition. (URL: www.algodoo.com/wiki/Algodoo_teacher_edition).

1. Einleitung

Algoryx selbst keinerlei Erklärung oder Informationen darüber gegeben wird.

In Kapitel 4 schließlich werden zahlreiche Beispiele erstellt und deren Entstehung beschrieben. Wie bereits erwähnt, wird darauf geachtet, wieviel Interaktivität seitens der Schüler möglich ist. Im ersten Teil wird auf die Mechanik eingegangen, da Algodoo hauptsächlich auf diese ausgelegt ist. Dementsprechend ist dies auch der größte Teil dieses Kapitels. Im Anschluss werden Beispiele zum Thema Optik vorgestellt. Die beiden Gebiete Elektronik und Atomphysik werden indess nur kurz behandelt, da sich hier höchstens Modelle erstellen lassen, die auf Mechanik basieren.

Abschließend wird ein Fazit darüber gezogen, wie gut sich Algodoo letzten Endes im Unterricht verwenden lässt.

2. Was sind Simulationsprogramme?

Um genau zu verstehen, was überhaupt Simulationsprogramme sind, werden in diesem Kapitel einige grundlegende Begriffe geklärt. Zunächst wird auf den in der Physikdidaktik so zentralen Begriff der Modellbildung eingegangen, vor allem auch in Bezug auf die anschließend erläuterten Begriffe Animation und Simulation.

Abschließend wird unter diesem Aspekt der Simulationsbaukasten Algodoo vorgestellt, sowie kurz einige weitere Programme angesprochen.

2.1. Modellbildung im Physikunterricht

Modelle sind vereinfachte Darstellungen der Realität, bei denen man nur einzelne Aspekte eines physikalischen Vorganges betrachtet. Sie sind damit ein wichtiger Bestandteil des Physikunterrichts, denn in der Physik als modellbildende Wissenschaft wird ausschließlich mit Modellen der Realität gearbeitet.

Doch wie entsteht ein solches Modell? GOLDKUHLE beschreibt die Modellbildung als eine Abbildung der Ebene der Realität auf die Ebene des Modells. Dabei unterscheidet sich die Modellebene von der Realitätsebene dadurch, dass das Modell versucht, die physikalischen Größen der Realität zu repräsentieren. Als Güte eines solchen Modells gibt GOLDKUHLE an, wie gut das Modell die Realität abbildet. Bei diesem Modellbildungsprozess ist allerdings auch darauf zu achten, die physikalischen Größen und Faktoren didaktisch sinnvoll zu reduzieren. Denn eine zu starke Reduktion führt automatisch eine Verringerung der Güte mit sich.³ BERGER nennt hierfür vier Stufen:

1. *umgangssprachliche Beschreibung des physikalischen Systems, seiner Elemente und ihrer Beziehungen,*
2. *Reduktion auf wesentliche, das System charakterisierende Elemente und Vereinfachungen ihres Beziehungsgefüges,*
3. *Abstraktion unter Verwendung korrekter Fachbegriffe*
4. *formale Beschreibung in der ausgewählten Entwicklungsumgebung*⁴

Der vierte Punkt bezieht sich hierbei schon auf die weitere Übertragung des Modells in ein Simulationsprogramm. Hierauf wird im nächsten Abschnitt eingegangen.

³Goldkuhle, Peter: Modellbildung und Simulation mit dem Computer im Physikunterricht. Aulis Verlag Deubner & Co KG Köln, 1997, Praxis Schriftreihe Physik S. 18ff.

⁴Berger, Veit; Mikelskis, Helmut F. (Hrsg.): Physik-Didaktik. Cornelsen Verlag, 2010 S. 142.

2. Was sind Simulationsprogramme?

Dieser Modellbildungsprozess soll nicht Gegenstand dieser Arbeit sein. Vielmehr werden bereits erarbeitete Modelle des Physikunterrichts verwendet und damit gearbeitet.

Ein Modell ist also eine abstrakte Abbildung der Wirklichkeit. Um Schülern nun das Modell eines physikalischen Sachverhaltes näherzubringen, gibt es mehrere Möglichkeiten. Zum einen gibt es das Experiment. In ein reales Experiment fließen allerdings auch immer diverse Störfaktoren mit ein und es kann sein, dass Größen, die man bewusst aus dem Modell entfernt hat, hier wieder eine Rolle spielen. Das reale Experiment ist damit oft näher an der Ebene der Realität als an der Ebene des Modells. Will man sich dabei wirklich nur auf die Aspekte des Modells beschränken, bietet es sich an, andere Medien zu nutzen, wie zum Beispiel vorgefertigte Animationen oder Simulationsprogramme.

2.2. Animationen und Simulationen

2.2.1. Definitionen

Eine Animation ist eine Darstellung einer scheinbaren Bewegung.⁵ Im Unterschied zu einer echten Bewegung werden bei einer Animation nur schnell hintereinander Standbilder gezeigt. Dies wird als stroboskopische Bewegung bezeichnet. Ab etwa 16 Bildern pro Sekunde nimmt das menschliche Auge eine flüssige Bewegung wahr. Dies ist allerdings nur eine sehr allgemeine Definition einer Animation. DOMAGK nutzt deshalb für ihre Definition von Animation zunächst die Begriffsbestimmung von BETRANCOURT und TVERSKY, die von Anwendungen reden, *“die eine Folge von Bildern generieren, so dass jedes Bild als eine Veränderung des vorherigen Bildes erscheint, wobei die Reihenfolge der Bilder entweder vom Designer oder dem Lernenden bestimmt wird.”*⁶ Sie nutzt anschließend eine andere Definition von SCHNOTZ und LOWE, um den Begriff weiter einzuschränken. Diese reden nämlich von einer bildhaften Darstellung, *“deren Struktur und Eigenschaften sich über die Zeit verändern und die Wahrnehmung einer kontinuierlichen Veränderung erzeugen”*.⁷

Diese Definition schließt auch allgemein Videos mit ein, insbesondere künstliche Bilder, deren Veränderung vom PC aufgrund eines Modells berechnet wurden. Auch Programme, die solche Bilder direkt berechnen, werden davon umfasst, unabhängig davon, welchen Einfluss der Nutzer des Programms auf diese Berechnung nehmen kann. Ab einem gewissen Grad der Einflussnahme spricht DOMAGK dann von einer Simulation.

⁵Domagk, Steffi; Niegermann, Helmut M. (Hrsg.): Kompendium des multimedialen Lernen. Springer-Verlag, 2008 S. 240.

⁶Domagk S. 240ff.

⁷Domagk S. 240ff.

Nach dieser Definition wäre eine Simulation also eine Animation, auf die man in gewissem Maße Einfluss nehmen kann.⁸

BERGER wählt eine geringfügig andere Definition für Simulation und Animation. Er spricht, speziell auf die Physik bezogen, von “*Visualisierung physikalischer Erscheinungen und Vorgänge*”, bei denen man kaum Interaktionsmöglichkeiten hat (Animationen) und solchen, bei denen man über diverse Parameter teils starken Einfluss nehmen kann (Simulation).⁹ Er sieht also die Simulation nicht als eine Untergruppe der Animationen an, sondern beide als gleichwertige Konzepte, denen beidesmal eine Visualisierung zugrunde liegt.

In Algodoo mit seinen zahlreichen Möglichkeiten, Parameter zu verändern, ist zunächst einmal jede Szene eine Simulation. In dieser Arbeit kommen sowohl Szenen vor, die zum Eingreifen und Ausprobieren durch Schüler gedacht sind, als auch solche, die nur zur Vorführung durch den Lehrer dienen. Bei diesen wäre ein Verändern der Parameter selbstverständlich auch möglich, zum Teil aber wenig sinnvoll, da zum Beispiel nur ein spezieller Sachverhalt dargestellt werden soll. In dieser Arbeit wird daher basierend auf der Definition von BERGER zwischen Animation und Simulation unterschieden, je nachdem, ob das Konzept der jeweiligen Szene eine Einflussnahme des Nutzers vorsieht oder nicht, unabhängig davon, ob diese möglich ist.

2.2.2. Umsetzung von Modellen in Simulationsprogrammen

Will man einen physikalischen Sachverhalt in einem Simulationsprogramm darstellen, so sind einige Überlegungen notwendig. Bereits in Abschnitt 2.1 wurde erläutert, wie der Modellbildungsprozess abläuft, nämlich durch eine Abbildung der Realitätsebene auf die Modellebene. GOLDKUHLE ergänzt diesen Prozess um einen weiteren Schritt. Mit einer weiteren Abbildung der Modellebene auf die Computermodellebene wird schließlich aus dem theoretischen Konstrukt des Modells eine anschaulichere Darstellung als Simulation.¹⁰ Die für das Modell relevanten Größen werden dem Simulationsprogramm übertragen, welches die Ergebnisse dann numerisch berechnet. Das bedeutet, dass das Simulationsprogramm meistens in einem kleinen Zeitintervall Δt die momentanen Kräfte berechnet und mit der daraus resultierenden Beschleunigung die Geschwindigkeitsänderung und Ortsänderung bestimmt.

⁸Domagk S. 260.

⁹Berger, Veit; Mikelskis, Helmut F. (Hrsg.): Physik-Didaktik. Cornelsen Verlag, 2010 S. 140.

¹⁰Goldkuhle S. 20.

2.2.3. Probleme bei Simulationen

Die Vorteile von Simulationsprogrammen wie Algodoo liegen auf der Hand: Ein physikalisches Modell kann umgesetzt und von den Schülern aktiv erforscht werden. Sie bekommen nur die für das Modell wichtigen Größen präsentiert und können die Auswirkungen von Veränderungen auf diese Parameter entdecken. Laut DOMAGK kann es hierbei aber auch zu Problemen kommen:¹¹ Wird den Schülern beim Erkunden allzu freie Hand gelassen, verlieren diese oft das wesentliche Ziel aus den Augen. Es kommt auch häufig vor, dass die Schüler ihre Simulationen so anpassen, dass die Ergebnisse mit ihren bisherigen Hypothesen übereinstimmen, wie zum Beispiel die Fehlvorstellung, dass schwere Gegenstände auf Grund ihrer Masse schneller zu Boden fallen als leichte, obwohl hier die Luftreibung die entscheidende Rolle spielt. Ein weiteres Problem sieht DOMAGK darin, dass Schüler oft Schwierigkeiten haben, selbstständig Annahmen zu formulieren und diese dann zu überprüfen.

Auf Grund dieser Probleme ist es ratsam, die Schüler nicht völlig frei an den Simulationen arbeiten zu lassen, sondern ihnen ein vernünftiges Maß an Instruktionen mitzugeben.

2.2.4. Simulationsarten

Der Einfachheit halber werden im Folgenden die Begriffe Simulation und Simulationsprogramm synonym verwendet. Es werden zwei verschiedene Arten von Simulationsprogrammen unterschieden.¹² Zum einen gibt es modellbildende Simulationsprogramme, bei welchen der Anwender sämtliche Eigenschaften, zum Beispiel Bewegungsgleichungen, selbst programmieren kann. Das Modell wird hier von Grund auf selbst erschaffen. Durch die eigenständige Programmierung hat der Anwender auch Kontrolle über alle Einstellungen und Eigenschaften des Modells. Hier steht die mathematische Formulierung des Modells im Vordergrund. Das Programm nimmt dem Anwender schließlich die Berechnungen ab und zeigt ihm das Ergebnis auf. GIRWIDZ nennt folgende drei pädagogischen Aspekte eines solchen Simulationsprogrammes:

1. *Der direkte Einblick und Zugriff auf das zugrunde liegende Modell mit Variationsmöglichkeiten verlangt eine größere Verarbeitungstiefe als rein deskriptive Erklärungen.*

¹¹Domagk S. 264.

¹²Domagk S. 261ff.

2. *Kausalzusammenhänge werden aus dem Modellverhalten erfahren und erlebt - sie werden nicht "erzählt". Dies hat einen Einfluss auf die Gedächtnishaftung.*
3. *Der Lernende hat die volle Kontrolle über Variations- und Lösungswege. Dies ermöglicht ein selbstbestimmtes Lernen und das Verfolgen eigener Ideen.*¹³

Zum anderen gibt es die modellanwendenden Simulationen. Hier ist die mathematische Grundlage bereits implementiert und der Anwender erstellt seine Simulation auf dieser. Dies schränkt natürlich die Möglichkeiten einer modellbildenden Simulation ein, macht diese aber gleichzeitig viel zugänglicher. Schüler können sich hier besser auf die Ergebnisse konzentrieren, da es nicht nötig ist, den vollständigen mathematischen Hintergrund zu beherrschen. Eigentlich bezeichnet der Begriff "modellanwendende Simulation" Simulationen, die sich auf ein Thema spezialisiert haben, wie zum Beispiel die zahlreichen Java-Applikationen im Internet, die auf einen einzigen physikalischen Sachverhalt zugeschnitten sind und speziell diesen präsentieren. Bei Algodoo hingegen verwischen die Grenzen zwischen modellbildender und modellanwendender Simulation, da zwar der physikalische Hintergrund bereits fest programmiert ist, der Anwender aber in der Gestaltung seines physikalischen Modells eine große Freiheit genießt. Algodoo ist also eher aus der physikalischen Sicht als aus der mathematischen Sicht modellbildend.

2.2.5. Multimodalität, Multicodierung und Interaktivität

Hier soll nun auf die drei Begriffe *Multimodalität*, *Multicodierung* und *Interaktivität* eingegangen werden, welche GIRWIDZ als interessante Aspekte von Multimedia und somit insbesondere auch von Simulationsprogrammen nennt.¹⁴

Multimodalität Multimodalität bedeutet, dass verschiedene Sinne des Lernenden, in erster Linie die akustischen und visuellen angesprochen werden. *"Der Einsatz akustischer und visueller Informationen in mehreren Beschreibungsformen kann unterschiedliche Aspekte eines Inhalts hervorheben, Zusammenhänge und Wechselbezüge verdeutlichen."*¹⁵ Nach dem Modalitätsprinzip ist dies besonders gut, denn es besagt, *"dass die besten Behaltens- und Verstehensleistungen von Lernenden dann erreicht werden, wenn*

¹³Kircher, Ernst/Girwidz, Raimund/Häußler, Peter; Häußler, Peter (Hrsg.): Physikdidaktik. Springer, 2009 S. 425.

¹⁴Kircher/Girwidz/Häußler S. 426.

¹⁵Kircher/Girwidz/Häußler S. 428.

2. Was sind Simulationsprogramme?

beide Sinneskanäle, der visuelle und der auditive, parallel die Informationen an den Lernenden vermitteln".¹⁶

Multicodierung Multicodierung meint, dass der Lernstoff auf verschiedene Art und Weise dargeboten wird, zum Beispiel durch Animationen, Tabellen, Diagramme, Texte usw. Simulationsprogramme bieten die Möglichkeit, Schülern ein physikalisches Modell nicht nur durch eine verbale Erklärung oder starre Bilder näher zu bringen, sondern bietet die Möglichkeit, es durch bewegte Animationen zu präsentieren, was den visuellen Sinneskanal viel deutlicher anspricht.

Interaktivität Interaktivität ist, wie oben bereits genannt, ein Hauptmerkmal von Simulationen. Die Schüler experimentieren und erforschen die Simulationsumgebung, sie können mehr oder weniger stark eingreifen und die Auswirkungen studieren. Echte Interaktivität bedeutet den Schritt weg *"von einem behavioristischen Lerndesign zu einer konstruktivistischen Lernumgebung"*.¹⁷ Nach GIRWIDZ kann erst von einer echten Interaktivität gesprochen werden, wenn:

1. *die Lernenden kreativ sein dürfen und Inhalte modifizieren bzw. selbst erstellen können;*
2. *das Programm dynamisch und adaptiv auf die Aktionen der Lernenden reagiert;*
3. *Lernende selbst die Lernkontrolle über ihre Lernprozesse übernehmen können [...]*
4. *den Lernenden vom Mediensystem bei Bedarf Hilfe bzw. Führung angeboten wird.*¹⁸

Bei Simulationsprogrammen und gerade speziell bei Algodoo treffen hier alle Punkte zu. Bei der Interaktivität werden in dieser Arbeit jedoch hin und wieder Abstriche gemacht bzw. wird ganz verzichtet, da sie gerade bei schwierigeren Sachverhalten oder Modellen, die nur etwas bildlich veranschaulichen sollen, nicht zum gewünschten Ziel führen würden.

2.3. Algodoo

Algodoo ist ein Simulationsbaukasten zur Erzeugung von Simulationen in einer virtuellen zweidimensionalen Welt. Ein hervorstechendes Merkmal ist seine einfache Bedienbarkeit.

¹⁶Domagk, Steffi; Niegermann, Helmut M. (Hrsg.): Kompendium des multimedialen Lernen. Springer-Verlag, 2008 S. 200.

¹⁷Kircher/Girwidz/Häußler S. 633.

¹⁸Kircher/Girwidz/Häußler S. 633.

Es ist möglich mit nur wenigen Klicks kleine Szenen zu erstellen und zu starten. Dazu stehen dem Nutzer grundlegende Zeichenwerkzeuge zur Verfügung, mit denen man verschiedene geometrische Objekte zeichnen kann. Diese können mit weiteren Werkzeugen verbunden werden, wie zum Beispiel durch drehbare Achsen, Federn und Seile. So kann im Handumdrehen aus einem Rechteck, zwei Kreisen und zwei Achsen ein primitives Fahrzeug erstellt werden, wie es in Abbildung 1 auf Seite 11 zu sehen ist.

Diese Zugänglichkeit und Benutzerfreundlichkeit macht Algodoo so interessant für den Unterricht. Schüler brauchen keine langwierige Einweisung, sondern können sofort anfangen, frei zu experimentieren. Gleichzeitig bietet Algodoo aber auch komplexere und tiefgreifendere Funktionen an, mit denen sich aufwendige Simulationen gestalten lassen.

2.3.1. Die Entwicklung von Algodoo

Die Geschichte von Algodoo beginnt mit der Veröffentlichung des Vorgängerprogramms *Phun*. Phun wurde von Emil Ernefeldt an der Umeå Universität in Schweden als Teil seiner Master-Arbeit entwickelt.¹⁹ Eine Beta-Version des kostenlosen Programms wurde am 17. Dezember 2007 veröffentlicht. Danach wurde Phun kontinuierlich weiterentwickelt, bis es schließlich am 15. Mai 2008 von *Algoryx Simulation AB* übernommen wurde. Emil Ernefeldt wurde bei Algoryx eingestellt und arbeitet seither weiter an seinem Programm. Mit der Übernahme wurde auch der Name in Algodoo geändert. Am 1. September 2009 schließlich wurde die erste Version von Algodoo veröffentlicht, diesmal kostenpflichtig.²⁰ Die Benutzer-oberfläche wurde von Phun übernommen. Große Neuerungen hingegen waren die Einführung von optischen Werkzeugen wie Lasern und anderen lichtdurchlässigen Objekten, sowie die Möglichkeit, Graphen von physikalischen Größen anzeigen zu lassen und Kräfte und Geschwindigkeiten durch Vektorpfeile darzustellen.²¹ Es kamen regelmäßig weitere Updates heraus, die Algodoo weiter verbesserten. Der nächste große Schritt kam mit dem Update von Version 1.8.5 auf Version 2.0.0. Dabei wurde die Benutzeroberfläche komplett überarbeitet und neu gestaltet. Algobox, ein Dienst, bei dem man selbst gestaltete Szenen hochladen und für andere Nutzer zugänglich machen kann (siehe Seite 30), wurde in das Programm integriert. Viele weitere Details wurden verbessert, zum Beispiel das Verhalten von Seilen.

Die aktuelle Version ist 2.0.1 (Stand 26. März 2012), bei der viele Verbesserungen bei der Skriptsprache *Thyme* vorgenommen wurden (siehe Seite 26).

¹⁹Wikipedia: Phun. [URL: en.wikipedia.org/wiki/Phun_\(2D_Physics_Sandbox\)](http://en.wikipedia.org/wiki/Phun_(2D_Physics_Sandbox)).

²⁰Wikipedia: Algodoo. [URL: en.wikipedia.org/wiki/Algodoo](http://en.wikipedia.org/wiki/Algodoo).

²¹Algoryx: Changes and updates in Algodoo. [URL: www.algodoo.com/wiki/Changes](http://www.algodoo.com/wiki/Changes).

2.3.2. Ausblick

Eine Beta-Version von Algodoo 2.0.2 kann bereits zum Testen heruntergeladen werden, wird aber erst nach Fertigstellung dieser Arbeit offiziell veröffentlicht. Mit diesem Update wird *Wind* zu Algodoo hinzugefügt. Bislang wurde zwar Luftreibung berücksichtigt, aber Wind konnte nicht simuliert werden. Zudem ist es mit dieser Version möglich, die genaue Position von Objekten zu bestimmen. In früheren Versionen war dies nicht möglich.

Da Algodoo auch in Zukunft immer weiter verbessert wird, ist die Wahrscheinlichkeit groß, dass viele Unzulänglichkeiten des Programms, wie sie zum Teil in Kapitel 3 beschrieben werden, behoben und noch viele neue Möglichkeiten hinzugefügt werden.

2.3.3. Kosten

Mit Version 2.0.0 hat sich auch das Bezahlmodell von Algodoo verändert. Algoryx hat drei verschiedene Versionen entworfen.

Algodoo Play Dies ist eine kostenlose Variante, die aber sehr stark eingeschränkt ist. Der Nutzer kann keine Szenen selbst erstellen, sondern nur fertige Szenen, zum Beispiel aus der Algodoo, herunterladen und nutzen. Ihm stehen nur das *Ziehen-* und *Löschen-*Werkzeug zur Verfügung (siehe Seite 13). Die ersten 15 Stunden allerdings stehen dem Nutzer alle Möglichkeiten der *Algodoo Physics*-Variante zur Verfügung. Nach Ablauf der Testphase wird das Programm auf *Algodoo Play* beschränkt.

Algodoo Physics Diese Version kostet 15 €, hat aber ebenfalls Beschränkungen. Der Nutzer kann hier zwar Szenen erstellen, aber didaktische Möglichkeiten wie Graphen, das Anzeigen von Kräften, Geschwindigkeiten und Impuls sowie die vorgefertigten Unterrichtsstunden, die Lektionen stehen nicht zur Verfügung. Diese Version ist eher zum Vergnügen als für den ernsthaften Unterrichtsgebrauch gedacht.

Algodoo for Education Dies ist schließlich die volle Version mit allen Werkzeugen und Extras, die Algodoo zu bieten hat. Eine einzelne Lizenz kostet hier 45 €. Kauft man gleich zehn Lizenzen auf einmal, kosten diese nur 350 €. Allerdings gibt es auch ein Angebot speziell für Schulen. Für einen Grundbetrag von 150 € pro Jahr ist es möglich, pro Schüler eine Lizenz für 1 € pro Jahr zu erwerben. Diese Lizenz erlaubt es den Schülern sogar, Algodoo zu Hause zu nutzen. Eine andere Version davon ist eine unbegrenzte Lizenz, die einmalig 450 € kostet und einmalig 5 € pro Schüler. Allerdings hat man hier nur für ein Jahr lang das Anrecht auf Updates.

3. Einführung in Algodoo

In diesem Kapitel wird nun eine kurze Einführung in Algodoo gegeben. Hierbei werden die Werkzeuge und die verschiedenen Einstellungen, die man an den Objekten vornehmen kann, erklärt, also eine Übersicht über alles, was man braucht, um selbst Szenen zu erstellen. Am Schluss wird auch auf die Skriptsprache *Thyme* eingegangen, die weitere Möglichkeiten bei der Szenenerstellung eröffnet.

3.1. Die Benutzeroberfläche

Mit dem Update auf Version 2 von Algodoo hat sich die Oberfläche geringfügig geändert. Alles wurde übersichtlicher angeordnet, aber wirklich tiefgreifende Änderungen gab es nicht. Deshalb wird sich jeder Nutzer, der bereits mit einer älteren Version gearbeitet hat, auch sofort in der neuen Version zurechtfinden.

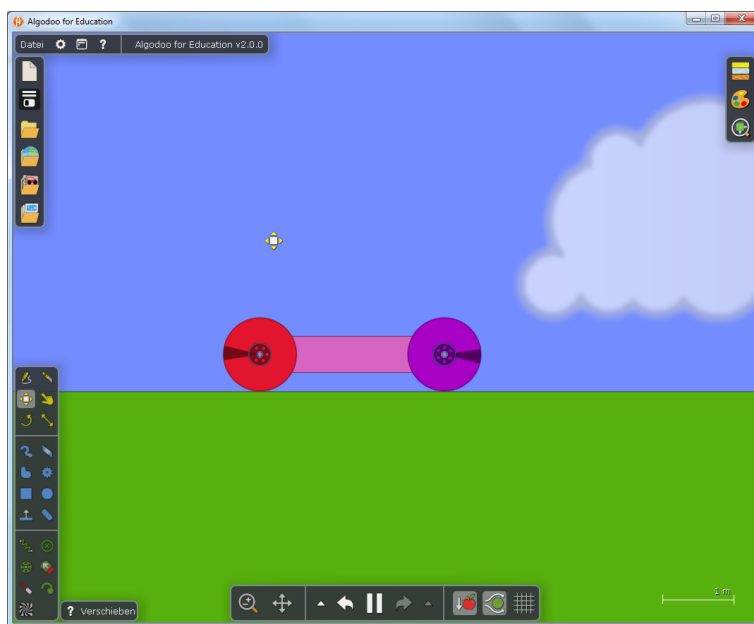


Abb. 1: Benutzeroberfläche von Algodoo

Öffnet man eine neue Datei, die in Algodoo als "Szene" bezeichnet wird, befindet sich dort standardmäßig eine Ebene (siehe Seite 16) vor einem blauen Hintergrund, über den Wolken ziehen. Es empfiehlt sich immer die Wolken durch *Rechtsklick* → *Hintergrundoptionen* zu deaktivieren, da diese nur störend sind.

In Abbildung 1 ist die Benutzeroberfläche und deren Aufbau zu sehen. Oben links befindet sich das Menü mit dem aus Windows bekannten *Datei-Reiter*, sowie *Optionen*, *Fenster anzeigen/verbergen* und *Hilfe*. Direkt darunter befindet sich die Szenenleiste,

3. Einführung in Algodoo

mit der man schnell Szenen oder Komponenten laden kann.

Im unteren linken Eck befindet sich die Werkzeugleiste und die Werkzeugoptionen. Damit werden sämtliche Objekte der Szene erstellt, seien es geometrische Figuren oder Achsen, Federn und Seile.

Oben rechts befinden sich die Materialeinstellungen. Hier wird festgelegt, welche Eigenschaften neu erstellte Objekte haben und wie sie aussehen. Neuen Objekten weist Algodoo eine zufällige Farbe zu, deren Bereich man hier festlegen kann. Je nach *Setting* sieht dieser Bereich anders aus und kann hier zusätzlich bearbeitet werden. Das *Setting* wird beim Erstellen einer neuen Szene festgelegt und hat nur Auswirkungen auf den Farbbereich.

Unten in der Mitte der Abbildung 1 befindet sich die Szenensteuerung. Hier kann die Simulation gestartet, pausiert und rückgängig gemacht werden. Ebenfalls befinden sich hier Zoom und Verschieben sowie die Funktion für das An- und Auschalten von Gravitation und Luftreibung.

In der Mitte sieht man ein zusammengesetztes Objekt. Es ist ein Rechteck, an das links und rechts zwei Kreise drehend mit Achsen befestigt wurden, womit man ein primitives Fahrzeug hat.



Menü Das Untermenü *Datei* entspricht dem, was man aus Windows kennt. Hier können neue Szenen gestartet, gespeichert und geöffnet werden. Erwähnenswert hierbei ist die Option *Alle Wasserpartikel löschen*. Algodoo stellt Wasser als viele kleine, elastische Kügelchen dar, was auf schwächeren Rechnern schnell zu einer Überlastung führen kann. Hiermit kann man diese wieder komplett entfernen.

Weiter gibt es noch die Optionen, in denen man sämtliche Einstellungen für Algodoo vornehmen kann. Diese sind alle selbsterklärend, weswegen hier nicht weiter darauf eingegangen wird.

Im Untermenü *Hilfe* befinden sich nützliche Tutorials, die Anfängern sehr zu empfehlen sind, denn hier wird ein kurzer Einstieg in die Funktionen von Algodoo gegeben.

Szenenmenü Die ersten drei Symbole sind nochmals Shortcuts für *Neue Szene*, *Szene speichern* und *Szene laden*. Darunter befindet sich das Symbol für die Algotbox. Die Algotbox ist ein Service von Algoryx, bei dem die Nutzer von Algodoo ihre selbsterstellten Szenen hochladen und so für andere zur Verfügung stellen können. Dieses System gibt es schon lange, allerdings wurde es erst mit Version 2 in die Benutzeroberfläche von Algodoo integriert. Mit wenigen Klicks kann man so eine fremde Szene laden und nutzen. Mehr

zur Albox erfährt man in Abschnitt 3.5 auf Seite 30.



Mit dem nächsten Symbol kann man *Komponenten* in seine Szene laden. Komponenten sind vorgefertigte Objekte, die man oft gebrauchen kann und mit wenigen Klicks in seine Szene einfügt. Algodoo stellt nicht allzuvielen davon zur Verfügung, man findet hauptsächlich Linsen für optische Szenen. Man kann allerdings selbst Komponenten erstellen und abspeichern.

