

PHYSIKSIMULATIONEN
MIT DER DYNAMISCHEN GEOMETRIESOFTWARE
CINDERELLA 2.0

SCHRIFTLICHE HAUSARBEIT
FÜR DIE ERSTE STAATSPRÜFUNG
FÜR DAS LEHRAMT AM GYMNASIUM

VON

SEBASTIAN KROHNE

VORGELEGT AM: 31. MÄRZ 2010

BETREUER: DR. THOMAS WILHELM
PRÜFER: PROF. DR. THOMAS TREFZGER

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

Inhaltsverzeichnis

1. Einleitung	1
2. Das Gedächtnis	3
2.1. Sensorisches Gedächtnis	3
2.2. Arbeitsgedächtnis	3
2.3. Langzeitgedächtnis	5
2.4. Zusammenarbeit zwischen Arbeitsgedächtnis und Langzeitgedächtnis	6
3. Bilder, Animationen und Simulationen	7
3.1. Multicodierung	7
3.2. Bildhafte Darstellungen	8
3.3. Animationen	10
3.4. Simulationen	12
4. Lerntheorien	15
4.1. Die Cognitive-Load-Theorie	15
4.1.1. Intrinsic Cognitive Load	15
4.1.2. Extraneous Cognitive Load	16
4.1.3. Germane Cognitive Load	18
4.1.4. Didaktische Gedanken	18
4.2. Die kognitive Theorie multimedialen Lernens	19
4.2.1. Didaktische Gedanken	21
4.3. Integriertes Modell des Text- und Bildverstehens	22
4.3.1. Mentale Repräsentationen	23
4.3.2. Das Text- und Bildverständnismodell	23
4.3.3. Das Integrierte Modell des Text- und Bildverstehens	24
4.3.4. Didaktische Empfehlungen	25
5. Dynamische Geometrie Systeme	27
5.1. Allgemeines	27
5.2. Euklid DynaGeo	29
5.3. Geogebra	30
5.4. Cinderella 2.0	32
5.4.1. CindyLab - Der Physikmodus von Cinderella 2.0	33

5.4.2. CindyScript - Die Scriptsprache von Cinderella 2.0	43
6. Ausgewählte Physiksimulationen	59
6.1. Die Wellenmaschine	59
6.1.1. Die Wellenmaschine für Transversalwellen	59
6.1.2. Wellenmaschine für stehende Wellen	62
6.1.3. Wellenmaschine für longitudinale Wellen	65
6.2. Elastische Stoßvorgänge	67
6.2.1. Der elastische Stoß	67
6.2.2. Impulserhaltung	70
6.3. Wurfbewegungen	71
6.3.1. Der waagerechte Wurf	72
6.3.2. Der schräge Wurf	75
6.4. Planetenbewegungen	77
6.4.1. Ellipsen als Planetenbahnen (Erstes keplersches Gesetz)	77
6.4.2. Zweites keplersches Gesetz	79
6.5. Elektrische Feldlinienbilder	82
6.5.1. Elektrische Feldlinien	83
6.6. Das homogene Magnetfeld	86
6.6.1. Einfacher Massenspektrograph	86
6.7. Strahlenoptik	89
6.7.1. Die optische Bank	89
6.8. Möglichkeiten und Grenzen von Cinderella 2.0	90
7. Fazit	92
8. Literaturverzeichnis	95
A. Anhang	99
Danksagung	101
Selbstständigkeitserklärung	103

1. Einleitung

In meiner Hausarbeit „Physiksimulationen mit der dynamischen Geometriesoftware Cinderella 2.0“ beschäftige ich mich mit dem Computereinsatz im Physikunterricht des Gymnasiums. Wie WILHELM¹ in einer aktuellen Erhebung festgestellt hat, nimmt der Einsatz des Computers im Physikunterricht des Gymnasiums stark zu. Dabei werden gerade Simulationen besonders häufig genutzt. In diesem Zusammenhang kann meine Arbeit einen Beitrag leisten, den Einsatz von Simulationen mit der dynamischen Geometrie Software Cinderella 2.0 zu erweitern.

Warum sind es gerade Simulationen, die beim Computereinsatz im Unterricht eine immer größere Verbreitung finden?

Die Antwort ist naheliegend: Sie sind besonders gut geeignet, einen motivierenden und effektiven Unterricht zu gestalten. Diese Annahme hat mich motiviert, dieses Thema in meiner Hausarbeit zu bearbeiten.

Während meiner Praktika an verschiedenen Gymnasien habe ich Unterricht gestaltet, bei dem auch der Computer, speziell Simulationen, zum Einsatz gekommen sind. Dabei ist mir aufgefallen, dass zwar eine Vielzahl von Simulationen im Internet zu finden ist, die aber oft nicht genau das darstellen, was ich mir gewünscht hätte. Das Programm Cinderella 2.0 ermöglicht es, physikalische Simulationen auf relativ unkomplizierte Art und Weise selbst zu erstellen. Außerdem ist das Programm Cinderella 2.0 interessant, weil es ein dynamisches Geometrieprogramm ist und somit auch im Mathematikunterricht eingesetzt werden kann. Deshalb habe ich dieses Programm für meine Arbeit ausgewählt.

Ist Cinderella 2.0 für einen Einsatz in der Schule geeignet?

Im Laufe der Arbeit wird untersucht, ob diese Frage positiv beantwortet werden kann, welche Möglichkeiten mit Cinderella 2.0 erstellte Simulationen bieten und wie sie programmiert werden sollten, um einen effektiven Unterricht zu ermöglichen. Dabei wird auf die psychologischen und pädagogischen Hintergründe eingegangen, die bei Simulationen berücksichtigt werden sollten.

Der erste Teil dieser Arbeit befasst sich mit diesen psychologischen und pädagogischen Hintergründen.

In Kapitel 2 werden die Funktionen des menschlichen Gedächtnisses auf der Basis des Mehrspeichermodells von ATKINSON & SHIFFRIN genauer erklärt. Ein Verständnis von der Arbeitsweise des Gedächtnisses ist die Grundlage dafür, die nachfolgenden Kapitel über „Bilder, Animationen und Simulationen“ sowie „Lerntheorien“ erfassen zu können.

Kapitel 3 der Arbeit befasst sich mit den verschiedenen Darstellungsformen von Lerninhalten.

¹vgl. WILHELM, THOMAS und TREFZGER, THOMAS (2010): *Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern*. In GRÖTZEBAUCH, HELMUTH und NORDMEIER, VOLKARD (Hrsg.): *Didaktik der Physik - Hannover 2010, Frühjahrstagung der DPG*. Hannover.

Ausgehend von dem Stichwort Multicodierung werden aufeinander aufbauend die Darstellungsformen Bilder, Animationen und Simulationen beschrieben. Die Simulationen werden dabei genauer betrachtet. Es wird darauf eingegangen, worin sich die verschiedenen Darstellungsformen von Simulationen unterscheiden und welche Bedingungen in einer Klasse erfüllt sein müssen, damit ein richtiger Umgang mit Simulationen möglich ist.

Im 4. Kapitel werden die wichtigsten Lerntheorien vorgestellt, die zum Thema Multimedia, also auch zum Einsatz von Simulationen, vertreten werden. Dabei werden zum einen der Begriff der kognitiven Belastung (Cognitive Load) und zum anderen der Zusammenhang zwischen Text- und Bild-Verständnis erläutert. Zu jeder Theorie werden didaktische Hinweise gegeben, die beim Einsatz von Multimediainhalten im Unterricht berücksichtigt werden sollten.

Der 2. Teil bzw. der Hauptteil dieser Arbeit befasst sich mit dem Programm Cinderella 2.0 und der Erstellung eigener Simulationen.

Im 5. Kapitel werden „Dynamische Geometrie Systeme“ (DGS) behandelt. Nach einem allgemeinen Überblick über die Geschichte und die Funktionen heutiger DGS werden die dynamischen Geometrie Programme Euklid DynaGeo, Geogebra und Cinderella 2.0 vorgestellt. Darüber hinaus wird die Möglichkeit aufgezeigt, mit diesen Programmen Physiksimulationen zu erstellen. Das Programm Cinderella 2.0 findet hierbei besondere Beachtung. Es werden die integrierten Programmumgebungen CindyLab und CindyScript genauer erklärt, weil sie für die Programmierung der besprochenen Simulationen verwendet werden.

In Kapitel 6 werden Simulationen vorgestellt, die mit Cinderella 2.0 erstellt worden sind. Es erfolgt zu jeder Simulation eine Einordnung in den Lehrplan, eine Konstruktionsbeschreibung und eine Diskussion der Vor- und Nachteile. Am Ende dieses Kapitel wird festgehalten, welche Möglichkeiten Cinderella 2.0 bietet und welche Verbesserungen wünschenswert wären.

Abschließend wird ein persönliches Fazit aus den Erkenntnissen der Arbeit gezogen.

2. Das Gedächtnis

2.1. Sensorisches Gedächtnis (Ultrakurzzeitgedächtnis)

Unter dem sensorischen Gedächtnis versteht man ein Aufnehmen externer Reize, die im Gehirn verarbeitet werden sollen. Die Reize können Gerüche, Geräusche, optische oder geschmackliche Reize oder etwas Gefühltes sein. Aufgabe des sensorischen Gedächtnisses ist es zu erkennen, welche dieser Reize für die entsprechende Person wichtig sind. Unwichtige Reize werden herausgefiltert und nur die wichtigen an das Arbeitsgedächtnis weitergeleitet.¹

2.2. Arbeitsgedächtnis (Kurzzeitgedächtnis)

Das Verständnis des Arbeitsgedächtnisses hat sich im Laufe des letzten Jahrhunderts stark verändert. ATKINSON & SHIFFRIN stellten 1968 ein Modell vor, in dem das Arbeitsgedächtnis einen Kurzzeitspeicher mit begrenzter Kapazität darstellt. Die Kapazität dieses Speichers wurde mit simplen Gedächtnisaufgaben festgelegt, z. B. durch das Wiederholen möglichst vieler vorher genannter Zahlen. Diese Testmethode wurde 1956 von MILLER erfunden und bis in die 1990er häufig verwendet, um die Gedächtniskapazität zu bestimmen.

HASSELHORN und SCHUMANN-HENGSTELER stellten 2001 folgende Definition vom Arbeitsgedächtnis auf: „Das Arbeitsgedächtnis ist ein internes kognitives System, das es ermöglicht, mehrere Informationen vorübergehend bewusst zu halten und miteinander in Beziehung zu setzen,“². Hierbei war es nicht mehr nur wichtig, wie viele Items jemand reproduzieren kann, sondern vor allem, in welcher Form und in welcher Geschwindigkeit diese Items wiedergegeben werden können. Man spricht hierbei von der funktionalen Kapazität des Arbeitsgedächtnisses.

Neuere neuropsychologische Ergebnisse haben ergeben, dass das Arbeitsgedächtnis kein einheitlicher Speicher ist. Es kann in mehrere Formen unterteilt werden. In einem Teil werden visuell-räumliche und in einem anderen phonologische Vorgänge bearbeitet. Diese Teile werden von einer zentralen Exekutive gesteuert. Die zentrale Exekutive ist eine Art Aufmerksamkeits-system und besitzt keine eigene Speicherkapazität. 2000 erweiterte BADDELEY seine Theorie um einen Verbindungsmechanismus zwischen der zentralen Exekutive und dem Langzeitgedächtnis. Dieser episodische Puffer dient der Optimierung. Er soll die Kapazität des Arbeitsgedächtnisses bestmöglich steuern, indem er die Informationen aus den visuell-räumlichen und phonologischen Vorgängen und dem Langzeitgedächtnis koordiniert.

Die visuell-räumlichen Vorgänge speichern zwei Arten von optischen Reizen. Der visuelle Speicher („visual cache“) speichert Muster und wird durch leichte Veränderungen der Farbe oder

¹vgl. HASSELHORN, MARCUS (2009): *Pädagogische Psychologie*. Stuttgart: Kohlhammer S. 50.

²HASSELHORN: *Pädagogische Psychologie*, S. 73 f.

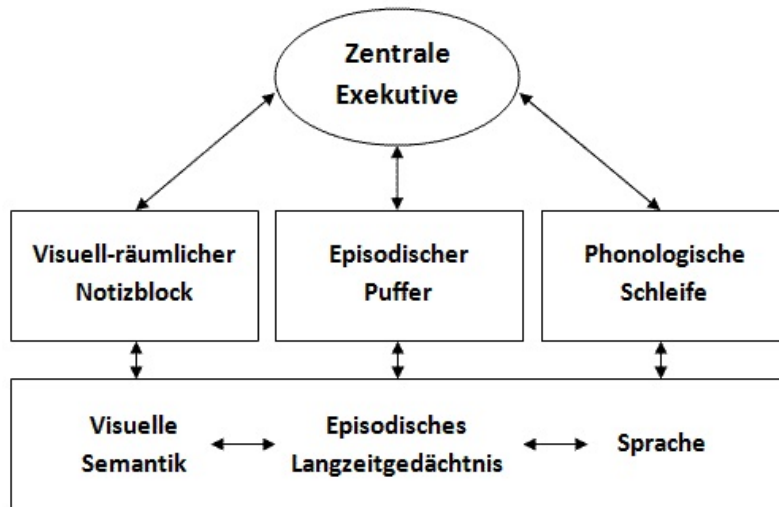


Abbildung 2.1.: Modell des Arbeitsgedächtnisses nach Baddeley ³

des Musters beeinträchtigt, nicht aber von Bewegungen, die während des Versuchs durchgeführt werden. Im Gegensatz hierzu steht ein Prozess zur Aufnahme räumlicher Bewegungssequenzen („inner scribe“). Die Speicherung dieser Bewegungsabläufe kann, z. B. durch Bewegungen, die während des Versuchs durchgeführt werden müssen, beeinträchtigt werden. Beim Aufnehmen räumlicher Sequenzen ist es nicht von Bedeutung, ob Farbe und Muster der Sequenz sich währenddessen ändern.⁴

Die phonologischen Vorgänge wurden auch von BADDELEY (1986) in zwei Komponenten aufgeteilt, in den phonetischen Speicher („phonological store“) und in den subvokalen Kontrollprozess („subvocal rehearsal“). Anders als bei MILLER ist der phonetische Speicher nicht auf eine bestimmte Anzahl von Items begrenzt, sondern auf eine Zeitspanne, in der die gespeicherte Information verfügbar ist. Man vergleicht diesen Speicher mit einem immer fortlaufenden Tonbandgerät. Auf diesem Tonbandgerät werden alle Informationen, die nicht für wichtig erachtet werden oder auf die keine Aufmerksamkeit gerichtet wurde, nach ungefähr zwei Sekunden wieder überschrieben. Die als wichtig erachteten Informationen werden weitergeleitet, weiterverarbeitet und auch für längere Zeit gespeichert. Da phonologische Reize oft länger anhalten als zwei Sekunden und diese Zeit nicht zum Verstehen des Aufgenommenen ausreicht, gibt es den subvokalen Kontrollprozess. *Inneres Sprechen* hält die Reize im Speicher, wodurch diese vor der endgültigen Löschung ständig neu in den Speicher gerufen werden können. Dieser subvokale Artikulationsprozess wird schon im Kindesalter automatisch von den Menschen benutzt. Darüber hinaus besitzt er aber noch weitere wichtige Eigenschaften. Eine weitere Funktion des subvokalen Kontrollprozesses ist es, bildliche Informationen so umzustrukturieren, dass sie vom phonetischen Speicher verstanden werden können. Gemeint ist das *bildliche Sprechen*: Ein Mensch sieht etwas

³vgl. HASSELHORN: *Pädagogische Psychologie*, S. 74.

⁴vgl. HASSELHORN: *Pädagogische Psychologie*, S. 76.

und spricht innerlich die Worte für das, was er gesehen hat. Dieser Prozess setzt sich fort, indem man Grapheme⁵ zu deuten lernt. Das bedeutet, dass man lernt, bestimmten Buchstabenkombinationen Bilder und Worte zuzuordnen, wodurch das geschriebene Wort in den phonetischen Speicher gelangen kann.

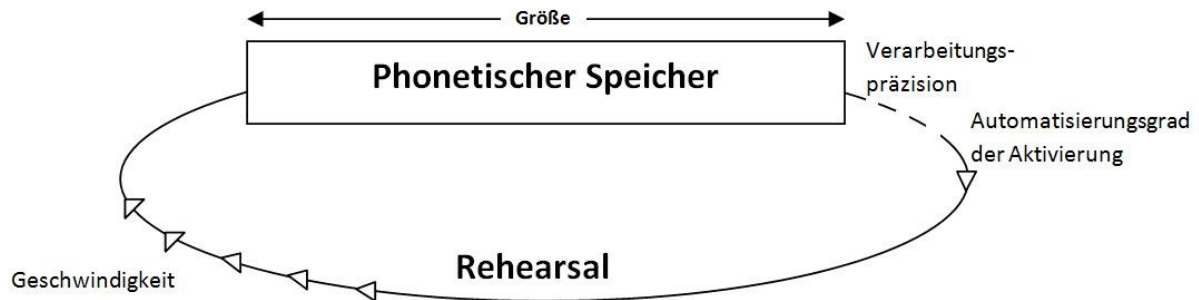


Abbildung 2.2.: Zwei-Komponenten-Modell des phonologischen Arbeitsgedächtnisses⁶

2.3. Langzeitgedächtnis

Das Langzeitgedächtnis wird als der permanente Wissensspeicher des Menschen beschrieben. Es werden Informationen, Zusammenhänge und Erfahrungen darin gespeichert, sodass zu einem späteren Zeitpunkt darauf zurückgegriffen werden kann⁷. Die Kapazität des Langzeitgedächtnisses scheint ebenso wie die Fähigkeit, bestimmte Informationen abrufen zu können, unbegrenzt zu sein⁸. Man unterscheidet drei verschiedene Teile des Langzeitgedächtnisses: das *semantische Gedächtnis*, in dem fachliches Wissen, bspw. die Elemente des Periodensystems, gespeichert werden, das *episodische Gedächtnis*, in welchem persönliche Erfahrungen gespeichert werden, sowie das *prozedurale Gedächtnis*, welches die Grundlage komplexer motorischer Fertigkeiten ist, z. B. des Schwimmens⁹. Wissen wird in sogenannten Schemata gespeichert. Unter einem Schema versteht man, wie unser Gedächtnis Informationen strukturiert und organisiert. Dies kommt z. B. bei der Organisation ähnlichen oder verwandten Wissens zum Tragen, etwa durch Ober- und Unterbegriffe, z. B. Pinguine, Emus und Kuckucke sind Vögel. Eine spezielle Form dieser Schemata sind Skripte. Skripte können eine Reihe bekannter aufeinanderfolgender Bewegungsabläufe sein, aber auch eine Verknüpfung bestimmten Wissens, welches häufig im Zusammenhang benötigt wird. Ein Beispiel für einen solchen Bewegungsablauf ist das Schleife binden oder ein Salto beim

⁵Graphem, auch Grafem *das*; -s, -e <aus gleichbed. *engl.-amerik.* grapheme zu *gr.* gráphema „Schrift“>; kleinstes bedeutungsunterscheidendes grafisches Symbol, welches ein oder mehrere Phoneme wiedergibt (Sprachw.); WISSENSCHAFTLICHER RAT DER DUDENREDAKTION (2000): *Duden, Das große Fremdwörterbuch*. Mannheim, Leipzig, Wien, Zürich.

⁶vgl. HASSELHORN: *Pädagogische Psychologie*, S. 79.

⁷vgl. NIEGEMANN, HELMUT und DOMAGK, STEFFI und HESSEL, SILVIA und HEIN, ALEXANDRA und HUPFER, MATTHIAS und ZOBEL, ANNETT (2008): *Kompendium multimediales Lernen*. Berlin: Springer Verlag S. 43.

⁸vgl. GAGE, NATHANIEL L. und BERLINER, DAVID C. (1996): *Pädagogische Psychologie*. 5. Auflage. Weinheim: Psychologie Verlags Union S. 281.

⁹vgl. HASSELHORN: *Pädagogische Psychologie*, S. 50 f.

Turnen. Auch Lernen ist als ein Skript anzusehen; jeder Mensch lernt unterschiedliche Dinge, aber häufig unter Anwendung der gleichen Methoden.¹⁰

2.4. Zusammenarbeit zwischen Arbeitsgedächtnis und Langzeitgedächtnis

SWELLER stellte 2005 fest, dass die Kapazität des Arbeitsgedächtnisses nur von neuen Informationen verbraucht werden kann. Informationen aus dem Langzeitgedächtnis können so oft und solange wie nötig in das Arbeitsgedächtnis gerufen werden. Wie gut jemand lernen kann, hängt also davon ab, wie schnell Informationen verarbeitet und in Schemata eingeordnet werden können. Die Kapazität des Arbeitsgedächtnisses wird also nur überschritten, wenn zu schnell zu viele neue Informationen aufgenommen werden. Der Prozess des Verstehens bezeichnet den Übergang vom Arbeitsgedächtnis in das Langzeitgedächtnis. Beim Verstehen geht es darum, die neuen Informationen so in das Langzeitgedächtnis weiter zu leiten, dass diese dort fest verankert werden. Dieser Vorgang ist nötig, um weitere Informationen aufnehmen zu können. Neue Informationen können nur aufgenommen werden, wenn das Arbeitsgedächtnis und das Langzeitgedächtnis simultan arbeiten, ohne dass die Kapazität des Arbeitsgedächtnisses überschritten wird. Beinhalten die Daten, die zu verarbeiten sind, zu viele neue Informationen, müssen diese in kleinere Lerneinheiten zerlegt werden. Das zu Lernende muss im Langzeitgedächtnis in vorhandene Schemata eingefügt werden können, ohne dass die Kapazität überschritten wird. Nur so kann sichergestellt werden, dass es im Langzeitgedächtnis auf Dauer gespeichert wird.¹¹

¹⁰vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 44.

¹¹vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 44.

3. Bilder, Animationen und Simulationen

3.1. Multicodierung

Nach WEIDENMANN¹ beschreibt Multicodierung Lernangebote, die verschiedene Symbolsysteme oder Codierungen aufweisen. Darunter versteht man eine Kombination aus piktoralen, verbalen oder zahlengebundenen Darstellungen. HEUER nennt speziell für den Physikunterricht fünf Ebenen der Codierung.² Die Codierung der ersten Ebene findet in Form von Sprache und Text statt. Sie ist die am häufigsten verwendete Art Lerninhalte zu vermitteln. Damit keine Missverständnisse auftreten, ist es von Bedeutung, dass der Lernende³ alle Begriffe, die zum Erklären benutzt werden, kennt. Auf der zweiten Ebene werden Abläufe durch Bilder dargestellt. Dies können sowohl Videos, Animationen oder Filme sein, die dem Lernenden präsentiert werden, als auch Abbildungen, die einen Vorgang stroboskopartig darstellen, wie in Abbildung 3.1. Mit diesen Stroboskopbildern lassen sich auch vergleichende Aussagen über die zeitliche Entwicklung von Gegenständen machen. Eine dritte Ebene stellen Strukturaussagen mit Hilfe von dynamisch ikonischen Repräsentationen (DIR)⁵ dar. Anhand von Vektoren, Flächen etc. werden physikalische Größen sowie ihre Abhängigkeiten dargestellt. Dabei wird ständig auf die Codierung aus Ebene zwei zurückgegriffen. Die vierte Ebene, die im Physikunterricht wohl am häufigsten

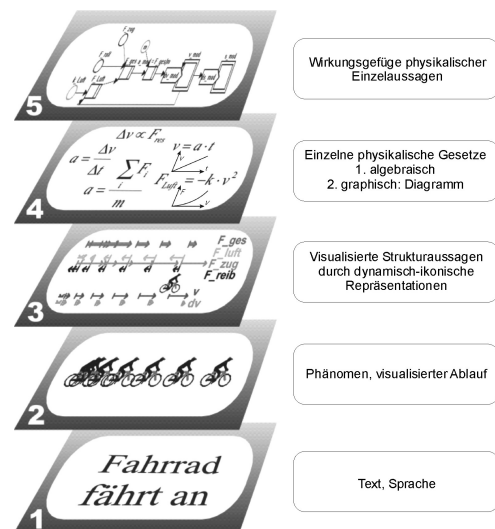


Abbildung 3.1.: Fünf Stufen der Codierung nach HEUER⁴

¹vgl. WEIDENMANN, BERND (2005): *Multicodierung und Multimodalität im Lernprozess*. In ISSING, LUDWIG J. (Hrsg.): *Information und Lernen mit Multimedia und Internet: Lehrbuch für Studium und Praxis*. Weinheim: Psychologie Verlags Union - Beltz Verlag S. 47.

²vgl. HEUER, DIETER (2003): *Physikunterricht gestaltet mit Multimedia. Praxis der Naturwissenschaften - Physik in der Schule*, 52/3 S. 2 f.

³Aus Gründen der leichteren Lesbarkeit verzichte ich auf eine geschlechtsspezifische Differenzierung, wie z. B. Schüler/innen. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung für beide Geschlechter

⁴vgl. WILHELM, THOMAS (2005): *Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung*. Dissertation an der Bayerischen Julius-Maximilians-Universität Würzburg, Würzburg, Deutschland, (URL: <http://www.physik.uni-wuerzburg.de/~wilhelm/veroeffentlichung/Dissertation.pdf>) S. 36

⁵nachzulesen bei WILHELM: *Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung*, S. 37 ff.

zum Einsatz kommt, sind algebraische Gleichungen oder Diagramme, die einzelne physikalische Gesetze bzw. Zusammenhänge darstellen. Die fünfte Ebene stellt die einzelnen physikalischen Abhängigkeiten, die einen Ablauf bestimmen, durch Symbole dar, welche durch Pfeile verknüpft werden. Durch diese Darstellung eines Wirkungsgefüges werden alle Zusammenhänge strukturell dargestellt und qualitativ verständlich.

In dieser Arbeit und den dargestellten Animationen und Simulationen werden hauptsächlich die Ebenen zwei bis vier Verwendung finden.

3.2. Bildhafte Darstellungen

Bilder lassen sich grundsätzlich in drei Oberkategorien einteilen, nach WEIDENMANN sind dies künstlerische, unterhaltende und informierende Bilder.⁶ Für den Physikunterricht sind speziell die informierenden Bilder interessant. Diese lassen sich nochmals unterteilen in Abbilder und logisch analytische Bilder.⁷ ISSING unterscheidet Abbildungen, analoge Bilder und logische Bilder, wobei die analogen Bilder als eine Sonderform der Abbildungen betrachtet werden können.⁸

Abbilder sind beispielsweise Photographien, Zeichnungen und Animationen, sie zeigen die visuell wahrnehmbaren Merkmale eines Objekts. Im Speziellen sind Zeichnungen gut geeignet für das Lernen, da hier eine Reduktion auf das Wesentliche stattfinden kann. Wichtige Elemente werden hervorgehoben und unwichtige vernachlässigt. Außerdem können Abläufe differenziert dargestellt werden, indem man Momentaufnahmen eines Ablaufs zeigt.⁹

Als Sonderform der Abbilder stellen die analogen Bilder auch existente direkt beobachtbare Objekte dar, sehen also aus wie Abbilder. Im Unterschied zu den Abbildern wollen sie aber Strukturen und Prozesse darstellen, die nicht direkt beobachtbar sind. Man unterteilt die analogen Bilder nochmals in funktionale und strukturelle Analogien. Dementsprechend helfen funktionale Analogien (z. B. Elektronenwolken zur Anzeige der Aufenthaltswahrscheinlichkeiten) die Funktionsweise und strukturelle Analogien (z. B. Elektronendrift als Bild für den elektrischen Strom in Metallen) die Struktur von etwas zu verstehen. Ein Nachteil der daraus entsteht, weil das Objekt bzw. der Prozess nicht direkt beobachtbar ist, dass unerwünschte Nebeninformationen aufgenommen werden können. Dies führt eventuell zu Fehlvorstellungen, z. B. lassen sich beim Atommodell nur schwer angemessene Größenrelationen darstellen.¹⁰

Logische Bilder sind solche, „[...] durch die Sachverhalte sichtbar gemacht werden, die in der Realität so nicht wahrgenommen werden können, und die keine Ähnlichkeit mit dem Dar-

⁶vgl. WEIDENMANN, BERND (1994): *Informierende Bilder*. In WEIDENMANN, BERND (Hrsg.): *Wissenserwerb mit Bildern. Instruktionale Bilder in Printmedien, Film/Video und Computerprogrammen*. Bern: Verlag Hans Huber S. 9.

⁷vgl. WEIDENMANN, BERND (1991): *Lernen mit Bildmedien: Psychologische und didaktische Grundlagen*. Weinheim, Basel: Beltz-Verlag S. 43ff.

⁸vgl. KIRCHER, ERNST und GIRWIDZ, RAIMUND und HÄUSSLER, PETER (2007): *Physikdidaktik - Theorie und Praxis*. Berlin: Springer S. 203.

⁹vgl. WILHELM, THOMAS (WS 2008/09): *Einführung in die Fachdidaktik I*. WS 2008/09 (URL: http://www.physik.uni-wuerzburg.de/~wilhelm/vorlesung/PPP_Kapitel15.pdf) 5.3.2.

¹⁰vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 204.

gestellten besitzen.“¹¹ Gemeint sind damit Diagramme oder grafische Darstellungen von Daten oder Funktionszusammenhängen.¹² Logische Bilder stellen also Zusammenhänge von Merkmalen eines Sachverhalts dar. Diese können sowohl qualitativer als auch quantitativer Art sein.¹³

Bilder in Medien sollen bestimmte Funktionen erfüllen. In der Literatur gibt es eine Vielzahl von Autoren, die Funktionen von Bildern nennen. Z. B. beschreibt BALLSTAEDT gesondert für jede Art von Bild die Funktionen.¹⁴ WEIDENMANN nennt allgemein die *Ersatz-*, die *Fokus-*, die *Konstruktions-* und die *Aktivierungsfunktion* von Bildern.¹⁵ KIRCHER, GIRWIDZ und HÄUSSLER nennen aufbauend hierauf Funktionen, die speziell für den Physikunterricht von Interesse sind. Auf diese Funktionen wird im Weiteren genauer eingegangen.¹⁶

Die Funktionen der Bilder, ermöglichen einen variantenreichen (didaktischen) Einsatz im Unterricht. Für die Wissensvermittlung sind die nachstehend beschriebenen Funktionen von besonderer Bedeutung.