Mit dem letzten Button öffnet man die *Lektionen*, die Algodoo bereit stellt. Dabei handelt es sich um vorgefertigte Szenen, die bereits zu einer Unterrichtseinheit aufbereitet wurden. Allerdings gibt es davon nur wenige (siehe Seite 31).



Werkzeugleiste und Werkzeugoptionen In der Werkzeugleiste befinden sich sämtliche Werkzeuge, die man zum Erstellen einer Szene benötigt. Im oberen Teil befinden sich diverse Werkzeuge zum Manipulieren von Objekten, wie zum Beispiel *Drehen*, *Skalieren* oder *Verschieben*. Im mittleren Teil befinden sich die Zeichenwerkzeuge, mit denen man Rechtecke, Kreise oder Freihandobjekte zeichnen kann, aber auch Zahnräder, Ebenen oder Seile. Im unteren Teil befinden sich zusätzliche Objekte, die man an den geometrischen Figuren anbringen kann, wie Federn, Achsen, Booster oder Laser.

Manche Werkzeuge haben noch zusätzliche Optionen, die rechts neben der Werkzeugleiste angezeigt werden. Wie die einzelnen Werkzeuge genau funktionieren, wird in Kapitel 3.2.1 erläutert.



Szenensteuerung Mit dieser Leiste kann die Szene gesteuert werden. In linken Teil befindet sich der *Zoom* und *Verschieben*. Bei gedrückter linker Maustaste auf

eines der beiden Symbole kann man es jeweils mit der Mausbewegung verändern. Bei der Handhabung vorteilhafter ist es, wenn man den Zoom mit dem Mausekranz und das Verschieben mit der mittleren Maustaste bedient.

Im mittleren Teil befinden sich der Start- und Pauseknopf. Wird er gedrückt, beginnt die eigentliche Simulation und kann auch wieder pausiert werden. Diese Funktion kann auch mit der Leertaste ausgeführt werden. Mit den beiden Pfeilen daneben können Veränderungen an der Simulation rückgängig gemacht (bzw. wiederhergestellt) werden oder man kann bei laufender Simulation zum letzten Pausepunkt zurückkehren.

Im rechten Teil schließlich befinden sich die Schaltflächen, um *Gravitation*, *Luftreibung* oder das *Gitternetz* ein- oder auszuschalten. Mit einem Rechtsklick auf die Symbole kann man zusätzlich die Stärke der Gravitation, das Verhalten der Luftreibung oder das

Aussehen des Gitternetzes einstellen.



Materialmenü - Aussehen - Darstellung Über die erste Schaltfläche dieses Menüs kommt man zu den Materialeigenschaften eines Objekts. Diese sind äußerst wichtig, da sie großen Einfluss auf die Simulation haben. Dichte, Elastizität, Reibung oder Lichtdurchlässigkeit werden hier eingestellt. Vorgefertigte Materialien haben spezielle Werte bei den Einstellungsmöglichkeiten. So hat Glas zum Beispiel eine hohe Lichtdurchlässigkeit.

Da es keinen Einfluss auf die Simulation hat, ist das Aussehen der Objekte weniger wichtig. Hier werden Farbe, Sichtbarkeit, Umrandung und ähnliches eingestellt. Man kann einem Objekt auch eine spezielle Textur zuweisen.

Im letzten Untermenü werden Einstellungen für das Anzeigen von Vektorgrößen vorgenommen. Dabei kann man Kräfte, Impuls und Drehimpuls sowie Geschwindigkeiten einstellen. Die Skalierung der Pfeile kann beeinflusst werden und speziell bei den Kräften kann man auswählen, welche Arten man angezeigt haben will.

3.2. Grundlagen

Kommen wir nun zum Herzstück von Algodoo, nämlich dem Erstellen einer Szene. Im Folgenden werden die wichtigsten Werkzeuge und die Einstellungen der Materialeigenschaften beschrieben und wie man all das zusammen einsetzt um eine Szene zu erstellen und zu simulieren.

3.2.1. Die Werkzeuge von Algodoo

In der Hilfe von Algodoo im Untermenü *Tools* werden die einzelnen Werkzeuge beschrieben und es ist empfehlenswert, sich dieses Tutorial anzuschauen. Im Folgenden werden die Werkzeuge noch einmal zusammengefasst.



Messer Dieses Werkzeug zerschneidet Objekte. Dies ist vor allem während Simulationen nützlich, wenn man beispielsweise ein Seil oder ähnliches durchschneiden will. Um Formen zurechtzuschneiden, ist das Messer allerdings etwas unpräzise. Dafür sollten besser die *CSG*-Optionen genutzt werden, die in Kapitel 3.3.4 auf Seite 25 näher beschrieben werden.



Verschiebe- und Ziehwerkzeug Diese beiden Werkzeuge verschieben Objekte in der Szene. Das *Verschieben*-Werkzeug macht dies direkt am Mauszeiger.

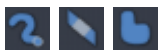
Allerdings übernehmen auch andere Werkzeuge diese Aufgabe. So kann man beispielsweise mit dem *Kreis*-Werkzeug ein Objekt anklicken und es verschieben. Es ist also nicht immer nötig, auf dieses Werkzeug zu wechseln. Das *Zieh*-Werkzeug hingegen funktioniert nur während einer laufenden Simulation. Es verschiebt die Objekte nicht direkt, sondern es verursacht eine Kraft auf diese, wodurch sie sich in Richtung Werkzeug bewegen. In den Optionen dazu kann man die maximale Kraft einstellen und die Stärke dieser Kraft. Ohne *Ziehe am Schwerpunkt* greift die Kraft an der Klickstelle an und *Deaktiviere Rotation* verhindert, dass sich das Objekt während des Verschiebens dreht.



Drehen und Skalieren Klickt man mit dem *Drehen*-Werkzeug auf ein Objekt, erscheint ein Kreis um den Schwerpunkt des Objekts und man kann mit der Maus die Ausrichtung des Objekts festlegen. Befindet man sich dabei im Inneren des Kreises, dreht man nur um 15°-Schritte; befindet man sich außerhalb, kann man den Winkel auf eine Nachkommastelle genau einstellen.

Klickt man auf ein markiertes Objekt mit der rechten Maustaste und hält diese gedrückt, hat das den selben Effekt wie das *Drehen*-Werkzeug. Das *Drehen*-Werkzeug an sich wird durch diese Rechtsklickoption nahezu obsolet.

Das *Skalieren*-Werkzeug ist jedoch sehr nützlich. Klickt man damit auf ein Objekt, erscheint ein rechteckiger Rahmen mit acht Angriffspunkten an Ecken und Kanten. Daran kann man das Objekt vergrößern oder verkleinern. Allerdings bleibt leider das Seitenverhältnis nicht erhalten, wenn man an den Eckpunkten zieht, wie das bei den meisten Bildbearbeitungsprogrammen der Fall ist.





Pinsel-, Lösch- und Polygon-Werkzeug Mit dem *Pinsel*-Werkzeug kann man Freihand zeichnen. So entstehen beliebige, aber auch unpräzise Figuren. Bei gedrückter Shift-Taste zeichnet man gerade Linien. Zusätzlich hat dieses Werkzeug die Option *Kalligraphie*, wodurch man mit einem breiten Pinsel zeichnet. Auch die Pinselstärke kann eingestellt werden. Die Option *Vereinigen* fügt das Gezeichnete direkt an ein bestehendes Objekt hinzu, die Option *An Geometrie kleben* heftet das neue Objekt an ein altes.


Das *Löschen*-Werkzeug ist genau das Gegenteil dazu. Damit kann man, wie mit einem Radiergummi, Teile von Objekten oder auch ganze Objekte wieder löschen. Auch von diesem Werkzeug kann man in den Optionen die Größe einstellen.


Auch mit dem *Polygon*-Werkzeug zeichnet man freihand, allerdings nur den Rand


eines neuen Objektes. Lässt man die Maustaste los, füllt Algodoo das Innere aus.

 **Rechteck- und Kreis-Werkzeug** Diese beiden Werkzeuge sind nahezu selbsterklärend. Man zeichnet mit ihnen Rechtecke und Kreise, wobei die Seitenlängen bzw. der Radius angezeigt werden. Für exakte Längen ist zu empfehlen, das Gitternetz zu aktivieren.

 **Ebenen-Werkzeug** Mit dem *Ebenen*-Werkzeug werden Ebenen erstellt, die im Prinzip Rechtecke mit unendlicher Ausdehnung in drei Richtungen darstellen. Der Winkel der Ebene kann wie bei dem *Drehen*-Werkzeug eingestellt werden. Eine Ebene hat alle Kollisionsgruppen (siehe *Kollisionsgruppen* in Abschnitt 3.3.1 auf Seite 23).

 **Zahnräder** Zahnräder werden ähnlich wie Kreise erstellt. Die Anzahl der Zähne richtet sich dabei nach der in den Optionen eingestellten *Zahnradgröße*. Jedes Zahnrad ist bereits mit einer Achse ausgestattet und wird an den Hintergrund oder an ein dahinter liegendes Objekt geheftet. Näheres zu Achsen wird auf Seite 17 dargelegt.

 **Seile** Seile waren vor Version 2 von Algodoo fast nicht zu gebrauchen, da sie ein seltsames unphysikalisches Verhalten an den Tag legten. Mit dem Update auf Version 2 wurden sie jedoch stark verbessert. Man zeichnet das Seil und es wird automatisch am Startpunkt befestigt. Liegt dieser auf einem Objekt, wird es daran befestigt, ansonsten am Hintergrund. Das Ende befestigt sich ebenfalls an Objekten, falls vorhanden, ansonsten bleibt es lose. In den Optionen lässt sich die *Stärke* des Seils einstellen. Ist dieser Wert niedrig, ist das Seil sehr dehnbar, ist er hingegen hoch, ist sie steifer, wird aber bei großen wirkenden Kräften instabil (sie fängt stark an zu zittern, was sie in früheren Versionen nahezu nutzlos gemacht hat). Nachdem ein Seil erstellt wurde, kann man durch einen Rechtsklick und dann über das Untermenü *Kette* auch eine Bruchgrenze eingeben und damit bestimmen, ab welchem Kraftstoß das Seil reißt.

 **Federn** Eine Feder wird zwischen zwei Punkten gespannt, wobei der Endpunkt an einem Objekt angeheftet werden muss. Nachdem eine Feder erstellt wurde, kann man über *Rechtsklick* → *Federn* die *Federkonstante*, *Dämpfung* sowie die *angestrebte Federlänge* einstellen.



Achsen Achsen werden mit einem Klick gesetzt und verbinden ein Objekt entweder mit dem Hintergrund oder einem dahinter liegenden Objekt. Objekte kollidieren nicht mehr miteinander und sind drehbar über die Achse verbunden. Die Größe der Achse hat keinerlei Bedeutung und hängt seltsamerweise vom eingestellten Zoom ab, kann aber zu Schönheitszwecken nachträglich mit dem *Skalieren*-Werkzeug verändert werden. In den Optionen gibt es, ähnlich wie beim *Seil*-Werkzeug, die Möglichkeit eine Brechgrenze einzustellen. Interessanter ist aber die Option *Motor*. Ist diese aktiviert, dreht sich die Achse in die vorgegebene Richtung mit der eingestellten maximalen Drehzahl. Das Drehmoment ist dafür ausschlaggebend, wie schnell die maximale Drehzahl erreicht wird. Wird eine Tastenbelegung für den Motor gewählt, wird dieser deaktiviert, da er jetzt über die Tasten gestartet wird.



Fixieren-Werkzeug Dieses Werkzeug heftet ein Objekt an den Hintergrund oder ein dahinter liegendes Objekt. Anders als bei Achsen ist das Objekt komplett befestigt und bleibt von Krafteinwirkungen völlig unberührt. Ähnlich wie beim *Achsen*-Werkzeug ist die Größe der Fixierung völlig unerheblich und vom Zoom abhängig. Ein unschönes Symbol ziert dann die Szene. Unter *Rechtsklick* → *Aussehen* kann der Regler *Lichtdurchlässigkeit* (siehe Seite 20) auf 0 % gestellt werden, wodurch das Symbol durchsichtig wird.





Schubdüse Die Schubdüse kam mit dem Update auf Version 2 zu Algodoo hinzu und wird einfach mit einem Klick an ein Objekt geheftet. Sie übt eine konstante Kraft (Voreinstellung 5 N) auf das Objekt aus. Diese kann unter *Rechtsklick* → *Schubdüse* verändert werden. Wird die Option *Schubdüsen folgen Geometrierotationen* deaktiviert, behält sie ihre Ausrichtung bei und überträgt dem Objekt eine Kraft in eine konstante Richtung. Ansonsten dreht sie sich mit dem Objekt mit.




Laser Laser werden ebenfalls mit einem Linksklick platziert, können jedoch auch an den freiem Hintergrund geheftet werden. Hält man die Maustaste gedrückt, kann man auch die Ausrichtung des Lasers bestimmen. Ansonsten nimmt man hierfür das *Dreh*-Werkzeug. Die Farbe des Lasers ist zunächst zufällig, kann aber über *Rechtsklick* → *Aussehen* nachträglich bestimmt werden. Unter *Rechtsklick* → *Laserstifte* kann man noch einige Einstellungen vornehmen, wie z.B. die *Abblenddistanz*, nach der das Licht des Lasers verblasst oder die *Lichtgeschwindigkeit*. Verringert man diese, breitet sich das Licht nicht mehr geradlinig aus, sondern wie beim schrägen Wurf. Die Größe ist auch bei diesem Werkzeug aus unerfindlichen Gründen beim Platzieren vom Zoom

abhängig und kann hier nachträglich korrigiert werden. Beim Laser allerdings hat die Größe Auswirkung auf die Breite des Lichtstrahls. Die Option *Cutter* bewirkt, dass der Laser wie das *Messer*-Werkzeug durch Objekte schneidet.

 **Leuchtspur** Eine Leuchtspur wird an ein Objekt geheftet, um dessen Bewegung besser verfolgen zu können. Sie ist zunächst ein runder Punkt, der eine Spur hinter sich herzieht (kein Stroboskopbild). Auch hier ist die Größe der Leuchtspur vom Zoom abhängig und sollte nachträglich unter *Rechtsklick* → *Leuchtspur* angepasst werden. Mit der *Leuchtzeit* kann man die Dauer einstellen, bis die Spur wieder verblasst.

 **Texturenwerkzeug** Hat man einem Objekt eine Textur zugewiesen, kann man diese mit diesem Werkzeug bearbeiten. Darauf wird in Kapitel 3.2.3 auf Seite 20 näher eingegangen, in dem das *Aussehen*-Menü besprochen wird.

 **Sketch-Werkzeug** Das *Sketch*-Werkzeug ist ein besonders nütliches Werkzeug. Es vereint nämlich alle bisher beschriebenen Werkzeuge in einem und ist von der Mausbewegung abhängig. Je nachdem, welche Figur gezeichnet wird, ist die Auswirkung unterschiedlich. Ein einfacher Stich nach unten beispielweise startet oder pausiert die Simulation. Eine genaue Beschreibung dieses Werkzeuges würde hier den Rahmen sprengen, jedoch gibt es unter *Hilfe* ein eigenes Tutorial für dieses Werkzeug, wo alle Formen beschrieben werden.

Das Sketch-Werkzeug wurde in erster Linie für den Einsatz auf interaktiven Whiteboards entwickelt, um praktisch alle Werkzeuge gleichzeitig zur Hand zu haben.

3.2.2. Das Material-Menü

Im Materialmenü werden sämtliche Einstellungen zu den physikalischen Eigenschaften getätigt.

Im oberen Teil stehen vorgegebene Materialien zur Auswahl. Diese Vorgaben stellen die einzelnen Eigenschaften auf für das jeweilige Material typische Werte und geben dem Objekt eine entsprechende Textur.

Diese Eigenschaften können natürlich auch separat eingestellt werden.

Dichte Wie in Abbildung 2 zu sehen ist, wird die Dichte in Algodoo zweidimensional berechnet und hat deswegen die Einheit $\frac{kg}{m^2}$. Das macht auch Sinn, da die Objekte in Algodoo nur eine Fläche aber kein Volumen besitzen. Aus der Dichte berechnet sich dann auch die Masse des Objektes.

Masse Masse und Dichte sind voneinander abhängig, das heißt, eine Veränderung des einen Wertes ergibt auch eine Veränderung des anderen Wertes. Die Masse eines Objektes ist sehr wichtig, da sie Einfluss auf Trägheit und Anziehungskräfte hat.

Reibung Dieser Wert gibt die Stärke der Reibung an. Ein Wert von 0 bedeutet ein reibungsfreies Objekt. Zu beachten ist, dass auch der Reibungswert des zweiten Objektes Einfluss hat.



Abb. 2: *Materialmenü*

Stoßelastizität Dieser Wert gibt an, wieviel Prozent der Energie bei einer Kollision zweier Objekte erhalten bleibt. Bei einem Wert von 0,9 bleibt beispielsweise 90 % der Energie erhalten. Stoßen zwei Körper mit je 0,9 bleiben $0,9 \cdot 0,9 = 0,81$, also 81 %, erhalten. Folglich steht ein Wert von 0 für einen vollkommen unelastischen Stoß und ein Wert von 1 für einen vollkommen elastischen Stoß. Bei Szenen, in denen keine Energie verloren gehen soll, muss also die *Stoßelastizität* auf 1 gestellt werden. Auch Werte größer 1 sind möglich, dann wird dem Körper Energie zugeführt, wie es zum Beispiel in Kapitel 4.1.8 auf Seite 49 genutzt wird.

Brechzahl Sind Objekte lichtdurchlässig (siehe nächstes Kapitel), werden die Lichtstrahlen von Lasern an der Oberfläche gebrochen. Die Brechzahl kann man hier einstellen. Legt man den Wert auf Unendlich, spiegelt die Oberfläche das Licht komplett.

Anziehung Die Anziehung repräsentiert die Gravitationskonstante G . Da in den Szenen allerdings nicht mit Massen gearbeitet werden kann, die groß genug sind, um realistische, aber spürbare Anziehungskräfte zu erhalten, muss man eben die Gravitationskonstante den Bedingungen anpassen, um vernünftige Kräfte zu bekommen. Der Wert kann auch negativ gewählt werden, wodurch es möglich wird, die Anziehungskräfte, die eigentlich Gravitationskräfte darstellen sollen, als elektrische Kräfte zu benutzen.

Killer und Unsterblich Ist die erste Option aktiviert, werden alle anderen Objekte, die der *Killer* berührt, entfernt. Die Begrenzungswände am Rand jeder Szene sind zum Beispiel Ebenen mit der *Killer*-Option, so dass alles, was den Rand berührt, gelöscht wird. Die *Unsterblich*-Option hingegen verhindert, dass ein Objekt durch einen *Killer* gelöscht wird.

3.2.3. Das “Aussehen-Menü”

Im Aussehen-Menü lassen sich optische Einstellungen vornehmen. Es ist über die Materialeigenschaften (siehe Materialeigenschaften in Kapitel 3.1 auf Seite 14) zu erreichen. Ist dabei ein Objekt markiert, bezieht sich das Menü auf dieses Objekts. Ist keines markiert, kann man Voreinstellungen für neue Objekte treffen. Dabei können Bereiche angegeben werden, aus denen die Werte für das neue Objekt gewählt werden. Das objektgebundene Menü kann auch über *Rechtsklick* → *Aussehen* auf ein Objekt aufgerufen werden.



Abb. 3: *Aussehen-Menü*

Farbwerte Im obersten Abschnitt des Menüs kann die Farbe des Objekts gewählt werden. Dafür sind die oberen drei Regler zuständig. Der vierte Regler stellt den *Alphawert* ein, also dessen Durchsichtigkeit. 0 % bedeutet völlig durchsichtig und 100 % bedeutet undurchsichtig. Dies hat nicht nur optische Auswirkungen, denn der *Alphawert* legt auch fest, wie viel Laserlicht durchgelassen wird. Glas und Eis, als vorgefertigte Materialien, haben beispielsweise einen niedrigeren *Alphawert*.

Kanten Diese beiden Optionen sind rein optischer Natur und beeinflussen die Umrahmung von Objekten. Welche Einstellungen hier vorgenommen werden, sind Geschmacksache.

Einblendungen Bei der Option *Kuchenstücke* wird ein Kuchenstück bei Kreisobjekten eingeblendet, um deren Drehung zu visualisieren. *Maßstab* wird nur bei Rechteckobjekten eingeblendet und fügt diesen ein Lineal hinzu, welches abhängig vom

Zoom der Szene ist. Zoomt man näher hinein, wird auch der Maßstab detaillierter. Ähnlich funktioniert die *Gradskala* für Kreisobjekte.

Visualisierung Bei diesen drei Optionen lassen sich Kräfte-, Geschwindigkeits- und Impulsvektoren bei Objekten anzeigen. Detailliertere Einstellungen hierzu finden sich im *Materialmenü* oben rechts auf der Arbeitsfläche. Dort kann eingestellt werden, welche Arten von Kräften angezeigt werden und diese können benannt werden.

3.2.4. Gravitation und Luftreibung

Gravitation Hier kann die Gravitationskraft an- und abgeschaltet werden. Zusätzlich können hier diverse Einstellungen getätigt werden.

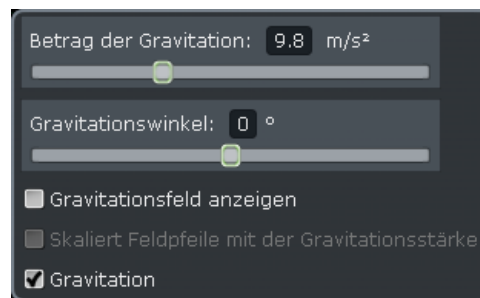


Abb. 4: Gravitationseinstellungen

Der Betrag des Ortsfaktors g kann eingestellt werden. Standardmäßig steht er natürlich auf $9,81 \frac{m}{s^2}$, kann aber beliebig eingestellt werden, sogar negativ. Der *Gravitationswinkel* legt den Winkel fest, um dem die Gravitation geneigt ist.

Eine nützliche Option versteckt sich hinter *Gravitationsfeld anzeigen*. Ist sie aktiviert, wird in der Arbeitsfläche das Gravitationsfeld als Vektorfeld angezeigt. Dies zeigt aber nicht nur die normale Gravitation an, sondern berücksichtigt auch Gravitationskräfte, die von Objekten mit Anziehungskräften ausgehen (siehe Abschnitt 3.2.2 auf Seite 18). Da man auch negative Anziehungskräfte vergeben kann, kann diese Option wie schon beschrieben dazu benutzt werden, elektrische Felder zu visualisieren.

Luftreibung Die Luftreibung ist in Algodoo sowohl linear als auch quadratisch von der Geschwindigkeit v des Objekts abhängig. Wie stark diese Abhängigkeit ist, kann man einstellen. Zusätzlich ist die Luftreibung vom Durchmesser D des Objekts in Bezug auf die Bewegungsrichtung abhängig.

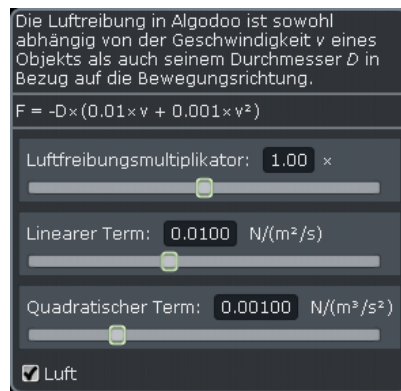


Abb. 5: *Luftreibung*

Die Formel für die Luftreibungskraft in Algodoo lautet also:

$$F = -D \cdot k \cdot (c_1 \cdot v + c_2 \cdot v^2)$$

c_1 und c_2 sind jeweils Vorfaktoren für den linearen und den quadratischen Term von v und können mit einem Regler verändert werden. Damit ist es möglich die Abhängigkeit von der Geschwindigkeit sehr genau zu bestimmen.

Mit dem Faktor k , dem Luftreibungsmultiplikator, ist ein zusätzlicher Einfluss auf die Luftreibungskraft möglich. Damit kann sie nochmals erhöht oder verringert werden.

3.3. Weiterführendes

In diesem Kapitel wird auf Funktionen eingegangen, mit denen man die Szenen detaillierter beeinflussen kann.

3.3.1. Kollisionsgruppen

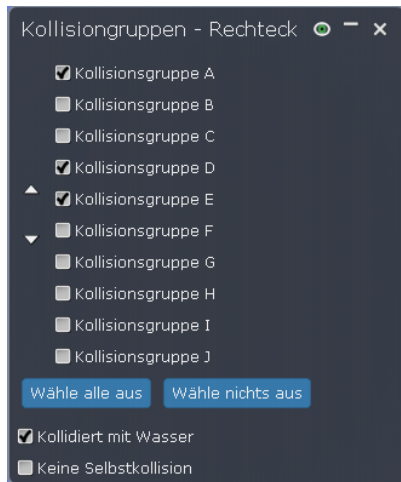


Abb. 6: *Kollisionsgruppen*

Über *Rechtsklick* → *Kollisionsgruppen* können den einzelnen Objekten verschiedene Kollisionsgruppen von A bis J zugewiesen werden. Gehören zwei Objekte der selben Gruppe an, kollidieren sie miteinander, ansonsten bewegen sie sich durcheinander hindurch. Sämtliche Objekte haben als Standardgruppe A eingestellt. Eine Ausnahme ist sowohl die Ebene, welche standardmäßig alle Kollisionsgruppen besitzt, als auch Wasser, welches gar keine Kollisionsgruppen besitzt, sondern mit allem kollidiert. Soll ein Objekt Wasser hindurch lassen, muss die Option *Kollidiert mit Wasser* deaktiviert werden.

Es kann eine beliebige Kombination aus Kollisionsgruppen ausgewählt werden, was bei größeren und komplexeren Szenen durchaus notwendig sein kann. Objekte, die mit dem *Fixieren*-Werkzeug zusammengeheftet wurden, kollidieren nie miteinander, selbst wenn sie den gleichen Gruppen angehören.

3.3.2. Graphen

Mit Graphen können verschiedene physikalische Eigenschaften von Objekten während einer laufenden Simulation beobachtet werden. Mit *Rechtsklick* → *Graph anzeigen* ruft man das entsprechende Fenster auf. Die Achsen können beliebig beschriftet werden. Standardmäßig wird zunächst ein Zeit-Schnelligkeits-Diagramm gezeigt.

Es bietet sich natürlich an, auf die x-Achse die Zeit oder die x-Position zu legen und auf die y-Achse die zu beobachtende Größe. Es können dabei auch mehrere Größen eingezeichnet werden, wobei dann zwei oder mehr Graphen übereinander gezeichnet werden. Hier kann allerdings keine beliebige Kombination gewählt werden. Das Achsen-Menü wurde dazu, den möglichen Kombinationen entsprechend, gruppiert. So kann man beispielsweise sämtliche Energien gleichzeitig betrachten, aber nicht Geschwindigkeit und Drehimpuls. Wurden mehrere zusammengehörige Größen gewählt, können diese anschließend beliebig ein- und ausgeblendet werden.

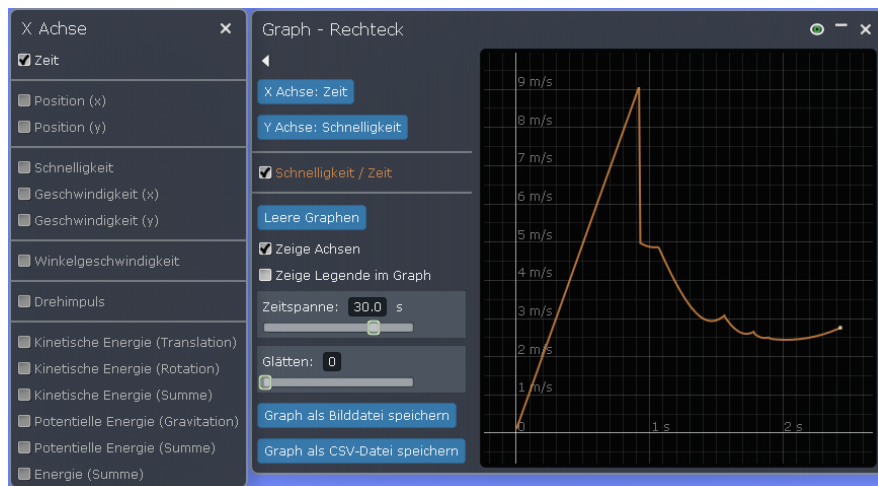


Abb. 7: Graphen-Fenster mit Achsenmenü

Die *Zeitspanne* gibt an, welches Zeitintervall angezeigt wird und mit dem *Glätten*-Wert kann der Graph ein wenig schöner gezeichnet werden, wobei hier von hohen Werten abzuraten ist, da diese das Ergebnis teilweise stark verfälschen.

Eine Schwachstelle der Graphen ist die Tatsache, dass man nicht die Graphen von verschiedenen Objekten gleichzeitig betrachten kann, um beispielsweise Bewegungen zu vergleichen.

3.3.3. Geometrieaktionen

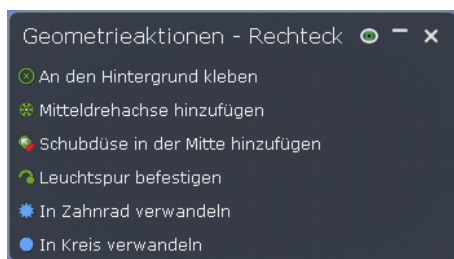


Abb. 8: Geometrieaktionen

Geometrieaktionen, die über *Rechtsklick* → *Geometrieaktionen* zu erreichen sind, erleichtern manche Dinge. *An den Hintergrund kleben* hat den selben Effekt wie das *Fixieren*-Werkzeug, mit dem Unterschied, dass kein Symbol zu sehen ist. Das hat zwar nur einen optischen Effekt, das Fixieren über das Rechtsklickmenü eines Objektes ist allerdings ein wenig komfortabler. Ein fixiertes Objekt kann über den gleichen Menüpunkt wieder gelöst werden.

Einen größeren Vorteil bringt *Mittelachse hinzufügen*. Gegenüber dem *Achsen*-Werkzeug platziert diese Option die Achse immer im Schwerpunkt des Objektes, was bei unregelmäßigen Objekten mit dem *Achsen*-Werkzeug sehr schwierig ist. Selbst wenn der Schwerpunkt außerhalb des Objektes liegt, wird auch die Achse außerhalb platziert.

Ähnlich funktioniert die *Leuchtspur hinzufügen*-Option, welche eine Leuchtspur in den Schwerpunkt eines Objektes setzt. Die Optionen *In Zahnrad verwandeln*, *In Rechteck*

verwandeln oder *In Kreis verwandeln* wandeln beliebige Objekte entsprechend um und versuchen, soweit möglich, die Größenverhältnisse beizubehalten.

3.3.4. Die CSG-Optionen

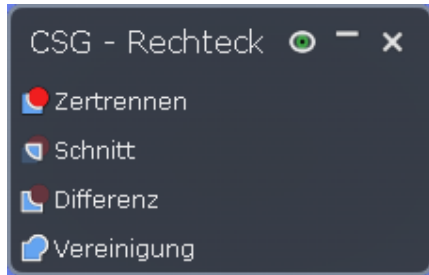


Abb. 9: CSG-Menü

Die *CSG*-Optionen (Abkürzung für *Constructive Solid Geometry*) sind sehr nützlich und teilweise sogar notwendig, um komplexe und exakte Figuren zu erstellen. Ein zweites Objekt funktioniert gewissermaßen als Schablone und wird mit dem zu bearbeitenden Objekt überlappt.

Zertrennen schneidet dann an den Rändern der Maske einen Teil des Objektes aus. *Schnitt* funktioniert ähnlich, nur dass dabei nur der überlappte Teil übrig bleibt, der Rest wird gelöscht. *Differenz* funktioniert genau anders herum und der überlappte Teil wird gelöscht, während der Rest übrig bleibt. *Vereinigung* fügt die Schablone an das Objekt an.

Die *CSG*-Optionen verbessern eine Unzulänglichkeit von Algodoo. Es ist nämlich kaum möglich, präzise Objekte zu zeichnen, da Zeichnen-Optionen wie man sie aus anderen Programmen wie zum Beispiel *GeoGebra* kennt, nicht vorhanden sind. Mit den *CSG*-Optionen wird dies immerhin indirekt möglich.

3.3.5. Komponenten

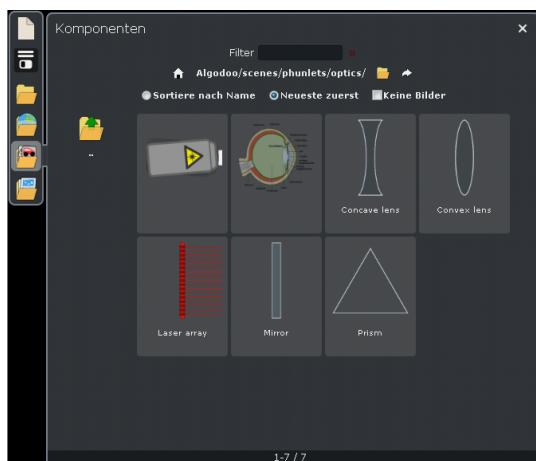


Abb. 10: Komponenten-Menü

Komponenten sind abgespeicherte, vorgefertigte Konstrukte. Man kann alles, was man selbst erstellt, in den Komponenten abspeichern, um es später mit wenigen Mausklicks wieder auf die Arbeitsfläche zu holen. Im Kapitel 4.2.1 auf Seite 52 wird ein solches Objekt abgespeichert und später in Kapitel 4.2.2 wieder verwendet.

Um dies zu tun, markiert man alles, was man abspeichern möchte, und verschiebt es über den *Komponenten-Button*. Das *Komponenten*-Menü öffnet sich und der zuletzt geöffnete Ordner wird angezeigt. Dort wird das zu speichernde Objekt zunächst namenlos abgelegt. Fährt

man mit dem Mauszeiger über das Symbol, erscheinen zwei Knöpfe. Der eine zum Löschen, der andere zum Umbenennen des Objekts.

Komponenten kann man einfach in einer neuen Szene platzieren, indem man wieder das Menü öffnet und das entsprechende Objekt anklickt. Es wird dann automatisch in der Szene platziert.

3.4. Die Skriptsprache Thyme

Die Skriptsprache Thyme (übersetzt: Thymian) ermöglicht es, die Szenen in Algodoo noch komplexer und präziser zu gestalten. Sehr wichtig ist dabei die Möglichkeit, jedem Objekt ein eigenes Skript zu geben, das bei einer Kollision ausgelöst wird, oder wenn es von einem Laser getroffen wird. Leider wird Thyme von Algoryx nicht offiziell unterstützt, weswegen man von Algoryx selbst keinerlei Informationen diesbezüglich bekommt. Die einzige Quelle bleibt deshalb das Algodoo-Forum, welches unter www.algodoo.com/forum zu erreichen ist. Im dortigen Unterforum **Thyme scripting** tauschen sich Nutzer des Forum untereinander aus. Wie Thyme funktioniert, haben die Forennutzer selbst herausgefunden und teilen ihre Erkenntnisse hier.

Ein paar nützliche Handgriffe mit Thyme werden nun in diesem Kapitel vorgestellt. Da sich viele Bezeichnungen an gängigen Programmiersprachen orientieren, ist es von Vorteil, aber nicht zwingend notwendig, wenn man sich bereits ein wenig in diesem Gebiet auskennt.

Grundsätzliches Es gibt zwei Möglichkeiten, in Algodoo ein Skript einzugeben. Zum einen kann man mit der F10-Taste die Konsole öffnen und dort das Skript hineinschreiben. Hier können Variablen erstellt und Funktionen und Befehle gegeben werden. Leider ist es nicht möglich, von der Konsole aus direkt ein Objekt in der Szene anzusprechen. Um mit Objekten zu interagieren, muss man die zweite Möglichkeit nutzen. Mit *Rechtsklick* → *Skriptmenü* öffnet sich ein Fenster mit einer ganzen Reihe von Variablen, die die Eigenschaften des Objekts festlegen. Alles, was man im *Material*-Menü oder *Aussehen*-Menü eingestellt hat, findet sich hier wieder, so gibt zum Beispiel *density* den Wert der Dichte an. Versteckt zwischen all den Variablen befinden sich die beiden Ausnahmen *onCollide* und *onHitByLaser*. In diese beiden Felder kann man Skriptbefehle eintragen, die entweder bei einer Kollision oder bei einem Treffer durch einen Laser ausgelöst werden. In diesen Befehlen kann man, anders als in der Konsole, direkten Bezug auf das Objekt nehmen. Wie dies genau funktioniert, wird weiter unten erklärt.

Variablen und der Scene.my-Container Variablen werden in Thyme wie üblich zugewiesen, zum Beispiel `x := 3`. Die Variablentypen orientieren sich dabei an gängigen Programmiersprachen, werden von Thyme aber automatisch angepasst, weswegen man sich darüber keine Gedanken machen braucht - außer man will der Variable einen *String*, also eine Verkettung von Zeichen, zuweisen. Dann muss diese Zeichenkette in Anführungszeichen stehen, damit Algodoo weiß, dass es sich bei dieser Variable um einen String handelt.

```
word := "Hallo Welt!"
```

In *Thyme* gibt es die zwei Zuweisungen `=` und `:=`. Es sollte immer letztere Möglichkeit gewählt werden, da es mit der ersten zu Problemen kommen kann.

Gibt man eine solche Variable ein, ist sie nur für diese Sitzung gespeichert und wird wieder gelöscht, sobald man das Programm schließt. Um eine Variable dauerhaft der Szene hinzuzufügen, muss man einen *Container* benutzen. Dieser heißt *Scene.my* und alles, was diesem Container hinzugefügt wird, wird auch mit der Szene gespeichert. Das heißt, auch wenn ein anderer Benutzer diese Szene lädt, zum Beispiel aus der Algobox, wird der *Scene.my*-Container und sein Inhalt mitgeladen und die Variablen sind verfügbar. Andernfalls würde die Szene nicht richtig funktionieren, da wichtige Variablen nicht im Container gespeichert wären. Also sollte die Zuweisung von oben besser so aussehen:

```
Scene.my.x := 3
```

Arrays Arrays sind spezielle Variablen, die wiederum eine gewisse Anzahl an Variablen zusammenfassen. So ist die Variable `pos = [2, 5]` ein Array, der die x- und y-Position als Zahlen speichert. Die Variablen werden immer in eckigen Klammern zusammengefasst und mit einem Komma getrennt. Die einzelnen Bestandteile eines Arrays lassen sich über die Positionen der Variablen ansprechen, wobei die erste Position mit 0 beginnt. `pos(0)` beispielsweise würde die x-Position, nämlich den Wert 2 wiedergeben, `pos(1)` den Wert 5. Es ist auch möglich, Arrays ineinander zu verschachteln. So wäre `bsp = [[3, 1], [4, 7, 2]]` ein Array, der wiederum zwei Arrays unterschiedlicher Längen enthält und `bsp(1)(1)` würde den Wert 7 zurückgeben.

Im Unterschied zu den meisten Programmiersprachen können Arrays in Algodoo Variablen verschiedener Typen speichern.

Objekte in die Szene einfügen Wie bereits erwähnt, gibt Algodoo keine Rückschlüsse auf die genauen Positionen von Objekten. Mit Thyme ist es jedoch möglich, Objekte an

3. Einführung in Algodoo

bestimmten Koordinaten zu platzieren. Jedes Objekt, für das es ein Werkzeug gibt, kann auch mit Thyme platziert werden. Hier wird allerdings nur auf die drei wichtigsten eingegangen - nämlich: Hinzufügen von Rechtecken, Kreisen und Laserstifen. Die Befehle hierfür lauten:

```
Scene.my.addBox({})
Scene.my.addCircle({})
Scene.my.addLaserPen({})
```

Wie man sieht, werden auch diese Objekte dem *Scene.my*-Container hinzugefügt. Dem Befehl folgen Klammern. Die runden Klammern enthalten die Argumente, welche das Objekt näher beschreiben. Diese Argumente werden nochmals mit den geschweiften Klammern gruppiert, welche selbst dann vonnöten sind, wenn man nichts einfügt. Mehrere Argumente werden mit einem Semikolon getrennt. Wichtige sind:

<code>pos = [x,y]</code>	Gibt die Position in x- und y-Koordinaten an
<code>angle = α</code>	Gibt den Drehwinkel α im Bogenmaß an
<code>colideSet = x</code>	Gibt die Kollisionsgruppe als natürliche Zahl x an, wobei diese eine Gruppenkombination repräsentiert.
<code>drawBorder = true/false</code>	Kanten zeichnen ja oder nein
<code>colorHSVA = [h,s,v,a]</code>	Die Farbe des Objekts in den HSVA-Werten aus dem <i>Aussehen</i> -Menü
<code>color = [a,b,c,d]</code>	Alternative Farbcodierung mit Zahlen a,b,c,d zwischen 0 und 1
<code>radius = r</code>	Gibt für <code>addCircle</code> den Radius des Kreises an
<code>size = [b,h]</code>	Gibt Breite und Höhe eines Rechtecks bei <code>addBox</code> an

Es gibt noch viele weitere Argumente. Im *Skriptmenü* eines Objektes kann man noch weitere Objektvariablen (nicht alle) ablesen. Dort kann man auch den Farbcode oder die Kollisionsgruppe ablesen. Man erstellt ein Objekt mit der gewünschten Farbe bzw. Kollisionsgruppe und kopiert dann deren Werte.

Folgender Befehl fügt zum Beispiel einen Kreis mit einem Radius von 0,5 m, der Farbe schwarz und ohne Kuchenstück an der Position (3,5 ; 5,76) hinzu:

```
Scene.my.addCircle({ pos := [3.5 , 5.76] ; radius := 0.5 ;
                    color := [0, 0, 0, 1] ; drawCake := false })
```

onCollide und onHitByLaser Im Skript-Menü eines Objektes stehen die beiden Einträge `onHitByLaser` und `onCollide` mit jeweils dem Eintrag (e)=>{}. Das bedeutet,

dass das Skript, das in den geschweiften Klammern steht, ausgeführt wird, wenn das Objekt entweder von einem Laser (`onHitByLaser`) oder von einem anderen Objekt (`onCollide`) getroffen wird. Dieses Ereignis `e` wird dem Skript zugeordnet, welches zunächst natürlich noch leer ist. In den geschweiften Klammern kann jede beliebige Art von Skript stehen. So würde bei `onHitByLaser` folgendes Skript einen Kreis in der Szene erstellen:

```
(e)=>{ Scene.my.addCircle({...}) }
```

Im Unterschied zur Konsole kann man hier auch das Ereignis `e`, welches das Skript ausgelöst hat, verwenden. Das Ereignis `e` besitzt ebenfalls Eigenschaften wie zum Beispiel den Ort des Treffers `pos`. Diese können abgefragt werden, indem sie mit einem Punkt mit `e` verbunden werden. `e.pos` gibt also den Positions-Array des Treffers zurück. Folgendes Skript erstellt einen Kreis direkt am Trefferpunkt:

```
(e)=>{ addCircle({ pos := e.pos }) }
```

Die Position, an der der neue Kreis erstellt wird, soll die gleiche wie die des Ereignisses sein. Ebenfalls können bei `onCollide` die beiden kollidierten Objekte und bei `onHitByLaser` Objekt und Laser angesprochen werden. Bei `onCollide` sind die Schlüsselwörter `this` für das Objekt mit dem Skript und `other` für das andere Objekt. So würde folgendes Skript die Farbe des geskripteten Objekts in weiß und die des anderen Objekts in schwarz verändern:

```
(e)=>{ e.this.color := [1,1,1,1] ;  
      e.other.color := [0,0,0,1]  
      }
```

`onHitByLaser` funktioniert genauso, nur lauten hier die beiden Schlüsselwörter `geom` für das getroffenen Objekt mit Skript und `laser` für den Laser.

If-Struktur Eine If-Struktur besteht aus einem Vergleich und zwei Skripten. Ist der Vergleich wahr, wird das erste Skript ausgeführt, ansonsten das zweite. Sie hat die Form:
Vergleich ? {...} : {...}

Ein Beispiel wäre:

```
x > 5 ? {'x ist größer 5'} : {'x ist kleiner oder gleich 5'}
```

Dies gibt entweder den String "x ist größer 5" oder "x ist kleiner oder gleich 5" zurück, je nachdem wie der Vergleich ausgefallen ist. Mathematische Vergleichsoperatoren sind: `<`, `<=`, `>`, `>=`, `==`, `!=`, wie aus anderen Programmiersprachen bekannt.

For-Schleife Eine For-Schleife wird so lange wiederholt, bis eine Zählvariable eine obere Grenze erreicht hat. In folgendem Beispiel werden fünf Laser im Abstand von jeweils 1 m platziert:

```
for(5, (k)=>{ addLaserPen({ pos := [0,k] }) } )
```

Die Zählvariable `k` wird von 0 bis 4 gezählt und jedesmal das Skript in den geschweiften Klammern ausgeführt. Das erste mal wird also ein Laser bei `[0,0]` platziert, beim nächsten Durchlauf ein zweiter bei `[0,1]` und so weiter. 5 ist hierbei nicht die Obergrenze, sondern zeigt an wie oft `k` erhöht wird. Da die Zählvariable aber immer bei 0 beginnt, zählt sie hier nur bis 4.

Die For-Schleife in Algodoo kann nicht für beliebig viele Wiederholungen verwendet werden, sondern nur für etwa 65. Bei zu vielen gibt *Thyme* statt dessen eine Fehlermeldung aus.

Thyme ab Version 2.0.1 Bislang war es nicht möglich, momentane Positionen von Objekten zu lesen. Die Variable `pos` konnte lediglich zur Platzierung genutzt werden. Ebenso war es nicht möglich, ein Objekt direkt im Skript anzusprechen, ohne es vorher umständlich über ein `onHitByLaser`-Skript in einer Variablen abzuspeichern.

Mit der Algodoo-Version 2.0.1, die kurz vor Fertigstellung dieser Arbeit veröffentlicht wurde, wurde auch *Thyme* stark verbessert. Das Skriptmenü wurde übersichtlicher gestaltet, die Variable `pos` kann jetzt jederzeit gelesen werden und gibt die momentane Position wieder und jedem Objekt wird bereits beim Erstellen eine Zahl zugewiesen, über die es im Skript ansprechbar ist. Dies konnte leider nicht mehr berücksichtigt werden, weshalb die Szenen in dieser Arbeit noch mit der alten *Thyme*-Version erstellt wurden.

3.5. Algobox

Algobox ist ein Dienst von Algoryx, bei dem man selbst erstellte Szenen auf einen Algoryx-Server hochladen kann. Andere Nutzer können diese Szene herunterladen und nutzen. Bislang war Algobox nur auf der Homepage von Algodoo erreichbar, aber ab Version 2.0.0 wurde Algobox direkt in das Programm integriert, was einen schnellen und bequemen Zugriff ermöglicht. Da allerdings jeder Szenen hochladen kann, ist Algobox mittlerweile sehr überfüllt und nur ein Bruchteil aller Szenen ist überhaupt für den Unterricht nutzbar. Es gestaltet sich mitunter etwas schwierig, in der Fülle an verschiedenen Szenen das herauszupicken, was man gerade sucht.

Man kann sich bei Algobox anmelden und hat dann die Möglichkeit, Szenen zu kommentieren und zu bewerten. Die Kommentare und Bewertungen anderer Nutzer geben immerhin einen kleinen Überblick über die Qualität der jeweiligen Szene. Grundsätzlich jedoch muss man schlichtweg verschiedene Szenen ausprobieren, um festzustellen, ob sie geeignet sind oder nicht.

Die Szenen eignen sich oft auch dazu, um sich Ideen zu holen. Das kann sich aber durchaus als problematisch erweisen, da vieles mit Thyme umgesetzt wird und es keine Möglichkeit gibt, lokale Variablen und Skript, die in die Konsole eingegeben wurden, einzusehen. Lediglich Skripte, die Objekten bei `onHitByLaser` oder `onCollide` zugeschrieben wurden, sind sichtbar. Manche Objekte sind oft auch versteckt, da sie zum Beispiel unsichtbar sind oder von anderen Objekten verdeckt und diese angeklebt sind.

Ein Beispiel für das Arbeiten mit Algobox wird in Kapitel 4.4.1 auf Seite 74 gezeigt.

3.6. Lektionen

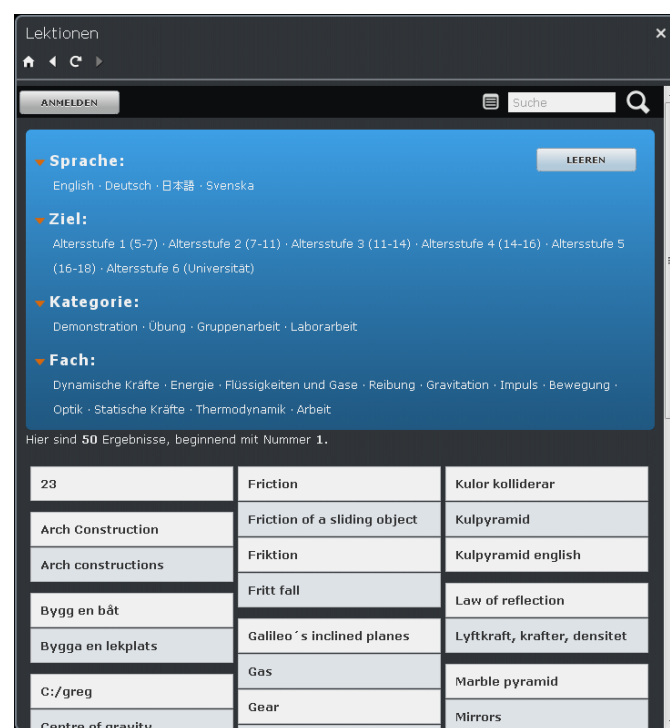


Abb. 11: Lektionen-Menü

Lektionen werden, wie in Algobox, hauptsächlich von Algodoo-Nutzern erstellt, aber auch von Algoryx selbst. Es handelt sich dabei nicht nur um vorgefertigte Szenen, sondern auch um Arbeitsaufträge für Schüler zu einem bestimmten Thema. Meistens ist

jedoch keine Szene dabei, sondern nur Aufträge. Lektionen können sozusagen als Arbeitsblätter genutzt werden.

Im *Lektionen*-Menü (siehe Abbildung 11) stehen viele Filter bereit, um die Suche einfacher zu machen. Man kann Sprachen filtern, das Alter der Zielgruppe, die Art der Lektion (Demonstration, Übung, Laborarbeit) sowie das Thema (Energie, Impuls, Optik, u.s.w.).

Zur Zeit stehen etwa 90 Lektionen zur Verfügung, davon 42 auf Englisch und nur eine auf Deutsch. Sofern die englischen Sprachkenntnisse der Schüler ausreichend sind, können die englischen Szenen im Unterricht verwendet werden. Zumindest kann man sich von diesen jedoch Ideen holen oder sie selbst übersetzen.

4. Beispielszenen

In diesem Kapitel werden einige Szenen vorgestellt und auf deren Erstellung eingegangen. Es wird vor allem auf Probleme eingegangen, die in Algodoo häufiger auftreten und auch auf einige Tricks und Kniffe, die das Erstellen solcher Szenen vereinfachen können oder mehr Möglichkeiten bieten.

Als Grundlage wird der G8 Lehrplan des bayerischen Gymnasiums benutzt. Bei den einzelnen Szenen wird auch erwähnt, wo im Lehrplan diese einzuordnen sind.²²

Es folgt eine kurze Beschreibung des Ziels der Szene und ob sie dazu gedacht bzw. geeignet ist, von Schülern genutzt zu werden oder ob sie vom Lehrer präsentiert werden sollte. Dann werden die Voraussetzungen genannt, die man zum Erstellen der entsprechenden Szene benötigt. Diese werden nur in Stichworten erwähnt und beziehen sich auf das Kapitel 3. Darauf folgt eine Anleitung zum Erstellen der entsprechenden Szene. Soll die Szene von Schülern erstellt werden, ist natürlich darauf zu achten, dass diese die dafür nötigen Kenntnisse in Algodoo besitzen. Wenn nichts anderes erwähnt wird, wird bei den Szenen, die die Schüler erstellen sollen, immer von einer leeren Algodoo-Szene ausgegangen, die die Schüler dann innerhalb gewisser Vorgaben selbst gestalten sollen, um auf die gewünschten Ergebnisse zu kommen. Dadurch werden die Stärken von Algodoo, die Anschaulichkeit und die verhältnismäßig leichte Bedienbarkeit, gut im Unterricht integriert.

Falls die der Szene zugrunde liegenden Gleichungen nicht zu kompliziert sind und man in der Szene gut die benötigten Werte ablesen kann, wird ein kurzer Vergleich zwischen dem theoretisch zu erwartenden Wert und dem in Algodoo erhaltenen Wert gezogen. Ist noch etwas zur Anwendung der Szene im Unterricht zu sagen, wird dies ebenfalls in einem eigenen Abschnitt behandelt.

4.1. Mechanik

Algodoo ist von seiner Funktion her deutlich auf die Mechanik ausgelegt. Es werden hauptsächlich die Auswirkungen von Kräften und Energien simuliert. Deshalb ist der erste Teil dieses Kapitels auch der größte.

²²ISB: Lehrpläne / Standarts Gymnasium G8. [URL: http://www.isb.bayern.de/isb/index.asp?MNav=0&QNav=4&TNav=0&INav=0&Fach=&LpSta=6&STyp=14](http://www.isb.bayern.de/isb/index.asp?MNav=0&QNav=4&TNav=0&INav=0&Fach=&LpSta=6&STyp=14).

4.1.1. Freier Fall

Einordnung in den Lehrplan Das Thema “Energie und Energieerhaltung” ist für die 8. Jahrgangsstufe vorgesehen. Die Schüler lernen erstmals den Begriff der Energie kennen und in welchen Formen diese auftritt. Der *freie Fall* ist hierfür ein Standardbeispiel.

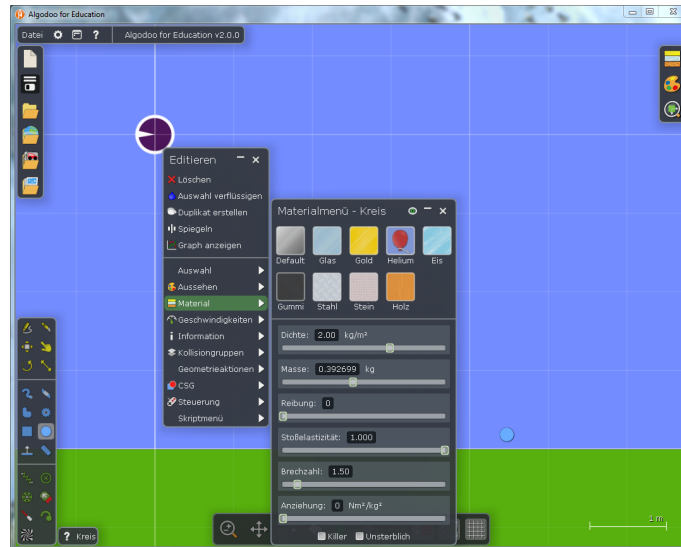


Abb. 12: Materialeigenschaften des Kreises

Ziel der Szene Bei dieser Simulation soll mit Hilfe eines Graphen gezeigt werden, wie sich beim freien Fall die kinetische und die potentielle Energie ineinander umwandeln und in der Summe erhalten bleiben. Da diese Szene sehr einfach zu erstellen ist, kann sie von den Schülern selbst erstellt und verwendet werden. Den Schülern sollte klar sein, dass für die Energieerhaltung Reibungsfreiheit und vollkommene Elastizität nötig sind. Sie können dann den freien Fall untersuchen und sich auch vom Energieverlust durch Reibung und unelastische Stöße überzeugen.

Voraussetzungen Grundlagen und Graphen

Erstellung Man erstellt eine neue Szene. Die bereits vorhandene Ebene benutzt man als Untergrund. Mit dem Kreiswerkzeug erstellt man in einer beliebigen Höhe einen Kreis. Da *Energieerhaltung* gilt, sollte jegliche Reibung und unelastische Stöße ausgeschaltet werden. Dazu klickt man mit der rechten Maustaste auf den Kreis und geht auf das *Material*-Menü. Dort stellt man die Reibung des Kreises auf 0 und die Elastizität auf 1, so dass er voll elastisch stößt. Das Gleiche muss man mit der Ebene tun, da sonst trotzdem Energie verloren ginge. Jetzt muss noch die Luftreibung deaktiviert werden, um eine

völlig reibungsfreie Umgebung zu erhalten (siehe Abbildung 12). Der Kreis springt jetzt ohne Energieverlust hoch und runter. Mit einem weiteren Rechtsklick auf den Kreis und dann auf *Graph* blendet man den zugehörigen Graphen ein. Auf der y-Achse wählt man *Kinetische Energie (Summe)*, *potentielle Energie (Summe)* und *Energie (Summe)*. Beim Zeitintervall werden 5 Sekunden eingestellt und schon ist die erste richtige Simulation fertig. Ein Druck auf die Leertaste startet die Simulation und der Kreis fängt an zu hüpfen, während parallel dazu seine Energien aufgezeichnet werden.