Die *Zeigefunktion* soll den Lernenden möglichst deutliche und angemessene bildhafte Vorstellungen vermitteln. Auch Abläufe von Vorgängen lassen sich durch eine Folge von Einzelbildern darstellen. Der Informationsgehalt von Bildern, die zum Lernen benutzt werden, ist in der Regel hoch, wenn unbekannte Sachverhalte vermittelt werden. Es empfiehlt sich, die Bilder mit weiteren methodischen Maßnahmen zu unterlegen und somit eine gezielte Aufnahme und Verarbeitung zu sichern, z. B. durch erläuternden Text oder mündliche Hinweise.

Die *Fokussfunktion* dient dazu, Details klar zu machen oder Fehlvorstellungen zu korrigieren. Hierzu benutzt man (Zeit-)Lupenaufnahmen oder Querschnitte, die auf spezielle Punkte hinweisen. Der Lernende muss beim Einsatz dieser Art von Bildern Vorkenntnisse besitzen, um den gezeigten Detailinhalt einordnen zu können.

Die *Konstruktionsfunktion* kombiniert bekannte Sachverhalte, Prozesse oder Vorgehensweisen und fördert somit neue Erkenntnisse. Zusätzlich kann man die Wirkung des Bildes noch dadurch verstärken, dass man symbolische Ergänzungen beifügt, um den neuen Sachverhalt deutlicher darzustellen, z. B. Kräftepfeile.

Die *physikspezifischen Visualisierungen* sind ein spezieller Punkt, den KIRCHER, GIRWIDZ und HÄUSSLER nennen. Bilder, die als Visualisierungen verstanden werden, sind optische Vorstellungshilfen für den Lernenden. Dies können zum Beispiel direkte Verknüpfungen mit Messergebnissen sein oder Visualisierungen zum besseren Verständnis abstrakter Sachverhalte, z. B. Feldlinienbilder.

Eine Kombination aus Bild und Text kann unterstützend wirken. Bilder können hier sowohl Er-

¹¹ SCHNOTZ, WOLFGANG (1994): *Wissenserwerb mit logischen Bildern*. In WEIDENMANN, BERND (Hrsg.): *Wissenserwerb mit Bildern. Instruktionale Bilder in Printmedien, Film/Video und Computerprogrammen*. Bern: Verlag Hans Huber S. 95.

¹² vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 204.

¹³ vgl. SCHNOTZ: *Wissenserwerb mit logischen Bildern*, S. 96.

¹⁴ vgl. BALLSTAEDT, STEFFEN-PETER (1997): *Wissensvermittlung: Die Gestaltung von Lernmaterial*. Weinheim: Psychologie Verlags Union - Beltz Verlag.

¹⁵ vgl. WEIDENMANN: *Lernen mit Bildmedien: Psychologische und didaktische Grundlagen*, S. 39.

¹⁶ vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 206 f.

gänzung zum Text als auch inhaltlicher Schwerpunkt mit Beschreibung sein. Oft dienen Bilder als *Ersatz für komplexe Beschreibungen*. Ist ein Sachverhalt zu kompliziert, als dass er ausformuliert werden könnte, kann man diesen Sachverhalt durch eine Grafik verdeutlichen, die einfacher zu verstehen ist, z. B. Magnetfelder. Ebenso verhält es sich häufig bei Situationsbeschreibungen. Soll das Hauptaugenmerk auf dem Text liegen, haben Bilder oft eine *Repräsentationsfunktion*, d. h. sie sollen den Inhalt des Textes widerspiegeln. Die *Interpretationsfunktion* dient dazu, einen bestimmten Textinhalt zu verstehen. Das Bild ist eine zusätzliche Hilfe für den zu lernenden Sachverhalt. Eine weitere bekannte Form ist die *Bildanleitung*, z. B. bei IKEA-Aufbauanleitungen. Der geschriebene Text wirkt hier nur unterstützend, die eigentliche Anleitung wird allein durch Bilder verständlich dargestellt. Häufig findet man in Schulbüchern Bilder, die vorrangig Anreize bieten sollen, aber keinen spezifischen Inhalt besitzen, dieses ist die *dekorative Funktion* von Bildern.

Bilder können dafür genutzt werden, den Lernstoff zu organisieren oder zu strukturieren. Sie können dazu dienen, wesentliche Ergebnisse herauszuarbeiten oder bestimmte Details hervorzuheben. Um diese *Organisations- & Strukturierungsfunktion* besser nutzen zu können, gibt es bestimmte Arten von Bildern, die dies unterstützen. Die sogenannten *Concept maps* sind Bilder, in denen Inhalte, Konzepte und Zusammenhänge strukturiert dargestellt werden. *Advance Organizer* sind Texte oder Bilder, die vor einem Lernabschnitt gezeigt werden, um Vorwissen zu aktivieren und zu strukturieren. Bilder, die einen bestimmten Wissensinhalt so darstellen, dass dieser sich besser merken lässt als reiner Text und die dem Gedächtnis sozusagen als „Eselbrücken“ dienen, haben eine *gedächtnisstützende Funktion*.

Eine wichtige Funktion von Bildern ist die *Motivierungsfunktion*. Damit Schüler sich mit einem Lerninhalt auseinandersetzen, müssen sie ausreichend motiviert sein. Diese Motivation kann mit Hilfe von Bildern erzeugt werden. Dabei ist es wichtig darauf zu achten, dass das Bild nicht nur dekorativ ist. Dies hat keine positiven Auswirkungen auf die Motivation des Lernenden. Vielmehr soll darauf Wert gelegt werden, dem Bild eine repräsentierende, organisierende oder interpretierende Funktion zuzuweisen. Dann können positive Lernerfolge erwartet werden.

3.3. Animationen

Das Wort Animation kommt vom lateinischen *animare* und bedeutet „zum Leben erwecken“. Diese Wortbedeutung lässt bereits erkennen, was man sich unter einer Animation vorzustellen hat: Einzelbilder werden zusammengefügt und abgespielt, was zu einer scheinbaren Bewegung des Objekts führt. „*Dieser Effekt wird durch eine schnelle Folge von Standbildern erzeugt und als stroboskopische Bewegung bezeichnet.*“¹⁷ Um einen kontinuierlichen Verlauf darstellen zu können, benötigt man mindestens 16 Bilder pro Sekunde.¹⁸ Eine modernere Definition basiert auf SCHNOTZ & LOWE, demnach sind „*Animationen [...] bildhafte Darstellungen, deren Struktur und Eigenschaften sich über die Zeit verändern und die Wahrnehmung einer kontinuierlichen*

¹⁷GOLDSTEIN (1996), zitiert in NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 240.

¹⁸vgl. BALLSTAEDT: *Wissensvermittlung: Die Gestaltung von Lernmaterial*, S. 189.

Veränderung erzeugen.“¹⁹ Bezogen auf den Physikunterricht beschreibt WILHELM Animationen als „bildhafte Repräsentationen, die sich dynamisch ändern. Aber sie müssen nicht unbedingt etwas über physikalische Größen aussagen. Wenn sie sich jedoch physikalischen Größen entsprechend bewegen, kann man sie auch zu den dynamisch ikonischen Repräsentationen rechnen. Dynamisch ikonische Repräsentationen (und Animationen) können sowohl bei Simulationen als auch bei Messungen verwendet werden.“²⁰

Animationen lassen sich in verschiedene Arten mit verschiedenen Funktionen einteilen, vergleichbar mit den Bildern. LOWE unterscheidet die Art einer Animation danach, auf welche Weise die Veränderungen ablaufen.²¹ Es entstehen drei Arten:

- transformierende Animation: Das Objekt ändert seine Form.
- translative Animation: Das Objekt verändert seine Position.
- transitive Animation: Die Elemente innerhalb der Animation werden verändert.

Die Funktionen, die sich bei Animationen ergeben, lassen sich in zwei Bereiche einordnen. Zum einen sind Animationen bewegte Bilder und können deshalb ähnliche Funktionen zugeordnet bekommen, wie statische Bilder (vgl. 3.2). Wie bei Bildern sollte auch darauf geachtet werden, dass die Animation eine didaktische Funktion erfüllt und nicht nur dekorative Funktion besitzt.²² Andererseits betrachten RASCH & SCHNOTZ die Funktionen in Relation zur kognitiven Belastung (vgl. 4.1 Cognitive Load).²³

Ist eine Aufgabe ohne Animation so schwierig, dass die Kapazität des Arbeitsgedächtnisses nicht ausreicht um den Inhalt vollständig zu verstehen, so bietet eine Animation eventuell die Möglichkeit den Inhalt verständlich zu machen. Die Animation hat somit einen *ermöglichenden* Effekt für das Verständnis der Aufgabe. Eine schwere Aufgabe kann durch eine Animation vereinfacht werden, indem sie beispielsweise einem möglichen Umdenken vorgreift und das Ereignis aus einer anderen Perspektive zeigt. Hierbei tritt ein *erleichternder* Effekt der Animation ein. Es können aber auch gegenteilige Effekte auftreten. Läuft eine Animation zu schnell ab oder ist sie zu komplex, so kann es vorkommen, dass nicht alle relevanten Informationen vom Arbeitsgedächtnis aufgenommen und in das Langzeitgedächtnis weitergeleitet werden. Es tritt dann eine *Erhöhung der Schwierigkeit* ein. Der umgekehrte Effekt, eine sogenannte *unpassende Erleichterung* der Aufgabe, kann dann auftreten, wenn die Animation für das Verstehen überflüssig ist. Der Lernende nimmt die wichtigen Inhalte dann nur oberflächlich auf. Die Folge ist, dass der Inhalt nicht oder in zu geringem Maße abgespeichert wird. Welche Effekte bei welchem Schüler eintreten, ist abhängig davon, welche Vorkenntnisse er besitzt. Dasselbe Animation kann zum

¹⁹SCHNOTZ & LOWE, zitiert in NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 241.

²⁰WILHELM: *Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung*, S. 37.

²¹vgl. LOWE in NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 241.

²²vgl. BALLSTAEDT: *Wissensvermittlung: Die Gestaltung von Lernmaterial*, S. 189.

²³vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 255.

Beispiel bei einem Schüler erleichternd wirken, bei einem anderen hingegen eine unpassende Erleichterung hervorrufen.

Es gibt keine eindeutigen Erkenntnisse, wie Animationen auf den Lernenden wirken. Allerdings kann man, ähnlich wie bei den Bildern, sagen, dass es Bedingungen gibt, unter denen der Einsatz von Animationen im Unterricht sinnvoll ist. HÖFFLER & LEUTNER führten hierzu eine Metaanalyse durch, in der Animationen im Vergleich zu statischen Bildern untersucht wurden. Dabei wurde festgestellt, dass Animationen statischen Bildern überlegen sind, wenn sie lernzielrelevante Dinge darstellen und nicht nur dekorative Funktion besitzen. Animationen sind nicht nur dann sinnvoll, wenn etwas Dynamisches dargestellt werden soll, sondern auch, wenn deklaratives Wissen vermittelt werden soll. Dies stützt die Vermutung, dass die Gemeinsamkeiten von statischen Bildern und Animationen stärker berücksichtigt werden sollten. Animationen mit geringem Realitätsgrad führen nicht zu schlechteren Lernergebnissen, sollten sich jedoch auf wesentliche Inhalte beschränken. Wie oben bereits erwähnt, können Animationen aber nur dann hilfreich sein, wenn das Vorwissen und die spezifischen Voraussetzungen des einzelnen Schülers berücksichtigt werden.²⁴

3.4. Simulationen

Simulationen dienen der Nachbildung ausgewählter Realitätsaspekte „mit Hilfe einer (häufig nicht sichtbaren) vorgefertigten Berechnung, wobei sich in der Regel einzelne Parameter verändern lassen oder man in den Ablauf eingreifen kann, so dass die Folgen der Handlung deutlich werden.“²⁵ Sie sind eine Art der Datengewinnung: Daten werden auf der Grundlage von angenommenen Zusammenhängen berechnet, ohne Messdaten darüber zu haben. Eine Simulation wird meistens mit Hilfe eines Computers erzeugt. Dadurch ergibt sich eine Alternative, den Begriff (Computer-)Simulation zu interpretieren. „Unter einer Computersimulation versteht man, die Nachahmung realer Prozesse mittels Computern auf der Grundlage formaler (analytischer, numerischer oder grafischer) Modelle.“²⁶ Der Nutzer soll hierbei durch die Interaktionen etwas über die dargestellten Phänomene und Abläufe lernen. Die Ergebnisse solcher Simulationen können auf unterschiedliche Weise dargestellt werden.

Eine Form der Darstellung der Ergebnisse von Simulationen sind die sogenannten *Simulationen mit Graphendarstellung*. Mit ihnen wird eine Berechnung durchgeführt und anschließend das Ergebnis in einem Graphen oder einem Diagramm dargestellt. Ein Beispiel für ein solches Simulationsprogramm ist das Tabellenkalkulationsprogramm Excel.

Als eine andere Form, die auf den in 3.3 beschriebenen Animationen aufbaut, können *animierte Simulationen* beschrieben werden als Animationen, die einen bestimmten Grad der Interaktivität

²⁴vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 256 ff.

²⁵WILHELM: *Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung*, S. 47.

²⁶BERGER, VEIT (2006): *Mit dem Computer im Unterricht modellieren*. In MIKELSKIS, HELMUT F. (Hrsg.): *Praxishandbuch für die Sekundarstufe I und II*. Berlin: Cornelson Scriptor Verlag S. 140.

überschreiten. Der Lernende sollte die Möglichkeit besitzen, Parameter innerhalb der Animation abzuändern, um so den Ablauf der Animation verändern zu können.²⁷

Um Simulationen im Unterricht nutzen zu können, müssen bestimmte Voraussetzungen erfüllt sein: Die Schüler kennen Simulationen als möglichen Weg zur Erkenntnisgewinnung bzw. können sich mit Hilfe der Simulation ein mentales Modell bilden (vgl. 4.3). Sie können Simulationen selbstständig durchführen und kennen die geeigneten Werkzeuge hierfür. Des Weiteren muss den Schülern bekannt sein, dass reale Prozesse simuliert werden. Die erworbenen Erkenntnisse müssen passend zum Kontext evaluiert werden.²⁸

Wie auch Bilder und Animationen können Simulationen ein reduziertes Abbild der Wirklichkeit wiedergeben. Dieser Aspekt kann sich in wissenschaftlichen Simulationen allerdings nachteilig auswirken, weil damit Unsicherheiten und Ungenauigkeiten innerhalb der Simulation entstehen. Zum Beispiel können Ungenauigkeiten in einem Flugsimulator schwerwiegende Folgen für das Flugverhalten in der Realität haben. Andererseits bieten didaktische Simulationen die Möglichkeit der Reduktion auf das Wesentliche. Der Lernende kann die wichtigen Einflussgrößen beeinflussen und erkennt somit auch leichter ihre Wirkung.²⁹ Ein weiterer wichtiger Aspekt von Simulationen ist, dass der Lernende den Inhalt durch Exploration erfasst. Fehler, die durch Einstellen der Einflussgrößen verursacht werden, können sofort erkannt werden und ihre Auswirkungen sind nur in der Simulation ohne Konsequenzen und nicht in der Realität sichtbar (wie im Beispiel des Flugsimulators). Beim Lernen durch Exploration muss aber darauf geachtet werden, dass die Schüler nicht ohne Anleitung experimentieren. Eine solche Herangehensweise führt dazu, dass Lernende versuchen, mit Hilfe eigener Experimente ihre alltäglichen Vorstellungen zu bestätigen. Die realen Vorgänge stimmen aber oft nicht mit den Vorstellungen der Schüler überein.³⁰

Es kann zwischen *modellierenden* und *modellbildenden* Simulationen unterschieden werden.³¹ In modellierenden Simulationen ist das Grundgerüst schon gegeben. Der Lernende hat die Möglichkeit, mit der Simulation zu arbeiten und einige Parameter zu verändern, um die Auswirkungen zu beobachten. Die wohl bekanntesten Simulationen dieser Art sind Flugsimulatoren. Die Piloten können ein Flugzeug steuern und Einstellungen in und an dem Flugzeug vornehmen. Im Gegensatz hierzu stehen die modellbildenden Simulationen. Der Lernende hat die Möglichkeit das System der Simulation selbst zu programmieren bzw. zu erstellen. Anhand der ablaufenden Simulation kann der Lernende dann überprüfen, ob alle Variablen bedacht worden sind und wie sich diese auswirken, z. B. eine selbsterstellte Simulation mit Cinderella 2.0, in der eine gedämpfte Federschwingung stattfindet. Dabei muss beachtet werden, dass ein globales Kraftfeld vorhanden ist, da sonst keine gedämpfte Schwingung stattfindet.

²⁷ vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 260.

²⁸ vgl. BERGER: *Mit dem Computer im Unterricht modellieren*, S. 139.

²⁹ vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 409.

³⁰ vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 264.

³¹ vgl. RIEBER, LLOYD P. (2005): *Multimedia Learning in Games, Simulations and Microworlds*. In MAYER, RICHARD E. (Hrsg.): *Cambridge Handbook of Multimedia Learning*. Cambridge: Cambridge University Press S. 555.

Einen Teil der modellbildenden Simulationen stellen die Modellbildungssysteme dar. „Bei einem Modellbildungssystem werden nicht nur einzelne Parameter variiert, sondern der Benutzer muss selbst die Zusammenhänge zwischen den Größen angeben. Vom Modellbildungssystem wird nach diesen Angaben das Programm, nach denen die abschließende Simulation abläuft, erst erstellt, um eine Simulation entsprechend den eigenen Vorstellungen sehen zu können.“ Man kann die verschiedenen Modellbildungssysteme nach der Art ihrer Ausgabe unterscheiden und erhält so zwei Arten von Modellbildungsprogrammen:

- Programme, die Simulationsergebnisse durch Diagramme darstellen. (Beispiel: Excel, Newton II)
- Programme, die Simulationsergebnisse auch in Animationen ausgeben können. (Cinderella 2.0, Modellus 4, VisEdit/PAKMA, JPAKMA)

Außerdem gibt es die virtuellen Welten, die sowohl modellanwendend als auch modellbildend sind. In ihnen sind die physikalischen Gesetze schon bekannt und man muss sich nur überlegen welche Körper vorkommen und welche Eigenschaften diese haben sollen. Beispiele für diese virtuellen Welten sind „Interactive Physics“³² und „Phun“³³.

Da mit Cinderella 2.0 eine einfache Möglichkeit besteht Konstruktionen als Applet zu exportieren, wird nun genauer betrachtet, wie Simulationen von Lehrern dargestellt werden. In einer aktuellen Erhebung von WILHELM über den Computereinsatz bei Physik-Gymnasiallehrern in Unterfranken stellte sich heraus, dass 55% der Physiklehrer Simulationen durch Applets darstellen.³⁴ Im Vergleich zu einer Studie in Rheinland-Pfalz auch von WILHELM aus dem Jahr 2004, bei der 33% der Lehrer angaben, Applets zur Darstellung von Simulationen zu verwenden, ist der Prozentsatz – zumindest in Unterfranken – signifikant angestiegen. „Die Stärkere Verwendung in Unterfranken 2009 gegenüber Rheinland-Pfalz könnte daran liegen, dass sich in den 4,5 Jahren die Computer-Kompetenzen der Lehrer verbessert haben bzw. jüngere Lehrer nachrückten, die dem Computer gegenüber aufgeschlossener sind.“³⁵ In beiden Studien waren Mehrfachnennungen möglich. Darüber hinaus ist auffällig, dass Programme, wie PAKMA oder JPAKMA (entwickelt an der Universität Würzburg), in Unterfranken viel häufiger vertreten sind als in Rheinland-Pfalz. Gleiches gilt umgekehrt für das Programm *Albert*.

In dieser Arbeit wird keine Form der Simulation als didaktisch besser geeignete Form angesehen. Die dargestellten Simulationen sollen vielmehr alle als Unterrichtshilfen dienen, aber auch dazu anregen, eigene Simulationen zu erstellen. Die in Kapitel 6 dargestellten Simulationen können auch eigenständig oder mit Hilfe der Konstruktionsbeschreibung von den Schülern selbst erstellt oder im Unterricht von ihnen benutzt werden.

³²<http://www.design-simulation.com/ip/index.php>

³³www.phunland.com

³⁴vgl. WILHELM und TREFZGER: *Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern*, S. 5.

³⁵WILHELM und TREFZGER: *Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern*, S. 4f.

4. Lerntheorien¹

4.1. Die Cognitive-Load-Theorie

Die Cognitive-Load-Theorie wurde 1991 von SWELLER & CHANDLER entwickelt, um Aussagen über die verschiedenen kognitiven Belastungen machen zu können. Ausgangspunkt dieser Theorie ist die begrenzte Kapazität des Arbeitsgedächtnisses. Während des Lernens werden bekannte Wissensinhalte aus dem Langzeitgedächtnis abgerufen und es wird versucht, neue Inhalte mit den alten, bereits vorhandenen Inhalten zu verknüpfen oder diese zu ergänzen. Ziel der Theorie ist es, dargebotene Lernmaterialien so zu gestalten, dass die Ressourcen des Arbeitsgedächtnisses optimal genutzt werden können, um den Lernprozess zu unterstützen. Hierbei werden drei Quellen der Belastung des Arbeitsgedächtnisses unterschieden. Man unterscheidet den Intrinsic Cognitive Load, den Extraneous Cognitive Load sowie den Germane Cognitive Load. Die einzelnen Quellen werden als additiv angesehen, was bedeutet, dass man sie zusammen betrachten muss, um die vollständige Kapazität des Arbeitsgedächtnisses ausschöpfen zu können.

4.1.1. Intrinsic Cognitive Load

Der Intrinsic Cognitive Load (ICL) ist eine Größe, die von der Schwierigkeit, der Komplexität und dem Umfang des zu lernenden Stoffes abhängig ist. Die sogenannte Element-Interaktivität (element interactivity) spielt hierbei eine erhebliche Rolle, da sie die Schwierigkeit einer Aufgabe bestimmt. Die Element-Interaktivität gilt als hoch, wenn eine Aufgabe sehr komplex ist. Komplex bedeutet hier, dass zum Lösen bzw. zum Verstehen des Problems nicht nur das Problem an sich verstanden werden muss, sondern auch die Zusammenhänge und Ursachen, die zum Auftreten dieses Problem führen. Folgendes Beispiel soll dieses Phänomen verdeutlichen: „Wie bremsst ein ICE?“ Aus dieser einfach erscheinenden Frage entstehen viele neue Fragen: „Welche Art von Bremsen besitzt ein ICE? Wann werden diese eingesetzt? Wie funktionieren die verschiedenen Bremsarten?“ Bezogen auf den ICE3 könnte die Antwort lauten: „Ein ICE3 besitzt sowohl eine Druckluftbremse in Verbindung mit einer Scheibenbremse als auch eine Wirbelstrombremse. Die Wirbelstrombremse darf nur auf bestimmten Streckenabschnitten oder zur Notbremsung verwendet werden. Die Druckluftbremse ist in Deutschland vorgeschrieben und darf daher überall genutzt werden. Die Druckluftbremse funktioniert ähnlich wie die Bremse in einem Auto. Durch Druckluft werden die Bremsattel an die Bremscheiben gepresst und durch die Reibung wird der Zug abgebremst. Die Wirbelstrombremse hingegen nutzt Elektromagneten auf der Unterseite des Zuges dazu, dass in der Schiene ein Wirbelstrom erzeugt wird. Es baut sich also in der Schiene ein

¹Die folgenden Gedanken sind hauptsächlich aus dem NIEGEMANN et al.: *Kompendium multimediales Lernen*, Kapitel 3 übernommen, daher wird nur bei anderen Quellen ein Verweis erstellt.

Magnetfeld auf. Mithilfe der Lenzschen Regel bewirkt dieses Magnetfeld, dass der Zug gebremst wird.“

Eine niedrige Element-Interaktivität bzw. ein niedriger ICL ist demnach bei einem Aufgabentyp gegeben, bei dem ein Problem ohne Hintergrundwissen gelöst werden kann. So weiß beispielsweise fast jeder, wie man einen Fernseher bedient, wie aber der Fernseher an sich bzw. die Fernbedienung funktionieren, das wissen nur die wenigsten und dieses Wissen ist zur Bedienung nicht erforderlich. Die Bedienung des Fernsehers ist also eine Aufgabe mit niedriger Element-Interaktivität.

Aus diesen Darstellungen wird deutlich, dass der ICL vom Vorwissen des Lernenden abhängt. Je mehr Grundverständnis und Hintergrundwissen für das Problem zur Verfügung steht, desto leichter wird für ihn das Lösen einer neuen Aufgabe. Die für den ICL benötigte Kapazität wird von der Gesamtkapazität abgezogen, sodass bei zu hohem Schwierigkeitsgrad die Kapazität des Arbeitsgedächtnisses schnell erreicht ist. Daher ist es für einen Lehrer wichtig, das Vorwissen seiner Schüler richtig einschätzen zu können, um zu gewährleisten, dass die gestellten Aufgaben nicht zu schwer sind. Schafft es der Lehrer, beim Schüler neue Schemata aufzubauen, wird das Lernen von darauf aufbauendem Wissen erleichtert, da dann bei diesen Aufgaben der Intrinsic Cognitive Load geringer ist.

4.1.2. Extraneous Cognitive Load

Der Extraneous Cognitive Load (ECL) beschreibt die Gestaltung der Lernmaterialien. Eine hohe Belastung des ECL bedeutet, dass der Lernende mit Materialien viele irrelevante, wenig zielführende und ineffektive kognitive Anstrengungen vollführen muss, um wenig relevante Informationen zu bekommen. Demnach wird wenig Kapazität benötigt, wenn man direkt und ohne größere Probleme das relevante Wissen herausfiltern kann. Es beschreibt also jene Art von Belastung, die Lernende von der ursprünglichen Problemstellung ablenken oder fernhalten. Im Folgenden werden spezielle Effekte genannt, die den ECL verringern oder erhöhen können.

- Beispielorientiertes Lernen beschreibt, wie man mit Hilfe einer herausgearbeiteten Lösung oder einer vom Lehrer vorbereiteten Lösung weitere analoge Aufgaben zu lösen versucht. Durch die Vorgabe einer Lösung steht für den Schüler nicht mehr das Problemfinden, sondern das Problemlösen im Vordergrund. Hierdurch wird freie Gedächtniskapazität geschaffen und der ECL wird gering gehalten. Man nennt diesen Effekt, den *Worked Example Effect*². Ein Worked Example wäre zum Beispiel: *Berechne für die rechte Gleichung die zugehörigen x-Werte:*

²vgl. ROST, DETLEF H. (2006): *Handwörterbuch Pädagogische Psychologie*. Weinheim, Basel, Berlin: Beltz Verlag S. 117.

$$\begin{array}{rcl}
0 = & x^2 + 6x + 5 & 0 = \quad x^2 + 7x + 6 \\
x_{1,2} = & \frac{-6 \pm \sqrt{6^2 - 4 \cdot 1 \cdot 5}}{2 \cdot 1} & x_{1,2} = \quad \dots \\
x_{1,2} = & \frac{-6 \pm 4}{2} & x_{1,2} = \quad \dots \\
x_1 = -1 \quad \wedge & x_2 = -5 & x_1 = \dots \quad \wedge \quad x_2 = \dots
\end{array}$$

- Der *Split Attention Effekt* tritt nur dann auf, wenn verschiedene visuelle Informationsquellen (z. B. Bild und Text) nebeneinander dargeboten werden, aber zum Verstehen des Lerninhalts gleichzeitig benötigt werden. Der Lernende muss in diesem Fall seine Aufmerksamkeit auf mehrere Informationsquellen gleichzeitig richten. Seine Aufmerksamkeit springt von einem Objekt zum anderen. Die Teilung der Aufmerksamkeit bewirkt eine Steigerung des ECL, wodurch das Lernen nicht mehr so effektiv sein kann. Nach AYRES ET AL. liegt die Lösung dieses Problems darin, Text- und Bildinhalte direkt zu kombinieren. Der Text soll möglichst neben dem Bild platziert werden, das von ihm beschrieben wird. In diesem Fall kann es sogar möglich sein, dass ein positiver Effekt auftritt und besser gelernt werden kann. Ist dies aus Platz- oder Zeitgründen nicht möglich, ist es erforderlich, dass Informationen in kleinere Informationsblöcke unterteilt und erst dann gelernt werden³.
- Der *Modality Effect* tritt dann auf wenn sehr viele visuell präsentierte Wissensinhalte gleichzeitig verarbeitet werden müssen. Er kann durch eine Kombination von Bildmaterial und gesprochenem Text verhindert bzw. reduziert werden.⁴ Der Effekt bezieht sich auf die Theorie des Arbeitsgedächtnisses nach BADDELEY, der zufolge sowohl der räumlich-visuelle als auch der phonologische Kanal jeweils überlastet werden kann. Bei einer auftretenden Überlastung eines Kanals können die Informationen nicht an das Langzeitgedächtnis weitergeleitet werden. Der Effekt lässt sich vermeiden, indem die neuen Informationen mit schon ausreichend vorhandenem Vorwissen beim Lernenden verankert werden. Der Effekt lässt sich besonders gut dadurch verringern, dass die visuellen Daten phonologisch unterstützt und somit beide Kanäle angesprochen werden.⁵
- Werden Texte simultan auditiv und visuell dargeboten, kann es passieren, dass die Informationen über das hinausgehen, was der Lernende zum Lernen benötigt. Dann wirkt sich diese gleichzeitige Darstellung negativ auf den Lernerfolg aus. Dies ist der *Redundancy Effect*, er ist stark an die Vorkenntnisse des Lernenden gebunden. Informationen die für einen Experten überflüssig sind, werden von Laien unter Umständen benötigt, um den

³vgl. MAZARAKIS, ATHANASIOS (2007): *Cognitive Load Theory und der split-attention effect: Ein empirischer Test kognitionspsychologischer Erweiterungen*. Diplomarbeit Universität Mannheim, (URL: http://madoc.bib.uni-mannheim.de/madoc/volltexte/2007/1612/pdf/DA_Mazarakis3.pdf) S. 13ff.

⁴vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 618.

⁵vgl. NIEGEMANN et al.: *Kompodium multimediales Lernen*, S. 47f.