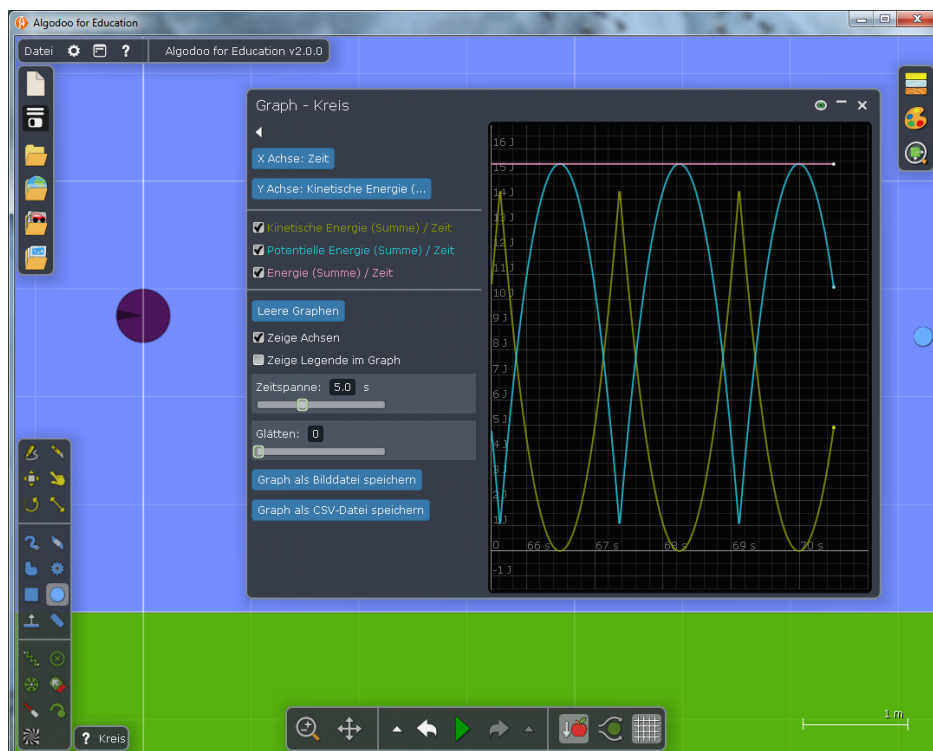


Abb. 13: *Energien beim freien Fall*

Vergleich von Theorie und Praxis In dem Graphen kann man die Zeit ablesen, die der Körper für den Fall braucht. Hier bietet es sich an, diese Zeit mit der theoretisch berechneten Zeit zu vergleichen. Aus der Energieerhaltung und $v = a \cdot t$ folgt:

$$E_{kin} = E_{pot} \quad \Rightarrow \quad \frac{1}{2}mv^2 = mgh \quad \Rightarrow \quad t = \sqrt{\frac{2h}{g}}$$

Bei einer Höhe von 4 m ergibt sich eine Fallzeit von $t = 0,90$ s. Verglichen mit der Zeit, die man am Graphen ablesen kann, passt dies gut zusammen.

4.1.2. Kräfte

Einordnung in den Lehrplan In der 7. Jahrgangsstufe kommt das Thema “Kräfte in der Natur und in der Technik” vor. Bei der Einführung der Kräfte wird unter anderem auf die Kraftvektoren eingegangen. Deren Angriffspunkte, Längen und Richtungen werden von Algodoo schön veranschaulicht.

Ziel der Szene In dieser Szene sollen Kräfte durch Vektorpfeile verdeutlicht werden. Algodoo kann diese direkt darstellen. Man kann Einfluss darauf nehmen, welche Kräfte angezeigt werden, wie lang die Vektorpfeile sind und wie sie benannt sind. Leider ist es nicht möglich, sie farblich zu unterscheiden. Die Szene, wie sie im Folgenden erstellt wird, ist als Präsentation für die Schüler gedacht. Allerdings können die Schüler auch problemlos damit selbst experimentieren. Sie können verschiedene Anordnungen selbst erstellen und dort das Zusammenspiel der wirkenden Kräfte, vor allem auch in Bewegung, betrachten,

Voraussetzungen Grundlagen und Anzeigen von Kräften

Erstellung In der ersten Situation soll die Gewichtskraft auf einen frei im Raum befindlichen Körper gezeigt werden. Man erstellt dazu ein Rechteck. Über *Rechtsklick* → *Aussehen* kommt man ins *Aussehen*-Menü, in dem man *Kräfte anzeigen* aktiviert. Über den Regler kann noch die Länge des Vektorpfeiles angepasst werden. Da die Gewichtskraft die einzig wirkende Kraft ist, stimmt sie mit der Gesamtkraft überein. Die Folge daraus ist, dass sich die Namen unschön überlagern. Dagegen kann man leider nichts machen. Da man später die Simulation kurz laufen lassen wird, bietet es sich an, das Rechteck vorerst mit dem *Fixieren*-Werkzeug zu befestigen, da es sonst aus der Szene fällt.

Die nächste Anordnung soll die Normalkraft verdeutlichen und aufzeigen, wie sich Gravitationskraft und Normalkraft, die von der Unterlage auf den Körper ausgeübt wird, ausgleichen. Dafür braucht man zunächst eine Unterlage. Man erstellt in der Nähe der ersten Anordnung ein längliches Rechteck, welches man ebenfalls an den Hintergrund anheftet. Da das *Fixieren*-Werkzeug allerdings ein unschönes Symbol hinterlässt, macht man das am besten per *Rechtsklick* → *Geometrieaktionen* → *An den Hintergrund anheften*. Darüber erstellt man einen Kreis. Da er auf der Unterlage aufliegen soll, startet man kurz die Simulation, wartet bis der Kreis still liegt und beendet die Simulation wieder. Wie bei der ersten Anordnung aktiviert man die Kraftvektoren des Kreises.

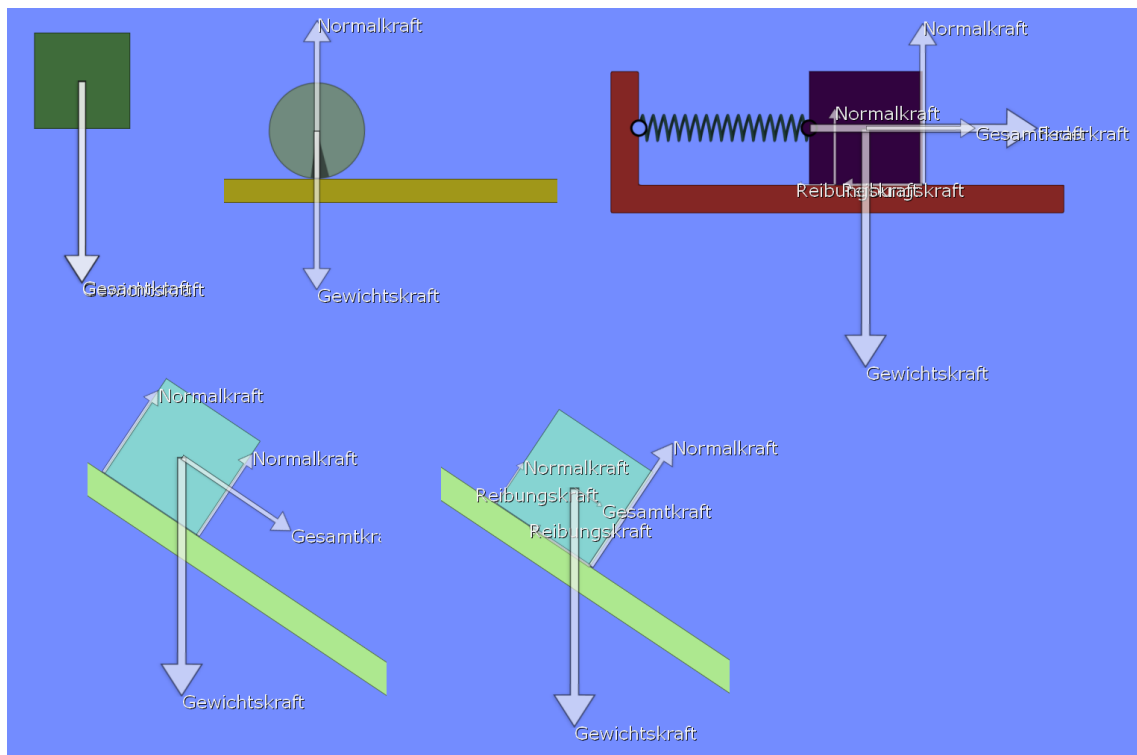


Abb. 14: Darstellung von Kraftvektoren

Bei der dritten Anordnung sollen Federkräfte betrachtet werden. Bei dieser Erstellung ist das Gitternetz hilfreich. Zunächst braucht man wie zuvor eine fixierte Unterlage. Zusätzlich benötigt man auf einer Seite noch eine Wand, an der die Feder befestigt wird. Dank des Gitternetzes kann man ein vertikales Rechteck erstellen, das passgenau mit der Unterlage überlappt. Mit *Rechtsklick* → *CSG* → *Vereinigen* fügt man das Rechteck der Unterlage hinzu und löscht es anschließend, da es nur als Schablone diente. Man erstellt ein Rechteck auf der Unterlage und lässt es durch kurzes Starten der Simulation zur Ruhe kommen. Anschließend verbindet man das neue Objekt über eine Feder mit der Wand. Noch wirken allerdings keine Federkräfte, daher verändert man über *Rechtsklick* → *Feder* die Ruhelänge. Die Kräfte des Objektes werden wie gewohnt visualisiert. Auch hier bietet es sich an, das Objekt vorläufig an den Hintergrund zu befestigen.

Die letzten beiden Beispiele behandeln die schiefe Ebene, einmal mit und einmal ohne Reibung. Es wird wieder eine fixierte Unterlage mit einem Rechteck erstellt. Beide Objekte werden markiert und mit dem *Drehen*-Werkzeug in einen beliebigen Neigungswinkel gedreht. Da man die gleiche Anordnung zweimal benötigt, kopiert man sie mit *Rechtsklick* → *Duplikat erstellen*. Bei der ersten Anordnung stellt man die Reibung beider Objekte auf 0 und fixiert das Rechteck am Hintergrund. Bei der zweiten Anordnung bleibt die Reibung natürlich bestehen. Da sich das Rechteck aber in Ruhe befindet,

treten noch keine Reibungskräfte auf. Deswegen muss die Simulation noch einmal kurz gestartet werden, sodass das Rechteck ein Stück rutscht und sich durch die Geschwindigkeit Reibungskräfte aufbauen. Hierbei ist ein wenig Fingerspitzengefühl gefragt. Bei pausierter Simulation kann man das Rechteck auch noch einmal an das obere Ende der Unterlage zurücksetzen, da es dabei seine Geschwindigkeit nicht verliert. Man sieht, dass beim Rechteck die Normalkräfte an den Auflagepunkten, in diesem Fall die beiden Ecken, angreifen.

Ist man schließlich mit allen Anordnungen und Vektorpfeilen zufrieden, werden die vorläufigen Fixierungen entfernt. Man beachte, dass auf fixierte Objekte keinerlei Kräfte wirken. Die komplette Szene ist eine Momentaufnahme, da einige Objekte beim Starten aus der Szene fallen würden.

4.1.3. Der waagrechte Wurf

Einordnung in den Lehrplan In der 10. Jahrgangsstufe wird der waagrechte Wurf im Thema “Die Mechanik Newtons” behandelt. Zum ersten Mal werden die Schüler mit einer zweidimensionalen Bewegung konfrontiert und damit, dass sich beide Bewegungsrichtungen unabhängig voneinander betrachten lassen.

Ziel der Szene Der waagrechte Wurf ist ein Standardthema in der Physik, welches aber durch einen realen Versuch schwer zu beobachten ist. Die beiden interessanten Größen, nämlich die Geschwindigkeiten in x-Richtung und y-Richtung, lassen sich kaum separat messen. Mit dieser Szene können sie aber dank der Graphen untersucht werden.

Häufig wird eine Aufgabe zum waagrechten Wurf bearbeitet, bei der die Energieerhaltung zur Bestimmung der Anfangsgeschwindigkeit v_{0x} genutzt wird. Dies soll auch in dieser Szene benutzt werden. Die Szene kann vom Lehrer vorbereitet werden, so dass die Schüler dann die verschiedenen Aspekte des waagrechten Wurfs überprüfen können, zum Beispiel die Abhängigkeit der Wurfweite von der Anfangshöhe oder der Masse des Objekts. Auch die Anfangsgeschwindigkeit v_{0x} lässt sich durch geschicktes Pausieren bestimmen. Zudem kann die Auswirkung der Luftreibung oder der Haftreibung betrachtet werden.

Voraussetzungen Grundlagen, CSG und Graphen

Erstellung Ziel dieser Simulation ist es, einen waagrechten Wurf nachzustellen, indem man ein Objekt (hier einen Kreis) über eine Rampe beschleunigt, von der es dann in waagrechtlicher Richtung wegfällt und eine Strecke der Höhe h_2 und der Länge x_w

zurücklegt. Die Anfangsgeschwindigkeit v_{0x} , mit der sich das Objekt bewegt, wird über einen freien Fall aus der Höhe h_1 ermittelt.

Um präzise arbeiten zu können, aktiviert man zunächst das Gitternetz und stellt in dessen Optionen die Segmente auf 10, da sich damit später besser Lineale anbringen lassen. Zur Konstruktion der Rampe erstellt man zunächst ein Rechteck und schneidet davon mit den *CSG*-Optionen einen Viertelkreis aus. Das Rechteck sollte nach der Rampe noch 1 *m* länger sein, da das Objekt ansonsten noch eine Geschwindigkeit in *y*-Richtung hätte. Man erstellt also ein Rechteck mit einer Höhe von 4 *m* und einer Breite von 5 *m*, sowie einen Kreis mit einem Radius von 3,5 *m*, platziert diesen wie in Abbildung 15 gezeigt und schneidet ihn mit der *CSG*-Option *Differenz* aus. Anschließend kann der Kreis gelöscht werden. Mit einem kleinen Rechteck und *CSG* oder einfach dem *Messer*-Werkzeug begradigt man noch das überstehende Endstück der Rampe.

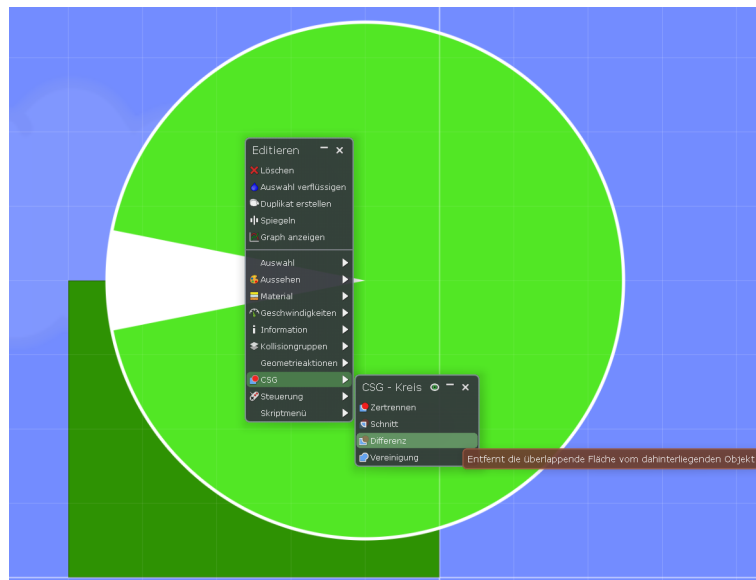


Abb. 15: Erstellung der Rampe

Nun werden die Lineale platziert. Es soll zunächst die für den waagrechten Wurf notwendige Höhe h_2 gemessen werden sowie die Fallhöhe h_1 , aus der die Geschwindigkeit v_{0x} zu berechnen ist. Zudem wird ein Lineal eingefügt, um die Wurfweite x_w zu bestimmen. Man erstellt zunächst ein Rechteck und fügt ihm über *Rechtsklick* → *Aussehen* → *Lineal* Maßeinheiten hinzu. Wahrscheinlich befinden sich diese auf der falschen Seite, sodass das Rechteck zunächst gedreht und neu skaliert werden muss. Es sollte unten auf gleicher Höhe wie das waagrechte Endstück der Rampe (siehe Abbildung 16) sein. Dieses Lineal kann einfach kopiert und unten an das Endstück angesetzt werden, um h_2 zu messen. Noch einmal kopiert und um 90° gedreht wird als Auftreffebene noch ein

Lineal platziert, das eventuell verlängert werden muss.

Als letztes platziert man einen kleinen Kreis wie in Abbildung 16 über der Rampe. Jetzt sind alle Teile der Simulation am richtigen Ort und man muss ihnen noch die notwendigen physikalischen Eigenschaften mitgeben. Zunächst werden alle Objekte außer dem Kreis über *Rechtsklick* → *Geometrieaktionen* fixiert. Die vertikalen Lineale sollen keinerlei Einfluss auf die Simulation haben, aus welchem Grund man deren Kollisionsgruppen komplett entfernt. Auf dem waagrechten Lineal soll der Kreis aufschlagen, weshalb dieses Gruppe A behält. Wenn der Kreis über die Rampe rutscht, soll er keine Energie durch Reibung oder Stöße verlieren, weswegen die Reibung beider Objekte auf 0 und die Stoßelastizität auf 1 gesetzt wird. Zum Schluss wird noch die Luftreibung deaktiviert. Damit ist die Szene fertig.

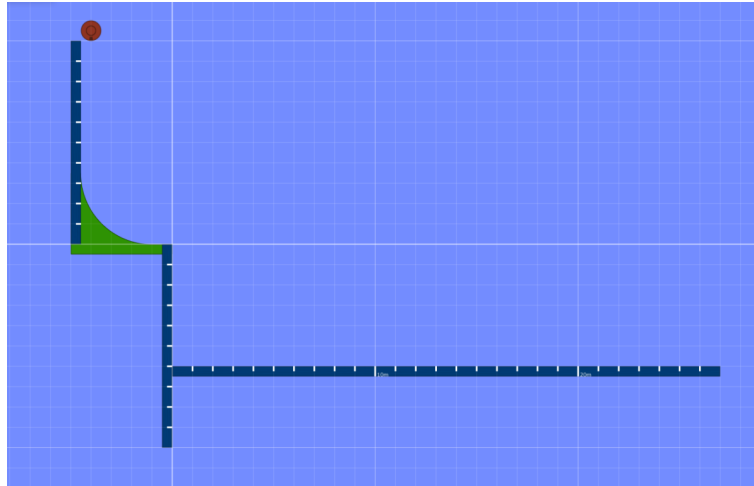


Abb. 16: *Fertige Szene*

Vergleich von Theorie und Praxis Aus der Fallhöhe h_1 , die mit dem oberen Lineal gemessen werden kann, lässt sich über Energieerhaltung v_{0x} berechnen:

$$E_{pot} = E_{kin} \Rightarrow mgh_1 = \frac{1}{2}mv_{0x}^2 \Rightarrow v_{0x} = \sqrt{2gh_1}$$

Mit den Formeln für beschleunigte und gleichförmige Bewegungen $x = \frac{1}{2}at^2$ und $x = vt$ ergibt sich schließlich:

$$h_2 = \frac{1}{2}gt^2 \Rightarrow t = \sqrt{\frac{2h_2}{g}} \Rightarrow x_w = v_x t = \sqrt{2gh_1 \cdot \frac{2h_2}{g}} = 2\sqrt{h_1 h_2}$$

In der hier erstellten Szene erhält man die Höhen $h_1 = 10 \text{ m}$ und $h_2 = 6 \text{ m}$. Damit lässt sich die Geschwindigkeit $v_{0x} \approx 14,0 \frac{\text{m}}{\text{s}}$ berechnen, sowie die Weite $x_w \approx 15,5 \text{ m}$.

Diese beiden Werte kann man mit den Werten der Simulation vergleichen. Indem man die Simulationsgeschwindigkeit verringert, kann man die Szene zum Zeitpunkt des waagrecht Abwurfes anhalten und über *Rechtsklick* → *Geschwindigkeiten* die momentane x-Geschwindigkeit ablesen. Diese betrug bei dieser Durchführung etwa $13,98 \frac{m}{s}$, was dem theoretischen Wert sehr nahe kommt. Allerdings fällt auch auf, dass noch immer eine gewisse y-Geschwindigkeit übrig bleibt. Hält man die Simulation ein zweites Mal beim Aufprall an, kann man am Lineal x_w ablesen, welche bei meinem Versuch etwa $15,8 m$ betrug.

4.1.4. Beschleunigte Bewegungen

Einordnung in den Lehrplan In der 9. Jahrgangsstufe wird das Thema “Kinematik und Dynamik einer geradlinigen Bewegung” behandelt. Dies umfasst sowohl die gleichförmige als auch die beschleunigte Bewegung. Diese Simulation dient in erster Linie dazu, die beschleunigte Bewegung zu untersuchen, kann aber, wie im Abschnitt “Alternative” beschrieben, auch für die gleichförmige Bewegung genutzt werden.

Ziel der Szene Da die Schüler zu dem Zeitpunkt, an dem die beschleunigte Bewegung eingeführt wird, noch keine Kenntnis über die Ableitung haben, ist es nicht möglich, die Zusammenhänge $a(t) = \frac{d}{dt}v(t) = \frac{d^2}{dt^2}x(t)$ einzuführen. Stattdessen muss auf die vereinfachte Form $a = \frac{\Delta v}{\Delta t}$ und $v = \frac{\Delta x}{\Delta t}$ zurückgegriffen werden und die Beschleunigung wird über das Steigungsdreieck bestimmt.

Die Graphen in Algodoo sind für eine solche Untersuchung ideal geeignet, da man an jeder Stelle die momentane Steigung und die x- und y-Werte ablesen kann. Die Schüler bekommen in dieser Szene ein vorgefertigtes Fahrzeug mit Anhänger und Ladung zur Verfügung gestellt. Dessen Zeit-Schnelligkeits-Graph kann betrachtet und daraus die Beschleunigung bestimmt werden. Die Auswirkung verschiedener Lasten oder verschiedener Steigungen auf die Beschleunigung können untersucht werden.

Erstellung Auf der beiliegenden CD befinden sich im Ordner *Texturen\Fahrzeuge* die Texturen für ein Fahrzeug, einen Anhänger und Räder. Man erstellt ein Rechteck, entfernt die Kanten und gibt ihm die Textur des Fahrzeuges. Mit dem *Messer*-Werkzeug schneidet man um das Fahrzeug herum die Form aus. Wahrscheinlich ist es noch nötig, die Proportionen des Autos mit dem *Skalieren*-Werkzeug nachzubessern. Anschließend werden an den Rädern zwei Kreise erstellt, die man mit *Rechtsklick* → *Geometrieaktionen* → *Mitteldrehachse hinzufügen* an den Achsen des Autos befestigt. Auch diesen beiden

Kreisen weist man die Textur der zugehörigen Reifen zu und passt sie mit dem *Textur*-Werkzeug an. Den Alphawert der Achsen stellt man im *Aussehen*-Menü auf 0, damit sie nicht zu sehen sind.

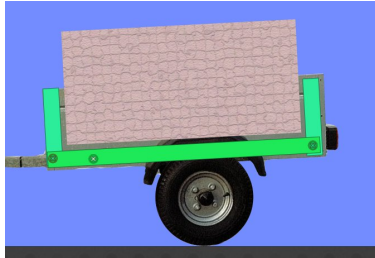


Abb. 17: *Der Anhänger*

Dasselbe wird mit dem Anhänger gemacht. Anschließend wird der Anhänger noch mit einer Ladefläche versehen. Mit dem *Rechteck*-Werkzeug erstellt man einen Boden und zwei Seitenwände, welche man mit dem *Fixieren*-Werkzeug verbindet. Die Kollisionsgruppe der Rechtecke stellt man auf B, damit die Ladung, die ebenfalls B bekommen wird, nur mit der Ladefläche, aber nicht mit dem Anhänger kollidiert. Man befestigt die Ladefläche am Anhänger und markiert anschließend diesen und das Rad. Über *Rechtsklick* → *Auswahl* → *Auswahl in den Vordergrund* legt man den eigentlichen Anhänger wieder in den Vordergrund, so dass das Hilfskonstrukt verborgen ist. Schließlich benötigt man noch eine Ladung. Dazu erstellt man ein Rechteck, das in die Ladefläche passt und etwas darüber hinausragt. Diesem gibt man das Material “Stein” und erhöht noch dessen Gewicht. Man platziert nun den Anhänger am Auto und verbindet beide mit dem *Achsen*-Werkzeug.

Das Auto soll sich auch noch bewegen. In den Drehachsenoptionen des Hinterrades macht man diese Achse zu einem Motor, indem man die *Vorwärtstaste* mit der *Pfeiltaste links* belegt und die *Rückwärtstaste* mit der *Pfeiltaste rechts*. Weiter setzt man die Drehzahl des Motors auf einen vernünftigen Wert. In diesem Beispiel wurden 500 *rpm* gewählt, da dies mit dem Radius des Reifens von 0,2 *m* eine Endgeschwindigkeit von etwa 11,5 $\frac{m}{s}$ ergibt, was für die Simulation völlig ausreichend ist. Das Drehmoment der Achse ist klein zu wählen, da ansonsten die Reifen durchdrehen. Außerdem ist bei einem kleinen Drehmoment die Beschleunigung geringer und damit auch schöner zu beobachten. Ein Wert von 3 *Nm* hat sich in dieser Simulation als vernünftig herausgestellt. Im Zweifelsfall sollte man ein wenig experimentieren. Die Bremse legt man am besten auf das Vorderrad. Hier belegt man die *Bremstaste* in den Drehachsenoptionen mit der *Pfeiltaste unten*. Dieser Achse gibt man nun ein großes Drehmoment, damit die Bremse auch richtig greift.

Zuletzt verändert man noch die Reibungskoeffizienten der Räder und des Untergrundes, damit das Auto auch Grip hat. Ein Wert von 1,5 bei Rädern und Ebene ist hierbei ausreichend. Damit man auch die Bewegung des Autos nachvollziehen kann, gibt man dem Untergrund noch eine Textur, am einfachsten eine der Materialtexturen, die Algodoo beiliegen. Mit *Rechtsklick* → *Auswahl* → *Ansicht folgt dem Objekt* heftet man noch die Kamera an das Fahrzeug und deaktiviert die Luftreibung.

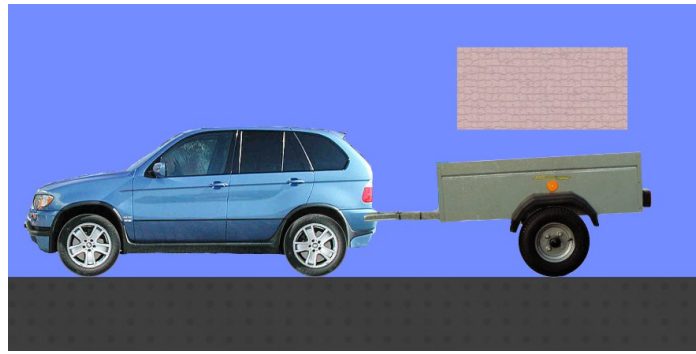


Abb. 18: Das fertige Fahrzeug mit Ladung

Anwendung im Unterricht In dieser Simulation lässt sich an einem Graphen bequem die Beschleunigung ablesen. Man muss nur noch den Graphen des Autos einblenden. Dieser steht bereits auf Zeit-Schnelligkeit. Man kann nun drei verschiedene Bewegungen aufzeichnen. Einmal die des unbeladenen Fahrzeuges, einmal des beladenen Fahrzeuges und man kann noch eine schiefe Ebene hinzufügen und das Fahrzeug hinauffahren lassen.

Man platziert den Graphen so, dass sowohl Fahrzeug als auch Graph gut sichtbar sind. Dann startet man die Simulation und beschleunigt das unbeladene Fahrzeug mit der Pfeiltaste. Hat es seine Maximalgeschwindigkeit erreicht, bremst man ab. Anschließend pausiert man die Simulation wieder und klickt in der Szenensteuerung auf *Spring zurück*, um das Fahrzeug wieder an seinen Startpunkt zurückzusetzen. Der Graph bleibt allerdings erhalten. Nun kann man das Auto beladen und die Szene erneut starten. Man sieht, dass die Zeit-Schnelligkeits-Kurve, je nach Last, deutlich langsamer ansteigt (siehe Abbildung 19).

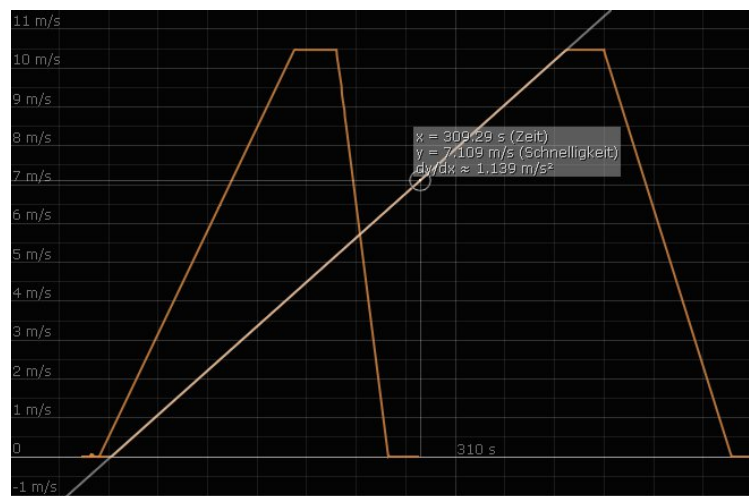


Abb. 19: Graph, an dem die Beschleunigung abgelesen wird

4. Beispielszenen

Den erhaltenen Graphen kann man nun analysieren. Bewegt man die Maus auf eine beliebige Stelle der Kurve, werden die entsprechenden Werte angezeigt, sowie die momentane Steigung $\frac{\Delta v}{\Delta t}$, welche der Beschleunigung a entspricht. In Abbildung 19 zum Beispiel kann man eine Beschleunigung von $a_1 = 1,139 \frac{m}{s^2}$ ablesen, welche zu dem beladenen Fahrzeug gehört. Die Kurve des unbeladenen Autos hingegen hat eine Beschleunigung von $a_2 \approx 2,66 \frac{m}{s^2}$. Auch die Beschleunigungen beim Bremsvorgang können abgelesen werden: $a_{2,brems} \approx -10,00 \frac{m}{s^2}$ und $a_{1,brems} \approx -3,97 \frac{m}{s^2}$.