Inhalt lernen zu können.⁶

- Der *Expertise Reversal Effect* wird durch zu viel Vorwissen erreicht. Besitzt der Lernende noch kein oder nur wenig Vorwissen, so ist es für ihn wichtig, dass die zu lernenden Informationen didaktisch sinnvoll aufbereitet werden. Das heißt, der Lernende benötigt zu Grafiken eventuell auch einen Begleittext oder zusätzlich noch eine gesprochene Information zum leichteren Verständnis. Ist aber schon genügend Vorwissen vorhanden und die Grafik könnte auch ohne Text und gesprochene Nachricht verstanden werden, so kann der *Expertise Reversal Effect* auftreten. Dieser bewirkt dann, dass wegen der didaktischen Aufbereitung die Informationen schwerer zu verstehen sind, als wenn die Information an sich zu lernen wäre. Bei der Präsentation der Informationen ohne didaktische Aufbereitung kann der *Expertise Reversal Effect* vermieden werden.
- Eine Überlastung des Cognitive Load tritt in der Regel nur bei hoher Komplexität des Lernstoffes auf, wenn verschiedene Inhalte in Bezug zueinander stehen oder aufeinander aufbauen. Dies ist der *Element Interactivity Effect*.⁷ Sind die einzelnen Lerninhalte jeweils für sich verständlich, so tritt auch keine Überlastung ein.

Beim Einsatz von Medien im Unterricht muss der Lehrer darauf achten, die Informationen möglichst unter Beachtung der beschriebenen Zusammenhänge zu präsentieren. Ziel sollte es sein, den Extraneous Cognitive Load möglichst gering zu halten, um die Kapazitäten der Lernenden nicht unnötig zu belasten.

4.1.3. Germane Cognitive Load

Der Germane Cognitive Load (GCL) stellt die kognitive Belastung dar, welche zur reinen Wissensaufnahme benötigt wird. Man sollte also versuchen, den ICL und den ECL so gering wie möglich zu halten, um dem GCL mehr freie Kapazität des Arbeitsgedächtnisses zu bieten. Auf diese Weise kann das Arbeitsgedächtnis effizienter arbeiten. Damit Schüler effektiv lernen können, muss der GCL möglichst groß sein. Bei einer durch den Lehrer vorgegebenen Aufgabe kann dies nur geschehen, indem der Lehrer, den ECL so gering wie möglich hält (vgl. 4.1.2).

4.1.4. Didaktische Gedanken

Die Cognitive-Load-Theorie sollte bei der Unterrichtsplanung eine entscheidende Rolle spielen. Denn beachtet ein Lehrer die Kapazität des Arbeitsgedächtnisses nicht, werden negative Lerneffekte bei den Schülern auftreten. Um den Schülern ein effizientes Lernen zu ermöglichen, müssen die einzelnen Punkte der Cognitive-Load-Theorie berücksichtigt werden. Es ist außerdem entscheidend, einen guten Überblick über das Vorwissen der Schüler zu haben. Schließlich muss die Element-Interaktivität an dem Kenntnisstand der Schüler ausgerichtet sein, um das

⁶vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 618.

⁷vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 618.

Lernmaterial nicht zu komplex zu gestalten. Zwischenzeitige Lernzielkontrollen sind sinnvoll, um den Wissensstand zu überprüfen. Um den ECL möglichst gering zu halten, ist es wichtig, sich Effekte wie *Worked Examples* zu Nutze zu machen, um einen leichteren Aufbau von Schemata zu gewährleisten. Außerdem hilfreich ist der Einsatz von leicht zu verstehenden *bildlichen Darstellungen*, z. B. sind Vektorpfeile für Lernende leichter verständlich als Liniendiagramme. Einen Kernpunkt sollte der Schemaerwerb bilden. Schemata erleichtern den späteren Wissenserwerb sehr. Mithilfe der *Worked Examples* oder *bildlicher Darstellungen* lassen sich solche Schemata relativ leicht erlernen und im Verbund mit einer hohen Kapazität für den GCL gute Lernergebnisse erzielen.

4.2. Die kognitive Theorie multimedialen Lernens (CTML)

Die Theorie wurde von MAYER als SOI-Modell (Selektion-Organisation-Integration) entwickelt und will den Vorgang des Text-Bild-Lernens erklären. Hierbei wird die Differenz zwischen dem erwarteten und dem tatsächlichen Lernerfolg betrachtet und eine Begründung für diesen Unterschied gesucht. MAYER geht in seiner Theorie von drei zentralen Annahmen aus:⁸

- Es gibt zwei Informationskanäle, einen visuellen und einen phonologischen. Informationen werden je nach Präsentationsform verarbeitet und an das Langzeitgedächtnis weitergeleitet. Diese Annahme basiert auf der *Dual Coding Theory* von PAIVIO⁹ sowie auf der *Cognitive-Load-Theorie* (vgl. 4.1).
- Die Kapazität des Arbeitsgedächtnisses und somit jedes einzelnen Kanals ist begrenzt. Die Grundlage dieser Aussage ist das Modell des *Arbeitsgedächtnisses* (vgl. 2.2) nach BADDELEY sowie die *Cognitive-Load-Theorie* (vgl. 4.1).
- Lernen ist ein aktiver Verarbeitungsprozess. Der Lernende wählt aktiv Informationen aus dem dargebotenen Stoff aus, die für ihn bedeutsam sind und erarbeitet eine für ihn schließliche mentale Repräsentation und baut Verbindungen mit bereits vorhandenem Wissen auf.

Ein aktiver Verarbeitungsprozess ist wichtig, um sich ein konkretes Modell des Lerngegenstandes aufzubauen. Der Lerninhalt muss gefiltert werden, wichtige Informationen müssen herausgearbeitet, strukturiert und mit vorhandenem Wissen verknüpft und anschließend abgespeichert werden. COOK & MAYER stellten 1988 eine Liste dieser mentalen Prozesse auf, welche später von CAMBLISS & CALFEE (1998) erweitert wurde.¹⁰

- Der Aufbau von Ursache-Wirkung-Ketten
Bsp: „Fahren eines Autos“ – Um ein Auto fahren zu können, bedarf es der Kenntnis, dass zum Beschleunigen das Gaspedal, zum Bremsen das Bremspedal und zum Schalten

⁸vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 611.

⁹vgl. WEIDENMANN, BERND und KRAPP, ANDREAS (2001): *Pädagogische Psychologie*. 4. Auflage. Weinheim: Psychologie Verlags Union S. 429.

¹⁰vgl. NIEGEMANN et al.: *Kompodium multimediales Lernen*, S. 51.

die Kupplung betätigt werden müssen – Das Gaspedal bewirkt, dass mehr Benzin in den Motor eingespritzt wird – Funktionsweise eines Motors – usw.

- Der Vergleich von Informationen
Bsp: „Stromkreislauf – Wasserkreislauf“
- Generalisieren von Begriffen
Bsp: „Adler-Pinguin-Emu“ sind alles Vögel
- Aufzählen und Zusammenstellen von Sachverhalten
Bsp: Lernen von deutschen Flüssen und Mittelgebirgen
- Klassifizieren bzw. Einordnen von Informationen
Bsp: Grammatik einer Sprache verstehen und anwenden

Die drei Grundannahmen lassen sich in drei Unterpunkte unterteilen, Selektion, Organisation und Integration (SOI-Modell).

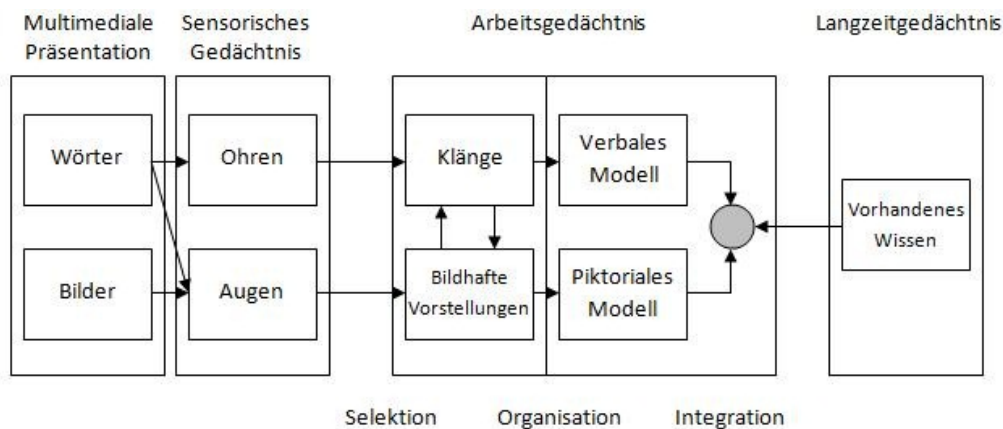


Abbildung 4.1.: Kognitive Theorie multimedialen Lernens (SOI-Modell) nach MAYER¹¹

Es existiert zum einen der Selektionsbegriff, mit dem verbunden ist, welche Wörter bzw. Bilder als bedeutend erachtet werden. Das sensorische Gedächtnis ist dafür zuständig, die relevanten von den irrelevanten Wörtern bzw. Bildern zu trennen und nur die wichtigen Informationen an das Arbeitsgedächtnis weiterzuleiten. Eine besondere Rolle nimmt hierbei der geschriebene Text ein, denn durch inneres Sprechen werden die optischen Impulse in auditive Impulse umgewandelt und somit durch den phonologischen Kanal an das Arbeitsgedächtnis weitergeleitet.

Zum anderen spielt der Organisationsbegriff eine Rolle, die Textinhalte werden in ein verbales Modell eingliedert und organisiert. Ebenso werden für wichtig erachtete Bildinhalte in einem verbalen Modell organisiert. Der Prozess des Organisierens findet im Arbeitsgedächtnis statt und stellt den Schritt von der Selektion zur Integration dar.

¹¹vgl. NIEGEMANN et al.: *Kompodium multimediales Lernen*, S. 53.

Schließlich werden im Integrationsprozess die neu organisierten Informationen mit dem im Langzeitgedächtnis bereits vorhandenem Wissen verknüpft oder, wenn noch kein Wissen vorhanden ist, als neues Wissen angelegt. Es ist wichtig, dass aus beiden Kanälen Informationen vorhanden sind, denn erst wenn sowohl visuelle als auch auditive Modelle der Information existieren, entsteht ein gemeinsames Modell, das im Langzeitgedächtnis abgespeichert wird und von dem Moment als Vorwissen benutzt werden kann.

MORENO erweiterte 2005 die Theorie um einen taktilen Sinneskanal, mithin einen Kanal, der das Fühlen beinhaltet. Allerdings ist dieser Kanal eher unbrauchbar für die Arbeit mit digitalen Medien und kann deshalb an dieser Stelle vernachlässigt werden.¹²

4.2.1. Didaktische Gedanken

Aus dem Modell lassen sich zwei wesentliche Konsequenzen in Bezug auf die Unterrichtsplanung mit Hilfe von Multimedia folgern. Zum einen muss darauf geachtet werden, dass nicht ein Kanal im Übermaß beansprucht wird und somit eine Überlastung des Arbeitsgedächtnisses auftritt. Zum anderen dürfen nicht zu viele Informationen über beide Kanäle gleichzeitig eintreffen, weil dann eine Überlastung des Arbeitsgedächtnisses stattfindet. Wenn nämlich zu viele Informationen gleichzeitig im Arbeitsgedächtnis vorübergehend gespeichert werden müssen, können neue Informationen nicht mehr aufgenommen werden.¹³

MAYER stellte hierzu eine Liste von Prinzipien auf, die beachtet werden müssen, um eine effiziente Arbeit mit multimedialen Inhalten zu gewährleisten.¹⁴

- *Das Multimediaprinzip:* Es besagt, dass insbesondere Lernende mit geringerem Vorwissen, aber nicht nur dann, effektiver lernen, wenn Text und Bild parallel dargeboten werden. Das Präsentieren von Grafiken aus dekorativen Gründen wirkt hingegen lernunwirksam.
- *Das räumliche Kontiguitätsprinzip:* Das Prinzip besagt, dass Text und zugehörige Bilder räumlich nahe beieinander stehen sollten. Ansonsten werden zu viele kognitive Ressourcen benötigt, um Text und Bild zu verknüpfen.
- *Das zeitliche Kontiguitätsprinzip:* Es resultieren bessere Lernergebnisse, wenn während einer Animation ein dazugehöriger gesprochener Text dargeboten wird. Wird die verbale Erklärung erst nach der Animation oder zu einem noch späteren Zeitpunkt angeboten, so ist der kognitive Aufwand wesentlich höher, was wiederum das Lernen erschwert.
- *Das Kohärenzprinzip:* Unwichtige Informationen sollten im Lernstoff gar nicht erst auftauchen. Es sollte nur gezeigt werden, was für das Verstehen und Lernen des Sachverhalts notwendig ist.

¹² vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 53f.

¹³ vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 53f.

¹⁴ vgl. KIRCHER, GIRWIDZ und HÄUSSLER: *Physikdidaktik - Theorie und Praxis*, S. 611 f. und NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 230 & S. 253 f.

- *Das Modalitätsprinzip:* Es werden größere Lernerfolge erzielt, wenn während einer Animation ein Text gesprochen wird, anstatt ihn zum Lesen anzubieten. Dieser Text kann auf zwei Arten dargeboten werden. Erstens kann der Text beschreibend gesprochen werden (z.B. „...Sie sehen...“) oder zweitens direktiv (z.B. „...achten Sie auf...“). Beide Formen zielen darauf ab, bestimmte Teile der Animation besonders zu beachten. Deshalb sind auch beide Formen gleichermaßen angebracht.
- *Das Redundanzprinzip:* Werden Animationen mit einer akustischen Erklärung hinterlegt, werden sie besser verstanden als bei einer Kombination aus Bild und Text. Sind mehrere Informationsquellen vorhanden, die den Sachverhalt komplett erklären, so kann dies sogar stören oder den Lernenden unnötig kognitive Ressourcen kosten. Es sollte also keine ergänzende Informationsquelle vorhanden sein, wenn bereits eine andere Quelle den Lernstoff komplett erklärt.
- *Das Vortrainingsprinzip:* Bevor ein neues Informationsangebot behandelt wird, sollte alles nötige Vorwissen bzw. der Umgang mit Geräten oder Systemkomponenten wiederholt werden, um keine kognitiven Ressourcen zu verschwenden.
- *Das Signalprinzip:* Das Signalprinzip besagt, dass der Lernende effektiver lernt, wenn Texte mit Hilfen versehen sind, die die Struktur oder die Ursache-Wirkungs-Zusammenhänge klären (z. B. Überschriften, Hervorhebungen, Signalwörter wie erstens, zweitens usw.).
- *Das Personalisationsprinzip:* Es wird ein größerer Lernerfolg erzielt, wenn der Lernende persönlich durch den Text angesprochen wird. Formale oder nüchterne Texte, z. B. in der dritten Person geschrieben sind, wirken sich eher negativ aus.

4.3. Integriertes Modell des Text- und Bildverstehens

Das Integrierte Modell des Text- und Bildverstehens (ITCP = Integrated Model of Text and Picture Comprehension) ist eine Weiterentwicklung des Modells vom Text- und Bildverständnis, welches SCHNOTZ & BANNERT 2003 vorgestellt haben. Es basiert auf der *Dual Coding Theory* von PAIVIO¹⁵, einer Begrenzung des Arbeitsgedächtnisses im Sinne der *Cognitiv-Load-Theory* (vgl. 4.1), der *kognitiven Theorie multimedialen Lernens* (vgl. 4.2), dem Mehrspeichermodell des Gedächtnisses nach ATKINSON & SHIFFRIN (vgl. 2) sowie den mentalen Repräsentationen beim Text- und Bildverstehen nach KOSSLYN und dem Text- und Bildverständnismodell nach SCHNOTZ & BANNERT. Auf die beiden letztgenannten Modelle wird im Folgenden näher eingegangen.

¹⁵vgl. WEIDENMANN und KRAPP: *Pädagogische Psychologie*, S. 429.

4.3.1. Mentale Repräsentationen

Mentale Repräsentationen sind Modelle von Bildern und/oder Texten im Arbeitsgedächtnis, die solange verändert und mit Wissen aus dem Langzeitgedächtnis ergänzt werden, bis ein komplexes Modell des Lerninhaltes entstanden ist. Dieses wird schließlich im Langzeitgedächtnis abgespeichert. Diese Repräsentationen sind anfangs sehr grob. Der Lernende strukturiert zunächst die Oberflächenmerkmale, wie z.B. Form, Farbe, Geruch oder Klang einer Information. Später werden dann die neu aufgenommenen Informationen mit schon vorhandenem Wissen verglichen und Gemeinsamkeiten sowie Unterschiede abgespeichert und entsprechend zugeordnet. So werden z. B. die Informationen über Kupfer mit anderen bekannten Elementen wie Eisen verglichen. Beide Elemente sind gute Wärmeleiter, glänzen und leiten elektrischen Strom, deshalb sind es beides Metalle. Sie unterscheiden sich unter anderem darin, dass Eisen ferromagnetisch und Kupfer diamagnetisch ist, sowie Eisen silbern und Kupfer rotbraun glänzt. Am Ende dieses Vergleiches besitzt der Lernende eine genaue Vorstellung über Kupfer; er hat sich ein mentales Modell gebildet.

4.3.2. Das Text- und Bildverständnismodell

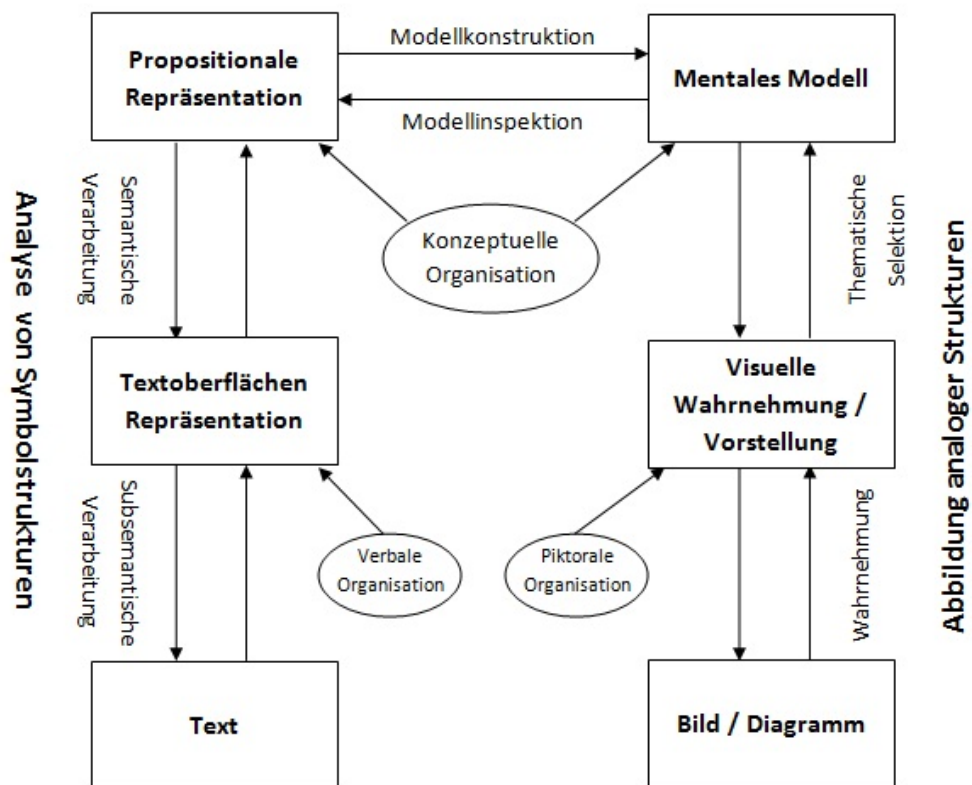


Abbildung 4.2.: Modell des Text- und Bildverstehens nach SCHNOTZ & BANNERT ¹⁶

¹⁶vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 57.

Das Text- und Bildverständnismodell baut auf der dualen Codierungstheorie sowie auf der Vorstellung der mentalen Repräsentation auf.¹⁷ Man unterscheidet zwischen einer beschreibenden Repräsentation (deskriptional), in der abstraktes Wissen repräsentiert wird, wie in dem obigen Beispiel die Art der Magnetisierung, und der darstellenden Repräsentation (depiktional), die einen konkreten Sachverhalt oder eine konkrete Sache beschreibt, so in Bezug auf das genannte Beispiel, Kupfer glänzt rotbraun. Zuerst betrachtet der Lernende die Oberflächenmerkmale und bekommt so eine erste Vorstellung von dem Inhalt. Bei einer genaueren Betrachtung der Text- oder Bildinhalte werden dann Verknüpfungen erstellt. Je nachdem, ob die Informationen visuell oder bildlich aufgenommen wurden, werden die Inhalte kanalgebunden abgespeichert. Ist ein mentales Modell komplett fertiggestellt, so besteht es sowohl aus Text- als auch aus Bildinhalten.¹⁸

4.3.3. Das Integrierte Modell des Text- und Bildverstehens

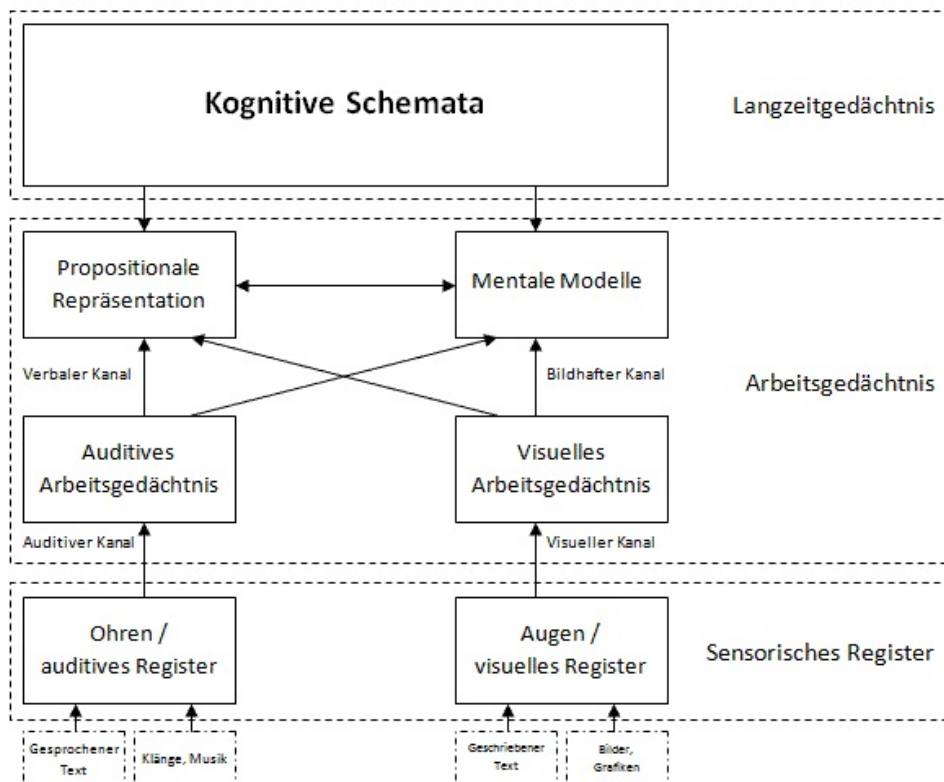


Abbildung 4.3.: Integriertes Modell des Text- und Bildverstehens nach SCHNOTZ ¹⁹

Das ITPC ermöglicht es, sowohl das Text- und Bildverständnis als auch eine Kombination aus

¹⁷vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 56.

¹⁸vgl. SCHNOTZ, WOLFGANG und BANNERT, MARIA (1999): *Einflüsse der Visualisierungsform auf die Konstruktion mentaler Modelle beim Text- und Bildverstehen*. *Zeitschrift für Experimentelle Psychologie*, Vol. 46, Nr. 3

¹⁹vgl. NIEGEMANN et al.: *Kompendium multimediales Lernen*, S. 58.

Musik bzw. Geräuschen mit Texten oder Bildern zu erklären. Um einen Text verstehen zu können, wird nach dem ITPC ein visueller Reiz in Form von Schrift mit dem Auge aufgenommen und über den visuellen Kanal in das visuelle Arbeitsgedächtnis weitergeleitet. Der Text wird durch das Auge oberflächlich wahrgenommen. Durch inneres Sprechen werden die Oberflächenmerkmale an den phonologischen Teil des Arbeitsgedächtnisses weitergeleitet und vertieft. Dieser Prozess der inneren Repräsentation wird vollendet, indem die gelernten Informationen mit themenbezogenen Schemata aus dem Langzeitgedächtnis verbunden werden. Auf diese Weise entsteht das fertige Modell.

Gesprochene Texte, Musik oder Geräusche werden ähnlich aufgenommen. Hier entscheidet zuerst das sensorische Gedächtnis, ob der gehörte Text wichtig genug erscheint, um in das Arbeitsgedächtnis weitergeleitet zu werden. Die Informationen gelangen über den phonologischen Kanal ins Arbeitsgedächtnis, wo ein Modell erstellt wird, das nur Oberflächenmerkmale besitzt. Nachdem dieses Modell vorhanden ist, wird mit Hilfe eines auditiven Filters herausgearbeitet, welche Informationen wichtig sind und im Langzeitgedächtnis gespeichert werden sollen. Am Ende der durch einen gesprochenen Text veranlassten Modellbildung steht die Verknüpfung mit bekannten Schemata und Bildern, wodurch das Modell auch bildhafte Informationen erhält und vervollständigt wird.

Bei bildhaften Informationen verläuft es analog. Der Selektionsprozess findet wieder im sensorischen Gedächtnis statt, bevor die Informationen über den visuellen Kanal an das visuelle Arbeitsgedächtnis weitergeleitet werden. Hier werden die bildhaften Informationen mit Hilfe eines visuellen Filters genauer betrachtet und die wichtigsten Informationen vertieft. Bei Bildern wird ein Modell durch Schemata und Vorwissen aus dem Langzeitgedächtnis in Form von Texten ergänzt.²⁰

4.3.4. Didaktische Empfehlungen

SCHNOTZ empfiehlt, zusätzlich zu den instruktionellen Konsequenzen der Cognitive Load Theorie und der kognitiven Theorie multimedialen Lernens noch einige weitere Punkte zu beachten.²¹

- *Präsentationssequenz-Prinzip*: Ist es dem Lehrenden nicht möglich, bildhafte und Textinformationen gleichzeitig zu präsentieren, so ist es besser, erst das Bild zu zeigen. Indem zuerst das Bild in den visuellen Speicher gelangt und ein oberflächliches Modell erstellt werden kann, wird versucht, eine Überlastung der kognitiven Kapazität zu verhindern. Die Textinformation würde den visuellen Speicher nutzen, um den Text in den auditiven Speicher zu überführen. Indem zuerst das Bild demonstriert wird, dient der Text als Ergänzung zu den bildlichen Informationen und bewirkt ein leichteres Erstellen eines mentalen Modells.

²⁰ vgl. NIEGEMANN et al.: *Kompodium multimediales Lernen*, S. 57ff.

²¹ vgl. SCHNOTZ, WOLFGANG (2005): *An integrated Model of Text and Picture Comprehension*. In MAYER, RICHARD E. (Hrsg.): *Cambridge Handbook of Multimedia Learning*. Cambridge: Cambridge University Press S. 65.

- *Strukturabbildungs-Prinzip*: Hat der Lehrende mehrere Bilder zur Auswahl, welche er dem Lernenden für den gleichen Lerninhalt zeigen könnte, so impliziert das Strukturabbildungs-Prinzip, dass nur ein Bild gezeigt werden sollte und zwar jenes, welches den zu lernenden Inhalt am adäquatesten repräsentiert.
- *Prozesskontroll-Prinzip*: Bei der Aufnahme von besonders schwierigen Texten mit Grafiken, soll der Text als Ergänzung zur Grafik dienen, wenn keine zeitliche Begrenzung vorgegeben ist. Dadurch entwickelt sich zuerst das mentale Modell des Bildes und anschließend kann der Text separat gelesen werden, um dieses Modell zu ergänzen. Eine gleichzeitige Darstellung bewirkt, dass der Text zusätzlich zu den Bildinformationen in Sprache umgewandelt werden müsste. Die auditiven Informationen müssten währenddessen in der phonologischen Schleife gehalten werden, bis der gesamte Inhalt gelernt ist. Dies hätte zur Folge, dass die Kapazität des Arbeitsgedächtnisses stark beansprucht werden würde.

5. Dynamische Geometrie Systeme

5.1. Allgemeines

Unter dem Begriff „Dynamische Geometrie Systeme“ (DGS) werden Programme verstanden, mit denen man geometrische Konstruktionen am Computer durchführen kann. Mit der Entwicklung von DGS wurde Ende der 1980er begonnen, z. B. das wahrscheinlich erste DGS „Cabri Géométrie“ wurde 1988 veröffentlicht.¹ Die Entwicklung hat sich seitdem ständig fortgesetzt. In den ersten Programmen wurden einfache Konstruktionen mit virtuellem Zirkel und Lineal erstellt. Man hat versucht, die komplette euklidische Geometrie durch DGS darzustellen.² Cinderella 2.0, als ein Beispiel für aktuelle Programme, unterstützt zusätzlich noch die Darstellung sphärischer und hyperbolischer Geometrie.³ Andere Programme, wie Geogebra, bieten zusätzlich zum Geometriemodus noch einen Algebramodus und stellen so eine Verknüpfung aus DGS und Computer Algebra Software (CAS) her.⁴ Außerdem besitzen die meisten neueren DGS-Programme die Möglichkeit, java-basierte Applets zu generieren und diese auf einer Internetseite einzubinden.

Unter geometrischen Konstruktionen versteht man in DGS Konstruktionen, die mit Hilfe von Zirkel und Lineal auch per Hand gezeichnet werden könnten. Dynamisch meint in diesem Zusammenhang, dass eine Konstruktion auch nach dem Zeichnen noch verändert werden kann. Diese Möglichkeit des Veränderns nennt sich Zugmodus.⁵ Jedes aktuelle DGS beinhaltet diese Eigenschaft. Denn dadurch, dass man Punkte verschieben kann, bieten sich viele Möglichkeiten zum Experimentieren. Konstruiert man abhängige Punkte, dann ändert sich beim bewegen des ursprünglichen Punktes die gesamte Konstruktion bzw. alle abhängigen Punkte, Strecken, etc.. Hiermit lässt sich z. B. sehr leicht prüfen, wie sich bestimmte Konstruktionen zu verschiedenen Gegebenheiten verändern, ob es Grenzbereiche oder gar Fälle gibt, bei denen die ursprüngliche Konstruktion nicht mehr vorhanden ist. Als Beispiel wird angenommen, man konstruiert ein

¹vgl. KOEPEL, ANDREAS und TÖNNIES, DIRK (2007): *Dynamische Geometrie im Mathematikunterricht der Sekundarstufe 1*. Köln: Aulis Verlag Deubner S. 11.

²vgl. ELSCHENBROICH, HANS-JÜRGEN und GAWLICK, THOMAS und HENN, HANS-WOLFGANG und HEINTZ, GABY und RICHTER-GEBERT, JÜRGEN (2001): *Dynamische Geometrie-Software: Stand der Forschung und Perspektiven*. In ELSCHENBROICH, HANS-JÜRGEN und GAWLICK, THOMAS und HENN, HANS-WOLFGANG (Hrsg.): *Zeichnung - Figur - Zugfigur*. Hildesheim Berlin: Verlag franzbecker S. 15.

³vgl. KORTENKAMP, ULRICH H. und RICHTER-GEBERT, JÜRGEN (2006): *Cinderella.2 Math in Motion*. 2006 (URL: <http://cinderella.de/files/flyer-web-de.pdf>).

⁴vgl. HOHENWARTER, MARKUS: *Was ist GeoGebra?* (URL: http://www.geogebra.org/cms/index.php?option=com_content&task=blogcategory&id=67&Itemid=63).

⁵vgl. LUDWIG, MATTHIAS und WEIGAND, HANS-GEORG (2009): *Konstruieren*. In WEIGAND, HANS-GEORG und FILLER, ANDREAS und SEBASTIAN KUNTZE, REINHARD HÖLZL ANS und LUDWIG, MATTHIAS und ROTH, JÜRGEN und SCHMIDT-THIEME, BARBARA und WITTMANN, GERALD (Hrsg.): *Didaktik der Geometrie für die Sekundarstufe I*. Heidelberg: Spektrum Akademischer Verlag S. 62.

Dreieck und verschiebt einen Punkt. Im Grenzfall entsteht eine Gerade. Es ist also nicht möglich ein Dreieck aus drei beliebigen Punkten zu konstruieren, nämlich dann nicht, wenn die Punkte auf einer Geraden liegen. Andere häufig vorkommende Operationen in DGS sind der Spurmodus und sogenannte Makros. Der Spurmodus oder „*die Ortslinienfunktion visualisiert die Bewegung von konstruierten Punkten beim Variieren von Ausgangsobjekten und ist ein heuristisches Hilfsmittel im Rahmen von Problemlöseprozessen.*“⁶ Ein Makro hingegen ist eine Zusammenfassung von Konstruktionsschritten, die in dem DGS nicht schon vorinstalliert ist. Zum Beispiel könnte man sich ein Makro erstellen, indem man nur mit Hilfe von virtuellem Zirkel und Lineal eine Parallele erstellt. Das Programm würde das Makro „Parallele“ abspeichern und mit Hilfe des Makros könnte dann eine Parallele zu einer Geraden mit nur einem Mausklick auf das Makro erstellt werden. DGS bieten aber noch wesentlich mehr Möglichkeiten. So werden je nach Programm immer mehr Mathematik unabhängige Funktionen eingebaut (z.B. der Physikmodus von Cinderella 2.0 vgl. 5.4.1) oder Konstruktionen werden fächerübergreifend genutzt, um Inhalte zu verdeutlichen (z.B. <http://www-m10.ma.tum.de/bin/view/MatheVital/WebHome>).

Der Einsatz von DGS im Unterricht ist jedem Lehrer selbst überlassen. Im Lehrplan sowie in den Bildungsstandards werden nur Inhalte und Kompetenzen angegeben, die die Schüler erlernen sollen.^{7,8} Im Physikunterricht dient die Simulation oft als Methode, um wichtige Inhalte zu elementarisieren bzw. zu verdeutlichen. Jeder Lehrer besitzt in seinem Unterricht Methodenfreiheit, d. h. er darf frei wählen, wie er Inhalte den Schülern präsentiert. Dies ist der Grund, warum Simulationen oder Computer im Allgemeinen fast gar nicht im Lehrplan bzw. den Bildungsstandards auftauchen. In Teil I des Lehrplans „*Das Gymnasium in Bayern*“ wird aber betont, dass es „*von großer Bedeutung ist, dass die Schulen im Hinblick auf methodische Fertigkeiten, die in verschiedenen Fächern zum Einsatz kommen, Konzepte für den Erwerb und die arbeitsteilige Vertiefung dieser Fertigkeiten entwerfen und umsetzen. Zu nennen sind insbesondere die Arbeitsfelder ‘Recherche, Analyse und Aufbereitung von Informationen’, ‘Einsatz moderner Medien und Informationstechnologien’ sowie ‘Präsentation und Dokumentation.’*“⁹ Für DGS bedeutet dies, dass sie in den Unterricht miteinbezogen werden sollen. Dabei muss den Schülern klar sein, dass ein Einsatz des Computers nicht eine bloße „Spielerei“ ist, sondern ein anderer Weg, Inhalte zu lernen und bestimmte Gegenstände oder Prozesse darzustellen.¹⁰ Im Folgenden sollen die Programme Geogebra, EUKLID DynaGeo – und Cinderella 2.0 genauer betrachtet werden.

⁶LUDWIG und WEIGAND: *Konstruieren*, S. 76.

⁷vgl. STAATSNSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrpläne Gymnasium G8*. (URL: <http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=1>).

⁸vgl. LANDESINSTITUT FÜR SCHULENTWICKLUNG - LANDESBILDUNGSSEKRETARIAT BADEN-WÜRTTEMBERG: *Bildungsstandards für Physik: Gymnasium - Klassen 6, 8, 10, Kursstufe*. (URL: http://www.bildung-staerkt-menschen.de/service/downloads/Bildungsstandards/Gym/Gym_Ph_bs.pdf).

⁹STAATSNSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrpläne Gymnasium G8*, 1.3 (4).

¹⁰vgl. ROTH, JÜRGEN (2008): *Dynamik von DGS – Wozu und wie sollte man sie nutzen?* In KORTENKAMP, ULRICH und WEIGAND, HANS-GEORG und WETH, THOMAS (Hrsg.): *Informatische Ideen im Mathematikunterricht. Bericht über die 23. Arbeitstagung des Arbeitskreises „Mathematikunterricht und Informatik“ in der Gesellschaft für Didaktik der Mathematik e. V. vom 23. bis 25. September 2005 in Dillingen an der Donau*. Hildesheim: Verlag Franzbecker S. 134.

5.2. Euklid DynaGeo

Nachdem mit Cabri Géométrie das erste auf DOS basierende DGS vorgestellt worden war, entschied sich Roland Mechling 1994 mit EUKLID ein unter Windows nutzbares DGS zu entwickeln. Der Zusatz DynaGeo kam im Laufe der Weiterentwicklung hinzu. EUKLID sollte vom Umfang und den Funktionen her mit Cabri Géométrie ebenbürtig sein.¹¹ Die aktuelle Version 3.5b besitzt alle wichtigen Eigenschaften von modernen DGS. Ähnlich wie in Geogebra bzw. Cinderella 2.0 gibt es eine Möglichkeit, Funktionen, Sehnen, Tangenten und Integralfächen zeichnen zu lassen. Mit dem „DynaGeoJ“-Viewer wurde dem Programm ein Java-basiertes Werkzeug zur Verfügung gestellt, mit dem plattformunabhängig dynamische Konstruktionen in Internetseiten integriert werden können. EUKLID DynaGeo ermöglicht es dem Benutzer mit einem Animationswerkzeug selbstdefinierte Bewegungen ablaufen zu lassen. Dieses Werkzeug ermöglicht es, Konstruktionen zu erstellen, die man im Physikunterricht nutzen kann. Einige Beispiele, wie EUKLID DynaGeo physikalische Sachverhalte darstellen kann, findet man unter <http://www.dynageo.de/discus/messages/5/424.html?1150234675>. Diese Physikanimationen werden ausgeführt, indem eine Art Schieberegler automatisch bewegt wird. Animationen

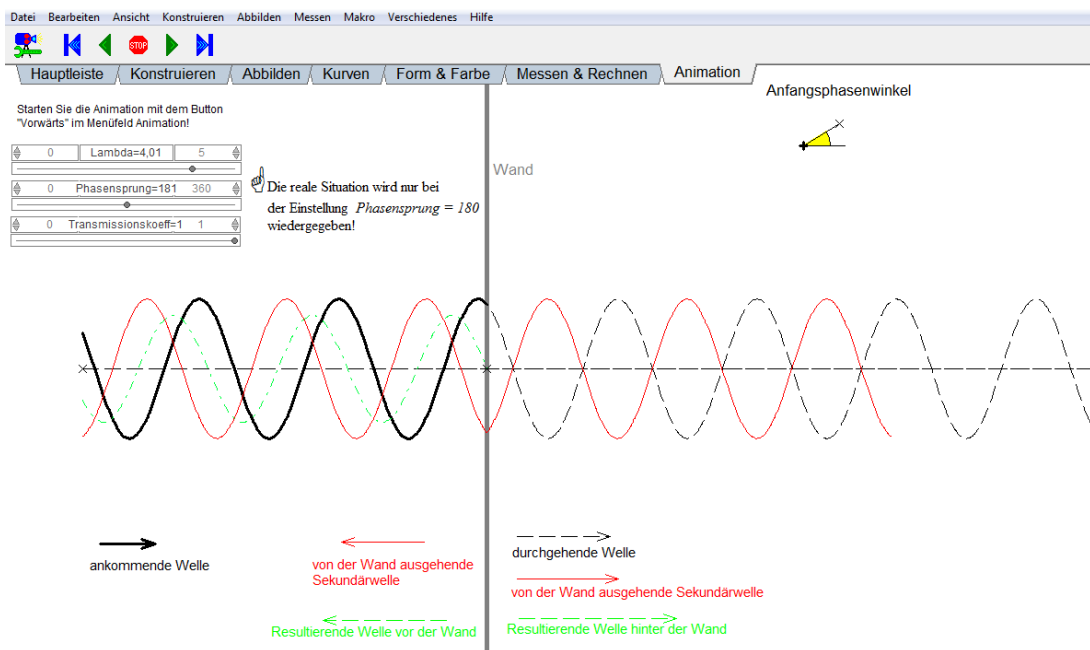


Abbildung 5.1.: Überlagerung von Wellen mit EUKLID DynaGeo¹²

mit physikalischem Inhalt sind also mit EUKLID DynaGeo durchaus möglich, wirken aber sehr statisch. Insgesamt ist EUKLID DynaGeo als DGS konzipiert worden und als dieses auch im Mathematikunterricht gut einzusetzen.

¹¹vgl. MECHLING, ROLAND: *EUKLID DynaGeo von innen*. (URL: <http://www.dynageo.de/scripte/discus/discus.pl>).

¹²Diese Simulation wurde von Stefan Rauscher auf <http://www.dynageo.de/discus/messages/5/424.html?1150234675> zur Verfügung gestellt.

5.3. Geogebra

Seit dem Erscheinen von Geogebra im Jahr 2002 wurde es fast jährlich mit zahlreichen nationalen und internationalen Preisen geehrt und scheint eines der vielversprechenden Projekte für den Computereinsatz im Schulalltag zu sein.¹³ Es wurde von Markus Hohenwarter im Rahmen seiner Diplomarbeit 2001/2002 an der Universität Salzburg entwickelt¹⁴ und ist ein javabasiertes plattformunabhängiges Programm, d. h. es kann auf jedem Betriebssystem genutzt und durch ein Java-Applet in alle aktuellen Internetbrowser integriert werden. Das Programm wird öffentlich zum Download bereitgestellt und ist uneingeschränkt kostenlos nutzbar.

Wie der Name andeutet, ist das Programm für den Einsatz in der Geometrie und der Algebra geeignet, kann darüber hinaus aber auch noch für die Analysis verwendet werden. Die enge Verknüpfung von Algebra und Geometrie zeichnet Geogebra aus, „*ein Ausdruck im Algebrafenster entspricht einem Objekt im Grafikfenster und umgekehrt.*“¹⁵ Für den Einsatzzweck von Geogebra bedeutet dies, dass man klassische Konstruktionen, wie Strecken, Geraden oder Kegelschnitte, erstellen, aber auch mit Vektoren und Matrizen arbeiten kann. Zudem kann es Funktionen differenzieren und als Funktionsplotter genutzt werden. Mit der Version 3.2 ist sogar ein eigener Tabellenkalkulationsmodus integriert worden.

Physikalische Animationen oder sogar Simulationen mit Geogebra zu erstellen, ist nicht so einfach wie mit Cinderella 2.0. Mit Geogebra erzeugte Physikanimationen müssen mathematisch genau definiert sein und sind nur bedingt dynamisch steuerbar. Animationen werden dadurch erzeugt, dass ein Schieberegler aus der Konstruktion in einem festgelegten Zeitintervall von einem Extremwert zum anderen abläuft.

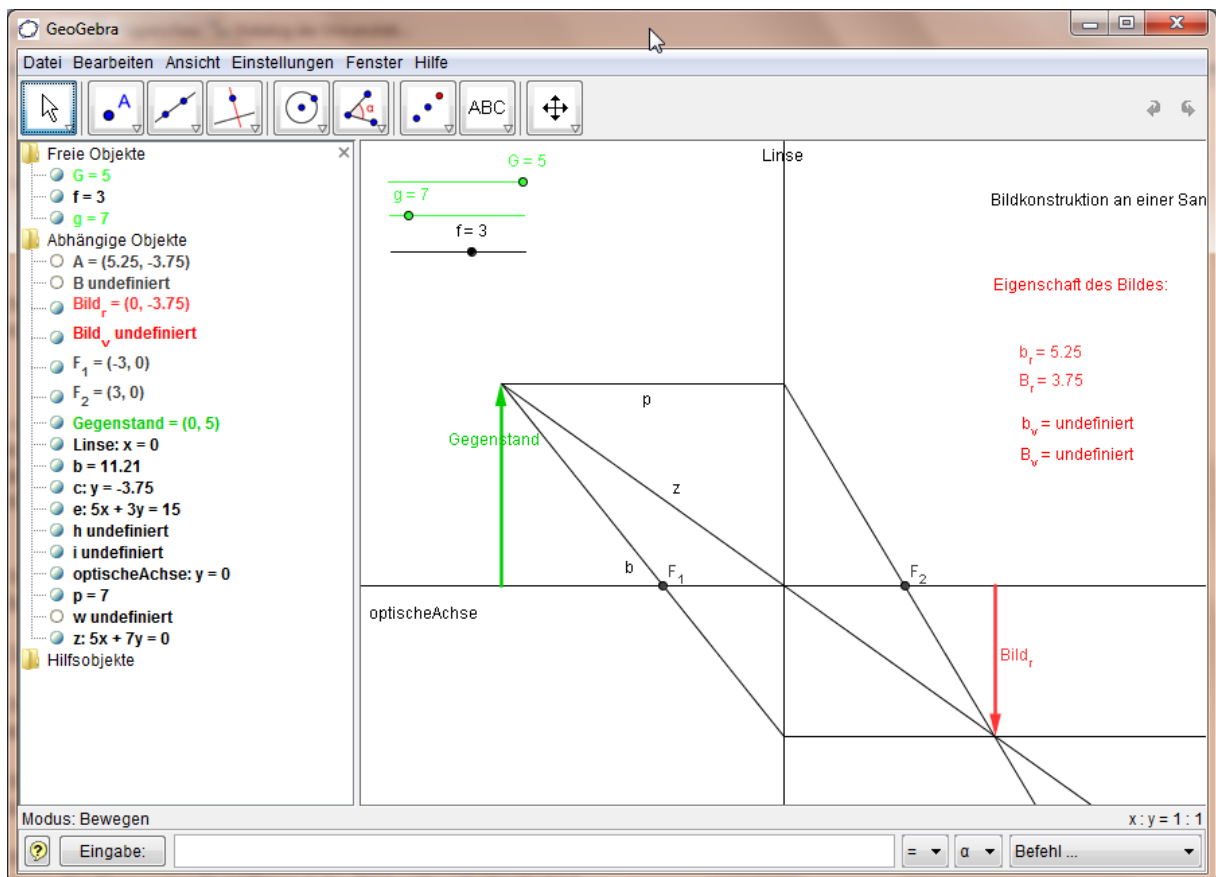
Auf <http://www.geogebra.org/de/wiki/index.php/Unterrichtsmaterialien> findet man eine große Anzahl von Unterrichtsmaterialien für das Fach Physik. Allerdings sind ein Großteil dieser Materialien keine Animationen, sondern bereits konstruierte Versuchsaufbauten. In diesen können die Schüler mit Schiebereglern den Versuchsaufbau verändern und bekommen dadurch andere Versuchsergebnisse. Beispielsweise werden Konstruktionen aus der Strahlenoptik betrachtet. Die Schüler müssen den Strahlengang durch eine Sammellinse nicht jedes Mal zeichnen, sondern können an den Schiebereglern die Einstellungen für einen anderen Versuchsaufbau vornehmen. Dadurch kann Unterrichtszeit gespart werden.

Simulationen, wie man sie in Cinderella 2.0 erstellen kann, sucht man in den Materialien für Geogebra vergebens. Dies liegt daran, dass Geogebra keinen internen Algorithmus zum Berechnen von dynamischen Veränderungen während einer Animation hat. Bewegte Objekte können sich also nur auf vordefinierten Bahnen bewegen und können deshalb nicht als dynamisch betrachtet werden.

¹³vgl. HOHENWARTER: *Was ist GeoGebra?*, http://www.geogebra.org/cms/index.php?option=com_content&task=blogcategory&id=67&Itemid=63.

¹⁴vgl. HOHENWARTER, MARKUS: *Geogebra*. (URL: <http://www.geogebra.org/de/wiki/index.php/GeoGebra>).

¹⁵vgl. HOHENWARTER, MARKUS: *GeoGebra - dynamische Mathematik für die Schule*. (URL: http://www.geogebra.org/help/geogebra_flyer_de.pdf).

Abbildung 5.2.: Beispiel einer Konstruktion mit Geogebra¹⁶

Abschließend soll aber erwähnt werden, dass Geogebra als DGS konzipiert wurde und es im Gegensatz zu Cinderella 2.0 nicht vorgesehen ist, mit dem Programm Physiksimulationen oder Animationen zu erstellen.

¹⁶vgl. HECKER, ANDREAS: *Konstruktion an Sammellinsen*. (URL: <http://physik.psi-online.de/geogebra/Sammellinsen.html>).

5.4. Cinderella 2.0

Das Projekt Cinderella entstand bei einer Kombinatorik-Konferenz in Schweden. Während einer Bootsfahrt, auf einem Schiff namens Cinderella, wurde von Henry CRAPO und Jürgen RICHTER-GEBERT der Entschluss gefasst, „[...] eine Software zu schreiben, mit deren Hilfe man mit wenigen Mausclicks geometrische Konstruktionen eingeben kann.“¹⁷

Die erste Version entstand 1993 auf einem NeXT-Rechner¹⁸. Sie baute auf projektiver Geometrie und der Invariantentheorie auf. Da die NeXT-Plattform an Popularität verlor, wurde 1995 eine zweite Version auf der plattformunabhängigen Sprache Java beschlossen. Die Möglichkeiten von Cinderella sollten um euklidische und nicht-euklidische Geometrie sowie Funktionen für Ortskurven erweitert werden. Des Weiteren sollte die Möglichkeit bestehen mit Cinderella erstellte Konstruktionen in einem Webbrowser zu integrieren, um online Aufgaben stellen zu können, die mit Hilfe eines programmeeigenen Beweisers automatisch überprüft werden sollten. 1997 wurden Jürgen RICHTER-GEBERT und Ulrich KORTENKAMP nach Zürich berufen, wo 1998 eine weitere Version entstand. Man entschloss sich, die dynamischen Berechnungen komplett in der komplexen Zahlenebene durchzuführen, um Fehler zu beseitigen, die während der dynamischen Bewegung einiger Konstruktionen entstanden sind.¹⁹ Seitdem wurde Cinderella mit etlichen Preisen ausgezeichnet, unter anderem mit dem European Academic Software Award 2000, sowie dem deutschen Bildungssoftwarepreis 2001. Cinderella 2.0 ist die neueste Version, sie entstand 2006. Speziell für den Physikunterricht wird Cinderella dadurch interessant, dass es seit der Version 2.0 einen speziellen Simulationsmodus CindyLab (siehe 5.4.1) gibt, in dem es möglich ist, frei mit Massen, Federn, Ladungen und Feldern zu experimentieren. Durch die dynamischen Eigenschaften von Cinderella ist es möglich, jederzeit in die Simulation einzugreifen und diese zu verändern.

Ebenso neu ab der Version 2.0 ist die integrierte Scriptsprache CindyScript (siehe 5.4.2). Sie unterstützt den Geometriemodus und CindyLab insofern, als dass sich durch sie eigene Befehle in Programmform erstellen lassen und diese in Simulationen eingebettet werden können. Hierdurch können bspw. Feldlinien wie in 6.5 sichtbar gemacht werden.²⁰

¹⁷RICHTER-GEBERT, JÜRGEN und KORTENKAMP, ULRICH H. (2001): *Benutzerhandbuch für die interaktive Geometrie-Software Cinderella*. Berlin, Heidelberg, New York: Springer-Verlag S. 1.

¹⁸vgl. <http://de.wikipedia.org/wiki/NeXT>.

¹⁹vgl. RICHTER-GEBERT und KORTENKAMP: *Benutzerhandbuch für die interaktive Geometrie-Software Cinderella*, S. 1f.

²⁰vgl. RICHTER-GEBERT, JÜRGEN und KORTENKAMP, ULRICH H.: *Cinderella.2 Documentation*. (URL: <http://doc.cinderella.de/tiki-index.php>).

5.4.1. CindyLab - Der Physikmodus von Cinderella 2.0²¹

Grundlagen

Die Grundlage des Physikmodus in Cinderella 2.0 sind die Newtonschen Axiome. Wenn man alle Kräfte kennt, die auf einen Körper wirken, sowie dessen Masse, Ort und Geschwindigkeit zu einem Zeitpunkt, kann man das weitere Verhalten des Körpers berechnen. Aus den Kräften und der Masse erhält man die Beschleunigung und damit die Geschwindigkeitsänderung. Zum besseren Verständnis werden alle wichtigen Gesetze noch einmal aufgeführt:^{22,23}

- 1. Newtonsches Axiom: Jeder Körper verharrt im Zustand der Ruhe oder der gleichförmigen geradlinigen Bewegung, solange keine Kraft auf ihn wirkt.
- 2. Newtonsches Axiom: Kraft ist als Impulsänderung definiert:

$$\vec{F} = \frac{d\vec{p}}{dt}$$

- 3. Newtonsches Axiom: Bei zwei Körpern, die nur miteinander, aber nicht mit anderen Körpern wechselwirken, ist die Kraft \vec{F}_1 auf den einen Körper entgegengesetzt gleich der Kraft \vec{F}_2 auf den anderen Körper. Newton formulierte dieses Gesetz als

$$actio = reactio$$

$$\vec{F}_1 = -\vec{F}_2$$

- Newtonsches Gravitationsgesetz:

$$\vec{F}(\vec{r}) = -G \cdot \frac{m_1 \cdot m_2}{r^2} \vec{e}_r$$

- Coulombsches Kraftgesetz:

$$\vec{F}(\vec{r}) = \frac{1}{4\pi\epsilon_0} \frac{Q_1 \cdot Q_2}{r^2} \vec{e}_r$$

Cinderella 2.0 nutzt bei diesen komplizierten Differentialgleichungen ein adaptives Verfahren zur numerischen Lösung der Anfangswertprobleme. Angewandt wird der ode-45-Algorithmus (auch: Dormand-Prince-Algorithmus) aus der Familie der Runge-Kutta-Verfahren. Bei solchen adaptiven Verfahren wird die Schrittweite vom Computer so gewählt, dass eine vorgegebene Genauigkeit der Berechnung erzielt wird. Dies bedeutet, dass für die Berechnung einer Simulation

²¹Aus RICHTER-GEBERT und KORTENKAMP: *Cinderella.2 Documentation*, <http://doc.cinderella.de/tiki-index.php?page=CindyLab> sowie auf eigenen Erfahrungen basierend.

²²vgl. DEMTRÖDER, WOLFGANG (2003): *Experimentalphysik 1: Mechanik und Wärme*. Berlin, Heidelberg, New York: Springer-Verlag S. 51 f.

²³vgl. DEMTRÖDER, WOLFGANG (2006): *Experimentalphysik 2: Elektrizität und Optik*. Berlin, Heidelberg, New York: Springer-Verlag S. 3.

der Rechenaufwand insbesondere bei Modellen mit langsamer Dynamik gering gehalten wird und das Ergebnis trotzdem genau ist.

Im weiteren Text wird im Zusammenhang mit Cinderella 2.0 ausschließlich von Simulationen gesprochen.

Die Elemente von CindyLab

Das Eigenschaftfenster eines beliebigen Elements in Cinderella lässt sich auf verschiedene Weise öffnen:

1. durch die *Tastenkombination Strg+I*
2. durch *Rechtsklick* auf das Element \rightarrow *Informationen einblenden*
3. durch den Reiter *Bearbeiten* \rightarrow *Informationen einblenden*

Zum besseren Verständnis wird nach jedem Abschnitt ein einfaches, wenn möglich fortlaufendes Beispiel gezeigt.



Der Massenpunkt: Ein Massenpunkt wird mit einem Mausklick erstellt, er unterscheidet sich von einem geometrischen Punkt dadurch, dass auf ihn Kräfte wirken können. Er besitzt folgende Eigenschaften:

- **Masse:** Da es keine physikalischen Einheiten in Cinderella gibt, wird die relative Masse des Punktes festgelegt. Mit Hilfe eines Schiebereglers kann sie in dem Intervall von null bis drei auf drei Stellen nach dem Komma eingestellt werden. Manuell kann jede beliebige Masse eingestellt werden. Die Eingabe muss mit ENTER bestätigt werden, ansonsten wird sie nicht übernommen.
- **Reibung:** Der Reibungskoeffizient kann zwischen null und eins gewählt werden, wobei eine betragsmäßig konstante Coulombreibung gemeint ist. Null beschreibt die minimale, eins die maximale Reibung.
- **Ladung:** Die Ladung ist verantwortlich für das Verhalten des Massenpunktes in Anwesenheit weiterer geladener Teilchen bzw. in einem elektrischen oder magnetischen Feld. Manuell kann jeder beliebige Wert für eine Ladung eingegeben werden.
- **Radius:** Cinderella versteht alle Massenpunkte, insofern sie nicht verändert worden sind, als punktförmige Massen. Die Einstellung Radius ermöglicht es, einem Massenpunkt einen festen Radius zuzuordnen, wodurch sich das Verhalten z. B. bei Stößen verändert.
- **Fixiert:** Wird ein Massenpunkt fixiert, wirken zwar Kräfte auf ihn bzw. er übt Kräfte auf andere Körper aus, aber er verändert seinen Ort nicht.

- **Geschwindigkeitsbegrenzung:** Dieser Punkt ist physikalisch nicht relevant. Er dient dazu, Fehler im Approximationsverfahren auszugleichen. Wird ein Massenpunkt unerwartet extrem schnell, so kann die Geschwindigkeitsbegrenzung zur Lösung des Problems führen.

Beispiel 5.1:

Es wird eine Masse erstellt und anschließend ihr Informationsfenster geöffnet. In dem Informationsfenster lassen sich auch Eigenschaften wie Größe, Farbe oder Spur einstellen.

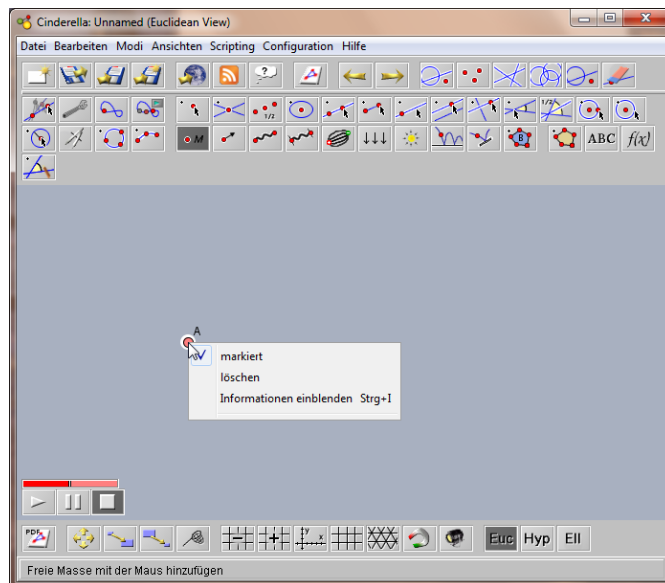


Abbildung 5.3.: Das Cinderella 2.0 Fenster

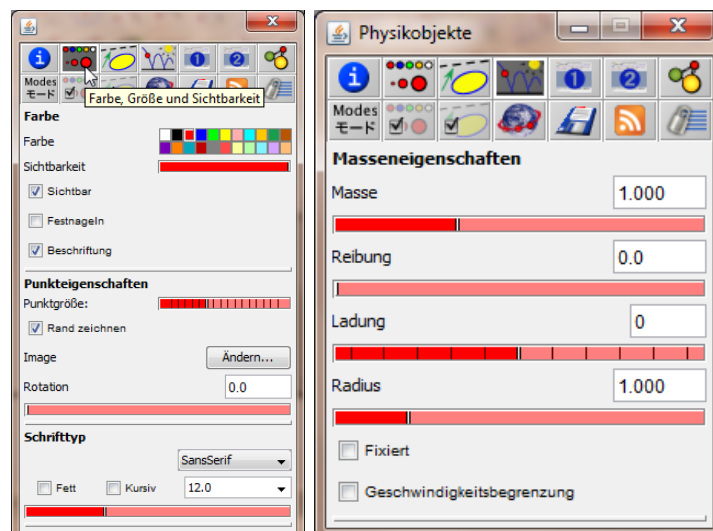


Abbildung 5.4.: Informationsfenster in Cinderella 2.0



Der Geschwindigkeitsvektor: Massenpunkte können in Cinderella mit einer gerichteten Geschwindigkeit versehen werden. Dieser Geschwindigkeitsvektor wird durch einen Mausklick an eine Masse angefügt und entsprechend der gewünschten Geschwindigkeit länger (schneller) oder kürzer (langsamer) gezogen. Ist keine Masse vorhanden, erstellt Cinderella automatisch einen Massenpunkt. Der Geschwindigkeitsvektor besitzt folgende Eigenschaften:

- **Faktor:** Bestimmt einen Faktor, mit dem der Betrag des Geschwindigkeitsvektors multipliziert wird, um die Flugbahn des Massenpunktes zu berechnen.
- **Bei Start lesen:** Ist dieser Punkt aktiviert, wird zu Beginn des Experiments die Anfangsgeschwindigkeit eingelesen und für den Versuch verwendet.