Alternative Man kann diese Simulation auch ein wenig vereinfacht zur Veranschaulichung der gleichförmigen Bewegung nutzen. Man erstellt wie gehabt das Fahrzeug, allerdings ohne Anhänger, da dieser hierbei nicht notwendig ist. Anstatt den Achsen Pfeiltasten zuzuweisen, bekommt eine Achse einen ständigen Motor und ein höheres Drehmoment. Die Drehzahl muss dabei nicht besonders hoch gewählt werden, da es hier nicht wichtig ist, dass der Beschleunigungsvorgang kurz ist. Das Fahrzeug fährt jetzt mit konstanter Geschwindigkeit, welche man mit einem Zeit-Orts-Diagramm untersuchen kann.

Man kann das Fahrzeug nun klonen und dem zweiten eine andere Drehzahl geben, so dass die beiden sich überholen. Dazu müssen allerdings noch die Kollisionsgruppen angepasst werden, da es sonst zum Zusammenstoß kommt. Leider ist es in Algodoo nicht möglich, die Bewegungskurven beider Objekte in einen gemeinsamen Graphen einzeichnen zu lassen.

4.1.5. Kreisbewegung: Karussell

Einordnung in den Lehrplan Ebenfalls zum Themenblock “Die Mechanik Newtons” gehört die Kreisbewegung mit konstanter Winkelgeschwindigkeit. Mit der Kreisbewegung einher geht die Zentripetalkraft.

Ziel der Szene Die Einführung der Zentripetalkraft ist sehr wichtig, da sie dem allgemein bekannten Konzept der Zentrifugalkraft widerspricht. Diese Animation soll noch einmal die tatsächlich wirkenden Kräfte einer Kreisbewegung anhand eines Karussells, das von oben betrachtet wird, veranschaulichen und ist zur Präsentation durch den Lehrer gedacht.

Voraussetzungen Grundlagen und Kräfte anzeigen

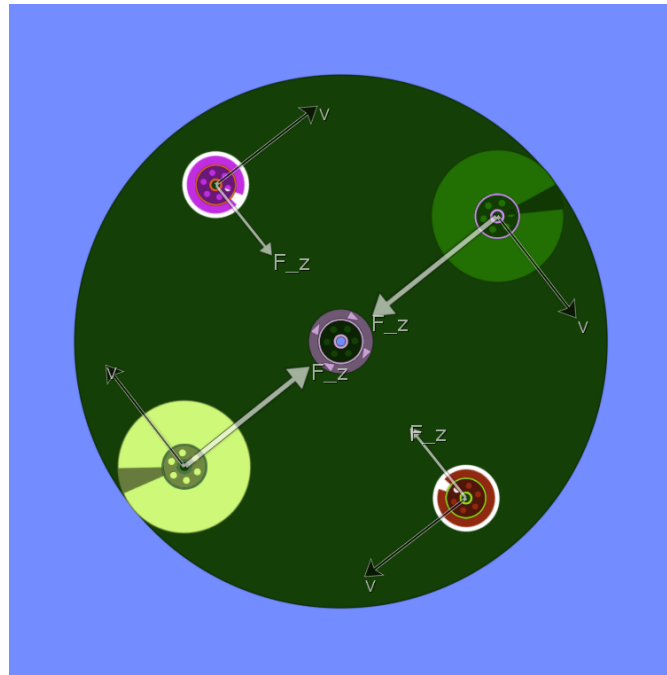


Abb. 20: Zentripetalkräfte und Geschwindigkeiten bei einem Karussell

Erstellung Die Erstellung dieser Szene ist recht einfach. Man erstellt einen großen Kreis, die Unterlage des Karussells, und gibt diesem über *Rechtsklick* → *Geometrieaktionen* → *Mittelachse hinzufügen* eine Achse. Nun erstellt man weitere Kreise, platziert diese über dem Karussell und gibt ihnen ebenfalls Mittelachsen. Bei der Achse des Karussells aktiviert man den Motor. Damit wäre diese Szene schon fast fertig. Nun visualisiert man noch die Kräfte an den Achsen (entweder im *Aussehen*-Menü der jeweiligen Achse oder im *Darstellungs*-Menü oben rechts) und nach Belieben kann man auch noch die Geschwindigkeiten visualisieren. Startet man die Simulation, werden jetzt Kräfte und Geschwindigkeiten angezeigt. Sind die Kraft- oder Geschwindigkeitsvektoren zu lang oder zu kurz, kann dies noch im *Darstellungs*-Menü geändert werden.

4.1.6. Kreisbewegung: Erde - Mond

Einordnung in den Lehrplan Nach Einführung der Kreisbewegung und der Zentripetalkraft in der 10. Jahrgangsstufe wird die Gravitationskraft behandelt.

Ziel der Szene Ähnlich wie im vorangegangenen Abschnitt soll diese Animation die Kräfte der Kreisbewegung verdeutlichen und in diesem Fall noch zusätzlich auf die Gravitationskräfte zwischen Körpern eingehen. Es soll daher in dieser Animation das System Erde - Mond nachgebildet werden. Da man in dieses System so gut wie nicht eingreifen

kann, ist diese Szene als Präsentation gedacht.

Voraussetzungen Grundlagen, Texturen und Skriptsprache

Erstellung Da es im Erde-Mond-System möglich ist, soll diese Simulation maßstabsgetreu nachgebildet werden. Der Radius der Erde beträgt ungefähr $r_E \approx 6378 \text{ km}$ und der des Mondes $r_M \approx 1738 \text{ km}$. Der Abstand der beiden Himmelskörper ist $r_{EM} \approx 373850 \text{ km}$. Es bietet sich ein Maßstab von 1:10 000 000 an, womit sich folgende Werte für die Simulation ergeben: $d_E = 0,64 \text{ m}$, $d_M = 0,17 \text{ m}$, $r_{EM} = 37,39 \text{ m}$. Wie bereits erwähnt, ist in Algodoo eine exakte Platzierung von Objekt per Hand kaum möglich, da Algodoo keine Informationen über Koordinaten Preis gibt. Es ist deshalb einfacher, die Skriptsprache zu benutzen, um die beiden Objekte zu platzieren. Mit der *F10*-Taste öffnet man die Konsole und gibt dort ein:

```
Scene.my.addCircle({ radius:=0.64 ; pos:=[0,0] })
```

Drückt man die Pfeiltaste nach oben, wird noch einmal das zuletzt eingegebene Kommando aufgerufen. Dieses ändert man ab zu:

```
Scene.my.addCircle({ radius:=0.17 ; pos:=[37.39,0] })
```

Erde und Mond wurden nun an den gewünschten Koordinaten mit den richtigen Maßen platziert.

Da sich die Szene im Weltall befindet, ändert man die Hintergrundfarbe auf schwarz und deaktiviert die Wolken. Über *Rechtsklick* → *Aussehen* gibt man den beiden Himmelskörpern die passenden Texturen. Auf der beigelegten CD im Verzeichnis **Texturen\ Planeten** befinden sich die Texturen für die Erde und den Mond. Mit dem *Textur*-Werkzeug werden diese angepasst und anschließend noch die Kanten im *Aussehen*-Menü entfernt.

Es macht keinen Sinn, bei den Massen und der Gravitationskonstante realistische Werte einzugeben, da man keine Umlaufdauer von einem Monat will. Man sollte die Masse der Erde noch beträchtlich erhöhen, da die Erde ansonsten nicht träge genug ist und die Wechselwirkung zwischen den beiden so groß ist, dass sich die Erde zu stark mitbewegt. Es wurde deshalb eine Masse von 100 kg und eine Anziehung von $5 \frac{\text{Nm}^2}{\text{kg}^2}$ gewählt. Den Mond kann man schließlich mit *Rechtsklick* → *Geschwindigkeiten* → *In den Orbit senden* auf eine Kreisbahn um die Erde schicken. Abhängig von den Massen und der Anziehungskraft berechnet Algodoo automatisch die Umlaufgeschwindigkeit.



Abb. 21: Erde und Mond mit Kraftvektoren

4.1.7. Impuls - Zentraler Stoß zweier Massen

Einordnung in den Lehrplan Impuls als Erhaltungsgröße kommt ebenfalls in der 10. Jahrgangsstufe im Thema “Die Mechanik Newtons” vor.

Ziel der Szene Diese Szene soll den voll elastischen Stoß behandeln. Über eine Betrachtung der Energieerhaltung und Impulserhaltung kann man die Geschwindigkeiten der beiden Körper nach dem Stoß berechnen. Spezialisiert man die Situation, beispielsweise durch gleiche Massen oder einen ruhenden Körper, vereinfachen sich die beiden Formeln teilweise erheblich. Für die Geschwindigkeiten u_1 und u_2 nach dem Stoß ergeben sich:

$$u_1 = \frac{m_1 v_1 + m_2 (2v_2 - v_1)}{m_1 + m_2} \quad ; \quad u_2 = \frac{m_2 v_2 + m_1 (2v_1 - v_2)}{m_1 + m_2}$$

Diese Simulation kann vom Lehrer dazu genutzt werden, die Ergebnisse dieser Vereinfachungen direkt und schnell zu überprüfen und für die Schüler zu visualisieren. Sie kann aber auch von den Schülern selbst genutzt werden, um eigene Vorhersagen zu überprüfen.

Voraussetzungen Grundlagen

Erstellung Diese Szene ist sehr schnell erstellt. Es ist sogar möglich, sie direkt im Unterricht zu konstruieren und den Schülern mitzuteilen, was und vor allem warum man es macht. Eine andere Möglichkeit wäre es die Schüler selbst erstellen zu lassen.

Die Ebene einer neuen Szene kann direkt verwendet werden. Man aktiviert das Gitternetz und erstellt zwei gleich große Körper, beispielsweise zwei Quadrate. Da man einen reibungsfreien und voll elastischen Stoß haben will, markiert man die beiden Körper und die Ebene und stellt deren Reibung auf 0 und die Stoßelastizität auf 1. Den Schülern sollte dabei klar werden, dass dadurch Impuls- und Energieerhaltung gewährleistet ist. Über

Rechtsklick → *Geschwindigkeiten* kann man den beiden Körpern jetzt gewünschte Anfangsgeschwindigkeiten geben. Im *Darstellungs*-Menü kann man die Geschwindigkeits- oder Impulsvektoren anzeigen lassen (beide gleichzeitig würden sich unschön überlagern). Lässt man sich die Werte der Impulse anzeigen, kann man sogar relativ leicht die Impulserhaltung nachrechnen.

Verwendung im Unterricht Man kann die einzelnen Spezialfälle überprüfen:

1. **Gleiche Massen und Körper 1 in Ruhe:** Für $m_1 = m_2 = m$ und $v_1 = 0$ vereinfachen sich die beiden Formeln zu:

$$u_1 = v_2 \quad ; \quad u_2 = 0$$

In der Simulation lässt sich das sehr gut überprüfen (siehe Abbildung 22).

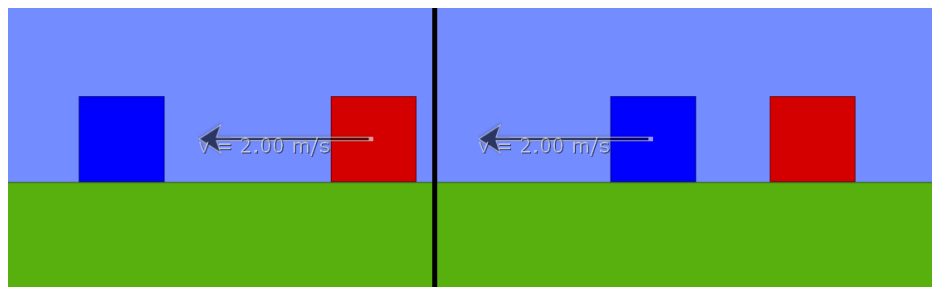


Abb. 22: Körper vor und nach dem Stoß

2. **Gleiche Massen, aber unterschiedliche Geschwindigkeiten:** Haben die Körper wieder die gleiche Masse m , fahren aber mit unterschiedlichen Geschwindigkeiten v_1 und v_2 aufeinander zu, ergibt sich:

$$u_1 = v_2 \quad ; \quad u_2 = v_1$$

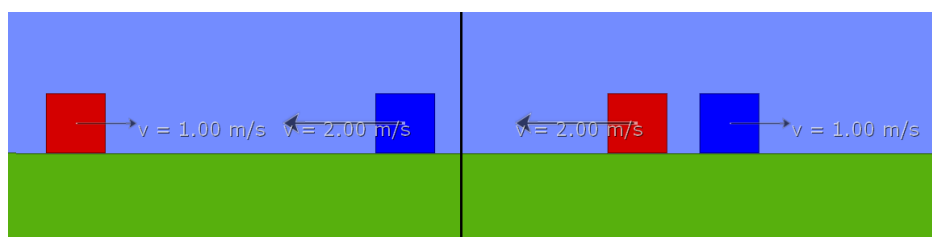


Abb. 23: Vor und nach dem Stoß

3. **Sehr kleine Masse, gegen sehr große, ruhende Masse:** Für $m_1 \gg m_2$ und $v_1 = 0$ folgt aus obiger Formel durch eine Abschätzung:

$$u_1 = 0 \quad ; \quad u_2 = -v_2$$

In der Simulation kann die große Masse am besten durch eine zweite, senkrechte Ebene dargestellt werden (Materialeigenschaften anpassen). Will man genauer auf die Abschätzung $m_1 + m_2 \approx m_1$ und damit auch $u_2 \approx -v_2$ eingehen, kann auch tatsächlich eine sehr große Masse erstellt werden. Da die zweite Masse trotz allem ein klein wenig Geschwindigkeit verliert, stimmt somit das Gleichheitszeichen bei $u_2 = -v_2$ nicht mehr. Nicht so bei der Ebene, da diese festgeheftet ist und selbst keine Energie aufnimmt.

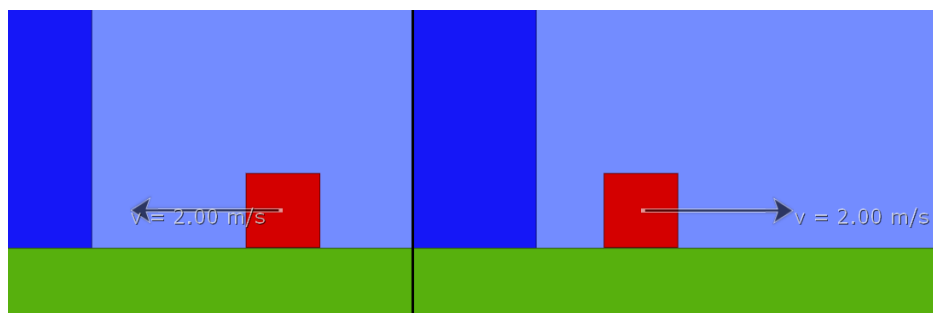


Abb. 24: Stoß an einer sehr großen Masse

Natürlich können auch noch viele weitere Fälle betrachtet werden, beispielsweise $m_1 = 4m_2$ und so weiter. Die Schüler können frei experimentieren.

4.1.8. Teilchenmodell des Gases

Einordnung in den Lehrplan Das Teilchenmodell wird in der 8. Klasse im Thema “Aufbau der Materie und Wärmelehre” behandelt. Anhand des Teilchenmodells soll der Begriff der inneren Energie veranschaulicht werden und wie sich diese durch Wärmezufuhr oder -abfuhr ändern kann.

Ziel der Szene In dieser Szene sollen zwei miteinander verbundene Gefäße dargestellt werden. Zunächst befinden sich nur in einem Gefäß Teilchen, die die Gasmoleküle darstellen. Durch einen Austauschkanal können diese auch in das andere Gefäß gelangen, womit ein Ausgleich stattfindet. In einem zweiten Schritt wird ein Gefäß abgekühlt und

4. Beispielszenen

das andere erwärmt. Auch hier findet sich nach einiger Zeit wieder ein Gleichgewichtszustand ein, wobei jetzt im erwärmten Gefäß weniger aber schnellere Teilchen zu finden sind. Diese Simulation ist eher dazu geeignet, vom Lehrer vorgestellt zu werden, um die interessanten Aspekte kontrolliert zu betrachten.

Voraussetzungen Grundlagen

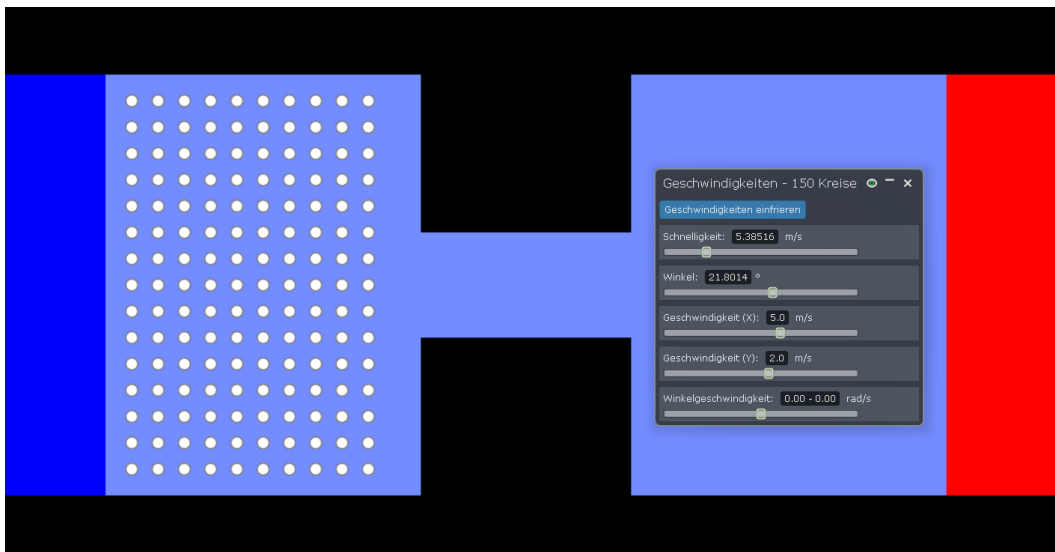


Abb. 25: Die verbundenen Gefäße mit Teilchen

Erstellung Die beiden Bereiche werden von vier Ebenen abgegrenzt. Zunächst erstellt man zwei Ebenen von links und rechts, welchen die Farben Rot und Blau gegeben wird (die Farben symbolisieren warm und kalt). Anschließend werden zwei Ebenen von oben und unten eingefügt, welche schwarz eingefärbt wurden, um sie von der abkühlenden und erwärmenden Seite zu unterscheiden. Weiter bringt man zwei gleich breite Rechtecke in der Mitte an, die den Wärmeaustauschtunnel beschränken. Diese müssen noch fixiert werden. Aus rein optischen Gründen kann man allen Objekten im *Aussehen*-Menü die Kanten entfernen (siehe Abb. 25). Da die Wände natürlich ohne Energieverlust mit den Teilchen interagieren sollen, setzt man wie gewohnt die Reibung auf 0 und die Stoßelastizität auf 1.

Nun fehlen noch die Teilchen. Man erstellt also einen kleinen Kreis in einem der Gefäße und macht diesen ebenfalls reibungsfrei und voll elastisch. Die Teilchen brauchen auch noch eine gewisse Anfangsgeschwindigkeit, damit sie sich beim Start der Simulation überhaupt bewegen. Mit *Rechtsklick* → *Geschwindigkeit* gibt man dem Teilchen eine

x- und y-Geschwindigkeit (in diesem Beispiel 5 m/s und 2 m/s). Mit gedrückter *Strg*-Taste kann man den Kreis klonen. Da nach diesem Vorgang gleich beide Kreise markiert sind, kann man dies gleich ein paar mal wiederholen und somit schnell und einfach eine Menge identischer Teilchen erstellen. Wieviele Teilchen man letzten Endes einfügen will, ist vor allem von der Leistung des PCs abhängig. Bei einer großen Anzahl beginnt die Simulation auf schwächeren Rechnern schnell zu ruckeln.

Die Szene wäre damit fertig. Die Abkühlung und Erwärmung wird im nächsten Abschnitt besprochen.

Anwendung im Unterricht Die Simulation dient der Veranschaulichung des Teilchenmodells und zeigt, wie sich Erwärmung und Abkühlung auf die innere Energie und somit auf die Bewegungsenergie der Teilchen auswirkt. Dementsprechend kann sie in Unterrichtseinheiten eingesetzt werden.

Startet man die Simulation, beginnen sich die Teilchen chaotisch zu bewegen und langsam auch in das zweite Gefäß zu diffundieren. Um diesen mitunter zeitaufwändigen Vorgang zu beschleunigen, kann die Simulationsgeschwindigkeit auf dem Startknopf kurzzeitig erhöht werden. Hat der Ausgleich stattgefunden, setzt man die Simulationsgeschwindigkeit am besten wieder auf 1x.

Nun kann man für die Erwärmung und Abkühlung sorgen. Dazu wird die Stoßelastizität der blauen Ebene auf 0,8 und die der roten Ebene auf 1,2 gesetzt. Teilchen, die mit der blauen Ebene kollidieren, wird jetzt kinetische Energie entzogen, wohingegen denen, die mit der roten Ebene kollidieren, kinetische Energie hinzugefügt wird. Beschleunigt man die Simulation wieder, stellt sich irgendwann wieder ein Gleichgewicht ein.

Es ist zusätzlich möglich, ein bestimmtes Teilchen mit einer Leuchtspur zu versehen, um dessen Bewegung verfolgen zu können, oder in einem Graphen die gesamte kinetische Energie der Teilchen zu betrachten.

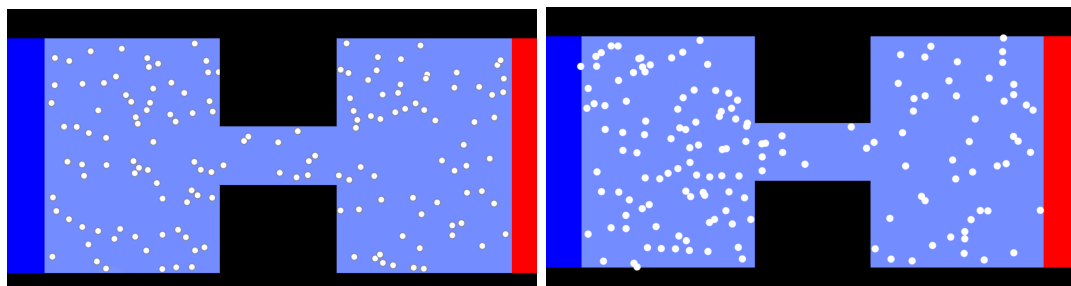


Abb. 26: Gleichgewichtszustände ohne und mit Temperaturänderung

4.2. Optik

Wie bereits eingangs in der Beschreibung von Algodoo erwähnt, sind auch optische Szenen möglich, da Algodoo Laserstifte beinhaltet, welche sich an durchsichtigen Objekten abhängig von deren Brechzahl brechen. Es ist also möglich, Linsen zu konstruieren und diese mit den Laserstifen zu nutzen. Obwohl sich einige interessante Szenen erstellen lassen, sind die Möglichkeiten leider noch sehr beschränkt. Mit den Lasern sind in Algodoo nämlich nur einzelne Lichtstrahlen realisierbar. Es geht nicht, breite Strahlen oder sogar radiale Lichtquellen wie zum Beispiel eine einfache Lampe darzustellen, weswegen die Möglichkeiten mit Algodoo optische Phänomene darzustellen, doch sehr eingeschränkt ist. Man kann zwar die Laser und damit auch die Strahlen größer machen, dennoch ist es nach wie vor ein einziger dicker Strahl, der nicht das gewünschte Ergebnis bringt.

Einige Szenen, die für diesen Teil geplant waren, mussten wieder entfernen werden, da sie leider nicht vernünftig umsetzbar waren, wie in Abschnitt 4.2.6 näher erleutert wird.

4.2.1. Schatten und Ausbreitung von Licht

Einordnung in den Lehrplan Optik wird am Anfang der 7. Klasse im Fach “Natur und Technik” durchgenommen. Es werden dort alle grundlegenden Themen der Optik angeschnitten. In den folgenden Abschnitten wird deshalb nicht mehr auf die Einordnung im Lehrplan eingegangen, da alle in derselben Jahrgangsstufe einzuordnen sind.

Ziel der Szene Hier sollen einfach Schattenwürfe demonstriert werden. Vorteil der Simulation ist hierbei, dass die Strahlen sehr genau beobachtet werden können. Außerdem soll hier gezeigt werden, dass man manche Unzulänglichkeit von Algodoo umgehen kann. Wie eingangs erwähnt, ist es nicht möglich, einen Lichtkegel zu erstellen. Trotzdem wird in dieser Simulation ein adäquater Ersatz gebastelt.

Mit der fertigen Szene können Schüler den Schattenwurf verschiedener Objekte betrachten.

Voraussetzungen Grundlagen und Skript

Erstellung Zunächst soll ein Lichtkegel mit einem Öffnungswinkel von 20° erstellt werden. Dazu platziert man 101 Laser gleicher Farbe in verschiedenen Winkel übereinander. Um einen schönen Lichtkegel zu erhalten, der erst spät auseinanderfächert, bietet sich ein Abstand von $0,2^\circ$ an. Per Hand ist dies unnötig aufwendig, weswegen man hier besser das Skript benutzt. Mit `addLaserPen` fügt man der Szene einen Laser hinzu. Als

zusätzliche Informationen geben wir ihm noch Position, Farbe und natürlich den Winkel mit. Dieser muss allerdings im Bogenmaß angegeben werden, weswegen man noch die Umrechnung über $\pi \backslash 180^\circ$ benötigt. Die Kreiszahl π kann man in Thyme über `math.Pi` Einfügen. Den Farbcode für Weiß erhält man, indem man einen weißen Laser erstellt und über *Rechtsklick* \rightarrow *Skriptmenü* den Eintrag bei `color` abliest. In diesem Fall lautet er: `[1,1,1,1]`. Der Code zum Erstellen eines Lasers und einem Winkel k lautet also:

```
Scene.my.addLaserPen({ pos := [0,0] ;
color := [1,1,1,1] ; rotation := math.Pi / 180 * k })
```

Doch auch so ist es noch recht umständlich, 101 Laser zu erzeugen. Deshalb nutzt man die *for-Schleife*, um den Winkel k automatisch laufen zu lassen. Da man schließlich einen Abstand von $0,2^\circ$ haben will, multipliziert man k noch mit 0,2. Zunächst erstellt man die Laser zwischen 0° und $+10^\circ$, da man hierfür bereits 51 Durchläufe der *for-Schleife* benötigt und diese, wie bereits erwähnt, nur etwa 65 Wiederholungen zulässt. Der Code für die oberen 51 Laser lautet also:

```
for(51, (k) => { Scene.my.addLaserPen({ pos := [0,0] ;
color := [1,1,1,1] ;
rotation := math.Pi / 180 * 0.2 * k }) })
```

Für die verbleibenden unteren 50 Laser nimmt man ein negatives k und lässt dieses erst bei 1 anstatt 0 beginnen, da sonst auf 0° zwei Laser übereinander lägen.

```
for(51, (k) => { Scene.my.addLaserPen({ pos := [0,0] ;
color := [1,1,1,1] ;
rotation := math.Pi / 180 * 0.2 * (-k-1) }) })
```

Damit wäre der Lichtkegel erstellt. Es ist möglich, dass diese Anzahl an Laser langsamere PCs überlastet. Ist dies der Fall, sollte man den Winkelabstand erhöhen und die Anzahl anpassen. Dadurch wird der Bereich, in dem der Kegel kontinuierliches Licht erzeugt, kleiner.

Nun kommt die eigentliche Szene. Da man hier mit Licht arbeitet, ist es angenehmer und übersichtlicher, die Hintergrundfarbe auf schwarz zu stellen, die Wolken zu deaktivieren und die untere Ebene zu löschen. Bevor man jetzt Objekte in die Szene setzt, empfiehlt es sich im *Material*-Menü für neue Objekte oben rechts die Standardeinstellung für die Brechzahl von 1,5 auf 1 zu setzen, da sonst alle neuen Objekte ein klein wenig Licht reflektieren würden, was in dieser Szene mit sehr vielen Lichtstrahlen zu unerwünschten Nebeneffekten führen würde. Jetzt kann man eine senkrechte Ebene platzieren und diese mit dem Lichtkegel anstrahlen.



Abb. 27: Schattenwurf eines unförmigen Objektes

Anwendung im Unterricht Man kann jetzt beliebige Objekte im Lichtkegel platzieren und deren Schattenwurf betrachten. Gerade mit dem *Polygon*- oder *Pinzel*-Werkzeug können kompliziertere Objekte erzeugt werden. Natürlich kann man auch bewegte Objekte betrachten. Ebenso können Spiegel in die Szene eingefügt werden. Ein Spiegel ist ein Objekt mit einer unendlich hohen Brechzahl. Die Einsatzmöglichkeiten sind vielfältig. Als Beispiel wurde in Abbildung 28 mit zwei Spiegeln und einer Röhre ein Periskop erstellt, das die Lichtstrahlen umleitet.

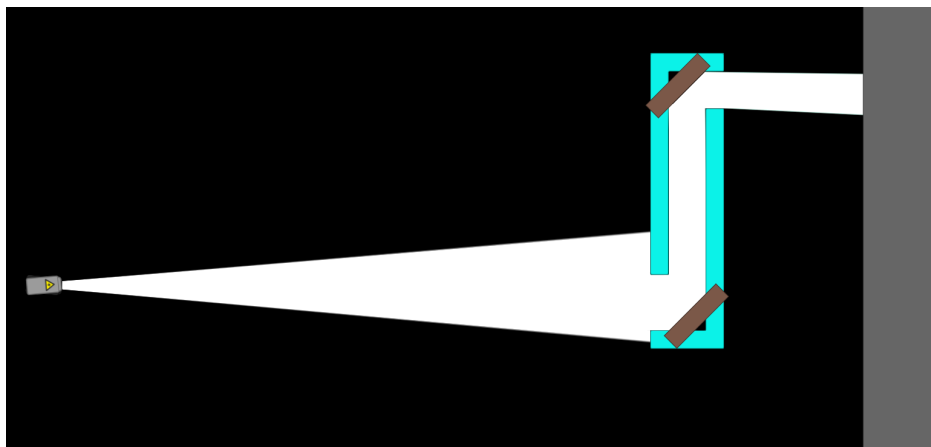


Abb. 28: Periskop

Mit dieser Simulation kann man die Schüler frei experimentieren lassen. Dazu ist es hilfreich, den erstellten Laserkegel als *Komponente* abzuspeichern. Dazu markiert man die Laser und zieht sie links oben auf das *Komponenten*-Symbol. Der Lichtkegel wird im zuletzt geöffneten Ordner abgespeichert und kann dort noch benannt werden. Von dort kann er auch noch in späteren Szenen, zum Beispiel in 4.2.3 oder in 4.2.5 verwendet werden.

4.2.2. Sonnen- und Mondfinsternis

Einordnung in den Lehrplan 7. Jahrgangsstufe Optik

Ziel der Szene In dieser Simulation soll anhand einzelner Lichtstrahlen der Sonne verdeutlicht werden, wie es zu einer Sonnen- bzw. Mondfinsternis kommt. Zudem soll eine Erklärung und Veranschaulichung des Mondzyklus gegeben werden. Bei dieser Szene handelt es sich um eine vorgefertigte Animation. Eine Einflussnahme ist hier nicht vorgesehen, die Szene sollte also den Schülern vom Lehrer präsentiert werden.

Voraussetzungen Grundlagen und Laser

Erstellung In dieser Szene soll es verschiedene Bereiche geben. Im ersten wird eine Sonnen- und Mondfinsternis dargestellt. Dazu platziert man drei Kreise, die jeweils Sonne, Erde und Mond darstellen (nicht maßstabsgetreu, aber in vernünftigem Verhältnis). Wie in Abschnitt 4.1.6 über die Gravitationskräfte zwischen Erde und Mond auf Seite 45 gibt man den Himmelskörpern Texturen, welche auf der beiliegenden CD im Ordner *Texturen\Planeten* zu finden sind. Auch in dieser Szene wird der Hintergrund auf schwarz gestellt, die Wolken deaktiviert und die Gravitation und Luftreibung ausgeschaltet.

Nun platziert man Lichtstrahlen an der Sonne. Dazu könnte man entweder einzelne Laser oben und unten an der Sonne anbringen, oder man greift auf den Lichtkegel zurück, der in Abschnitt 4.2.1 erstellt und als Komponente abgespeichert wurde. Über die *Komponenten* platziert man zwei Lichtkegel, wie in Abbildung 29 zu sehen ist, oben und unten an der Sonne und richtet sie mit dem *Drehen*-Werkzeug auf die Erde aus. Damit die Sonne nicht einen Teil der Strahlen verdeckt, sollte man deren Kollisionsgruppen komplett entfernen. Erde und Mond reflektieren jetzt einen Teil der Strahlen, was hier nicht in die Szene passt. Deswegen stellt man deren Brechzahlen auf 1. Den Mond platziert man jetzt so vor der Erde, dass sein Kernschatten die Oberfläche trifft. Sind Erde und Mond weit genug von der Sonne entfernt, weiten sich die Strahlen der beiden Lichtkegel bereits. In diesem Fall ist das sogar erwünscht, da man so noch den Halbschatten der Mondfinsternis erkennen kann, wie man in Abbildung 29 sieht.

Wenn man möchte, kann man den Mond noch in eine Umlaufbahn um die Erde senden. Dazu gibt man der Erde eine gewisse *Anziehungskraft* und heftet sie am besten an den Hintergrund (ebenso die Sonne). Mit *Rechtsklick* → *Geschwindigkeiten* → *In Orbit senden* lässt man den Mond die Erde umkreisen. Dabei durchläuft er nicht nur die Sonnenfinsternis, sondern auf der anderen Seite auch die Mondfinsternis. Dabei ist den

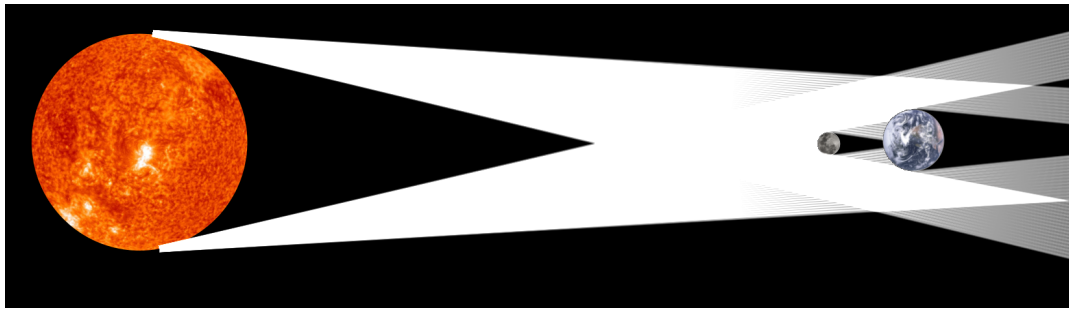


Abb. 29: *Sonnenfinsternis*

Schülern zu erklären, dass dies in der Realität, wegen der Dreidimensionalität nur selten passiert.

Der nächste Bereich der Szene wird ein wenig abseits des vorherigen platziert. Hier sollen vier Mondphasen (Neumond und Vollmond sowie zunehmender und abnehmender Halbmond) dargestellt werden. Hierfür aktiviert man am besten wieder das Gitternetz, um die vier Monde einfacher zu platzieren. Man erstellt wieder eine Erde und platziert jetzt mit Hilfe des Gitternetzes oben, unten, links und rechts jeweils einen Mond, diesmal aber mit der *Halbmond*-Textur. Mit dem *Textur*-Werkzeug verschiebt man es so, dass die helle Seite nach links schaut. Da die Erde aus dem vorherigen Bereich eine Anziehungskraft hat, die diesen Bereich natürlich nicht beeinflussen soll, heftet man alle fünf Objekte an den Hintergrund. Jetzt platziert man wie in Abbildung 30 zu sehen noch Laser an der Erdoberfläche, die den Blickwinkel auf den Mond darstellen.

Im dritten und letzten Bereich sollen die Mondphasen noch einmal an einem bewegten Modell dargestellt werden. Die Laser, die den Beobachter darstellen, sollen sich zusammen mit dem Mond um die Erde bewegen. So muss sich sowohl der Beobachter als auch der Mond drehen. Da dies nur mit der Gravitationskraft und einer drehenden Erde kaum zu synchronisieren ist, muss man sich hier ein wenig behelfen. Dazu wurden hier gleich zwei große Kreise erstellt und einer mit einer Mittelachse am Hintergrund befestigt. An diesem wurden dann zwei Laser wie im Beispiel zuvor angebracht. Damit diese immer exakt auf den Mond ausgerichtet bleiben, erstellt man ein stabförmiges Rechteck und befestigt dieses mit dem *Fixieren*-Werkzeug an der Drehscheibe, direkt unter den Lasern (siehe Abbildung 31). Am anderen Ende erstellt man nun den Mond und verbindet ihn mit einer Mittelachse an der Stange. Jetzt kann man die Laser noch justieren. Die Achse der Drehscheibe wird zum Motor gemacht und man gibt ihm noch eine bestimmte Drehzahl. In diesem Beispiel wurden 2 *rpm* gewählt. Damit die helle Seite des Mondes jetzt noch immer der Sonne entgegen zeigt, wird dessen Achse ebenfalls zum Motor gemacht und eine Drehzahl von 2 *rpm* gegeben, allerdings in entgegengesetzter Richtung.

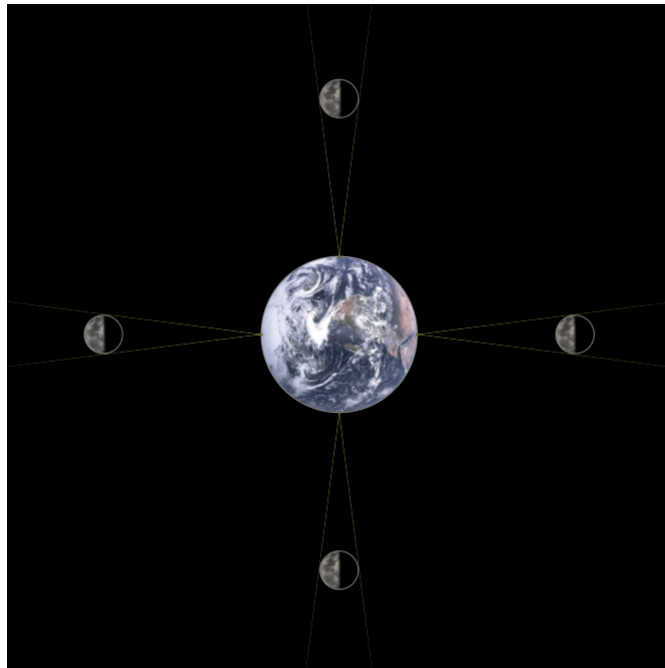


Abb. 30: Vier Mondphasen

Zuguterletzt entfernt man bei dem zweiten der am Anfang erstellten Kreise die Kollisionsgruppen und gibt ihm die Erdtextur. Man platziert ihn so über der Drehscheibe, dass diese verdeckt ist, macht den Stab und die Achse am Mond im *Aussehen*-Menü unsichtbar und die Szene ist fertig.

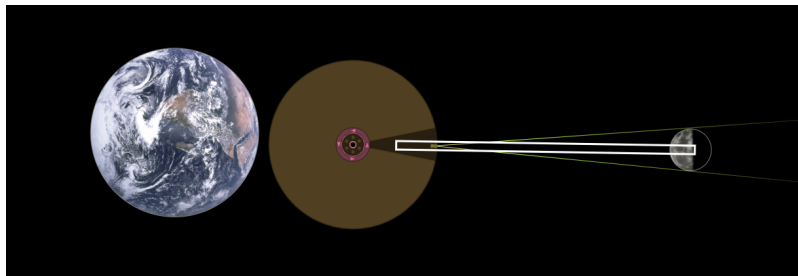


Abb. 31: Versteckter Aufbau der Mondphasen-Szene

4.2.3. Prisma und Linsen

Einordnung in den Lehrplan 7. Jahrgangsstufe Optik

Ziel der Szene Diese Simulation ist dazu gedacht, den Schülern das Verhalten von Licht näherzubringen, insbesondere wie es sich an verschiedenen Medien bricht. Die Schüler können hier völlig frei experimentieren und mit einer leeren Szene starten.

Voraussetzungen Grundlagen und Laser

Erstellung Es empfiehlt sich aus Gründen der Sichtbarkeit, die Hintergrundfarbe auf schwarz zu stellen und die Wolken zu entfernen, da die Schüler so die Lichtstrahlen besser sehen können.

In den Komponenten von Algodoo im Unterordner **optics** befinden sich bereits diverse Objekte, mit denen experimentiert werden kann. Dort sind Prismen, Sammell- und Zerstreuungslinsen sowie ein Spiegel vorhanden und mit dem *Skalieren*-Werkzeug kann man die Proportionen und somit auch die Eigenschaften der Linsen verändern.

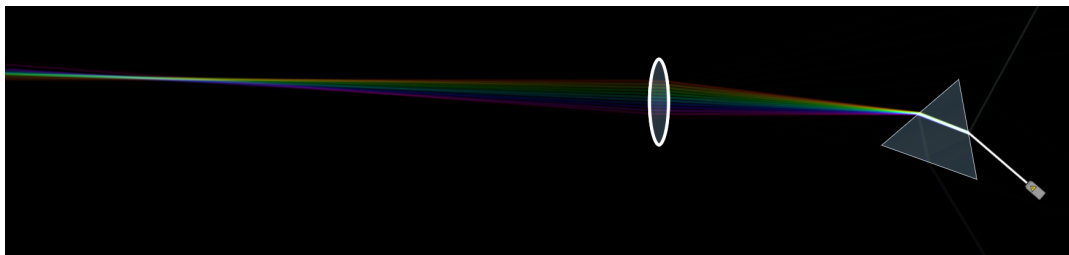


Abb. 32: *Eine von vielen möglichen Versuchsanordnungen*

Anwendung im Unterricht Je nachdem, welche Lernziele man verfolgt, sollte man die Schüler auf spezielle Dinge hinweisen oder ihnen konkrete Arbeitsanweisungen geben. Zum Beispiel kann man die Anweisung geben, weißes Licht mit einem Prisma aufzuspalten und mit einer beliebigen Linse wieder zu bündeln, wie in Abbildung 32 zu sehen ist.

Grundsätzlich kann den Schülern hier freie Hand gelassen werden, um diese grundlegende Optik zu entdecken.

4.2.4. Ein Brillant

Einordnung in den Lehrplan 7. Klasse Optik

Ziel der Szene Ein Brillant ist ein speziell geschliffener Diamant, der eine hohe Brillanz hat. Das heißt, es wird viel Licht des Brillanten in die Richtung des Betrachters gebrochen, sei es durch Spiegelung an der Oberfläche oder Reflexionen und Totalreflexionen im Inneren des Brillanten. Er ist daher interessant für den Physikunterricht, da an ihm viele optische Eigenschaften untersucht werden können.

In dieser Szene soll ein Modell eines Brillanten erstellt werden, an dem mithilfe der Laser experimentiert werden kann.

Voraussetzungen CSG

Erstellung Da beim Brillant die Abmessungen sehr wichtig sind, muss hier präzise gearbeitet werden. Die obere, glatte Seite sollte etwa 56 % der Gesamtbreite einnehmen. Die oberen, schrägen Seiten haben einen Neigungswinkel von etwa $32,2^\circ$, die unteren Seiten von $40,8^\circ$. Das kleine Zwischenstück hat etwa 2 % der Breite.²³

Man erstellt also ein Rechteck. Ein zweites Rechteck dient als Schablone für die CSG-Optionen. Man dreht die Schablone jetzt um den entsprechenden Winkel, legt sie mit Hilfe des Gitternetzes exakt an und schneidet so den Brillanten Ecke für Ecke aus. Am Ende sollte er wie in Abbildung 33 aussehen. Nun weist man dem Brillant noch das Material "Glas" zu und erhöht seinen Brechungsindex auf 2,4, wie bei einem Diamanten.

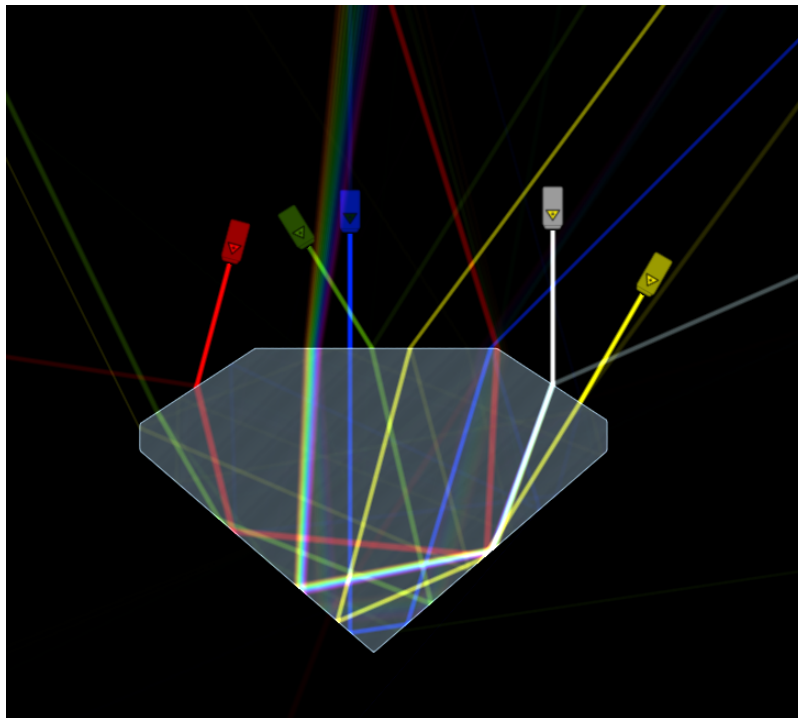


Abb. 33: *Brillantmodell mit einfallendem Licht*

Lässt man nun von oben Laserstrahlen in den Brillanten fallen, sieht man, dass so gut wie kein Licht unten aus dem Brillanten herausfällt. Dabei können verschiedene Farben und Winkel ausprobiert werden. Der Brillant kann auch mit dem Skalieren-Werkzeug bearbeitet werden, wodurch er seine Seitenverhältnisse verliert. Dann ist zu beobachten, wie nicht mehr alles Licht nach oben hin herausstrahlt.

²³Wikipedia: Brillant. (URL: de.wikipedia.org/wiki/Brillant).

4.2.5. Das menschliche Auge

Einordnung in den Lehrplan 7. Klasse Optik

Ziel der Szene Es soll ein menschliches Auge erstellt und dessen optische Fähigkeiten dargestellt werden. In erster Linie wird in dieser Simulation auf Kurz- und Weitsichtigkeit eingegangen und dargestellt, wie eine Brille die Sehschwäche beheben kann. Die Schüler können die Linse des Auges austauschen und Brillengläser davor setzen.

Voraussetzungen Grundlagen und Optik

Erstellung Auf der beigelegten CD im Ordner **Textures** befindet sich die Textur eines Auges. Man erstellt ein Rechteck, gibt ihm diese Textur und passt mit dem *Skalieren*-Werkzeug die Größe an. Bei optischen Szenen bietet sich normalerweise ein schwarzer Hintergrund an, da man darauf die Laserstrahlen am besten sieht. Hier ist aber ein grauer Hintergrund besser, da man ansonsten die zusätzliche Beschriftung des Auges nicht mehr lesen könnte. Im *Aussehen*-Menü werden noch die Kanten des Rechtecks entfernt. Da das Auge nur als Hintergrundbild agieren und nicht den Laser blockieren soll, entfernt man im *Kollisionsgruppen*-Menü alle Gruppen. Um die Funktion der Pupille zu verdeutlichen, zeichnet man mit dem *Polygon*- oder *Pinzel*-Werkzeug die obere und untere Iris grob nach. Auch die Aderhaut zeichnet man nach, damit die Lichtstrahlen nicht einfach durch das Auge hindurch laufen. Alle drei Objekte versteckt man anschließend mit *Rechtsklick* → *Auswahl* → *Auswahl in den Hintergrund* hinter dem Auge. Da man das Laserlicht auf den hellen Farben der Textur nur schwer erkennen würde, wird die Textur noch verdunkelt. Im *Aussehen*-Menü stellt man den *V-Wert* auf etwa 60 %

Als nächstes braucht man einfallendes Licht. Man erstellt dazu einen Laser und passt dessen Größe und Farbe an. In diesem Beispiel wird rotes Licht verwendet. Mit gedrückter *Strg*-Taste klonst man den Laser, bis man die gewünschte Anzahl erreicht hat und platziert diese untereinander, um eine parallele Lichtfront zu erhalten. In diesem Beispiel sind acht Laser ausreichend (siehe Abbildung 34). Man kann die Laser noch ein wenig drehen, um einen gewissen Einfallswinkel zu haben und platziert sie anschließend so, dass sie in die Pupille hineinstrahlen.

Nun kommt der wichtigste Teil: Die Linsen. Aus den von Algodoo vorgegebenen Komponenten im Unterordner *optics* fügt man eine konkave und vier konvexe Linsen der Szene hinzu. Drei der konvexen ergeben Linsen für das Auge, die anderen beiden ergeben Brillengläser. Man platziert nun eine konvexe Linse über dem Auge und verändert so lange mit dem *Skalieren*-Werkzeug die Größe und Proportionen der Linse, bis das

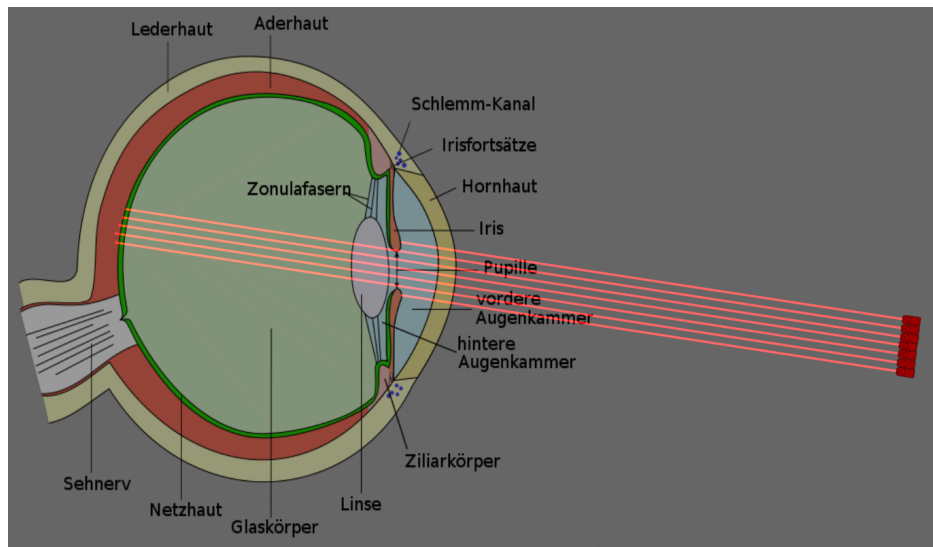


Abb. 34: Auge, noch ohne Linse

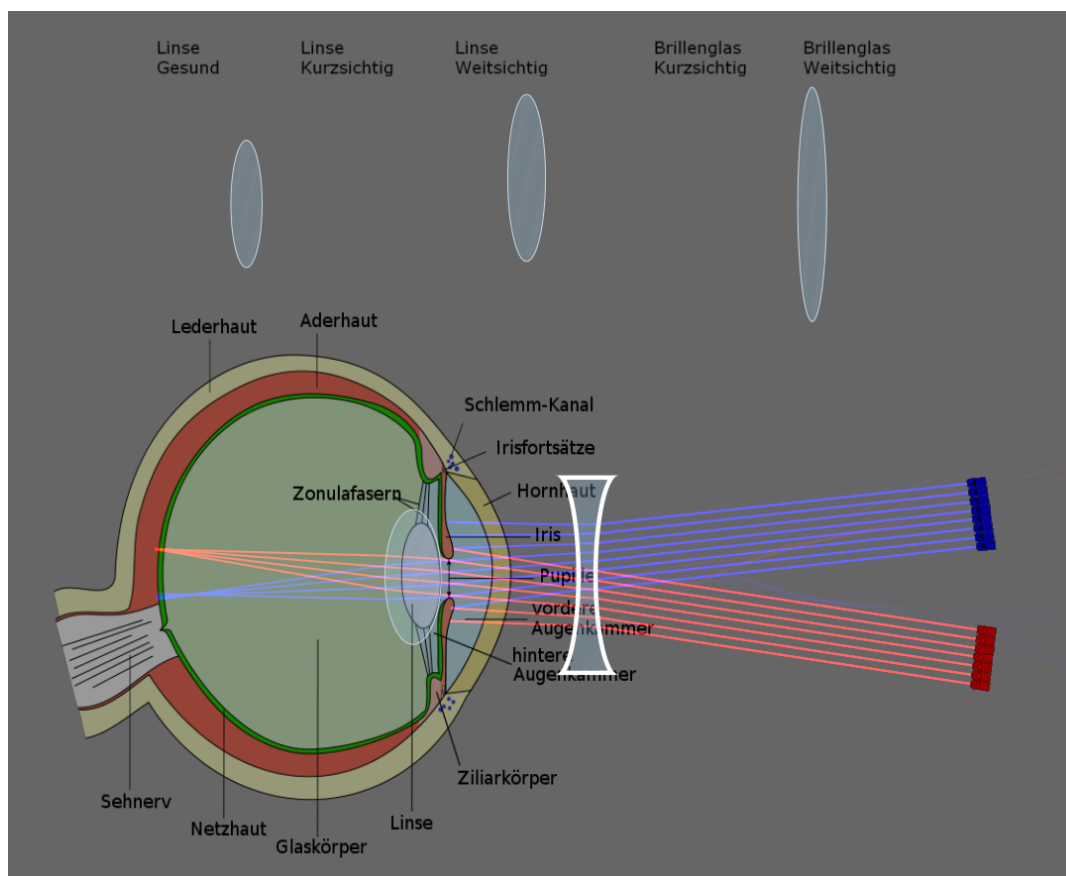


Abb. 35: Fertige Szene mit kurzsichtiger Linse und Brille

4. Beispielszenen

Laserlicht auf der Netzhaut gebündelt wird. Dies ist nun die gesunde Linse. Das gleiche wiederholt man nun mit zwei weiteren konvexen Linsen, nur formt man eine so, dass die Strahlen deutlich vor der Netzhaut gebündelt werden und die andere so, dass die Strahlen hinter der Netzhaut gebündelt werden. Somit hat man eine kurzsichtige und eine weitsichtige Linse.

Jetzt konstruiert man noch die Brillengläser. Dazu setzt man die kurzsichtige Linse ein, platziert die konkave Linse vor dem Auge und skaliert auch diese so lange, bis das Licht wieder auf der Netzhaut gebündelt wird. Das gleiche wiederholt man mit der weitsichtigen Linse und der letzten konvexen Linse.

In diesem Beispiel wurden noch fünf Rechtecke der Szene hinzugefügt, diese über *Rechtsklick* → *Text* mit der jeweiligen Linse beschriftet, die Kanten entfernt und ihnen die gleiche Farbe wie dem Hintergrund zugewiesen. Weiter wurde noch eine zweite Laserfront unter einem anderen Winkel und mit einer anderen Farbe platziert. Dadurch wird deutlich, dass die Bilder kopfüber in das Auge projiziert werden.

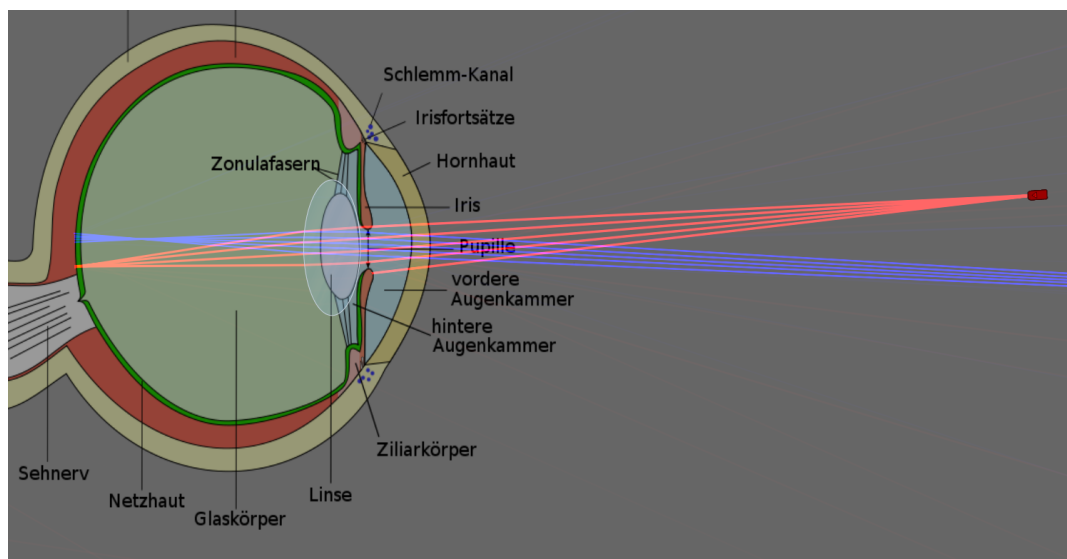


Abb. 36: Kurzsichtiges Auge mit nahem Gegenstand (rot) und fernem Gegenstand (blau)

Anwendung im Unterricht Diese Simulation ist hervorragend dazu geeignet, die Schüler experimentieren zu lassen. Sie können selbst die Eigenschaften der Linse mit dem *Skalieren*-Werkzeug verändern und die Auswirkungen beobachten, und die Unterschiede zwischen Kurz- und Weitsichtigkeit erforschen. Zusätzlich bietet es sich an, anstatt der Laserfronten, die schließlich einen unendlich weit entfernten Gegenstand darstellen, Lichtkegel mit unterschiedlichen Öffnungswinkeln zu erstellen, um unterschiedliche

Entfernungen darzustellen. In Abbildung 36 wurde ein roter Lichtkegel mit einem Öffnungswinkel von 4° erstellt (fünf Laser, die auf -2° , -1° , 0° , 1° und 2° gedreht und übereinander gelegt wurden), der einen nahen Gegenstand darstellt, der im Auge scharf abgebildet wird. Der viel weiter entfernte blaue Lichtkegel mit einem Öffnungswinkel von $0,8^\circ$ wird dagegen unscharf abgebildet.