Beispiel 5.2:

Dem Massenpunkt wird eine Geschwindigkeit über den Knopf Masse mit Geschwindigkeit zugeordnet, indem von der Masse aus ein Geschwindigkeitsvektor gezogen wird. Je länger der Vektor, desto schneller der Massenpunkt.

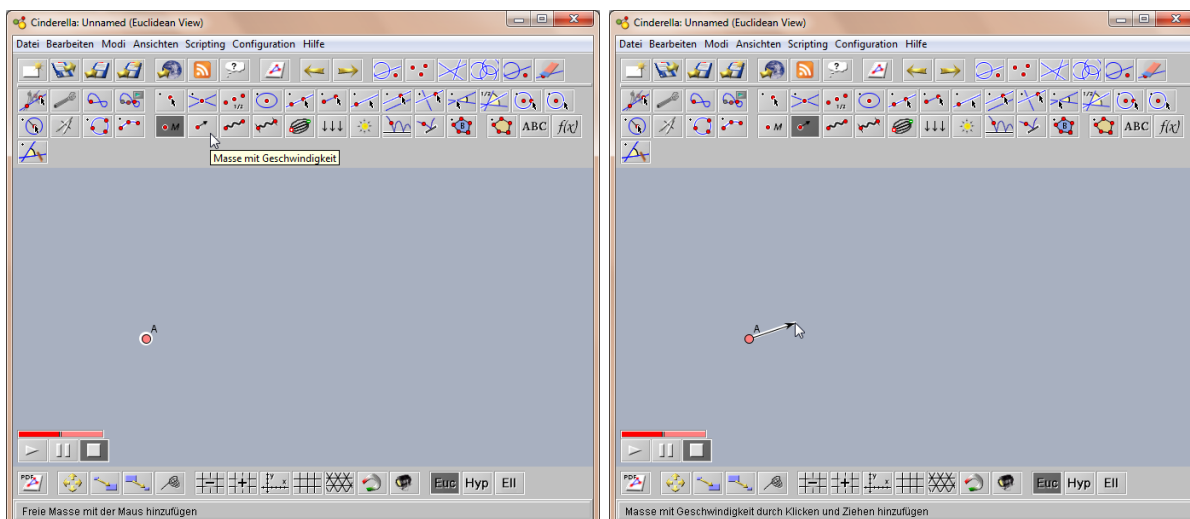


Abbildung 5.5.: Geschwindigkeitszuordnung in Cinderella 2.0



Das homogene Gravitationsfeld, konstante Richtung: Das Feld wird durch einen Vektor dargestellt. Länge und Richtung des Vektors beschreiben die Lage und Richtung des Gravitationsfeld. In den Eigenschaften gibt es die Option Faktor, mit welcher der Vektor wie bei der Geschwindigkeit multipliziert wird.

Beispiel 5.3:

Man erzeugt ein homogenes Gravitationsfeld, indem ein Vektor über den Knopf Schwerkraft erzeugt wird, und beobachtet die Flugbahn des Massenpunktes beim Ablaufen der Simulation, indem man die Play-Taste drückt.²⁴

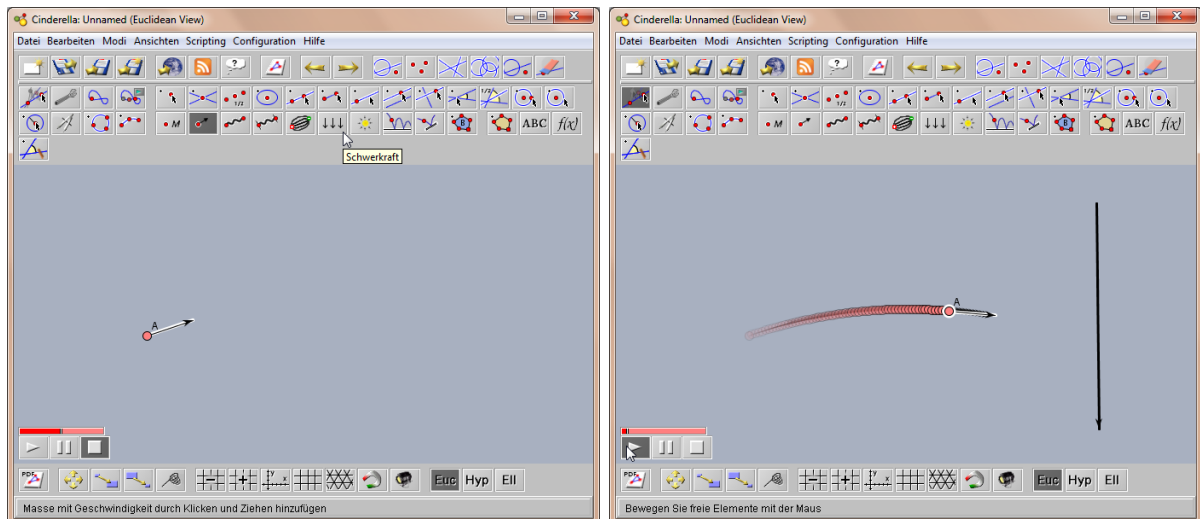


Abbildung 5.6.: Homogenes Gravitationsfeld in Cinderella 2.0



Das zentrale Gravitationsfeld, abstandsabhängig: Das zentrale Gravitationsfeld wird durch eine Sonne dargestellt. Im Gegensatz zum homogenen Gravitationsfeld, wirkt die Sonne als Fixstern mit einem abstandsabhängigen Gravitationsfeld. Die einzige Einstellmöglichkeit der Sonne ist ihre Masse und die wird genauso eingegeben wie bei einem Massenpunkt.

Beispiel 5.4:

Man konstruiert einen Massenpunkt mit einem Geschwindigkeitsvektor, in dessen Nähe ein zentrales Gravitationsfeld gesetzt wird und beobachtet dessen Flugbahn.

²⁴In den Screenshots wurde zur Verdeutlichung die Spur des Massenpunktes aktiviert. Einzustellen unter Informationen → Objektspuren → Spur.

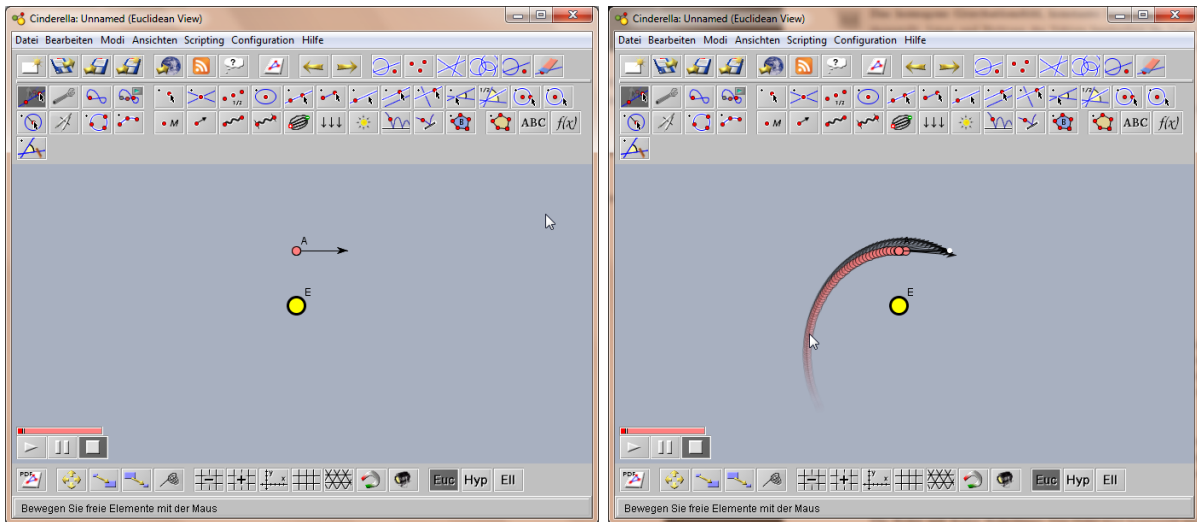


Abbildung 5.7.: Zentrales Gravitationsfeld in Cinderella 2.0



Das homogene Magnetfeld: Es wird zwischen mehreren festen Punkten aufgespannt und durch eine farblich markierte Fläche dargestellt. Als Einstellmöglichkeiten sind Polung und Stärke des Feldes sowie eine betragsmäßig konstante Coulombreibung vorhanden, die wie bei einem Massenpunkt eingestellt werden.

Beispiel 5.5:

Man erstelle einen Massenpunkt mit einem Geschwindigkeitsvektor, dessen Richtung auf ein homogenes Magnetfeld zeigt. Das homogene Magnetfeld wird erzeugt, indem man bspw. vier geometrische Punkte konstruiert und es anschließend mit dem Magnetfeld-Button und durch Anklicken jedes Punktes erstellt.

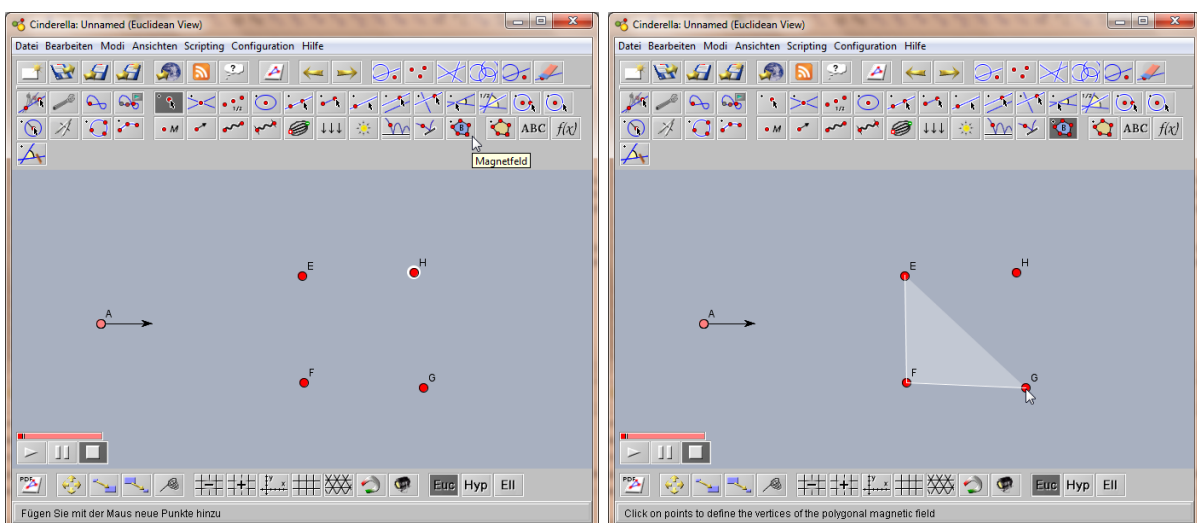


Abbildung 5.8.: Konstruktion eines Magnetfeldes in Cinderella 2.0

Anschließend bestimmt man über das Informationsfenster eine Ladung für den Massenpunkt und beobachtet die Flugbahn.

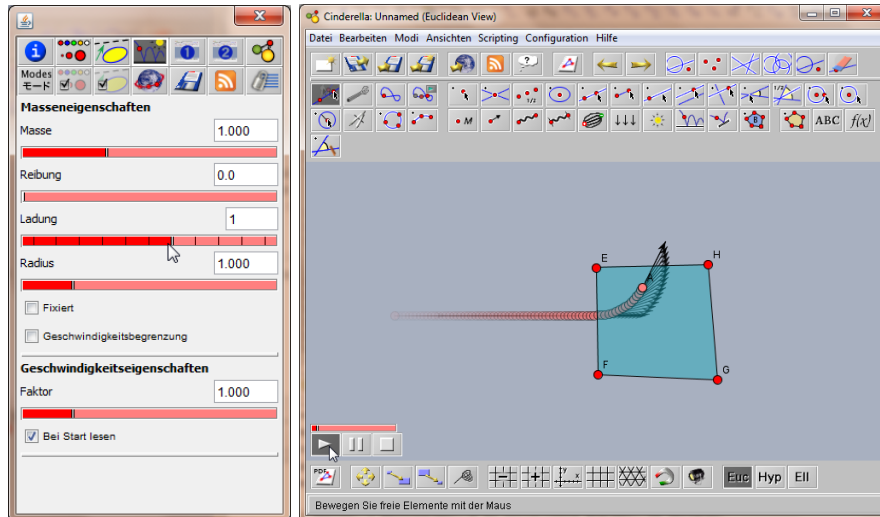


Abbildung 5.9.: Einstellungen eines Magnetfeldes in Cinderella 2.0

Die Feder mit fester Ruhelänge: Eine Feder wird konstruiert wie eine Strecke zwischen zwei Punkten. Sie kann zwischen zwei beweglichen Körpern oder einem beweglichen Körper und einem festen Punkt, der Wand, gespannt werden. Haben die Körper als Abstand zwischeneinander die Ruhelänge der Feder, wirkt keine Kraft auf sie. Die Kraft, die außerhalb der Ruhelänge auf die Körper wirkt, berechnet sich mit $\vec{F} = D \cdot (\text{Auslenkung} - \text{Ruhelänge})$. In den Eigenschaften gibt es den Punkt Federart, dieser hat vier Auswahlmöglichkeiten, die aber nicht alle Federn beschreiben. Stattdessen beschreiben sie, welche Kraft, ausgedrückt durch das Federsymbol, zwischen den Massen berechnet werden soll. Man hat folgende Möglichkeiten:

- **Gummiband:** Eine Feder mit der Ruhelänge null. Die Kraft zwischen den Endpunkten berechnet sich durch $\vec{F} = -D \cdot \vec{L}$.
- **Feder mit einer Ruhelänge:** Sie kann entweder manuell oder über einen Schieberegler in den Eigenschaften eingestellt werden. Zusätzlich gibt es die Option, dass die Ruhelänge aus der Konstruktion ausgelesen wird. Speziell für diese Einstellung gibt es die im Deutschen „Sonnenanregung“ genannten Einstellungen. Gemeint ist hiermit, dass die Ruhelänge der Feder periodisch angeregt wird. Man kann hierzu die Amplitude und die Phase einstellen, mit der die Federlänge schwingen soll.
- **Gravitationskraft:** Die Kraft, die zwischen den Massen der Endpunkte wirkt.
- **Coulombkraft:** Die Kraft, die zwischen den Ladungen der Endpunkte wirkt.

Beispiel 5.6:

Man erstelle einen geometrischen Punkt, einen Massenpunkt und ein homogenes Gravitationsfeld. Anschließend spanne man eine Feder zwischen den beiden Punkten. In den Informationen soll Ruhelänge aus Zeichnung ausgewählt sein. Man starte die Simulation, lenke die Masse mit einem Mausklick aus und beobachte, wie die Masse anfängt zu schwingen.²⁵

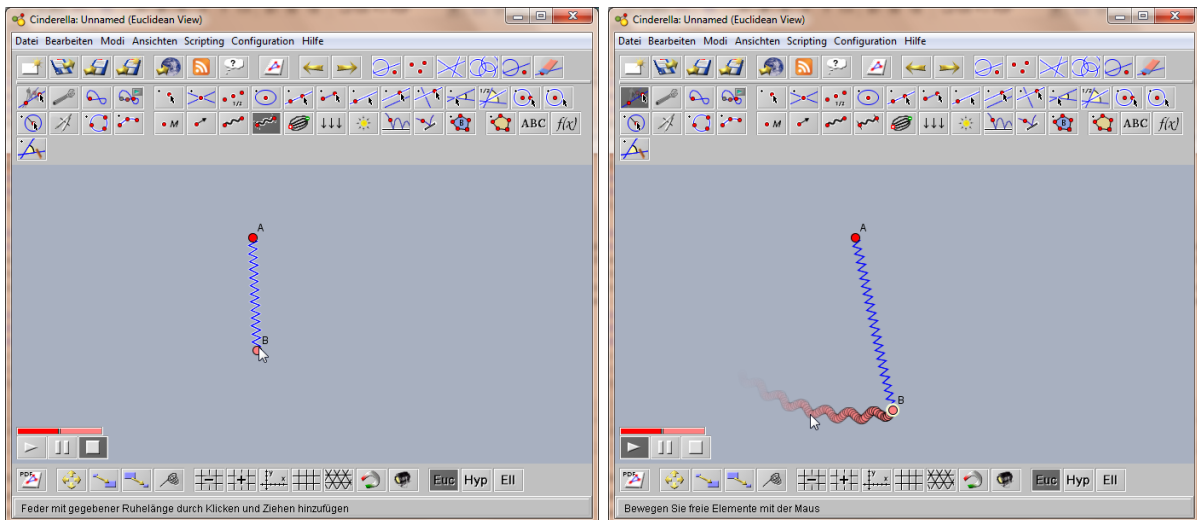


Abbildung 5.10.: Federn in Cinderella 2.0

Der Boden: Der Boden wird durch einen Mausklick erstellt. Er wird durch eine schwarze horizontale Linie dargestellt, die von Massenpunkten nicht passiert werden kann. Bewegte Körper werden am Boden reflektiert, hierzu kann man in den Einstellungen die Dämpfung des Bodens einstellen. Die jeweilige Geschwindigkeitskomponente eines Körpers multipliziert mit dem Dämpfungsfaktor ergibt die neue Geschwindigkeitskomponente. Die Dämpfung kann durch einen Schieberegler eingestellt werden, dabei steht der Faktor null für keine und eins für maximale Dämpfung, der Körper bleibt auf dem Boden.



Die Wand: Sie wird wie eine Strecke konstruiert. Sie besitzt die gleichen Eigenschaften wie der Boden. Allerdings können Wände in die gesamte Konstruktion und in jeglicher Richtung eingebaut werden.

²⁵Die anderen Federn bzw. Kräfte werden ebenso erstellt. Es muss allerdings darauf geachtet werden, dass z. B. bei der Coulombkraft die Massen eine Ladung besitzen müssen, um eine Kraft aufeinander wirken lassen zu können.

Beispiel 5.7:

Da der Boden und die Wand sehr ähnlich sind, wird ein Beispiel für beide Element verwendet. Zuerst wird ein Boden durch einen Mausklick konstruiert, dann zwei möglichst senkrechte Wände. Zwischen die Wände setzt man einen Massenpunkt mit einem Geschwindigkeitsvektor. Bevor man die Simulation startet, kann man noch ein homogenes Gravitationsfeld einfügen, um einen hüpfenden Ball ohne Reibung zu simulieren. Der Boden und die Wände besitzen standardmäßig eine Dämpfung. Im Informationsfenster lässt sich diese beliebig verstellen.

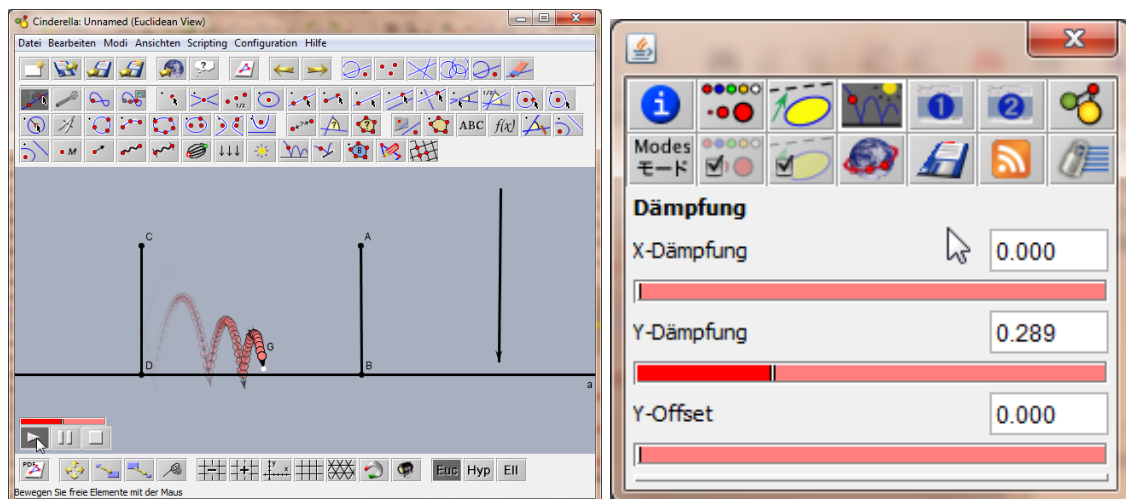


Abbildung 5.11.: Böden und Wände in Cinderella 2.0



Die Simulationseinstellungen: Die Simulationseinstellungen beschreiben globale Einstellungen, die für die gesamte Konstruktion gelten, unabhängig davon, welche Einstellungen in den jeweiligen Elementen getätigt worden sind.

- **Genauigkeit:** Sie gibt an, wie groß die Zeitabstände sind, die Cinderella in der Simulation benutzt, um die Bahn eines Körpers zu berechnen. Eine Vergrößerung der Genauigkeit bewirkt, dass die Simulation langsamer abläuft. Ein Teilstrich des Schiebereglers bewirkt jeweils eine Halbierung des Zeitintervalls oder, anders ausgedrückt, eine Verdopplung der Wegpunkte.
- **Bilderrate:** Sie gibt an, wie viele Bilder pro Sekunde in der Simulation gezeigt werden. Eine Erhöhung des Schiebereglers um einen Teilstrich bewirkt, dass sich die Bilderzahl pro Sekunde halbiert. Dies führt dazu, dass die Simulation bei einer maximalen Einstellung von Genauigkeit und Bilderrate genauso schnell abläuft wie bei der minimalen Einstellung beider Schieberegler. Der Unterschied liegt darin, dass die maximalen Einstellungen genauere Ergebnisse liefern.
- **Federkonstante:** Sie ist ein globaler Multiplikator, mit dem jede Federkraft in der Simulation multipliziert wird. Ist dieser Faktor gleich null, wirkt keine Federkraft. Der Faktor

eins ist die neutrale Einstellung.

- **Schwerkraft:** Sie ist eine globale Kraft, die nach unten gerichtet ist. Sie wirkt auf alle Körper. Im Gegensatz zum homogenen Gravitationsfeld als Element wird diese Kraft nicht durch einen Vektor dargestellt, sondern wirkt im Hintergrund.
- **Reibung:** Sie beschreibt eine globale Coulombreibung, unabhängig davon, ob die Elemente eine Reibung besitzen.
- **Anziehung zwischen Massen:** Zwischen allen Massenpunkten in der Konstruktion wirken Gravitationskräfte.
- **Ladungen erzeugen Kräfte:** Zwischen allen Ladungen in der Konstruktion wirken Coulombkräfte.
- **Massen sind Bälle:** Massenpunkte werden nicht mehr punktförmig angenommen, sondern als Kugel bzw. Kreis mit der Größe, die in der Konstruktion angegeben ist.
- **Antriebseigenschaften:** Sie beschreiben Faktoren, mit denen die bei der Feder beschriebene „Sonnenanregung“ multipliziert wird.

Beispiel 5.8:

Man kann in den oberen Beispielen die globalen Einstellungen testen. Hierzu wird das Informationsfenster geöffnet und der Reiter Globale Simulationseigenschaften ausgewählt.

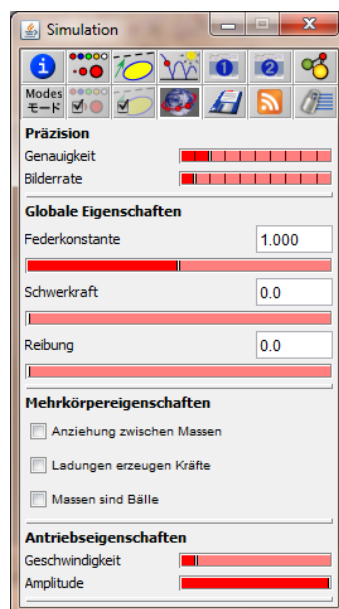


Abbildung 5.12.: Globale Einstellungen in Cinderella 2.0

Die Möglichkeiten von CindyLab

CindyLab ist erst in Cinderella 2.0 integriert worden und steckt noch in den Kinderschuhen. Es ist schon jetzt möglich, viele physikalische Simulationen durchzuführen. Interessant wären allerdings auch Darstellungsmöglichkeiten von Beschleunigung, Impuls, Energie usw. Diese sind nur bedingt darzustellen und nicht durch CindyLab einstellbar. Die Möglichkeit diese Größen zu veranschaulichen ist durch die integrierte Scriptsprache CindyScript gegeben und wird aus diesem Grund anschließend besprochen.

5.4.2. CindyScript - Die Scriptsprache von Cinderella 2.0²⁶

Grundlagen

CindyScript ist eine speziell für Cinderella entwickelte Scriptsprache, die den Umgang mit der Geometrie, der Funktionentheorie und den Simulationen effektiver und umfangreicher gestalten soll. Sie kann auf drei verschiedene Arten eingesetzt werden. Es können mathematische

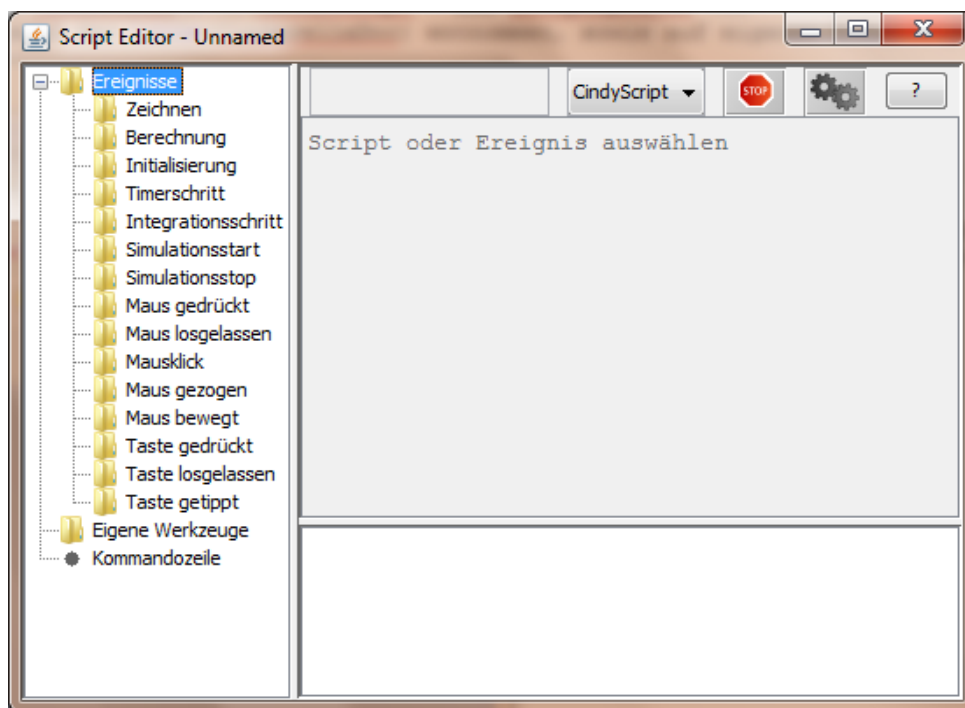


Abbildung 5.13.: Das Eingabefenster von CindyScript

Funktionen in eine Konstruktion integriert, Eigenschaften von Elementen der Konstruktion mit CindyScript verändert und eigenständige Programme geschrieben werden. Ein integrierter Script Editor ermöglicht die einfache Eingabe der Befehle. Er ist unter *Scripting* → *Scripte erstellen* zu finden. Das Scriptfenster besteht aus drei Teilen. In der rechten oberen Ecke sind eine Play-

²⁶Aus RICHTER-GEBERT und KORTENKAMP: *Cinderella.2 Documentation*, <http://doc.cinderella.de/tiki-index.php?page=CindyScript> sowie auf eigenen Erfahrungen basierend.

(dargestellt durch zwei Zahnräder) und eine Stop-Taste. Die Play-Taste ist nach jeder Änderung bzw. Scripteingabe neu zu betätigen, um die Befehle zu implementieren. Ist ein Befehl falsch eingegeben oder entsteht eine Endlosschleife beim Ausführen der Scripte, dient die Stop-Taste zum Anhalten der Operation. In der rechten unteren Ecke ist ein Fenster, in dem die von Cinderella erzeugten Ausgaben dargestellt werden, z. B. Fehlermeldungen, berechnete Ergebnisse oder allgemeine Hinweise. Scriptbefehle können auf der rechten Seite eingegeben werden. Um klarzustellen, bei welchem Ereignis das jeweilige Script ausgelöst werden soll, gibt es auf der linken Seite eine Auswahl an Unterordnern, denen Ereignisse zugeordnet sind.

Optionen von CindyScript

Im Folgenden wird die Funktion der Unterordner des Scripteingabefensters erläutert:

- **Zeichnen:** In diesem Verzeichnis werden die Befehle direkt ausgeführt und wiederholt, sobald etwas Neues zur Konstruktion hinzugekommen ist.
- **Berechnung:** Befehle werden ausgeführt, wenn etwas bewegt wird. Soll sich also z. B. die Farbe eines Punktes beim Bewegen verändern, muss der Befehl in diesem Verzeichnis eingefügt werden.
- **Initialisierung:** Ein Script an dieser Stelle wird immer ausgeführt, sobald etwas zur Konstruktion hinzugefügt oder aus ihr entfernt wird. Es ist nützlich, Punkte an dieser Stelle einzufügen, um Variablen zurückzusetzen oder Massenpunkte an ihren Ursprungsort zurückzusetzen.
- **Timersschritt:** Das Script wird in regelmäßigen Zeitabständen während einer Simulation ausgeführt.
- **Integrationsschritt:** Das Script wird nach einer bestimmten Schrittweite wieder ausgeführt.
- **Simulationsstart:** Scripte, die hier enthalten sind, werden am Anfang einer Simulation ausgeführt. Hier sollte die Ausgangssituation der Simulation erstellt werden.
- **Simulationsstop:** Wird am Ende einer Simulation ausgelöst. Hier können die Ergebnisse ausgegeben werden.
- **Maus gedrückt:** Scripte werden ausgeführt, während die linke Maustaste gedrückt wird.
- **Maus losgelassen:** Scripte werden ausgeführt, sobald die linke Maustaste losgelassen wird.
- **Mausklick:** Scripte werden ausgeführt, wenn die linke Maustaste kurz angeklickt wird.
- **Maus gezogen:** Scripte werden ausgeführt, solange die linke Maustaste gehalten und die Maus bewegt wird.

- **Maus bewegt:** Scripte werden ausgeführt, solange die rechte Maustaste gehalten und die Maus bewegt wird.
- **Taste gedrückt:** Scripte werden ausgeführt, während eine Taste auf der Tastatur gedrückt wird.
- **Taste losgelassen:** Scripte werden ausgeführt, sobald eine Taste auf der Tastatur losgelassen wird.
- **Taste getippt:** Scripte werden ausgeführt, nachdem eine Taste kurz gedrückt wurde.

Um ein besseres Verständnis von CindyScript zu erhalten, wird im Folgenden auf einige Befehle eingegangen. Dabei werden nach jedem neuen Element einfache Beispiele eingefügt.

Variablen und Funktionen müssen nicht explizit definiert werden, sondern können direkt als Ausdruck eingegeben werden. Eine Funktion wird definiert, indem

$$\text{Funktion}(\text{Argument1}, \text{Argument2}) := \text{Ausdruck};$$

eingegeben wird. Das Semikolon dient dazu, die Zeile bzw. den Befehl abzuschließen.

Beispiel 5.9:

Man erstelle eine Funktion $f(x)$ und lasse sich diese durch den Befehl `plot()` zeichnen:

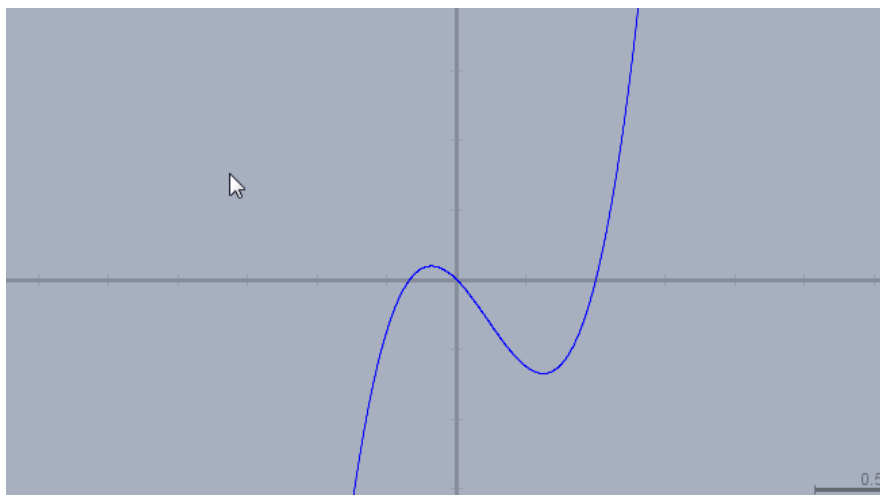
$$f(x) = 3x^3 - 2x^2 - x; \text{plot}(f(x));$$


Abbildung 5.14.: Funktionsplot einer kubischen Parabel

Operationen werden in Cinderella nacheinander ausgeführt. Die erste Eingabe wird also auch als erstes ausgeführt und kann von nachfolgenden Operationen überschrieben werden. Cinderella ist auch im Stande rekursive Funktionen zu berechnen. Sie werden mit einer `if`-Bedingung (siehe S.50) eingegeben.

Beispiel 5.10:

Die Fakultätsfunktion kann durch eine rekursive Darstellung definiert werden:²⁷

```
Fakultaet:=if(n==0,1,n*Fakultaet(n-1);
```

Die Definition einer Variablen findet in CindyScript an der Stelle statt, an der sie zum ersten Mal im Programmcode eingegeben wird. Der Befehl `createvar(Variable);` erzeugt und `removevar(Variable);` entfernt eine Variable. Möglichkeiten für Variablen sind in CindyScript: Zahlen, Zahlenreihen, Boolesche Aussagen (0,1), ganze Listen, geometrische Punkte und Programme. Definierte Variablen sind global nutzbar, d. h. sie können in jeder Zeile des Programmcodes wieder aufgerufen werden. Eine Ausnahme bilden die lokalen Variablen, die z. B. in Summen als Laufoperatoren genutzt werden. Sie können mit `local(Variable1, Variable2,...);` erzeugt und mit `removevar(Variable1, Variable2,...);` entfernt werden. Diese Möglichkeit muss nur angewandt werden, wenn man explizit lokale Variablen benötigt. Lokale Variablen in Summen oder Aufzählungen müssen nicht extra definiert werden, sondern werden von CindyScript nur an dieser Stelle genutzt. Man kann einzelne Variablen auf ihren ursprünglichen Wert zurücksetzen, indem man den Befehl `clear(Variable);` nutzt. Um alle Variablen wiederherzustellen, muss die Klammer leer bleiben. Einige wichtige konstante Größen und mathematische Variablen sind in CindyScript schon vordefiniert, z. B. π , das imaginäre i oder die Booleschen Aussagen (0,1), wobei 0 durch False und 1 durch True ausgedrückt wird.

Beispiel 5.11:

Es wird ein Punkt auf einem Kreis bewegt, um eine Sinuskurve darzustellen. Hierzu definiert man sich eine Variable c , die sich je nach Koordinate eines bewegten Punktes ändert und lässt die Bewegung in y -Richtung graphisch darstellen. Den Punkt und den Kreis erzeugt man mit Cinderella und legt mit dem Automatische Animation-Knopf die Bewegung auf dem Kreis fest. Die Variable sowie die graphische Darstellung werden mit CindyScript erzeugt. Der Befehl `drawcurves` wird auf Seite 57 genauer erläutert.

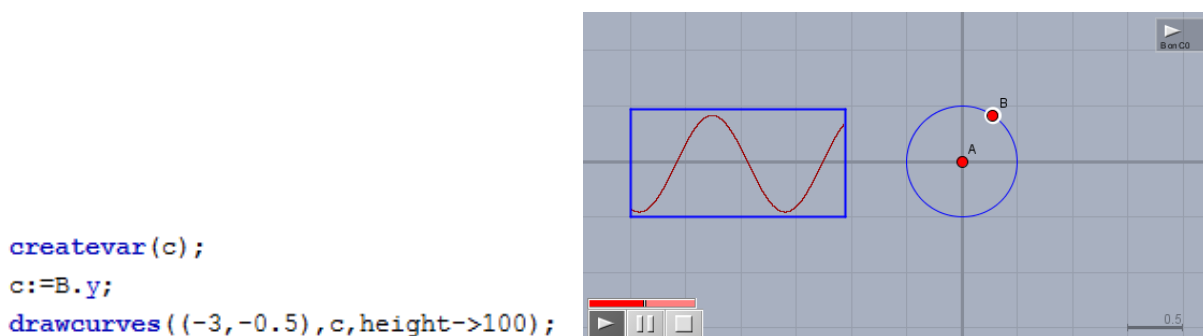


Abbildung 5.15.: Darstellung einer Sinuskurve

²⁷vgl. RICHTER-GEBERT und KORTENKAMP: *Cinderella.2 Documentation*, <http://doc.cinderella.de/tiki-index.php?page=Variables+and+Functions>.

Physikalische Eingabemöglichkeiten

Alle Einstellungen, die in Cinderella 2.0 über das Informationsfenster festgelegt werden, kann man auch in CindyScript eingeben. Es werden hier nur Befehle vorgestellt, die Konstrukte aus CindyLab beschreiben. Für alle nicht aufgeführten Befehle wird die Onlinedokumentation auf <http://doc.cinderella.org> empfohlen. Befehle, die sich auf Punkte, Geraden, Federn oder Felder beziehen, werden mit `Name.Befehl=Gewünschte Aktion` aufgerufen, z. B. `A.friction=0.3` setzt die Reibung des Punktes A auf 0.3. Eigenschaften, die nur eingelesen werden sollen, werden ohne Gleichheitszeichen und die gewünschte Aktion eingegeben. Eingelesene Daten können von CindyScript nicht verändert werden. Sie dienen dazu, z. B. die Zeichnung eines Graphen zu ermöglichen. Für Massenpunkte stellt CindyScript die in Tabelle 5.1 dargestellten Befehle zur Verfügung.

Befehl	Bedeutung
<code>mass</code>	Eine reelle Zahl, die die Masse angibt.
<code>charge</code>	Eine ganze Zahl, die die Ladung angibt.
<code>friction</code>	Eine reelle Zahl, die die Reibung angibt.
<code>radius</code>	Eine reelle Zahl, die den Radius des Massenpunktes als Ball beschreibt.
<code>vx</code> bzw. <code>vy</code>	Eine reelle Zahl, die die Geschwindigkeit in x- bzw. y-Richtung angibt.
<code>v=[x,y]</code> (=Vektordarstellung)	Wird als zweidimensionaler Vektor eingelesen, der die Geschwindigkeit beschreibt.
<code>fx</code> bzw. <code>fy</code>	Eine reelle Zahl, die die Kraft in x- bzw. y-Richtung angibt.
<code>f</code>	Wird als zweidimensionaler Vektor eingelesen, der die Kraft angibt.
<code>kinetic</code> oder <code>ke</code>	Liest die kinetische Energie des Massenpunktes ein.

Tabelle 5.1.: Befehle für Massenpunkte

Zusätzlich zu den Einstellungsmöglichkeiten gibt es noch die Option, dass ein Massenpunkt nicht von den Einstellungen in der Simulation betroffen ist. Dieser ist dann zwar fixiert, wirkt aber trotzdem auf andere Elemente in der Simulation. Dieser Befehl lautet: `Name.simulate=False` bzw. `Name.simulate=True`, falls der Massenpunkt nicht fixiert sein soll.

Beispiel 5.12:

Es werden drei Massenpunkte in gleichem Abstand zueinander konstruiert. In den globalen Simulationseinstellungen wird der Punkt Ladungen erzeugen Kräfte aktiviert. Anschließend werden ihnen mit CindyScript Ladungen und Reibung zugeordnet, sowie die Geschwindigkeit, die Kraft, die auf einen Massenpunkt wirkt, und seine kinetische Energie graphisch dargestellt. Der Punkt C bekommt auch eine Ladung, wird aber von der Simulation ausgeschlossen, um die Funktion

dieser Option darzustellen.

```

A.charge=1;
A.friction=0.3;
B.charge=-1;
drawcurves((0,-10),[B.vx,B.vy,B.fx,B.fy,B.ke],
texts->[
"Geschwindigkeit von v in x-Richtung="+B.vx,
"Geschwindigkeit von v in y-Richtung="+B.vy,
"Kraft auf B in x-Richtung="+B.fx,
"Kraft auf B in y-Richtung="+B.fy,
"Kinetische Energie von B="+B.ke, ],
width->400,height->50);
C.charge=1;
C.simulate=false;

```

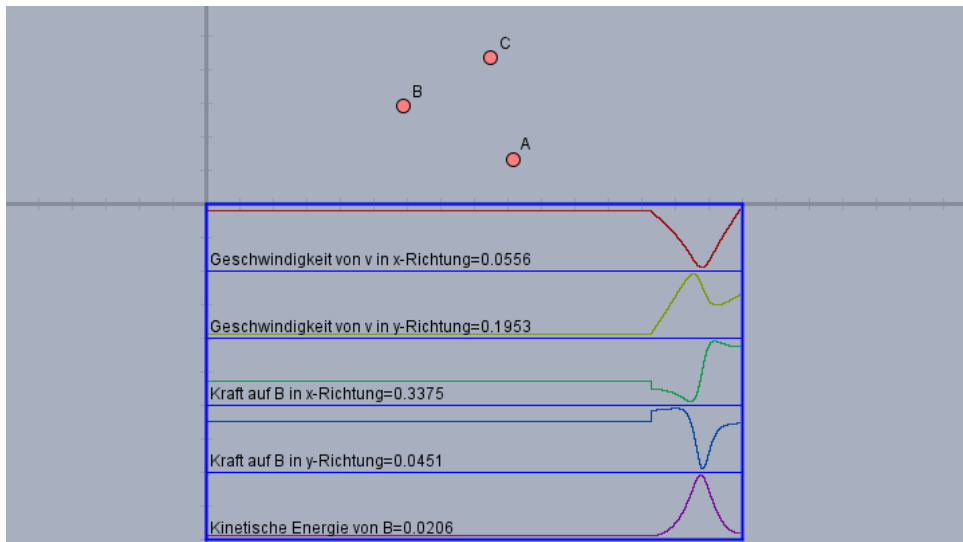


Abbildung 5.16.: Verhalten von Massen unter verschiedenen Kräfteinwirkungen

Analog zu den Einstellungen für Massenpunkte werden die Möglichkeiten für Federn, Anziehung zwischen Massenpunkten und Coulombkräfte in Tabelle 5.2 angegeben.

Beispiel 5.13:

Man erstelle sich zwei geometrische Punkte als feste Enden für zwei Federn. Eine Feder wird mit der Ruhelänge null (Gummiband) und die andere mit einer festen Ruhelänge konstruiert. Es werden zwei Graphen gezeichnet, einer, der die potentielle Energie der Feder ohne Ruhelänge darstellt, und ein zweiter, der die Auslenkung der anderen Feder darstellt. Der Befehl `simulation().friction=0.02;` erzeugt eine Coulombreibung, die auf alle Elemente der Simulation wirkt. (vgl. Tabelle 5.4)

Befehl	Bedeutung
<code>l</code>	Liest die momentane Länge der Feder ein.
<code>lrest</code>	Liest die Ruhelänge der Feder ein.
<code>ldiff</code>	Liest die momentane Auslenkung der Feder ein.
<code>strength</code>	Eine reelle Zahl, die die Stärke der Federkonstanten angibt.
<code>f</code>	Wird als zweidimensionaler Vektor eingelesen, der die Kraft angibt.
<code>amplitude</code>	Eine reelle Zahl, die angibt, wie groß die Amplitude der Eigenschwingung ist.
<code>speed</code>	Eine reelle Zahl, die angibt, wie hoch die Erregerfrequenz der Eigenschwingung ist.
<code>potential</code> oder <code>pe</code>	Liest die potentielle Energie der Feder ein.

Tabelle 5.2.: Befehle für Federn, Anziehung zwischen Massen und Coulombkräfte

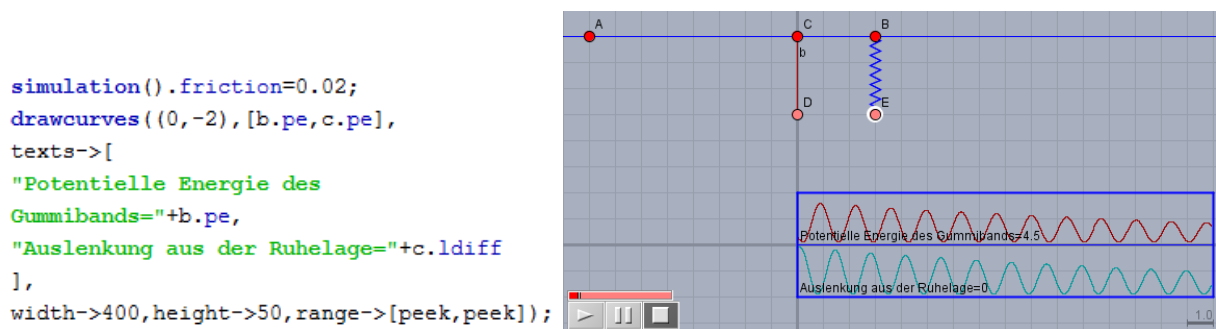


Abbildung 5.17.: Beispiel von Federschwingungen

Die Befehle für die verschiedenen Kraftfelder sind in Tabelle 5.3 dargestellt.

Beispiel 5.14:

Ein zentrales Gravitationsfeld sowie zwei bewegte Körper, die um das Zentrum kreisen sollen, werden mit Cinderella erzeugt. Die Einstellungen für das Kraftfeld werden mit CindyScript vorgenommen. Dem Kraftfeld werden eine Masse für das Zentrum und eine Kraft zugewiesen. Außerdem werden die kinetischen Energien der bewegten Körper und die potentielle Energie des gesamten Systems aufgezeichnet.

Befehl	Bedeutung
<code>factor</code>	Eine reelle Zahl, die den Multiplikator zwischen der graphischen Darstellung und der aktuellen Geschwindigkeit angibt.
<code>strength</code>	Eine reelle Zahl, die die Stärke des Kraftfeldes angibt.
<code>potential</code> oder <code>pe</code>	Liest die potentielle Energie aller Körper in einem Gravitationsfeld ein.
<code>mass</code>	Eine reelle Zahl, die die Masse des Mittelpunkts eines zentralen Gravitationsfeldes bestimmt.
<code>friction</code>	Eine reelle Zahl zwischen null und eins, die die Reibung innerhalb eines Kraftfeldes festlegt.
<code>xdamp</code> bzw. <code>ydamp</code>	Eine reelle Zahl, die angibt, wie groß die Dämpfung an Wänden und Böden in x- bzw. y-Richtung ist.

Tabelle 5.3.: Befehle für Gravitations- und Magnetfelder

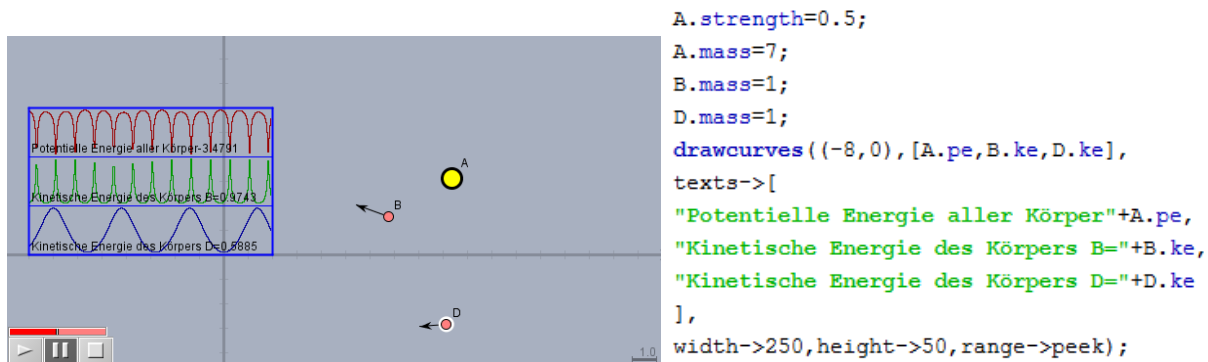


Abbildung 5.18.: Beispiel eines zentralen Gravitationsfeldes

Da die globalen Simulationseinstellungen (Tabelle 5.4) für die gesamte Simulation gelten, wird anstatt des Variablennamens `simulation()` eingegeben, z. B. setzt `simulation().friction=0.3` die Reibung für alle Elemente auf 0.3.

Bedingungen und Schleifen

Bedingungen und Schleifen sind wichtig für Simulationen, um z. B. bestimmte Einstellungen auszuschließen, die physikalisch keinen Sinn ergeben oder mathematisch nicht berechenbar sind. Sie können ineinander verschachtelt werden, um exakte Aussagen treffen bzw. Simulationen so genau wie möglich beschreiben zu können.

Die `if`-Bedingung wird eingesetzt, falls es eine feste Bedingung gibt, die nur wahr oder falsch sein kann und die je nach Ergebnis eine Reaktion hervorrufen soll. Bei der Eingabe müssen eine Bedingung und zwei Befehle festgelegt werden.

Befehl	Bedeutung
<code>gravity</code>	Eine reelle Zahl, die die Stärke eines in der gesamten Simulation wirkenden Gravitationsfeldes angibt.
<code>friction</code>	Eine reelle Zahl, die eine Coulombreibung für jeden Körper festlegt.
<code>potential</code> oder <code>pe</code>	Liest die potentielle Energie aller Körper in der Simulation ein.
<code>kinetic</code> oder <code>ke</code>	Liest die kinetische Energie aller Körper in der Simulation ein.

Tabelle 5.4.: Befehle für die globalen Simulationseinstellungen

Beispiel 5.15:

Man erstellt sich einen Körper, dargestellt durch einen Massenpunkt in Cinderella. Im Scripteditor wird der Quelltext

```
if(A.y>0,
drawtext([0,0],„Y ist größer als null!"),
drawtext([0,0],„Y ist kleiner als null!")
);
```

eingegeben. Er bewirkt, dass im Ursprung des Koordinatensystems angezeigt wird, ob der Punkt A sich oberhalb oder unterhalb der X-Achse befindet. Wird die Simulation gestartet und der Massenpunkt bewegt, ändert sich die Aussage entsprechend.²⁸

Der `trigger`-Operator bezieht sich auch auf eine Aussage, die wahr oder falsch sein kann. Es wird ein Ereignis eingegeben, welches beim Eintreffen der Aussage ausgeführt werden soll.

Beispiel 5.16:

Analog zum vorherigen Beispiel erstellt man einen Massenpunkt und benutzt folgenden Quelltext:

```
trigger(A.y>0,drawtext([0,0],„Y ist nun größer als null!"));.
```

Der `trigger`-Operator bewirkt in diesem Beispiel, dass nur dann eine Ausgabe gemacht wird, wenn der Y-Wert des Punktes A vom negativen in den positiven Bereich übergeht. Der `drawtext`-Befehl wird auf S.52 genauer erläutert.²⁹

Die `while`-Schleife beinhaltet eine Boolesche-Bedingung und ein Ereignis, welches ausgeführt wird, solange diese Bedingung wahr ist. Sie ist also eine Zählschleife, die ausgeführt wird, bis die Bedingung nicht mehr zutrifft.

²⁸vgl. RICHTER-GEBERT und KORTENKAMP: *Cinderella.2 Documentation*, <http://doc.cinderella.de/tiki-index.php?page=Control+Operators#if2>.

²⁹vgl. RICHTER-GEBERT und KORTENKAMP: *Cinderella.2 Documentation*, <http://doc.cinderella.de/tiki-index.php?page=Control+Operators&highlight=trigger>.

Beispiel 5.17:

Ein Beispiel einer `while`-Schleife ist, die Berechnung der Fakultätsfunktion in folgender Form:

```
x=1;           //x mit 1 initialisieren
y=5;           //y mit 5 initialisieren
fak=1;         //fak mit 1 initialisieren
erg=while(     //erg mit while-Schleife initialisieren
x<y, x=x+1;    //x erhöht sich nach jedem Durchlauf um eins, bis y
               erreicht ist
fak=fak*x);    //fak errechnet sich in jedem Durchlauf neu
println(erg);  //Ausgabe von erg im Scripteditor aus
```

Hierbei bestimmt der Wert von `y` die Zahl, bis zu welcher die Fakultät ausgerechnet werden soll.

Die `repeat`-Schleife ist eine reine Zählschleife, die einen Befehl so oft ausführt, wie es festgelegt wurde. Sie baut sich auf, indem man angibt, bis zu welcher natürlichen Zahl die Schleife wiederholt und welches Ereignis jeden Durchlauf bearbeitet werden soll.

Beispiel 5.18:

Ein einfaches Beispiel wäre, sich Punkte einer Funktion im ersten und vierten Quadranten zeichnen zu lassen. Es wird als Beispiel eine Sinusfunktion betrachtet. Der Operator `draw` wird auf Seite 52 genauer erläutert.

```
repeat(720,i,draw[i/360*pi,sin(i/360*pi)]);
```

Die Schleife wird 720 mal durchlaufen. Bei jedem Durchlauf erhöht sich die Zählvariable `i` um eins. Es werden somit 720 Punkte gezeichnet, bis die Schleife komplett abgearbeitet ist.

Die Zeichenoperatoren

Eine weitere Möglichkeit, Eigenschaften in den Simulationen zu visualisieren ist, Zeichenoperatoren zu nutzen. Diese Operatoren ermöglichen es, z. B. Daten direkt in der Simulation auszugeben. Der `draw`-Operator zeichnet Objekte, die keine Auswirkungen auf die Simulationen haben. Es kann aber von Vorteil für die Simulation sein, bestimmte Punkte oder Geraden einzuzichnen, die Zusammenhänge verdeutlichen. Der `draw`-Operator besitzt mehrere Eingabemöglichkeiten. Es können Punkte, Strecken und Geraden gezeichnet werden. Die Befehle lauten:

```
draw([x,y],Option);           zeichnet einen Punkt
draw([x1,y1],[x2,y2],Option); zeichnet eine Strecke
draw(line([a,b,c],Option));    zeichnet eine Gerade der Form ax+by+c=0
draw(join([x1,y1],[x2,y2],Option)); zeichnet eine Gerade durch zwei Punkte
```

Es gibt eine Reihe von Einstellungsmöglichkeiten von Punkten, Strecken und Geraden, die in Tabelle 5.5 aufgeführt sind. Diese Optionen werden aneinandergereiht am Ende des Befehls

eingetragen. Zudem ist es mit dem Befehl `drawpoly([Liste],Farbe)`; möglich Polygone beliebig einzeichnen zu lassen. Liste bedeutet, dass jeder Eckpunkt des Polygons angegeben werden muss. Ein weiterer Befehl zum Ausgeben eines Textes bzw. eines ausgelesenen Wertes ist `drawtext([Startpunkt], "Text");`.

Befehl	Bedeutung
<code>pointsize->Zahl</code>	Setzt eine Punktgröße zwischen 1 und 20 fest.
<code>linesize->Zahl</code>	Setzt eine Liniengröße zwischen 1 und 20 fest.
<code>size->Zahl</code>	Setzt sowohl Punkt- als auch Liniengröße fest.
<code>pointcolor->[Zahl,Zahl,Zahl]</code>	Legt die Punktfarbe im RGB-Format fest. Die Zahlen müssen zwischen 0 und 1 liegen.
<code>linecolor->[Zahl,Zahl,Zahl]</code>	Legt die Linienfarbe im RGB-Format fest. Die Zahlen müssen zwischen 0 und 1 liegen.
<code>color->[Zahl,Zahl,Zahl]</code>	Legt Linien- und Punktfarbe fest.
<code>alpha->Zahl</code>	Legt die Durchlässigkeit des Objekts fest. Der Wert 0 ist komplett transparent, 1 bedeutet völlig undurchlässig
<code>noborder->>true</code> bzw. <code>border->>true</code>	Der Punkt besitzt keinen bzw. besitzt einen Rand.

Tabelle 5.5.: Optionen für den draw-Operator

Beispiel 5.19:

Mit Hilfe des Scriptcodes wird das „Haus vom Nikolaus“ mit einem gelben Dach entwickelt.

```
draw([0,0],[0,2]);
draw([0,0],[2,2]);
draw([0,0],[2,0]);
draw([2,0],[2,2]);
draw([2,0],[0,2]);
draw([0,2],[2,2]);
drawpoly([[0,2],[1,3],[2,2]],color ->
(1,1,0));
drawtext([3,2],"Das ist das Haus vom
Nikolaus")
```



Abbildung 5.19.: Eine Demonstration der draw-Operatoren

Der plot-Operator ermöglicht es, Funktionen zu zeichnen. Sie werden durch den Befehl

```
plot(Funktion(#),Option);
```

initialisiert. Die Raute ist in CindyScript als Laufvariable vordefiniert und wird benötigt, um die Funktion zeichnen zu können. Optionen wie Farbe, Größe und Durchlässigkeit werden wie beim

`draw`-Operator eingestellt. Es gibt darüber hinaus einige Optionen im Umgang mit geplotteten Funktionen, die in Tabelle 5.6 dargestellt werden.

Befehl	Bedeutung
<code>start->Zahl</code>	Legt den Startwert für das Zeichnen fest.
<code>stop->Zahl</code>	Legt den Endwert für das Zeichnen fest.
<code>pxlres->Zahl</code>	Legt die Pixelgröße für das Zeichnen fest.
<code>extrema->True oder False</code>	Extrempunkte werden (nicht) eingezeichnet.
<code>minima->True oder False</code>	Minima werden (nicht) eingezeichnet.
<code>maxima->True oder False</code>	Maxima werden (nicht) eingezeichnet.
<code>zeros->True oder False</code>	Nullstellen werden (nicht) eingezeichnet.
<code>inflections->True oder False</code>	Umkehrpunkte werden (nicht) eingezeichnet.

Tabelle 5.6.: Optionen für den `plot`-Operator

Beispiel 5.20:

Es wird ein Polynom 4. Grades gezeichnet, bei dem Extremstellen und Wendepunkte markiert werden. Der Programmcode lautet:

```
plot(x^4-2*x^2,inflections->>true,extrema->>true);
```

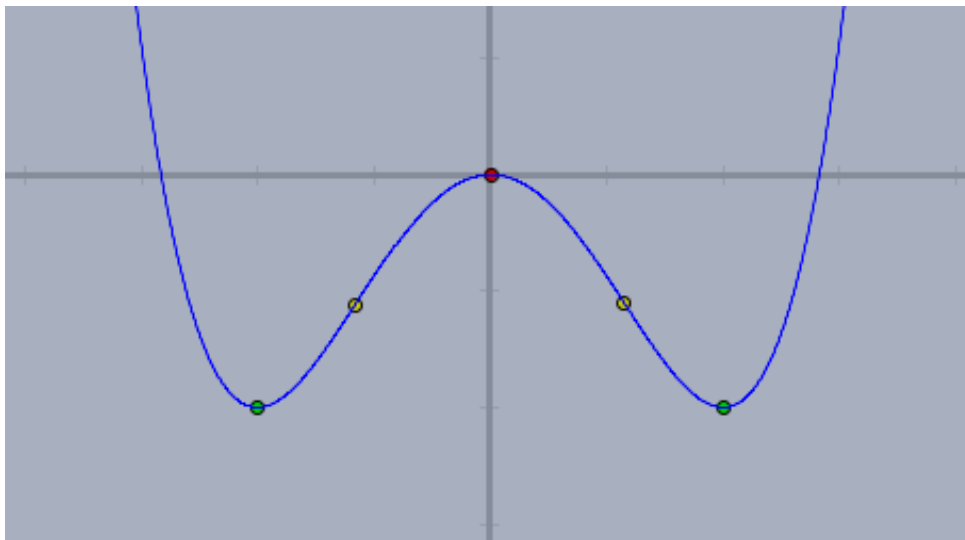


Abbildung 5.20.: Der `plot`-Befehl in Cinderella 2.0

Der `colorplot`-Operator ermöglicht es dem Nutzer, einem zweidimensionalen Graphen durch farbliche Abstufungen räumliche Tiefe zu verleihen. Der Befehl

```
colorplot([Farbe im RGB-Format],Eckpunkt1,Eckpunkt2)
```

benötigt eine Funktion, nach der sich die Farbe ändern soll und zwei Eckpunkte, zwischen denen ein Rechteck festgelegt wird, in dem der Graph gezeichnet wird. Mit der Option `pxlres` können die farblichen Abstufungen weicher bzw. pixeliger gestaltet werden.