Somit kann der Einsatz der Brillengläser erforscht werden. Die Schüler finden selbstständig heraus, welcher Abstand der Brillengläser gut ist.

4.2.6. Teleskop

Einordnung in den Lehrplan 7. Jahrgangsstufe Optik

Ziel der Szene Hier sollen die Schüler selbst ein Teleskop erstellen. Dazu bekommen sie als Arbeitsanweisung, welche Linsen sie verwenden sollen. Das heißt, die Schüler erstellen zunächst Linsen mit einer vorgegeben Brennweite und fügen diese zu einem Teleskop zusammen.

Voraussetzungen Grundlagen und Optik

Erstellung Ein einfaches Fernrohr besteht aus zwei Linsen, einem Objektiv mit einer Brennweite f_{Ob} und einem Okular mit Brennweite f_{Ok} . Dabei sollte f_{Ob} um einiges größer sein als f_{Ok} . Diese werden dann so platziert, dass die Brennpunkte beider Linsen zusammenfallen, also in einem Abstand von $t = f_{Ok} + f_{Ob}$, welcher der Tubuslänge entspricht.

Um die Brennweite von Linsen zu bestimmen, können die Schüler wie in Beispiel 4.2.3 eine Linse aus den Komponenten verwenden und diese mit dem *Skalieren*-Werkzeug bearbeiten. Mit Hilfe eines angelegten Maßstabs und Lasern kann die Brennweite passend gewählt werden. Alternativ können auch zwei Kreise überlappt und mit den *CSG*-Optionen die Schnittmenge ausgeschnitten werden. So erhält man ebenfalls eine Linse, die man mit dem *Skalieren*-Werkzeug bearbeiten kann.

Wurden die Linsen gesetzt, werden noch einige parallele Laser davor gesetzt, die ein weit entferntes Objekt darstellen. Wahrscheinlich muss alles noch ein wenig angepasst werden, bis die aus dem Okular austretenden Strahlen wieder parallel verlaufen.

Probleme Beim Justieren des Teleskops fällt auf, dass die Laserstrahlen nicht sehr exakt laufen. Verschiebt man sie leicht nach oben oder unten, wird der Brennpunkt teilweise unscharf und die Strahlen nach dem Okular laufen nicht mehr parallel. Das sollte

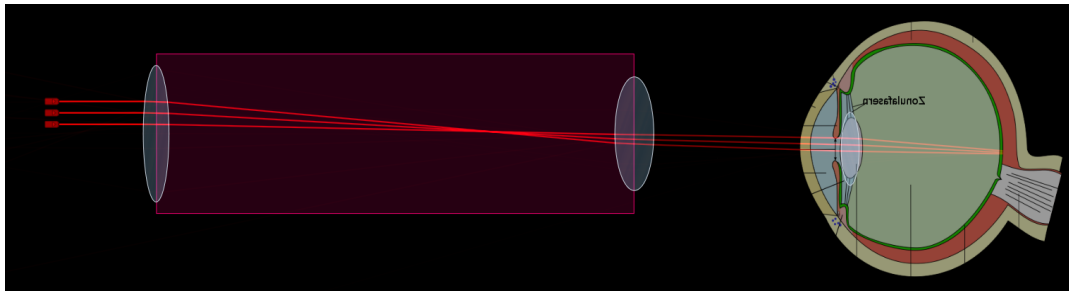


Abb. 37: *Teleskopmodell*

eigentlich nicht vorkommen, ist jedoch technisch bedingt. Die Oberflächen der Linsen sind nicht perfekt rund, sondern setzen sich aus kurzen geraden Stücken zusammen und sind deswegen ganz leicht kantig, wie es in Abbildung 38 ansatzweise zu sehen ist. Dadurch werden Laser teilweise sehr unterschiedlich abgelenkt. Der Brennpunkt einer Linse wird dabei stark verzerrt und ist nicht mehr eindeutig auszumachen.

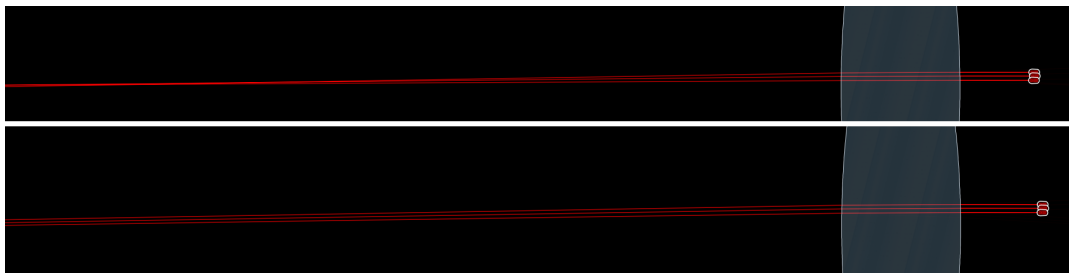


Abb. 38: *Ausschnitt einer Linse. Im unteren Bild wurden die Laser ein kleines Stück vertikal versetzt*

Auf Grund dieser Tatsache sind präzise optische Szenen in Algodoo leider kaum möglich. Das Teleskop ist mit nachträglichem Justieren der Strahlen noch umsetzbar, aber ein geplantes Beispiel über eine optische Bank wurde in dieser Arbeit verworfen, da sich die Fehler bei den Strahlen durch mehrere Linsen aufaddieren.

4.3. Elektronik

Zum Thema Elektronik kann man in Algodoo leider nur sehr wenig zusammenstellen, da das Programm dafür nicht ausgelegt ist und deswegen auch keinerlei Werkzeuge angeboten werden. Es können aber auf Mechanik basierende Ersatzmodelle erstellt und die Gravitationskraft als Ersatz für die elektrische Kraft genutzt werden.

4.3.1. Elektrische Felder

Einordnung in den Lehrplan In der 11. Jahrgangsstufe werden zum Thema “Statisches elektrisches Feld” Feldlinienbilder betrachtet. Deren Aussehen kann in Algodoo, dank der Möglichkeit Kraftfelder darzustellen, untersucht werden.

Ziel der Szene In dieser Szene können die Schüler sehr frei das Aussehen von elektrischen Felder verschiedener Ladungsverteilungen untersuchen. Da es in Algodoo allerdings keine elektrische Kraft gibt, muss man hierfür die Gravitationskraft nutzen. Da man in Algodoo die Gravitationskonstante auch negativ wählen kann, ist es möglich, positive und negative Ladungen darzustellen. Allerdings gehen die Feldlinien des Gravitationsfeldes von “negativer Masse” zur “positiven Masse”, also andersherum als die elektrischen Feldlinien. Den Schülern muss klar sein, dass eine positive *Anziehung* im *Material*-Menü einer negativen elektrischen Ladung entspricht.

Vorraussetzung Grundlagen

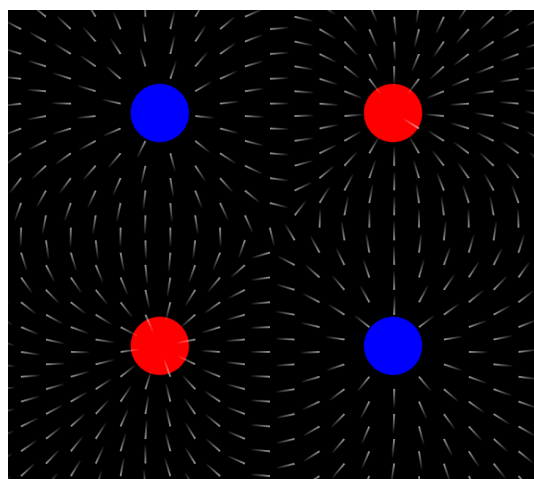


Abb. 39: Ladungsverteilung mit elektrischem Feld

Erstellung Den Schülern kann hier eine völlig neue Szene präsentiert werden. Sie müssen nur die Gravitationskraft deaktivieren und die Ebene löschen, die von Anfang an vorhanden ist. Man kann nun verschiedene Ladungsanordnungen erstellen. Dafür bieten sich Kreise an, da sie meistens eine Punktladung repräsentieren sollen. Man sollte darauf achten, dass alle Kreise die gleiche Masse haben, da die genutzte Gravitationskraft auch massenabhängig ist. Es bietet sich an, dass man den ursprünglichen Kreis dupliziert. Nun kann man den Kreisen unter *Rechtsklick* → *Material* eine *Anziehung* geben. Im *Gravitations*-Menü aktiviert man noch *Gravitationsfeld anzeigen*.

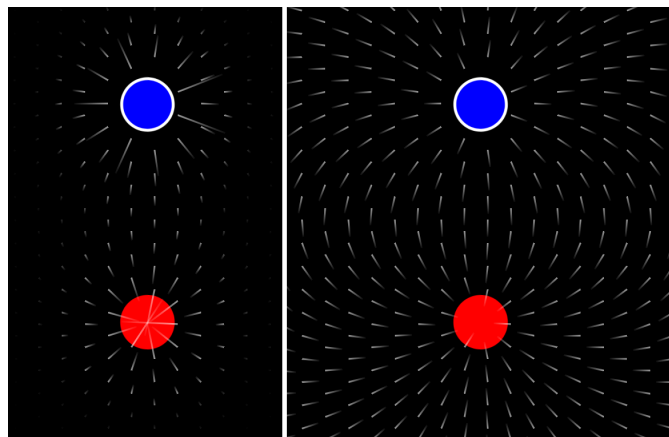


Abb. 40: Elektrischer Dipol mit und ohne Skalieren-Option

Von der Option *Skaliert Feldpfeile mit der Gravitationsstärke* ist abzuraten, da Algodoo dann versucht, die Größe der Gravitationskraft anhand der Feldlinienlängen darzustellen und diese Länge abhängig von der längsten Feldline skaliert. Dadurch verschwinden leider fast alle anderen Feldlinien komplett, außer diejenigen, die direkt beim Objekt sind, wie man in Abbildung 40 sehen kann.

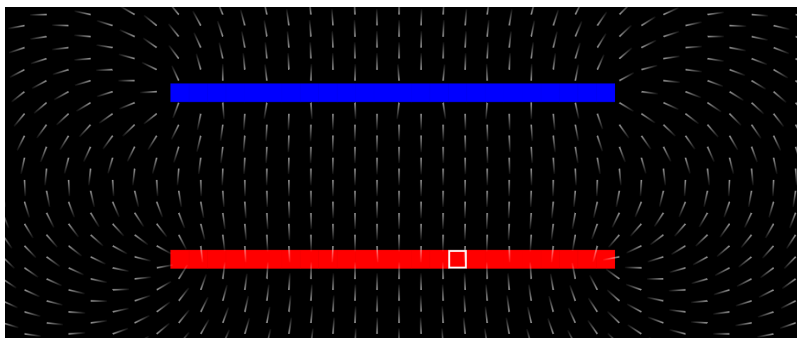


Abb. 41: Plattenkondensator

Weiterhin ist zu beachten, dass Algodoo die Kräfte unabhängig von der Form des Objekts berechnet und schlichtweg den Schwerpunkt als Punktladung betrachtet. Will man

also etwas Kompliziertes konstruieren, setzt man es aus kleinen Einzelteilen zusammen. Möchte man beispielsweise einen Kondensator erstellen, platziert man ein kleines Viereck, gibt diesem die gewünschten Eigenschaften, dupliziert es einige Male und platziert sie nebeneinander. Anschließend dupliziert man die so erstellte “Platte” und ändert das Vorzeichen der Anziehung.

4.3.2. Bewegung im elektrischen Feld

Einordnung in den Lehrplan Das vorherige Beispiel weiterführend, wird in der 11. Jahrgangsstufe das Verhalten geladener Teilchen in einem homogenen elektrischen Feld besprochen. Ein spezielles Beispiel hierfür ist das geladenen Teilchen in einem Plattenkondensator.

Ziel der Szene Die Schüler sollen in dieser Szene die Bewegung eines geladenen Teilchens im Plattenkondensator nachstellen. Der Plattenkondensator kann dabei vorkonstruiert und als Komponente abgespeichert werden. Mit angelegten Maßstäben können die Schüler die y-Ablenkung und die Auswirkungen unterschiedlicher Startgeschwindigkeiten oder Massen (was der Ladung entspricht) bestimmen.

Vorraussetzung Grundlagen und Gravitation

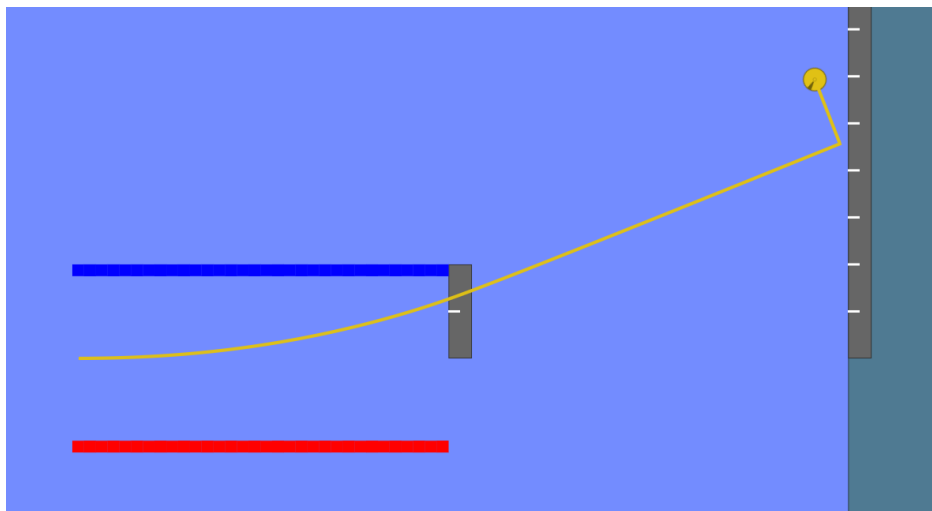


Abb. 42: Geladenes Teilchen im Plattenkondensator (Maßstab in dieser Zoomstufe nicht sichtbar)

Erstellung Die Schüler starten eine neue Szene und entfernen zunächst die Ebene und deaktivieren Gravitation und Luftreibung. Nun konstruieren sie entweder nach An-

4. Beispielszenen

weisung selbst einen Kondensator oder man hat zuvor einen Kondensator wie in 4.3.1 beschrieben erstellt und als *Komponente* abgespeichert, den die Schüler direkt nutzen können.

Nun müssen die Schüler nur noch eine entsprechend große Kugel erstellen, der sie über *Rechtsklick* → *Geschwindigkeiten* eine Geschwindigkeit in x-Richtung geben können. Es bietet sich an, dieser Kugel eine Leuchtspur anzuheften, um die Bewegung besser verfolgen zu können. Optional kann noch ein Schirm in Form einer Ebene hinter dem Kondensator aufgestellt werden. In Abbildung 42 wurden noch zwei Maßstäbe hinzugefügt, die keine Kollisionsgruppe haben, um die Szene nicht zu behindern. Anhand der beiden Maßstäbe kann die y-Ablenkung im Kondensator und die gesamte y-Ablenkung am Schirm bestimmt werden.

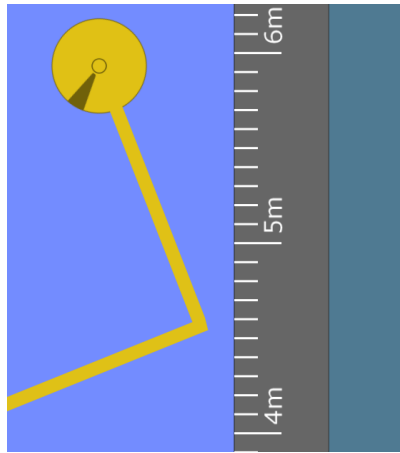


Abb. 43: Bestimmung der y-Ablenkung

Vergleich von Theorie und Praxis Auf das Teilchen mit der Ladung q wirkt die elektrische Kraft $F_{el} = q \cdot E$. Mit $F = m \cdot a$ erhält man für die Beschleunigung in y-Richtung: $a_y = \frac{qE}{m}$. In x-Richtung gilt $x = v_{0x} \cdot t$ und in y-Richtung $y = \frac{1}{2} \cdot a_y \cdot t^2$. Ineinander eingesetzt ergibt sich:

$$y = \frac{1}{2} a_y t^2 = \frac{qE}{2m} \cdot t^2 = \frac{qE}{2mv_{0x}^2} \cdot x^2$$

Bei dieser Simulation ist nun problematisch, dass man weder Informationen über q noch über E hat, da beide Größen eigentlich gar nicht existieren, sondern mit der Gravitationskraft dargestellt werden. Allerdings könnte man sagen, dass die Masse des Teilchens auch seiner Ladung entspricht. Den Schülern muss aber unbedingt klar sein, dass dies nur dem Zweck der Simulation dient, da sowohl Masse als auch Ladung in der Formel

vorkommen. Die Schüler müssen dies strikt auseinander halten.

Man kann diesem Umstand jedoch auch nutzen und zum Beispiel den Arbeitsauftrag erteilen, mit Hilfe der übrigen messbaren Größen das E-Feld zu bestimmen.

4.3.3. Leitermodell

Einordnung in den Lehrplan Bereits in der 7. Jahrgangsstufe im Fach “Natur und Technik” wird das Thema “elektrischer Strom” behandelt. Die Modellvorstellung eines elektrischen Stromkreises steht hier im Mittelpunkt.

Ziel der Szene Diese Szene soll eine vom Lehrer präsentierte Animation sein, die den Schülern das Leitermodell des elektrischen Stromes darlegt. Man kann zwar beschränkt Einfluss auf *Spannung* und *Stromstärke* nehmen, das Hauptaugenmerk liegt aber in der Präsentation des Modells.

Unabhängig vom Einsatz im Unterricht soll diese Szene auch zeigen, wie man mit *Thyme* und durch Laser ausgelöste Skripte um geht und wie man fehlende Möglichkeiten im Skript ersetzen kann.

Erstellung In dieser Szene soll ein waagrecht liegender Leiter erstellt werden. In dem Leiter befinden sich positiv geladene Atomrümpfe und freie Elektronen. Durch eine angelegte Spannung wirkt eine elektrische Kraft auf die freien Elektronen, die durch die Gravitationskraft ersetzt wird. Die Elektronen bewegen sich durch den Leiter. Wichtig ist dabei, einen konstanten Nachschub an Elektronen zu liefern, was mit dem Skript realisiert wird.

Zunächst erstellt man den Leiter. Dazu werden zwei längliche Rechtecke als oberer und unterer Rand platziert und angeheftet. Anschließend wird ein Atom erstellt. In Abbildung 44 wurde den Atomen noch eine Textur gegeben. Diesem Atom gibt man nun eine *Anziehung*. Der Wert hierfür sollte sich nach dem Größenverhältnis der Szene richten und ist am besten nachträglich so einzustellen, dass die Elektronen vernünftig durch das Leitermodell fließen. Man heftet das Atom an, dupliziert es einige Male und verteilt es gitterförmig im Leiter. Das Gitternetz ist hierbei sehr hilfreich. An einer Seite des Leiters, wo später der “Ausgang” sein soll, wird eine senkrechte Ebene erstellt und ihr die Option *Killer* im *Material*-Menü gegeben. Diese Ebene sorgt dafür, dass später Elektronen, die bereits durch den Leiter geflossen sind, direkt gelöscht werden.

Damit die Elektronen eine waagrechte Kraft erfahren, wird die Gravitationskraft um 90° gekippt und ein wenig verringert. Zusätzlich wird bei der Luftreibung der *Luftreibungsmultiplikator* auf einen Wert von etwa 60 erhöht, da sonst die Elektronen zu stark

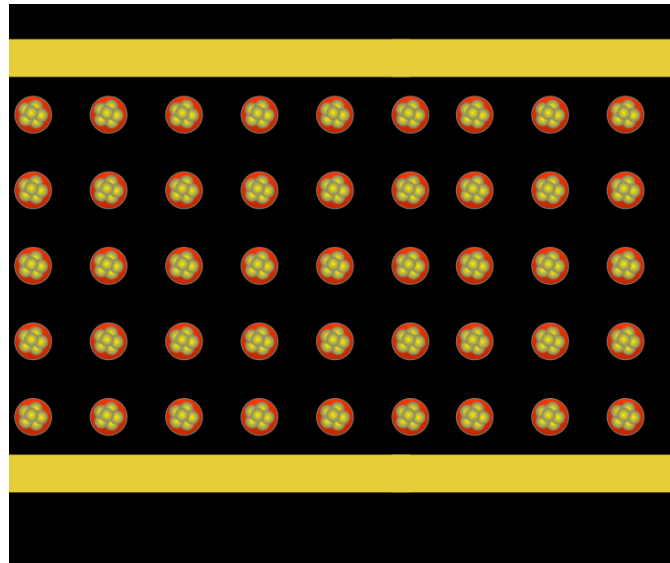


Abb. 44: *Ausschnitt des Leiters*

beschleunigt werden würden. Die genauen Werte für Gravitation und Luftreibung sollte man ebenfalls am Ende anpassen.

Nun wird eine Art “Generator” gebaut, der ständig Elektronen erstellt, um einen konstanten Strom aufrecht zu halten. Es wird das `onHitByLaser`-Skript genutzt. Dazu braucht man einen Laser und ein beliebiges weiteres Objekt, dem das Skript zugeschrieben wird. Man erstellt am “Eingang” des Leiters einen Laser und ein Objekt, zum Beispiel ein Rechteck. Um die Häufigkeit der Elektronenentstehung steuern zu können, wird ein Objekt, wie in Abbildung 45 zu sehen ist, erstellt, das sich um den Laser dreht. Dazu erstellt man über dem Laser einen Kreis und schneidet mit den *CSG*-Optionen einen kleineren Kreis aus. Dem übrigen Ring gibt man über *Rechtsklick* → *Geometrieaktionen* eine Mittelachse. Diese wird genau im Schwerpunkt platziert. Nun schneidet man noch ein Teil des Ringes ab, entweder mit dem *Lösch*-Werkzeug oder den *CSG*-Optionen. Die Drehachse macht man nun zum Motor und kann über dessen Drehzahl regeln, wie häufig das Rechteck vom Laser getroffen wird.

Das Skript, das dem Rechteck in das `onHitByLaser`-Feld im Skript-Menü geschrieben wird, soll ein Elektron erstellen, also einen kleinen Kreis mit einem bestimmten Radius und einer einheitlichen Farbe. Das Skript sieht also folgendermaßen aus:

```
(e)=>{ Scene.my.addCircle({
    radius := 0.1; pos := e.pos;
    color := [0,0,1,1]; drawCake := false
}) }
```

Dieses Skript erstellt nun am Treffpunkt des Lasers einen kleinen, blauen Kreis. Dieser würde nun aber weggeschossen werden, da er dieselbe Kollisionsgruppe wie das Rechteck besitzt und zur Hälfte in diesem erstellt wird. Man gibt also dem Rechteck Kollisionsgruppe *B*. Es wird immer noch vom Laser getroffen, das Elektron interagiert aber nicht mehr mit ihm.

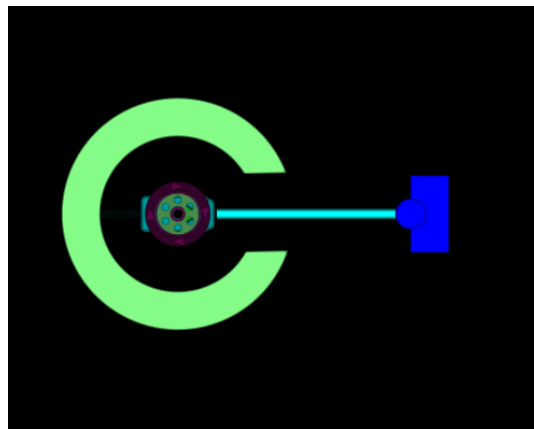


Abb. 45: Der *“Elektronengenerator”*

Dieser Generator erstellt aber noch sehr langsam einzelne Elektronen. Es gibt natürlich mehrere Möglichkeiten, die Anzahl zu erhöhen. Hier werden zwei Varianten vorgestellt, wobei beide zu den gleichen Ergebnissen führen, die zweite Variante aber eher dazu gedacht ist, tiefer in die Skriptsprache einzutauchen.

Die einfachere Möglichkeit wäre, dem Laser die Kollisionsgruppe *A* zu entfernen und die des Drehobjektes auf *B* zu ändern. Der neu erstellte Kreis blockiert damit nicht mehr den Laser und das Rechteck wird permanent angestrahlt, wodurch ein ganzer Stoß Kreise erstellt wird, die sich alle gegenseitig abstoßen. Ist allerdings die Drehzahl des Drehobjektes zu gering oder der Spalt zu groß, kann es zu einer wahren Flut von Elektronen kommen. Dabei ist ein wenig Fingerspitzengefühl gefragt. Man kann den Generator noch beliebig duplizieren.

Die zweite Möglichkeit besteht darin, zu der y-Koordinate der Erstellungsposition eine zufällige Zahl zwischen -2,5 und +2,5 zu addieren. Da aber keine Funktion zum Generieren zufälliger Zahlen bereit steht, muss man sich selbst aushelfen. In einem solchen Fall empfiehlt es sich, das Algodoo-Forum zu besuchen und im Unterforum **Thyme-scripting** nach einer Lösung zu suchen. In diesem Beispiel wird der Umstand ausgenutzt, dass neu erstellten Objekten eine zufällige Farbe zugewiesen wird, wenn keine speziell angegeben

wird. Der Farbcode besteht aus vier Zahlen zwischen 0 und 1 mit sieben Kommastellen. Ziel ist es also, einen weiteren Kreis mit zufälliger Farbe zu erstellen und eine der vier Zahlen im Farbcode einer Variablen zuzuweisen, die dann als Zufallszahl verwendet wird. Das ursprüngliche Skript kann erweitert werden, so dass ein zusätzlicher Kreis ein Stück außerhalb des für die Szene interessanten Bereichs erstellt wird. Diesem muss nun wiederum ein `onHitByLaser`-Skript gegeben werden, welches den Farbcode speichert. Zunächst platziert man also einen weiteren Laser, am besten ebenfalls per Skript, um ihn exakt platzieren zu können:

```
Scene.my.addLaserPen({ pos := [0,20] })
```

Damit der Farbcode in eine Variable gespeichert werden kann, muss diese zunächst initialisiert werden:

```
Scene.my.random := 0
```

Nun kann das Skript folgendermaßen erweitert werden:

```
(e)=>{ Scene.my.addCircle({  
  pos := [2,20];  
  onHitByLaser := (e)=> {Scene.my.random = e.geom.color(0)}  
});  
Scene.my.addCircle({  
  ...  
  pos := e.pos + [0, 5*(Scene.my.random - 0.5)];  
  ...  
})  
}
```

Der neu erstellte Kreis wird direkt vom Laser getroffen und der Variablen wird der erste Eintrag des Farbcodes zugewiesen. `e.geom.color(0)` bedeutet hierbei: Erster Eintrag des `color`-Arrays (Array-Indizes beginnen bei 0) der Geometrie des Ereignisses `e`. Schließlich wird zu der Position des Elektrons ein Array addiert, der die x-Position unverändert lässt, aber die y-Position um eine Zahl zwischen -2,5 und +2,5 verändert. Die Variable `Scene.my.random` wird hier sozusagen zunächst um 0,5 auf den Bereich [-0,5 ; 0,5] verschoben und auf [-2,5 ; 2,5] gestreckt.