Beispiel 5.21:

Es wird ein Zentralpotential einer homogenen Kugel durch farbliche Abstufungen räumlich dargestellt. Hierzu erzeugt man einen Massenpunkt. Der `colorplot`-Befehl lautet dann

```
colorplot([dist(#,C)/10,dist(#,C)/10,dist(#,C)/10],[-50,-50],[50,50]);
```

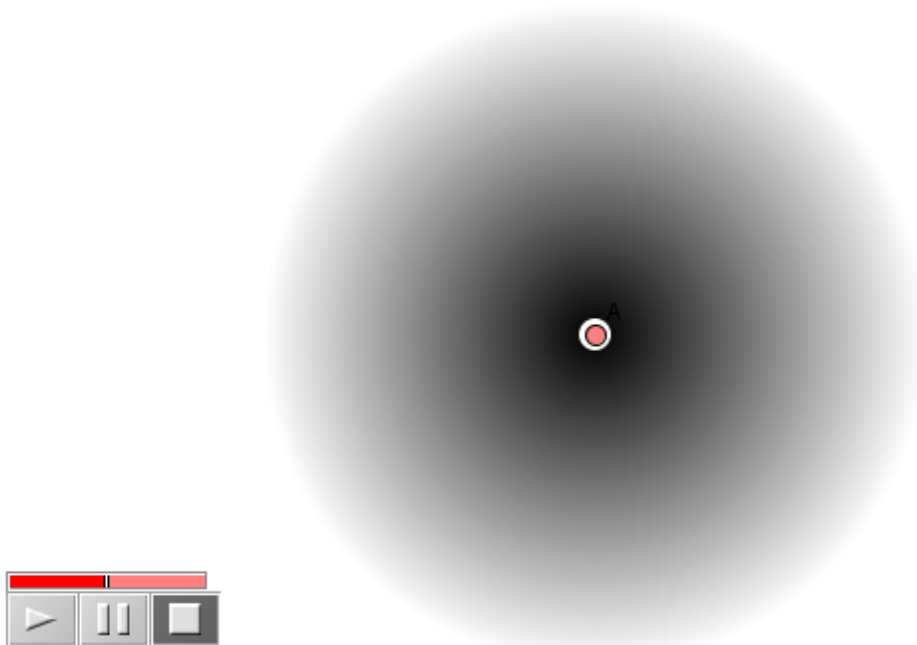


Abbildung 5.21.: Der `colorplot`-Befehl

Der `drawfield`-Operator dient zum Zeichnen von Vektorfeldern. Der Befehl

```
drawfield(Vektorfunktion,Option)
```

erstellt ein Vektorfeld mit rot-blauen Nadeln, die sich in Richtung des Vektorfeldes ausrichten. Nutzt man die Option `stream->true`, wird das Vektorfeld durch bewegte Punkt dargestellt. Diese ziehen eine Spur hinter sich her. Die Länge der Spur gibt dabei die Feldstärke an. Weitere Optionen sind in Tabelle 5.7 dargestellt.

Beispiel 5.22:

Es wird ein entfernungsabhängiges Vektorfeld konstruiert, ähnlich zu einem zentralen Gravitationsfeld. Hierzu erstellt man einen Massenpunkt. Im CindyScript-Eingabefenster dient die Befehlszeile

```
drawfield([(#.x-A.x)/dist(#,A)^2, (#.y-A.y)/dist(#,A)^2], stream->>true);
```

dazu, den normierten Vektor in die Richtung von A zeigend, durch die Entfernung zum Punkt A zu dividieren und somit die Vektorfeldkomponente an der Stelle # zu erhalten.

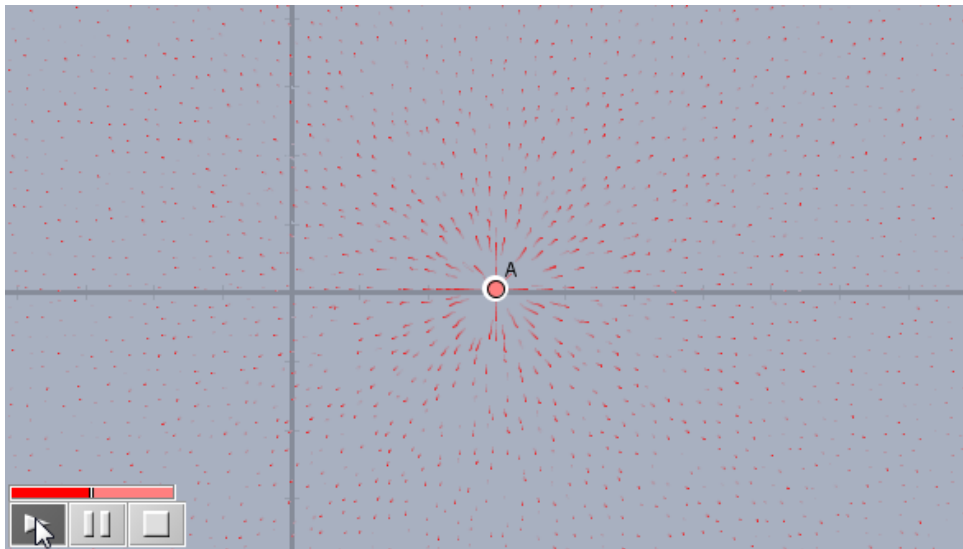


Abbildung 5.22.: Der drawfield-Operator

Befehl	Bedeutung
resolution->Zahl	Legt die Größe ein Feldes fest, in dem Nadeln zufällig verteilt werden.
stream->True oder False	Legt fest, ob das Feld durch bewegte Nadeln dargestellt werden soll.
jitter->Zahl	Legt fest, wie lang die Spur der Punkt ist.
needlesize->Zahl	Legt die Größe der Nadeln fest.
factor->Zahl	Bestimmt einen Skalierungsfaktor für das Feld.
move->Zahl	Legt die Geschwindigkeit der Punkte fest.
color->[Zahl,Zahl,Zahl]	Legt die Farbe der Punkte bzw. einer Seite der Nadeln fest.
color2->[Zahl,Zahl,Zahl]	Legt die zweite Farbe der Nadeln fest.

Tabelle 5.7.: Optionen für den drawfield-Operator

Der `drawforces`-Operator stellt die vorherrschenden Kräfte innerhalb der Simulation dar. Hierzu muss allerdings in den globalen Einstellungen „Anziehung zwischen Massen“ oder „Ladungen erzeugen Kräfte“ aktiviert sein. Die Optionen zur Einstellung des `drawforces`-Befehls sind die Gleichen wie bei dem `drawfield`-Operator.

Beispiel 5.23:

Es werden zwei Massenpunkte erstellt. In den Optionen wird für jeden Massenpunkt eine Ladung sowie die Einstellung „fixiert“ festgelegt. Außerdem wird in den globalen Einstellungen „Ladungen erzeugen Kräfte“ aktiviert. Anschließend wird im Scriptfenster der `drawforces`-Befehl ohne Optionen eingegeben.

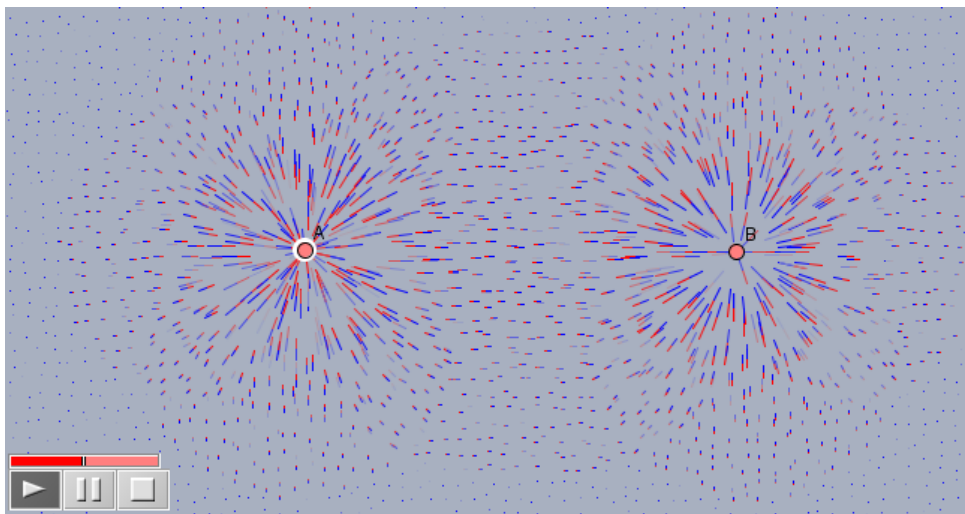


Abbildung 5.23.: Der `drawforces`-Befehl

Der `drawcurves`-Operator ermöglicht es, während der laufenden Simulation Größen in Form von Graphen darzustellen. Die Befehlszeile `drawcurves([Punkt],Zeichenobjekt,Option);` bewirkt, dass an der Stelle `[Punkt]` ein Fenster entsteht, in dem das `Zeichenobjekt` mit den gewünschten `Optionen` gezeichnet wird. Die Optionen werden in Tabelle 5.8 vorgestellt. Es können auch mehrere Zeichenobjekte dargestellt werden. Diese müssen dann in dem Befehl in eckigen Klammern geschrieben werden.

Beispiel 5.24:

Es wird ein Federpendel konstruiert, welches sich in einem homogenen Gravitationsfeld befindet. Reibung wird in diesem Fall vernachlässigt, um eine saubere Sinuskurve zu erhalten. Mit CindyScript werden die Graphen des y -Werts, der kinetischen und der potentiellen Energie des Körpers gezeichnet.

Befehl	Bedeutung
width->Zahl	Legt die Breite des Fensters fest.
height->Zahl	Legt die Höhe des Fensters fest.
border->>true oder false	Legt fest, ob das Fenster einen Rand besitzt.
back->>true oder false	Legt fest, ob das Fenster einen Hintergrund besitzt.
back->[Zahl,Zahl,Zahl]	Bestimmt die Farbe des Hintergrunds in RGB-Form.
backalpha->Zahl	Legt die Durchlässigkeit des Hintergrunds fest.
colors->[Liste]	Legt die Farbe, in RGB-Form, des jeweiligen Zeichenobjekts fest.
texts->[„Text1“, „Text2“, ...]	Legt die Beschriftung der Graphen fest.
showrange->>true oder false	Zeigt den maximalen und minimalen Ausschlag des Graphen an.
range->„peek“ oder „auto“	Setzt die Zeichenhöhe der Graphen entweder auf die Maximalwerte oder automatisch fest.
range->[Liste]	Gibt die Zeichenhöhe für jeden Graphen individuell vor.

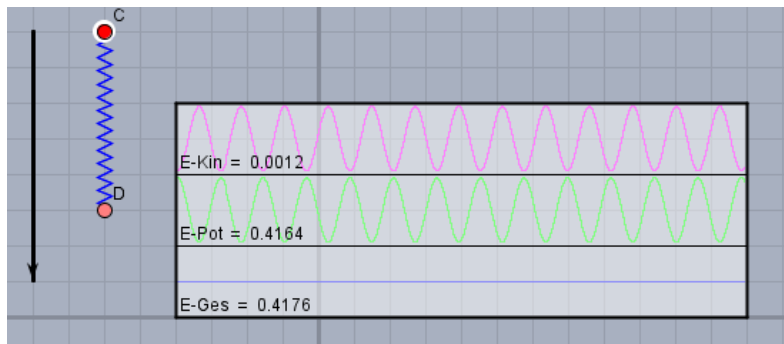
Tabelle 5.8.: Optionen für den *drawcurves*-Operator

Abbildung 5.24.: Beispiel von Graphen innerhalb einer Simulation

```

linecolor([0,0,0]);textcolor([0,0,0]);
drawcurves([0,0],[D.ke,simulation().pe,D.ke+simulation().pe],
height->50,back->[1,1,1],backalpha->0.5, range->["auto","auto"],
width->400, colors->[[1,0.5,1],[0.5,1,0.5],[0.5,0.5,1]],
texts->["E-Kin = "+D.ke, "E-Pot = "+simulation().pe,
"E-Ges = "+(D.ke+simulation().pe)]);

```

6. Ausgewählte Physiksimulationen

In diesem Kapitel werden mit Cinderella 2.0 erstellte Simulationen vorgestellt. Diese sind bis auf drei gekennzeichnete Ausnahmen selbst erstellt worden. Dabei wird insbesondere auf die Probleme, Möglichkeiten und Konstruktionsbeschreibungen eingegangen, die sich bei der Arbeit mit Cinderella 2.0 ergeben. Vor jeder Simulationsbeschreibung wird Bezug auf den Lehrplan genommen und auf didaktische Überlegungen eingegangen. Nach jeder Beschreibung werden Vor- und Nachteile der Simulationen diskutiert. Am Ende dieses Kapitels werden allgemeine Vor- und Nachteile von Cinderella 2.0 diskutiert.

6.1. Die Wellenmaschine

Im Lehrplan des achtjährigen Gymnasiums in Bayern wird das Thema Wellenlehre in der 10. Klassenstufe behandelt: „*Die Schüler erarbeiten sich grundlegende Kenntnisse über Wellen und deren Ausbreitung.*“¹ Dabei sollen die Schüler sowohl Transversal- als auch Longitudinalwellen kennenlernen. In der 11. Klassenstufe wird das Thema wieder aufgegriffen. Es werden elektromagnetische Wellen behandelt, insbesondere das Phänomen der stehenden Wellen.

Die Wellenlehre wird im Schulalltag häufig durch die Veranschaulichung an einer mechanischen Wellenmaschine eingeführt. Diese Wellenmaschinen sind allerdings sehr kostspielig in der Anschaffung. Außerdem sind sie aufgrund zahlreicher aneinandergereihter Oszillatoren fehleranfällig und der Ablauf der beobachteten Wellen kann nicht in seiner Geschwindigkeit gesteuert werden. Einige der Wellenmaschinen ermöglichen zwar das Festhalten eines bestimmten Bewegungszustandes der Welle, allerdings ist es nicht möglich, die beobachtete Welle danach weiterlaufen zu lassen. Ein Vorteil der Simulation einer Wellenmaschine ist es, einen Bewegungszustand festzuhalten bzw. in der Geschwindigkeit zu regeln und danach die Simulation fortsetzen zu können. Es werden drei Typen von Simulationen für Wellenmaschinen dargestellt: Die Simulation einer einfachen Transversalwelle mit festem sowie losem Ende, eine Simulation speziell für stehende Transversalwellen und eine für Longitudinalwellen.

6.1.1. Die Wellenmaschine für Transversalwellen

Das Cinderella-Applet „Wellenmaschine“ für transversale Wellen wurde von Jürgen RICHTER-GEBERT entworfen.² Es stellt eine Wellenmaschine dar, die durch eine Sinusschwingung eine

¹STAATSMINISTERIUM FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrplan Physik 10.* (URL: <http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=26439>).

²Quelle: <http://blog.cinderella.de/archives/75-WaveMachine.cdy.html>.

Transversalwelle anregt. Es ist möglich, die Reflexion am losen und am festen Ende zu simulieren. Hierzu wurde ein Knopf eingefügt, der das Ende fixiert. Die Frequenz der Anregung wird durch einen Schieberegler festgelegt. Die Konstruktion eines Applets kann leider nicht Schritt

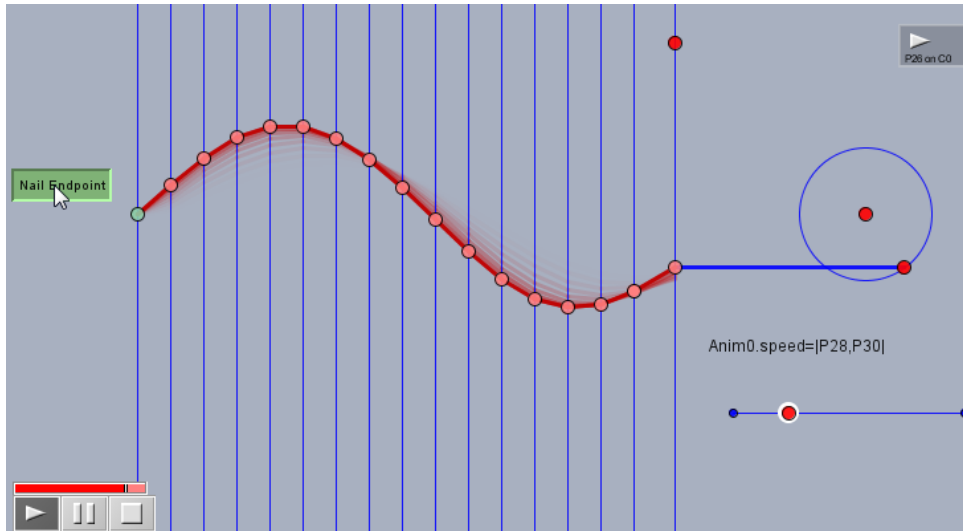


Abbildung 6.1.: Wellenmaschine mit festem Ende

für Schritt nachvollzogen werden, da es nicht möglich ist exportierte Applets mit Cinderella 2.0 zu öffnen. Es soll hier aber dennoch die Darstellung einer möglichen Konstruktion stattfinden, um es dem Leser zu ermöglichen, eigene ähnliche Konstruktionen leichter erstellen zu können.

Wellenmaschine:

1. Erstelle eine vertikale Gerade.
2. Erstelle äquidistante parallele Geraden.
3. Setze eine Masse auf die zuerst erstellte vertikale Gerade.
4. Erstelle auf jeder weiteren Geraden einen Massenpunkt.
5. Verbinde alle Massenpunkte mit Federn der Ruhelänge null.
6. Zeichne eine Strecke zwischen dem Punkt auf dem Kreis und dem Massenpunkt der ersten vertikalen Geraden.

Frequenzanreger:

1. Erstelle einen Kreis um einen Punkt.
2. Erstelle einen Punkt auf dem Kreis.
3. Lege eine automatische Animation des Punktes auf dem Kreis fest.

Schieberegler:

1. Zeichne eine Strecke unter dem Kreis.
2. Setze einen Punkt auf diese Strecke.

Knopf:

1. Erstelle einen Text zum Fixieren des Endpunktes.
2. Im Informationsfenster des Textes in dem Tab „*Info*“ einen Haken bei „*als Knopf benutzen*“ setzen.

Die weiteren Eingaben müssen im Scriptfenster von Cindyscript eingegeben werden. Die Befehle können im Unterordner „*Zeichnen*“ eingegeben werden. Die Simulation berechnet alle Eingaben unter „*Zeichnen*“ jeweils neu, wenn die Simulationsumgebung neu gezeichnet werden muss, mithin wenn sich etwas in der Konstruktion bewegt. Es werden drei Eingaben gemacht:³

1. `if(Text0.pressed, moveto(P3, [-1,0]))`; bewirkt, dass der Punkt P3 an die Koordinaten (-1,0) verschoben wird, sobald der Knopf zum Fixieren des Endpunktes angeklickt wird. Er bleibt dort solange fixiert, bis der Knopf ein zweites Mal geklickt wird.
2. `Anim0.speed=|P4,P6|`; legt die Animationsgeschwindigkeit mit Hilfe des Schiebereglers fest. Dabei wird der Abstand des Punktes auf der Geraden zum linken Ende der Geraden gleichgesetzt mit der Animationsgeschwindigkeit.
3. `moveto(R, [11,B.y])`; bewirkt, dass der erste Massenpunkt immer die gleiche Ordinate hat, wie der Punkt auf dem Kreis, der eine Sinusschwingung erzeugt.

In der obigen Konstruktionsbeschreibung sind nur die grundlegenden Schritte erklärt. Um Beschriftungen, Größen oder Farben einzustellen, kann entweder das Informationsfenster des einzelnen Objekts oder der Scripteditor genutzt werden.

Ein positiver Aspekt dieser Simulation ist, dass es keine Überlagerungen durch ungewollte Reflexionen gibt. Ebenso positiv hervorzuheben ist die Darstellung einer Spur der Federn, wodurch eine Bewegung sogar auf einem Screenshot deutlich wird. Ein weiterer klarer Vorteil ist, dass man sowohl das lose als auch das feste Ende in einer Simulation zur Verfügung hat und so die Ablaufgeschwindigkeit frei regeln kann. Im Umgang mit der Simulation ist negativ aufgefallen, dass zwar periodisch erzeugte Wellen entstehen können, man aber nur schwer einzelne Wellenfronten aussenden kann. Da während der Simulation der erste Oszillator die vertikalen Bewegungen der Kreisbewegung ausübt und an diese gebunden ist, kann dieser Punkt nicht manuell ausgelenkt werden. Möchte man eine einzelne Wellenfront aussenden, sollte man die Erregerfrequenz mit dem Schieberegler ganz abstellen. Nun kann man den zweiten Oszillator manuell auslenken und eine einzelne Wellenfront erzeugen. Außerdem ist es nicht möglich in diesem Applet die Kugelmasse bzw. die Federhärte zu verstellen. Wird die Simulation allerdings selbstständig erzeugt, ist es möglich die Masse bzw. die Federhärte, entweder direkt mit CindyScript einzustellen oder

³Die Bezeichnungen und die Koordinaten können in eigenen Konstruktionen abweichen und müssen dementsprechend geändert werden.

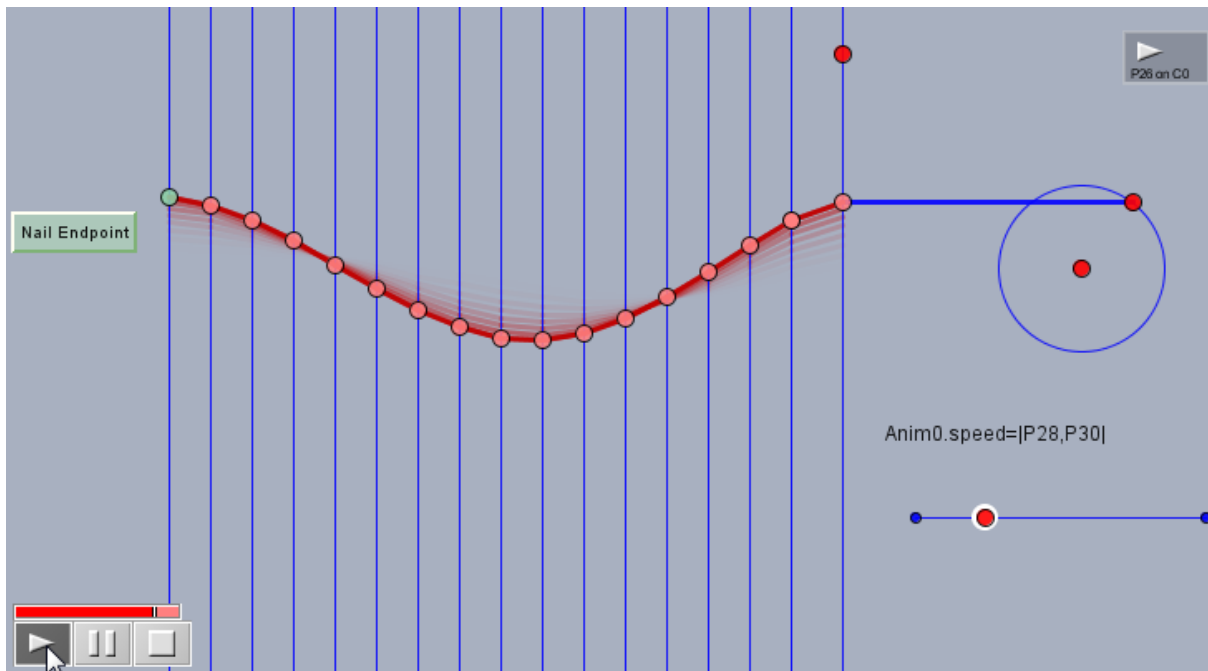


Abbildung 6.2.: Wellenmaschine mit loseem Ende

indirekt, indem ein Schieberegler eingebaut wird, dem diese Einstellungen zugeordnet werden. Für den Schüler verwirrend könnte die Tatsache sein, dass in diesem Applet die Wellenbewegung von rechts nach links abläuft. Eine solche Simulation wird leichter verstanden, wenn die Bewegung in der Leserichtung von links nach rechts abläuft. Wünschenswert wäre es außerdem, die Auslenkungsgeschwindigkeit der Kugelmassen durch Vektoren zu visualisieren. Dies ist leider mit Cinderella 2.0 nur bedingt möglich. Es wird aber unter 6.2 eine Möglichkeit dargestellt, solche Geschwindigkeitsvektoren nachträglich einzufügen. Es entsteht allerdings ein Problem, wenn viele dieser Vektoren eingeführt werden sollen. Zum einen wird der Rechenaufwand und damit die Simulationsgeschwindigkeit schon bei einigen Kugelmassen so anspruchsvoll, dass die Simulation merkbar langsamer und stockender abläuft. Zum anderen ist die Erstellung solcher Vektorpfeile ebenso aufwändig in der Berechnung wie das Hinzufügen neuer Massenpunkte. Deshalb wird empfohlen, nur mit Computern, die eine hohe Rechenleistung erbringen, eine solche aufwändige Simulation zu erstellen.

6.1.2. Wellenmaschine für stehende Wellen

Auf Basis der Simulation „Wellenmaschine“ wurde eine Cinderellakonstruktion entworfen, in der auf beiden Seiten der Oszillatoren Frequenzanreger eingebaut wurden. Hierdurch wird es enorm erleichtert, eine stehende Welle zu erzeugen. Die Konstruktionsbeschreibung würde wie folgt aussehen:

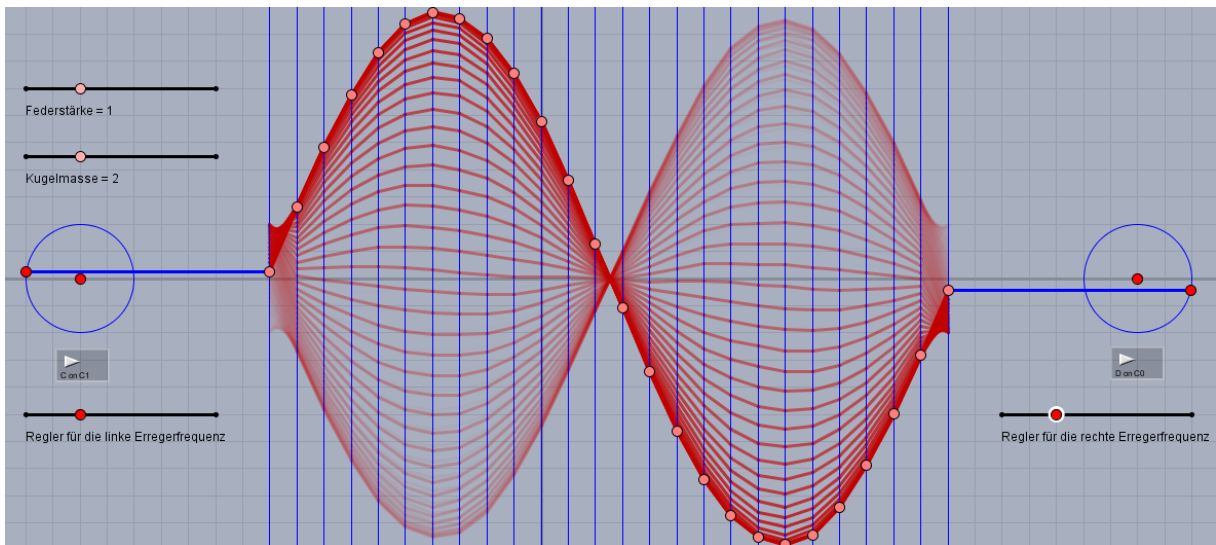


Abbildung 6.3.: Stehende Welle

Oszillatoren:

1. Erstelle eine vertikale Gerade.
2. Erstelle äquidistante parallele Geraden.
3. Setze eine Masse auf die zuerst erstellte vertikale Gerade.
4. Erstelle auf jeder weiteren Geraden einen Massenpunkt.
5. Verbinde alle Massenpunkte mit Federn der Ruhelänge null.

Frequenzanreger:

1. Erstelle einen Kreis um einen Punkt.
2. Erstelle einen Punkt auf dem Kreis.
3. Lege eine automatische Animation von dem Punkt auf dem Kreis fest.
4. Zeichne eine Strecke zwischen dem Punkt auf dem Kreis und dem Massenpunkt der ersten vertikalen Geraden.

Schieberegler:

1. Zeichne eine Strecke.
2. Setze einen Punkt auf diese Strecke.

Die restlichen Einstellungen wurden mit CindyScript festgelegt. Es wurden die gleichen Befehle wie bei 2. und 3. der vorherigen Simulation genutzt, um die Geschwindigkeit der Erregerfrequenzen zu steuern und die Endpunkte an diese Frequenzen zu binden. Zusätzlich wurde zum besseren Verständnis unter dem Schieberegler eine Erklärung in Form eines Textes angebracht. Diese Erklärungen wurden mit dem `drawtext`-Operator 5.4.2 eingezeichnet. Zusätzlich wurden noch zwei Schieberegler eingebaut, einer, der die Kugelmasse festlegt und ein anderer, der die

Federhärte einstellt. Die Eingabe lautet:

```
a:=[P13,...,P38];
forall(a,#.mass=|P39,P41|);
drawtext([-19,3.5],"Kugelmasse = "+ P13.mass);
b:=[S0,...,S25];
forall(b,#.strength=|P42,P44|/2);
drawtext([-19,6], "Federstärke = "+ S1.strength);
```

Zuerst wird eine Liste **a** definiert, in der alle Kugelmassen stehen. Die drei Punkte müssen im Programmcode mit den jeweiligen Namen der Massenpunkte ersetzt werden. Anschließend wird mit dem **forall**-Operator jedem Element der Liste eine Masse als Abstand des Schiebereglers zugeordnet. Der Befehl **drawtext** wird unter dem Schieberegler angezeigt und gibt die aktuelle Masse der Körper an. Die Einstellungen der Federstärke erfolgen analog.

Bei der Konstruktion der Simulation sind einige Probleme aufgetreten. Es wurde versucht, eine Ausgabe für den Phasenverschiebungswinkel ϕ einzufügen, allerdings kann Cinderella 2.0 nur Winkel ausgeben, die als Schnittwinkel zweier Geraden definiert werden. Dies hatte zur Folge, dass der Winkel nicht größer als π werden konnte. Außerdem war keine Kreisbewegung des Punktes mehr möglich, da die Animation des Punktes auch nach π wieder zurück auf den Startpunkt gesprungen ist. Für eine Phasenverschiebung der Erreger innerhalb der Simulation bedeutet dies, dass keine genauen Werte für den Phasenverschiebungswinkel angegeben werden können. Es ist aber möglich, die Winkel $\frac{\pi}{2}$, π , $\frac{3\pi}{2}$ und $2 \cdot \pi$ einzustellen, indem man die Option „*automatisch an Koordinaten heften*“ aktiviert und den Punkt verschiebt. Voreingestellt in der Simulation ist ein Phasenverschiebungswinkel von π , da hiermit die besten Ergebnisse für stehende Wellen erzeugt werden konnten.