Da der Laser auf das Rechteck nun ebenfalls nicht mehr blockiert wird, entstehen pro Umdrehung des Drehobjektes einige Elektronen und damit auch einige Kreise, deren Farbcode gespeichert wird. Diese sollen aber sofort wieder verschwinden. Dazu platziert

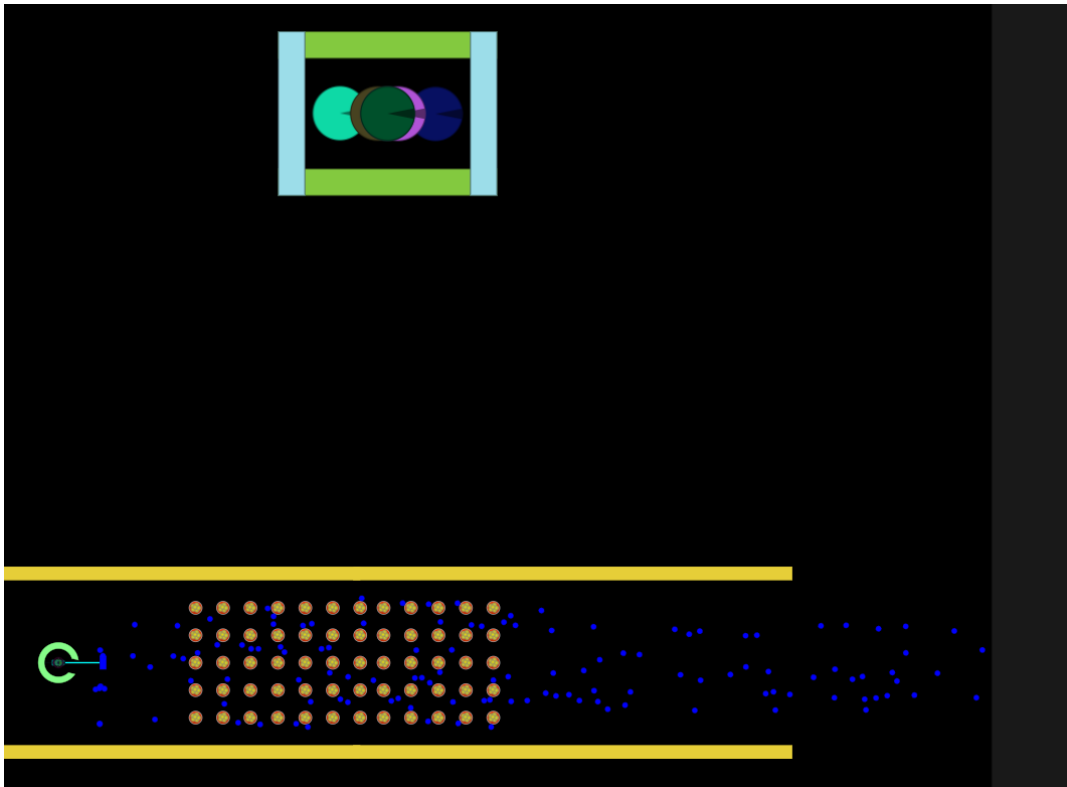


Abb. 46: Gesamtüberblick über die Szene

man einfach eine Art Käfig aus Rechtecken um den Bereich, in dem die Kreise entstehen, befestigt diese und gibt ihnen die Option *Killer*.

In Abbildung 46 ist die gesamte Szene zu sehen, inklusive der Kreise mit zufälligem Farbcode. Interessant ist davon allerdings nur der Bereich mit den Atomrümpfen im Inneren des Leiters. Der Generator würde nur vom Wesentlichen ablenken und kann noch von einem kollisionsgruppenfreien Rechteck überdeckt werden. Da auch hier die Gravitationskraft als elektrische Kraft benutzt wird, kann indirekt die Spannung am Leiter erhöht werden, indem man den Betrag der Gravitation erhöht.

4.4. Atomphysik

Ähnlich wie in der Elektronik ist es schwierig, in diesem Bereich Szenen zu erstellen, da Algodoo hier an seine Grenzen stößt. Die Vorgänge in der Atom- und Kernphysik lassen sich nicht mehr so einfach mit Mechanik darstellen.

4.4.1. Atommodelle

Einordnung in den Lehrplan In der 9. Jahrgangsstufe werden die einfacheren Atommodelle durchgenommen. In der 12. Jahrgangsstufe werden die Atommodelle der Quantenphysik eingeführt, auf welche aber hier in diesem Beispiel nicht eingegangen wird.

Ziel der Szene In diesem Abschnitt soll der Umgang mit der Algotbox gezeigt werden, denn es ist nicht immer notwendig, eine Szene selbst zu erstellen. In der Algotbox werden zahlreiche Szenen zur Verfügung gestellt. Allerdings können passende Szenen, wie auf Seite 30 schon beschrieben, schwer zu finden sein, oder nicht den Erwartungen entsprechen. Letzten Endes kommt es darauf an, was genau man mit der Szene bezwecken bzw. aufzeigen will und ob das durch eine Szene in Algotbox gut wiedergegeben wird.



Abb. 47: Algotbox mit dem Suchbegriff "Atom"

Erstellung Unter dem Suchbegriff “Atom” bekommt man eine ganze Reihe verschiedener Szenen angeboten, wie in Abbildung 47 zu sehen ist. Man sieht gleich, dass viele der Suchergebnisse nicht brauchbar sind, weil sie etwas anderes darstellen, aber zum Beispiel den Begriff “Atom” im Namen haben. Viele Szenen sind auch einfach nur Spielerei und nicht für den Unterricht brauchbar. Man sieht den Szenen ihre Qualität nicht an, da man nur einen Ausschnitt davon sieht und den Titel kennt. Manchmal verbirgt sich hinter einer scheinbar unpassender Szene eine sehr gute und umgekehrt kann eine vielversprechend aussehende Szene völlig nutzlos sein. Hier ist die einzige Lösung, es auszuprobieren.

Die erste Szene, die offensichtlich etwas mit einem Atommodell zu tun hat, ist “Lithium-7 atom”.²⁴ Klickt man auf *Play*, wird sie automatisch heruntergeladen und geöffnet. Zu sehen ist ein Atomkern, bestehend aus Protonen und Neutronen sowie sehr kleine, kaum sichtbare Elektronen. Startet man die Szene, beginnen die drei Elektronen auf elliptischen Bahnen um den Kern zu kreisen.

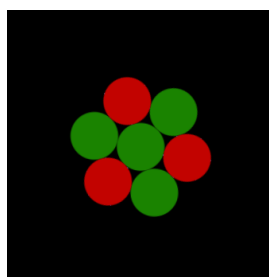


Abb. 48: Die Szene “Lithium-7 atom”

Diese Szene repräsentiert das Rutherford’sche Atommodell, bei dem die Elektronen auf Grund der Coulombkraft um den Kern kreisen wie Planeten um die Sonne. Der Autor hat sich Mühe gegeben, die wichtigen Aspekte des Modells umzusetzen. Elektronen haben eine negative Ladung und werden von den Protonen mit positiver Ladung angezogen und auf ihrer Bahn gehalten. Die Protonen und Neutronen wiederum ziehen sich aufgrund der Kernkräfte an, die durch Gluonen simuliert werden. Diese wurden als sehr kleine Kreise an die Protonen geklebt und mit einer sehr großen Anziehungskraft versehen.

Ein anders Beispiel ist die Szene “Bohr Neon Model”.²⁵ Der Autor versucht hier auf die Welleneigenschaften der Elektronen und auf deren Energieniveaus einzugehen. Allerdings zeigen sich hier die eingangs erwähnten Grenzen Algodoo in Bezug auf die Atomphysik.

²⁴xx.jason.xx xx.jason.xx: Lithium-7 atom. (URL: <http://www.algodoo.com/algobox/details.php?id=56802>).

²⁵MrBPhysics: Bohr Neon Model. (URL: <http://www.algodoo.com/algobox/details.php?id=37801>).

Elektronen und Kern bewegen sich in dieser Szene nicht mehr auf Grund von Kräften, sondern sind komplett durch Hilfsmittel gesteuert. Markiert man die gesamte Szene, wird sichtbar, dass alle Elektronen durch Federn mit Achsen, die sich drehen, im Mittelpunkt verbunden sind. Unterschiedliche Federhärten sorgen für verschiedene Wellenlängen und unterschiedliche Federlängen für verschiedene Bahnen.

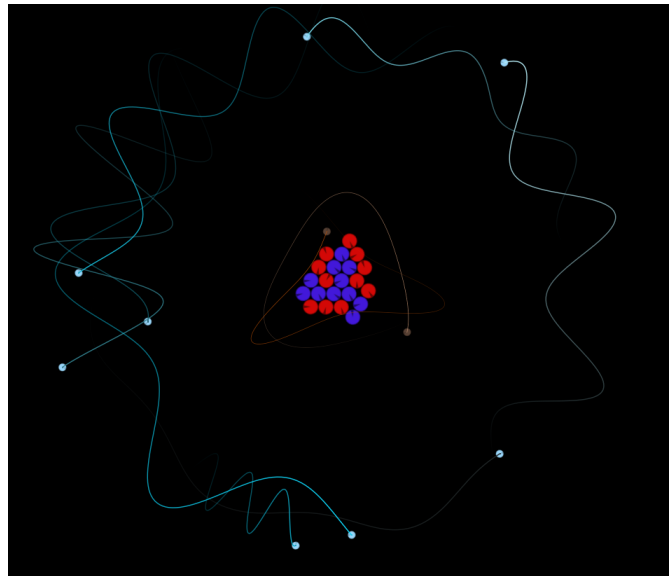


Abb. 49: Die Szene “Bohr Neon Model”

Ein drittes Beispiel ist die Szene “Atom Sim”.²⁶ Hier wurden neun verschiedene atomare Teilchen erstellt, die in einem abgetrennten Bereich präsentiert werden (siehe Abbildung 50, linke Seite). Um dies zu realisieren, wurden pro Teilchen bis zu vier verschiedene Kreise übereinander geheftet, die jeweils unterschiedliche Anziehungskräfte ausüben. Ähnlich wie bei dem Beispiel zuvor stoßen sich Protonen gegenseitig ab, wenn sie weit genug voneinander entfernt sind, ansonst ziehen sie sich an und können zusammen mit Neutronen einen Kern bilden.

Startet man die Szene, formt sich neben der Box ein Magnesium-Atom aus je zwölf Protonen, Neutronen und Elektronen. Nachdem sich der Kern gebildet hat, schweben die Elektronen um diesen herum. Sie befinden sich aber nicht korrekt auf Bahnen, da sie nur lose im Raum schweben. Man könnte dies jedoch als Aufenthaltswahrscheinlichkeit interpretieren.

Natürlich gibt es noch viele weitere Beispiele und Szenen zu diesem Thema und es werden immer neue generiert und zur Verfügung gestellt. Will man den Schülern also dieses Thema mit Hilfe von Algodoo näherbringen, bietet es sich an, immer mal wieder

²⁶Else, Someone: Atom Sim. (URL: <http://www.algodoo.com/algobox/details.php?id=51275>).

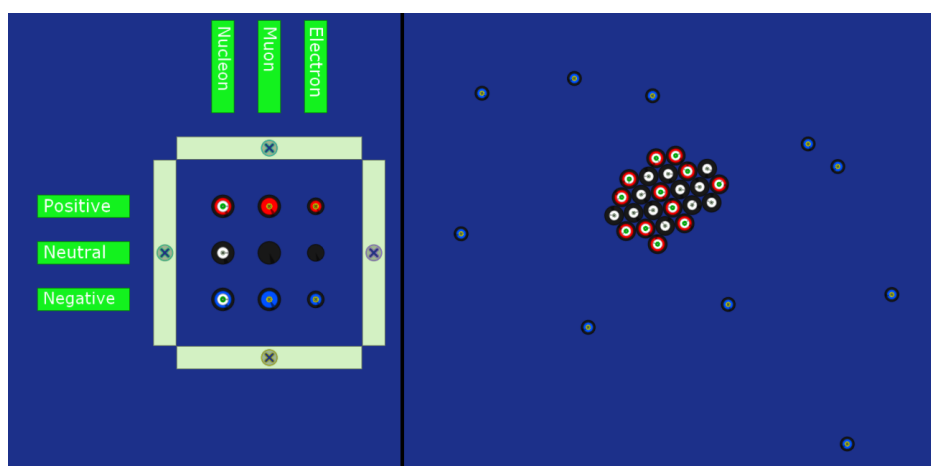


Abb. 50: Die Szene "Atom Sim"

einen Blick in die Algodoo zu werfen.

4.4.2. Rasterkraftmikroskop

Einordnung in den Lehrplan In der 9. Jahrgangsstufe kann beim Thema "Atome" das Rasterkraftmikroskop erwähnt werden. Seine Funktionsweise soll im folgenden Modell beschrieben werden.

Ziel der Szene Es soll ein Modell eines Rasterkraftmikroskops erstellt werden. Dieses funktioniert so, dass eine sehr kleine Spitze über eine Oberfläche fährt und von dieser Oberfläche aufgrund atomarer Kräfte ausgelenkt wird. Diese Auslenkung wird gemessen und ist somit ein Maß für die Oberflächenbeschaffung. In diesem Beispiel wird ein Wagen erstellt, der eine Probe über eine Oberfläche führt, die die Probe durch Gravitationskräfte anzieht. Die Probe wird an einer Feder befestigt und mit einem x-y-Graphen kann deren Auslenkung beobachtet werden. Die Schüler können mit dieser fertig erstellten Szene verschiedene Oberflächen abtasten.

Diese Szene soll auch ein Beispiel dafür sein, wie man mit Problemen umgeht, die bei komplexeren Szenen in Algodoo auftreten.

Voraussetzung Grundlagen, Kollisionsgruppen und Anziehungskräfte

Erstellung Auf der bereits bestehenden Ebene einer neuen Szene kann man gleich den zu scannenden Untergrund platzieren. Da in Algodoo das Gravitationsfeld schließlich unabhängig von der Form nur vom Schwerpunkt eines Objektes ausgeht, muss man mehrere kleine Objekte nutzen. Dafür bieten sich Quadrate an. In diesem Beispiel wurde

hierfür eine Seitenlänge von $0,5\text{ m}$ gewählt. Dank des Gitternetzes (eingestellt auf 4 Segmente) wird das erste Quadrat platziert, an den Hintergrund geheftet und mit einer Anziehungskraft von $5\frac{\text{Nm}^2}{\text{kg}^2}$ versehen. Anschließend kann das Quadrat beliebig oft kopiert und platziert werden wie in Abbildung 51 zu sehen.

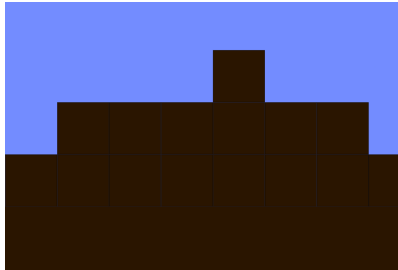


Abb. 51: *Beispiel für einen zu scannenden Untergrund*

Als nächstes kommt die eigentliche Arbeit: Das Modell des Rasterkraftmikroskops. Hier treten einige Probleme auf: An dem Wagen soll eine Feder befestigt werden, an welcher der Probekopf, eine Kugel, hängt. Da sie nur in y-Richtung ausgelenkt werden soll, braucht man eine Art Führungsschiene. Da der gesamte Rest des Mikroskopkopfes nichts mit der Anziehungskraft der Oberfläche zu tun haben soll, muss diesem verschiedene Kollisionsgruppen zugeordnet werden. Desweiteren reicht ein einfacher Wagen aus einem Rechteck und zwei Kreisen nicht aus, da er durch die Kräfte auf die Probe ebenfalls verrutschen würde. Abhilfe schaffen hier eine Schiene und Zahnräder.

Zunächst erstellt man den Wagen. Dafür wurde ein Rechteck mit einer Höhe von $0,5\text{ m}$ und einer Breite von 2 m gewählt, sowie zwei Zahnräder mit einer *Zahnradgröße* von $0,1\text{ m}$ (in den *Zahnradoptionen*) und einem Radius von $0,5\text{ m}$. Zahnräder sind allerdings immer etwas größer als der gewählte Radius und sollten daher noch kurz mit dem *Skalieren*-Werkzeug in Form gebracht werden. Diese werden, wie in Abbildung 52 zu sehen ist, an dem Rechteck angebracht und mit einem Motor versehen, dem Pfeiltasten zugewiesen werden. Nun erstellt man eine Unterlage, heftet diese an den Hintergrund und bringt dort in regelmäßigem Abstand kleine Zacken an, in die die Zahnräder greifen können. Bei den in diesem Beispiel erstellten Zahnrädern passte ein Abstand von $0,125\text{ m}$ und einer halb so großen Breite. Das lässt sich sehr bequem mit dem Gitternetz realisieren. Direkt über dem Rechteck wird noch eine weitere Schiene platziert und an den Hintergrund befestigt, die dafür sorgt, dass der Wagen auch in Kontakt mit der unteren Schiene bleibt, da er ansonsten manchmal einfach darüberhüpfen würde. Damit der Wagen ohne Störung in der Schiene entlang geführt wird, stellt man deren Reibung und die des Rechtecks auf 0.

Nun wird die Probe angebracht. Zunächst erstellt man ein langes Rechteck (in Abbil-

dung 52 markiert, aber unsichtbar) und schneidet mit einem schmalen Rechteck und den *CSG*-Optionen ein Mittelstück heraus, in dem dann die Probe geführt wird. Man setzt es über das Rechteck des Wagens und befestigt es mit dem *Fixieren*-Werkzeug daran. Da sie für die eigentliche Simulation nicht wichtig ist, kann man sie unsichtbar machen (im *Aussehen*-Menü den Alpha-Wert auf 0 und die Kanten entfernen). Innerhalb dieser Führungsschiene wird nun ein Kreis erstellt: die eigentliche Probe. Sie sollte exakt so breit sein wie das ausgeschnittene Stück (Gitternetz). Zuletzt wird der Kreis mit einer Feder an dem Rechteck des Wagens befestigt.

Nun werden die Kollisionsgruppen bearbeitet. Da der Wagen nicht vom Untergrund beeinflusst werden soll, entfernt man zunächst die Kollisionsgruppe A von allen Objekten des Wagens außer der Probe. Zahnräder und untere Schiene sollen interagieren, also bekommen sie Kollisionsgruppe B (auch die Zacken der Schiene). Die obere Schiene soll mit dem Rechteck interagieren, aber nicht mit den Zahnrädern, also bekommen Rechteck und obere Schiene Kollisionsgruppe C. Die Führungsschiene des Probekopfes soll eigentlich nur mit diesem interagieren, also bekommt sie Gruppe D und der Probekopf bekommt ebenfalls zusätzlich zu A die Gruppe D.

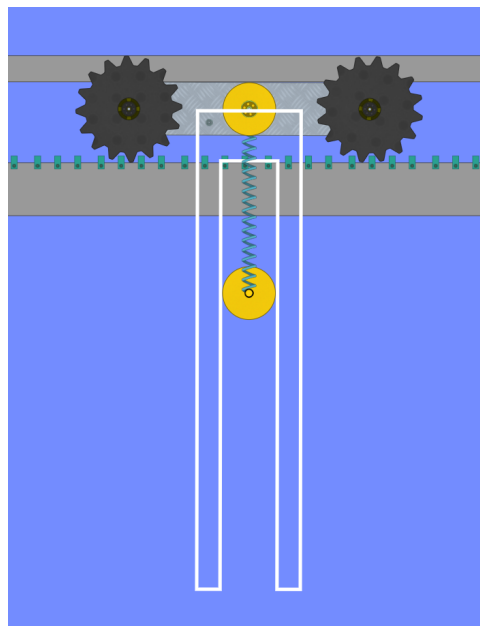


Abb. 52: *Der Mikroskopkopf*

Nun muss man noch ein paar Feineinstellungen vornehmen. Dazu platziert man die gesamte Messvorrichtung, falls nicht schon geschehen, über der Oberfläche. Man lässt sich den Graphen der Probe anzeigen und stellt diesen auf x-Position und y-Position, sowie die Dauer auf unendlich. Nun kann man die Simulation starten und lässt den

4. Beispielszenen

Mikroskopkopf einmal darüber fahren. Im Idealfall hat man gleich ein einigermaßen schönes Bild, das die Oberfläche widerspiegelt, wie in Abbildung 53 zu sehen ist. Die Oberfläche wird verkehrt herum und nur sehr undeutlich wiedergegeben. Man sollte daher bei der Oberfläche nicht zu viele Details einbauen.

Hat man noch kein vernünftiges Bild erhalten, muss man so lange die Federhärte und die Anziehungskräfte verändern, bis man ein besseres Bild erhält. Notfalls muss man sogar die komplette Anordnung der zu scannenden Oberfläche verändern und vereinfachen.

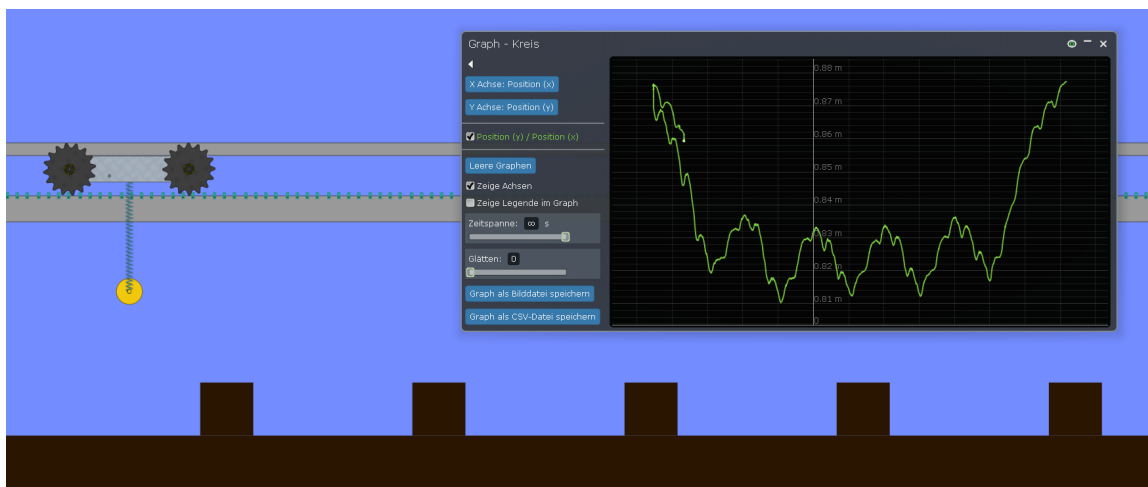


Abb. 53: Die gesamte Szene nach einem Scandurchlauf

Anwendung im Unterricht Das Rasterkraftmikroskop kann entweder einfach nur durchgeführt werden oder die Schüler können selbst damit experimentieren. Man kann zum Beispiel eine kollisionsgruppenfreie Blackbox über die Oberfläche legen und die Schüler diese scannen und Rückschlüsse auf deren Beschaffenheit ziehen lassen. Den Rechtecken können auch unterschiedliche Gravitationskräfte gegeben werden, so dass sie unterschiedliche Atome repräsentieren. Damit kann man die Verwendungsmöglichkeit richtiger Rasterkraftmikroskope nachahmen.

5. Fazit

Beim Erstellen der Beispiele in Kapitel 4 wurden die Möglichkeiten und Stärken von Algodoo aufgezeigt, aber auch dessen Schwächen. Es bleibt die Frage zu klären, ob Algodoo zum Einsatz im Unterricht geeignet ist und sich eine Anschaffung lohnt.

Algodoo ist deutlich auf Benutzerfreundlichkeit ausgelegt. Die Oberfläche wirkt freundlich und nicht so steril und kalt wie es bei vielen andern Programmen der Fall ist. Sehr schnell sind einfache Szenen erstellt und man kann Ergebnisse direkt beobachten. Diese Einfachheit lädt zum Experimentieren ein. Dies birgt allerdings sowohl Vor- als auch Nachteile. Auf der einen Seite ist es natürlich sehr gut, wenn die Schüler motiviert werden, eigenständig die Physik zu erkunden und kennenzulernen, auf der anderen Seite können Schüler bei so viel Freiheit schnell das Ziel aus den Augen verlieren und sich in Spielereien verlieren. Gerade in Situationen, in denen Schüler mit einer komplett neuen Szene starten, ist es notwendig, eindeutige Instruktionen zu geben. Wie bereits auf Seite 6 beschrieben, verlieren sich Schüler in zu großer Freiheit oder passen ihre Experimente unbewusst so an, dass sie ihren Fehlvorstellungen entsprechen. Algodoo mit seinen vielen Möglichkeiten ist hierfür besonders anfällig.

Steigt man tiefer in Algodoo ein, findet man hier und da Unzulänglichkeiten, wie zum Beispiel das Fehlen besserer Werkzeuge zum geometrischen Erstellen von Objekten, oder die in Kapitel 4.2.6 auf Seite 63 beschrieben, Fehler in der Optik. Allerdings muss erwähnt werden, dass dies meistens nur Detailfehler sind, die an sich nicht schwer in Gewicht fallen. Es werden auch immer wieder Mängel durch die zahlreichen Updates behoben. Während diese Arbeit entstand, wurde die Version 2.0.0 von Algodoo veröffentlicht, die beispielsweise die Benutzeroberfläche zugänglicher oder die Seile überhaupt erst benutzbar gemacht hat. Diese haben sich vor der Verbesserung bereits bei geringen Lasten völlig unphysikalisch verhalten und waren deshalb kaum zu gebrauchen.

Bei den oben erstellten Beispielen wird klar, dass Algodoo gerade mechanische Sachverhalte sehr anschaulich präsentieren kann. Anhand der Graphen und Vektorpfeile von Kräften, Geschwindigkeiten und Impulsen lassen sich wichtige physikalische Größen schön darstellen und untersuchen. Diese Anschaulichkeit gepaart mit der einfachen Bedienbarkeit machen Algodoo zu einem guten Werkzeug für den Physikunterricht. Dadurch, dass Algodoo dazu geeignet ist, sowohl als Experimentierbaukasten für Schüler eingesetzt zu werden, als auch vom Lehrer zur Präsentation verwendet zu werden ist es relativ flexibel und oft einsetzbar.

Wer Algodoo einmal selbst ausprobieren möchte, kann dies mit der kostenlosen Testversion auf der beigelegten CD tun oder die aktuellste Testversion von der Algodoo

5. *Fazit*

Homepage unter www.algodoo.com herunterladen. Insgesamt kann also eine Empfehlung für den Simulationsbaukasten Algodoo ausgesprochen werden.

Literatur

Algoryx: Algodoo teacher edition (URL: www.algodoo.com/wiki/Algodoo_teacher_edition), abgerufen am 28.03.2012 um 14:45.

Algoryx: Changes and updates in Algodoo (URL: www.algodoo.com/wiki/Changes), abgerufen am 28.03.2012 um 14:35.

Berger, Veit; Mikelskis, Helmut: Physik-Didaktik. Heidelberg 2010.

Domagk, Steffi: Bewegtbilder: Animation, Video und Simulation. In: Kompendium multimediales Lernen. Hg. v. **Helmut Niegemann et al.** Berlin, Heidelberg 2008, S. 239 - 273.

Goldkuhle, Peter: Modellbildung und Simulation mit dem Computer im Physikunterricht. In: Praxis Schriftreihe Physik 54. Köln 1997.

ISB : Lehrpläne / Standards Gymnasium G8 (URL: <http://www.isb.bayern.de/isb/index.asp?MNav=0&QNav=4&TNav=0&INav=0&Fach=&LpSta=6&STyp=14>), abgerufen am 31.03.2012 um 16:45.

Kircher, Ernst et al. : Physikdidaktik. Berlin 2009.

MrBPhysics : Bohr Neon Model (URL: www.algodoo.com/algobox/details.php?id=37801), abgerufen am 27.03.2012 um 16:45.

Someone Else: Atom Sim (URL: <http://www.algodoo.com/algobox/details.php?id=51275>), abgerufen am 27.03.2012 um 16:30.

Wikipedia : Algodoo (URL: en.wikipedia.org/wiki/Algodoo), abgerufen am 28.03 um 14:00.

Wikipedia : Brillant (URL: de.wikipedia.org/wiki/Brillant), abgerufen am 28.03.2012 um 17:00.

Wikipedia : Phun (URL: [en.wikipedia.org/wiki/Phun_\(2D_Physics_Sandbox\)](http://en.wikipedia.org/wiki/Phun_(2D_Physics_Sandbox))), abgerufen am 28.03.2012 um 14:00.

Wilhelm, Thomas und Trefzger, Thomas : Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern. Didaktik der Physik. Hannover 2010.

xx.jason.xx : Lithium7 atom (URL: www.algodoo.com/algobox/details.php?id=56802), abgerufen am 27.03.2012 um 16:35.

B. Anhang

Inhalt der beigelegten CD:

\Komponenten		7 - Impuls
Anhaenger.phz		8 - Gasmodell
Auge.phz		\Optik
Fahrzeug.phz		1 - Schatten.phz
Ladung.phz		2 - Sonnen-Mond-Finsternis.phz
Lichtkegel.phz		4 - Brillant.phz
\Szenen		5 - Auge.phz
\Atomphysik		6 - Teleskop.phz
1-1 - Lithium7 Atom.phz		\Texturen
1-2 - Bohr Neon Model.phz		\Fahrzeuge
1-3 - Atom-Sim.phz		anhaenger.jpg
2 - Rasterkraftmikroskop		anhaenger_reifen.jpg
\Elektronik		suv bearbeitet.jpg
1 - Elektrisches Feld.phz		suv_reifen.jpg
2 - Kondensator.phz		\Planeten
3 - Leitermodell.phz		erde.jpg
\Mechanik		Halbmond2.jpg
2 - Kräfte.phz		mond.jpg
3 - Waagrechtter Wurf.phz		Sonne.jpg
4 - Beschleunigung.phz		Atom.png
5 - Karussell.phz		Auge.png
6 - Erde-Mond.phz		Algodoo_2_0_1-Win32

Danksagung

Ich möchte mich bei allen Bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben.

Mein Dank gehört denjenigen, die diese Arbeit Korrektur gelesen haben, vor allem Sabine, Katharina und Rieke.

Zudem möchte ich mich bei Prof. Wilhelm, der mich bei der Erstellung dieser Arbeit betreut hat, bedanken. Er hatte immer ein offenes Ohr für Fragen und sonstige Probleme und gab mir viele wichtige Ratschläge.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die Arbeit in allen Teilen
selbstständig gefertigt und keine anderen als die in der Arbeit
angegebenen Hilfsmittel benutzt habe.

Die Zeichnungen, Kartenskizzen und bildlichen Darstellungen
habe ich selbst gefertigt.

Ort, Datum

Unterschrift