Zur besseren Darstellung der Simulation sollten die Bewegungen durch Spuren (Schlieren) verdeutlicht werden. Bei der Konstruktion misslang es jedoch, die Federbewegungen durch eine Spur darzustellen. Es sollte durch einen Knopf die Möglichkeit bestehen, diese Spur an- bzw. auszuschalten. Leider gibt es keinen Befehl bei CindyScript, der eine solche Funktion ermöglicht. Möchte man also, dass die Federn während der Bewegung eine Spur hinterlassen, müssen alle Federn ausgewählt und in ihrem Informationsfenster die Spur aktiviert werden.

Leider ist es auch in dieser Konstruktion nur über den in 6.1.1 beschriebenen Umweg möglich, eine Auslenkung mit der Maus durchzuführen. Prinzipiell kann diese Simulation auch als einfache Wellenmaschine mit festem Ende genutzt werden, indem eine der Erregerfrequenzen komplett abgestellt wird. Somit bekommt man einen fixierten Körper und kann eine Reflexion am festen Ende darstellen. In der rechten oberen Ecke sind zusätzlich zwei Animationsknöpfe, ob die Punkte sich überhaupt auf dem Kreis bewegen sollen. Hierdurch kann man auch ein Ende fixieren. Diese Animationsknöpfe sollten direkt unter den Kreisen stehen, allerdings ändert Cinderella 2.0 beim Neustart den Ort der Knöpfe wieder auf die rechte obere Ecke.

6.1.3. Wellenmaschine für longitudinale Wellen

Es wurden einige Möglichkeiten getestet, Longitudinalwellen mit Cinderella 2.0 darzustellen. Alle basierten auf der Idee, dass ein Massenpunkt einen anderen durch eine $\frac{1}{r}$ bzw. $\frac{1}{r^2}$ abhängige Kraft von sich wegstoßen sollte. Cinderella 2.0 bietet hierzu drei Möglichkeiten an:

1. Die Darstellung von Federn mit Ruhelänge zwischen den einzelnen Massenpunkten.

Der Vorteil dieser Versuchsanordnung liegt darin, dass es beim Start der Simulation nicht ohne äußere Einwirkung zu einer Bewegung der Massenpunkte kommt. Man kann eine einzelne Wellenfront gut beobachten. Der Nachteil ist allerdings, dass jede Feder, die auseinandergezogen wird, eine rückstellende Kraft ausübt, um in den Zustand ihrer Ruhelänge zurückzukehren.

2. Die Darstellung mit Federn ohne Ruhelänge zwischen den einzelnen Massenpunkten.

Dieser Versuchsaufbau funktioniert nur mit zwei festen Enden, da sonst alle Massenpunkte auf einen Punkt hin gezogen werden. Hier tritt zudem das Problem auf, dass eine Auslenkung eines Massenpunktes auch eine rückstellende Kraft nach sich zieht, wodurch es zu ungewollten Eigenschwingungen kommt.

3. Die Darstellung durch gleiche Ladungen, die sich gegenseitig abstoßen.

In diesem Versuchsaufbau tritt das Problem auf, dass alle Ladungen Kräfte aufeinander wirken, wodurch sie sich auseinander bewegen, soweit es keine festen Eckpunkte gibt.

Will man eine Bewegung ohne festes Ende darstellen, wird empfohlen, die erste Variante zu benutzen. Bei einer Simulation mit festem Ende wird empfohlen, die dritte Variante zu nutzen, da hiermit die besten Ergebnisse erzielt worden sind. Da alle Konstruktionen aber analog ablaufen, wird nur die erste Variante ausführlich dargestellt:

Oszillatoren:

1. Zeichne eine horizontale Gerade.
2. Setze Massenpunkte in gleichen Abständen, beginnend über dem Punkt auf dem Kreis, auf diese Gerade.
3. Verbinde alle Massenpunkte mit Federn.

Frequenzanreger:

1. Erstelle einen Kreis.
2. Setze einen Punkt auf diesen Kreis.
3. Lege eine Animation des Punktes auf dem Kreis fest.

Schieberegler:

1. Erstelle eine Strecke unter dem Kreis.
2. Setze einen Punkt auf diese Strecke.

Knopf:

1. Erstelle einen Text mit der Eingabe „Endpunkt fixieren“.
2. Im Informationsfenster des Textes in dem Tab „Info“ einen Haken bei „als Knopf benutzen“ setzen.

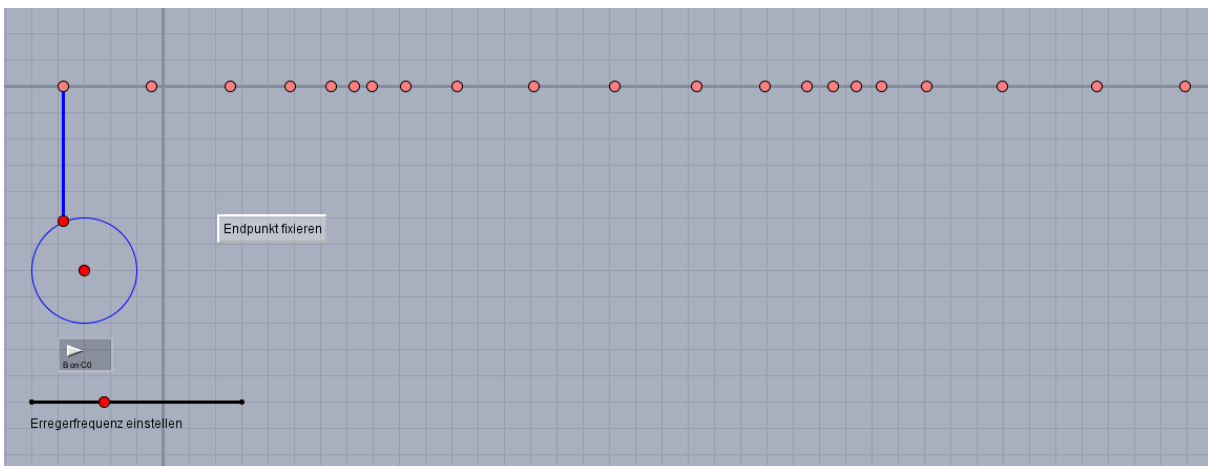


Abbildung 6.4.: Eine Longitudinalwelle

Es werden folgende Einstellungen mit CindyScript festgelegt.

1. `if(Text0.pressed, moveto(P0,[39,0]))`; Bei gedrückter Taste wird der Endpunkt fixiert.
2. `moveto(D,[B.x,0])`; Bewegt den Punkt über dem Kreis.
3. `Anim0.speed=|P1,P3|`; Die Animationsgeschwindigkeit des Punktes auf dem Kreis wird durch den Schieberegler bestimmt.

Im Umgang mit der Simulation wird deutlich, dass sie in dieser Form besser nicht im Unterricht eingesetzt werden sollte. Es ist nämlich nicht möglich gewesen, mit den Elementen aus CindyLab und der Hilfe von CindyScript eine Simulation zu konstruieren, die mehrere Wellenfronten ohne Komplikationen aussendet. Sicherlich ist es möglich, nur mit CindyScript ein Programm zu schreiben, welches Longitudinalwellen angemessen darstellen kann. Solche Programme existieren aber bereits und sind im Internet auf entsprechenden Websites zu finden.⁴

⁴z. B. <http://www.schulphysik.de/java/physlet/applets/welle01.html>.

6.2. Elastische Stoßvorgänge

Der Lehrplan der 10. Klassenstufe sieht im Rahmen der newtonschen Mechanik vor, dass die Schüler die Gesetzmäßigkeiten der Impuls- bzw. Energieerhaltung kennenlernen. Diese Erhaltungssätze sollen auf einfache Beispiele angewandt werden können. Der Lehrplan sieht explizit vor, dass die Schüler nur eine eindimensionale Vorstellung bei der Impulserhaltung bekommen. Trotzdem wird hier eine Simulation für zweidimensionale Stöße vorgestellt, um dem Lehrer die Möglichkeit zu geben das Gesetz der Impulserhaltung zweidimensional darzustellen.

Im Physikunterricht werden Stöße oft durch die Nutzung einer Luftkissenbahn oder eines Luftkissentisches eingeführt. Leider sind die Luftkissentische in Schulen nicht so häufig vorhanden wie die Luftkissenbahnen. Diese Luftkissentische ermöglichen die Beobachtung von zweidimensionalen Stößen, während mit Luftkissenbahnen nur eindimensionale Stöße dargestellt werden können. Mit Hilfe von Cinderella 2.0 wird ein solcher Luftkissentisch simuliert und zum besseren Verständnis werden zusätzlich Geschwindigkeitsvektoren angezeigt. Für Schulen, die nur mit einer Luftkissenbahn ausgestattet sind, ermöglichen die Simulationen also ergänzend die Darstellung zweidimensionaler Stöße und dienen somit als Ersatz für den Luftkissentisch.

An dieser Stelle werden zwei Simulationen vorgestellt. In der ersten soll nur das Phänomen des elastischen Stoßes veranschaulicht werden, während in der zweiten Simulation der Impulserhaltungssatz aufgezeigt werden soll.

6.2.1. Der elastische Stoß

Diese Simulation stellt eine Art Baukasten dar, mit Hilfe dessen die Stoßvorgänge kreisförmiger Körper demonstriert werden können. Man kann die Körper mit der Maus in den farbigen dargestellten Kasten verschieben, um sie in die Simulation zu integrieren. Die Simulationseinstellungen wirken sich nur auf Körper innerhalb dieses Kastens aus. Man hat zwei Körper mit einer Anfangsgeschwindigkeit zur Auswahl. Diese Anfangsgeschwindigkeit kann in ihrer Richtung und Stärke durch ein Verschieben der Vektorspitze verstellt werden. Den anderen beiden Körpern wurden durch geeignete Befehle in CindyScript auch Geschwindigkeitsvektoren zugeordnet. Diese werden sichtbar sobald diese durch einen Stoß eine Geschwindigkeit erhalten. Es wurden für jeden Körper zwei Schieberegler eingebaut, wobei einer die Masse, der andere die Größe des Körpers reguliert. Die folgende Konstruktionsbeschreibung wird nicht für jede Option einzeln dargestellt. Es wird für jeweils einen Körper die Erstellung der Schieberegler und der Geschwindigkeitsvektoren erklärt.

Schieberegler:

1. Zeichne eine Strecke.
2. Setze einen Punkt auf diese Strecke.

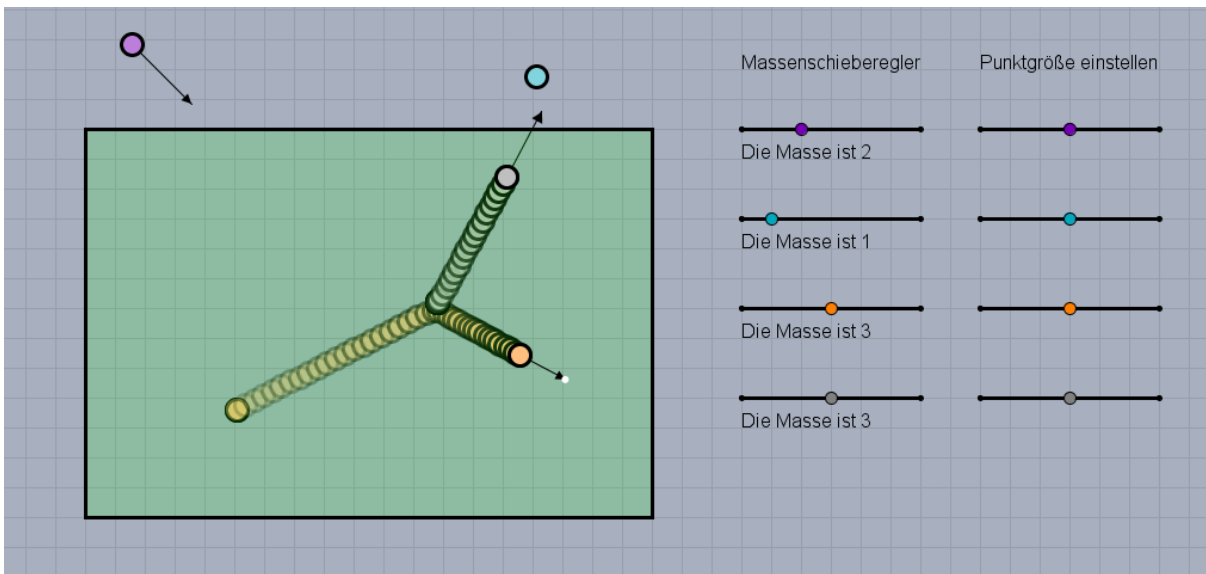


Abbildung 6.5.: Beispiel eines elastischen Stoßes

Lufkissentisch und Kugelmassen:

1. Konstruiere ein Viereck aus Wänden.
2. Erstelle Massenpunkte.
3. Ordne mindestens einem Massenpunkt einen Geschwindigkeitsvektor über den Button „Masse mit Geschwindigkeit“ zu.

Alle nachfolgenden Beschreibungen bzw. alle verwendeten Namen für Punkte sind der oberen Simulation entnommen. In eigenen Simulationen können die Namen und Koordinaten abweichen. Die CindyScript-Befehle lauten:

1. `E.mass=|N,Y|`; legt die Masse als Abstand zwischen dem linken Punkt eines Schiebereglers und dem verschiebbaren Punkt fest.
2. `E.size=3*|W,P7|`; legt die Größe als Abstand zwischen dem linken Punkt eines Schiebereglers und dem verschiebbaren Punkt fest. Der Faktor 3 ist eingeführt worden, um den Schieberegler nicht unnötig lang gestalten zu müssen.
3. `if(E.y<5,if(-8<E.y,if(-6<E.x, if(E.x<13,E.simulate=true))),E.simulate=false)`; legt fest, dass nur Punkte innerhalb des konstruierten Vierecks von der Simulation berücksichtigt werden.
4. `drawtext([16,1],"Die Masse ist "+ E.mass)`; erzeugt unter dem jeweiligen Schieberegler eine Ausgabe über die Masse des Körpers.

5. `drawtext([24,7], "Punktgröße einstellen", size->16)`; erzeugt eine Überschrift für eine Spalte der Schieberegler.
6. `drawtext([16,7], "Massenschieberegler", size->16)`; erzeugt eine Überschrift für die andere Spalte der Schieberegler.

Die Erstellung eines Vektorpfeils in einer Simulation ist komplexer und wird an dieser Stelle gesondert erklärt, damit an späterer Stelle hierauf verwiesen werden kann.

```
SpitzeG:=[G.x+G.vx,G.y+G.vy];
SpitzeG2:=SpitzeG-0.35*G.v/|G,SpitzeG|;
SeitenpktG1:=SpitzeG2+0.2*[G.vy,-G.vx]/|G,SpitzeG|;
SeitenpktG2:=SpitzeG2-0.2*[G.vy,-G.vx]/|G,SpitzeG|;
draw(G,SpitzeG,color->(0,0,0));
drawpoly([SpitzeG,SeitenpktG1,SeitenpktG2],color->(0,0,0));
```

Alle G s in diesem Script sind spezifisch für den Punkt G , für andere Punkte muss das G mit dem Namen des neuen Punktes entsprechend ersetzt werden. Die Idee dieses Scripts ist es, einen dynamisch visualisierten Vektorpfeil zu definieren, indem eine Strecke vom Punkt selbst zu der Spitze des Vektors gezeichnet wird. Die Pfeilspitze wird dadurch konstruiert, dass in einem festen Abstand von der Spitze ein Dreieck gezeichnet wird. Dieses Dreieck bewegt sich mit der Spitze mit und verändert seine Größe nicht.

In der ersten Zeile wird die Spitze des Geschwindigkeitsvektors definiert, indem zum Punkt G die Geschwindigkeitskomponenten addiert werden. Die zweite Zeile definiert den Mittelpunkt der Grundseite des Dreiecks. Es wird ein Teil von der Spitze abgezogen. Diese Strecke hat immer die gleiche Länge, da der Vektor von G zur `SpitzeG` normiert wurde, ehe er abgezogen wird. Die dritte und vierte Zeile definieren die beiden Eckpunkte der Grundseite des Dreiecks. Hierzu wird ein Teil des normierten Vektors, der orthogonal zu dem Vektor von G nach `SpitzeG` ist, subtrahiert bzw. addiert. Die Zeilen fünf und sechs zeichnen den visualisierten Vektorpfeil.

Es ist mit Cinderella 2.0 zwar grundsätzlich möglich, Geschwindigkeitsvektoren darzustellen, allerdings müssen diese vor dem Start der Simulation festgelegt werden. Soll ein Körper sich am Anfang der Simulation in Ruhe befinden, kann kein Geschwindigkeitsvektor festgelegt werden, da er sich sonst bewegen würde. Wird der ruhende Körper durch einen Stoß in Bewegung versetzt, ist es nicht möglich, durch die in Cinderella 2.0 integrierten Mechanismen einen Vektorpfeil zeichnen zu lassen. Deshalb ist die Konstruktion eines solchen Vektorpfeils nötig, da Cinderella 2.0 es zwar ermöglicht, die Geschwindigkeit bzw. die Kraft eines Körpers auszulesen, dagegen nicht, diese Geschwindigkeit bzw. die zur Kraft gehörige Beschleunigung durch einen Vektorpfeil darstellen zu lassen.

In der Durchführung ist aufgefallen, dass Cinderella 2.0 alle Kugelmassen, egal wie groß ihre Visualisierungen sind, als Punktmassen berechnet, wodurch es so aussieht, als wenn zwei Ku-

gelmassen die einen Stoß vollführen sich erst überlappen und sich dann wieder auseinander bewegen.

Es wäre wünschenswert für jedes Teilchen eigenständig eine Spur zu aktivieren bzw. zu deaktivieren. Wird die Simulation mit Cinderella 2.0 geöffnet ist dies kein Problem, da über das *Informationsfenster* jeder Kugelmasse individuell eine Spur zugewiesen werden kann. Wird die Simulation allerdings in einem Browser als Applet genutzt ist diese Möglichkeit nicht gegeben. Eine ständige Spur kann aber vor allem bei schnelleren Bewegungen zu Verwirrung führen, deshalb wurde darauf verzichtet standardmäßig eine Spur eingestellt zu haben.

Diese Simulation kann von Schülern genutzt werden, um sich mit dem Phänomen des elastischen Stoßes im zweidimensionalen vertraut zu machen und ein Gefühl dafür zu bekommen, wie sich Körper verhalten, wenn ein Impuls auf sie ausgeübt wird. Da der Lehrplan nur den eindimensionalen Stoß vorsieht, kann eine Simulation auch analog zu dieser erstellt werden, bei der die Kugelmassen auf einer Geraden erstellt werden.

6.2.2. Impulserhaltung

Die Basis dieser Simulation ist die gleiche, wie in der zuvor beschriebenen. Allerdings werden hier nur zwei Kugelmassen betrachtet. Zusätzlich wurde noch eine Ausgabe erzeugt, welche die Impulse der beiden Massenpunkte sowie den Gesamtimpuls ausgibt.

```
drawtext([0,-8], "Der Impuls der roten Masse ist: p="+E.mass*|E.v|);  
drawtext([0,-9], "Der Impuls der grünen Masse ist: p="+F.mass*|F.v|);  
drawtext([0,-11], "Der Gesamtimpuls ist,  
p="+|E.mass*E.vx+F.mass*F.vx, E.mass*E.vy+F.mass*F.vy|);
```

Cinderella 2.0 kann entweder die kinetische Energie eines Körpers oder die Geschwindigkeit auslesen. Da die Geschwindigkeit proportional zu dem Impuls ist, wurde die Geschwindigkeit ausgelesen und mit der Masse des Körpers multipliziert, um den Impuls zu erhalten. Der Gesamtimpuls wird als die Summe aller Impulse angegeben.

Das Problem bei dieser Simulation ist, dass der Algorithmus zur Berechnung der Geschwindigkeit bei zu schnellen Stoßvorgängen ungenau wird. Je schneller die Simulation abläuft, desto größer wird die Schrittweite zwischen den einzelnen Berechnungen. Findet nun ein Stoßvorgang statt, wird ein Körper beschleunigt und ein anderer abgebremst. Dieser Vorgang ist mit einem enormen Rechenaufwand verbunden, wodurch in diesem Moment die Genauigkeit gesenkt wird. In Abbildung 6.6 ist zu erkennen, dass sich hierdurch der Gesamtimpuls ändert. Es sollte in dieser Simulation aber der Impulserhaltungssatz demonstriert werden. Die Verwendung dieser Simulation ist also nur bedingt möglich. Man kann die Ausgabe der Nachkommastellen durch den Befehl `format(Ausdruck,Nachkommastellen)`; auf beliebig viele Nachkommastellen reduzieren, z. B. `format(pi,2)`; gibt die Zahl π auf zwei Nachkommastellen aus. Die in der Simulation vorherrschende Rechenungenauigkeit bewirkt trotzdem, dass keine Impulserhaltung

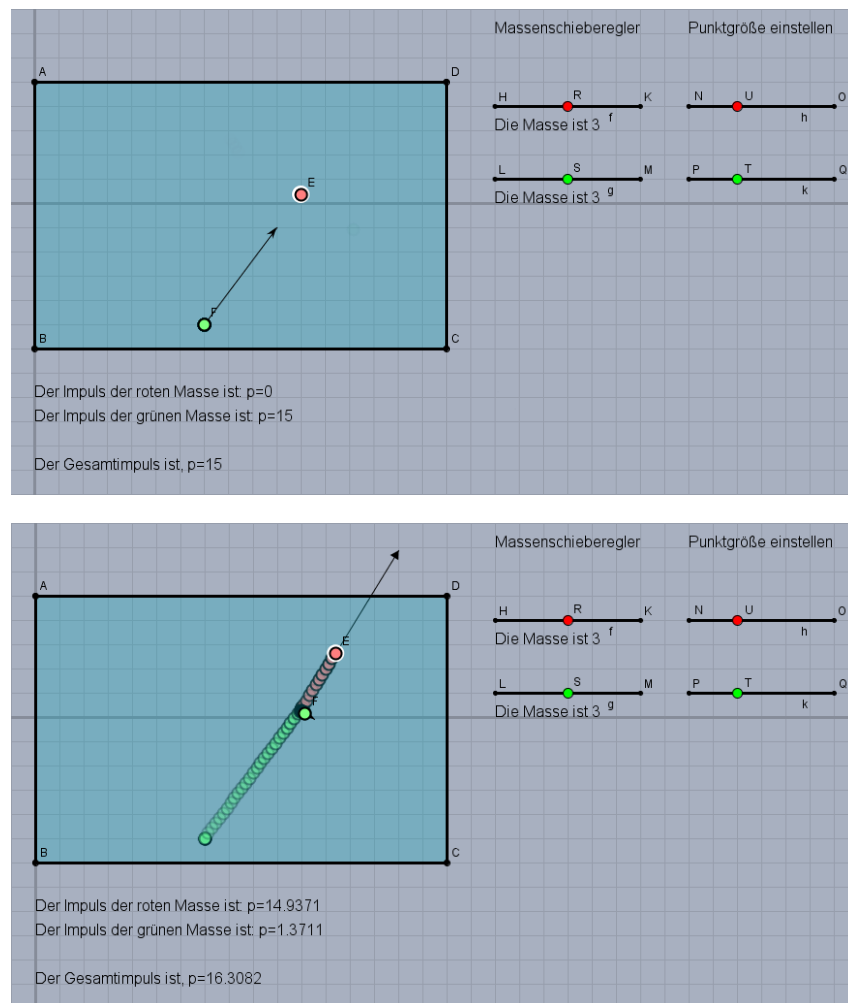


Abbildung 6.6.: Simulation zur Impulserhaltung

gilt und dies nach längerer Zeit ersichtlich wird. Positiv hervorzuheben ist aber die Visualisierung der Geschwindigkeitsänderung durch die Veränderung des Geschwindigkeitsvektors. Individuell erwünscht könnte eine genauere Beschreibung des Impulses in Form einer Formel sein. Hierzu könnte man anstatt direkt den Impuls auszugeben, zuerst die Formel angeben, danach die einzelnen Größen einsetzen und am Schluss die Ausgabe machen lassen, wie groß der Impuls ist.

6.3. Wurfbewegungen

Der Lehrplan der 10. Jahrgangsstufe sieht vor, den Schülern die newton'sche Mechanik zu vermitteln. Dabei lernen die Schüler, wie sie Bewegungen durch die Gesetze von Newton beschreiben und berechnen können. Aufbauend auf eindimensionalen Bewegungen können die zweidimensionalen Bewegungen anhand des waagerechten Wurfs eingeführt werden.

Hier werden zwei Simulationen vorgestellt, eine zum waagerechten und eine zum schrägen Wurf. Der schräge Wurf wird nicht explizit im Lehrplan erwähnt, kann aber von jedem Lehrer im

Rahmen seiner didaktischen Freiheit bei ausreichend vorhandener Zeit behandelt werden, denn „im Unterricht können und dürfen aus zeitökonomischen und didaktischen Gründen nicht alle Inhalte in gleicher Ausführlichkeit behandelt werden. Bei der Bildung von Schwerpunkten und der Auswahl der Vertiefungen ist darauf zu achten, dass die grundlegenden Ziele in ausreichendem Maße berücksichtigt werden.“⁵

Im Unterricht werden Wurfbewegungen oft in Realexperimenten gezeigt. Ein Problem dabei ist, dass diese Wurfbewegungen sehr schnell ablaufen und schlecht beobachtet werden können. Die Darstellung mit Simulationen bietet den Vorteil, dass bspw. die Flugbahn des Objekts angezeigt werden kann.

6.3.1. Der waagerechte Wurf

In dieser Simulation wird eine Kugelmasse waagrecht abgeschossen bzw. es wird ihr eine Geschwindigkeit in x-Richtung zugeordnet. Aufgrund eines homogenen Kraftfeldes wird sie nach unten beschleunigt. Es wurde ein Schieberegler eingebaut, der die Startgeschwindigkeit in x-Richtung bestimmt. Zusätzlich wurden vier Knöpfe integriert. Ein Knopf ermöglicht es, die Spur der Kugelmasse zu zeichnen, ein anderer erstellt Graphen von der Beschleunigung, der Geschwindigkeit und der Koordinate in y-Richtung. Durch die zwei anderen Knöpfe kann der Geschwindigkeitsvektor bzw. der Beschleunigungsvektor der Kugelmasse angezeigt werden. Die Konstruktion ist folgendermaßen aufgebaut:

Umgebung:

1. Zeichne einen Boden.
2. Erstelle eine waagerechte Wand auf Höhe des Abwurfs.
3. Erstelle eine senkrechte Wand vom Boden zur waagerechten Wand.
4. Erstelle an der Kante wo sich waagerechte und senkrechte Wand treffen eine Kugelmasse.

Schieberegler und Knöpfe:

1. Zeichne eine Strecke.
2. Erstelle einen Punkt auf der Strecke.
3. Erstelle jeweils ein Textfeld pro Knopf und beschrifte es entsprechend.
4. Aktiviere im *Informationsfenster* des Textfeldes „als Knopf benutzen“.

Die restlichen Eingaben werden mit CindyScript erstellt.

```
Ab1=|E,G|;
```

```
drawtext([E.x,E.y-1], "Die Abwurfgeschwindigkeit ist "+format(Ab1,1));
```

⁵STAATSINSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrpläne Gymnasium G8*, <http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=26439>.

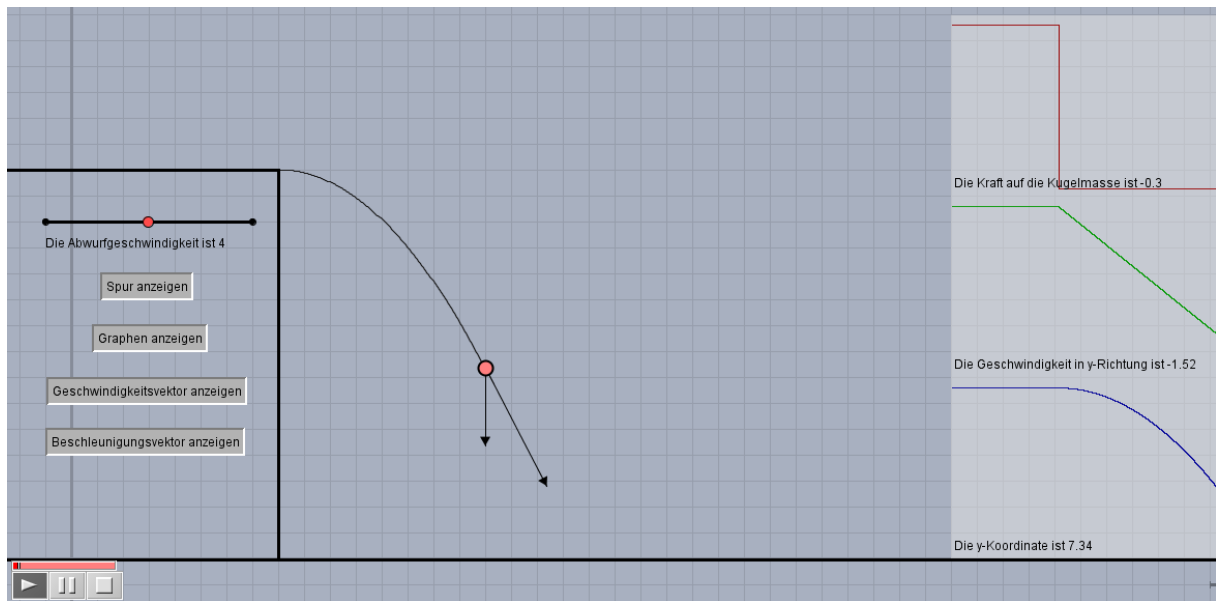


Abbildung 6.7.: Der waagerechte Wurf

Der Abstand des beweglichen Punktes auf dem Schieberegler wird mit `Ab1` bezeichnet. Unter dem Schieberegler wird eine Ausgabe erstellt, in welcher die Abwurfgeschwindigkeit auf eine Stelle nach dem Komma genau angegeben wird. Damit die Geschwindigkeit der Kugelmasse nach dem Start nicht gleich bleibt, wird die Geschwindigkeit in dem Unterordner `Simulationsstart` des CindyScript-Eingabefensters festgelegt. Somit wirkt sich eine Änderung des Geschwindigkeitschiebereglers erst nach einem Neustart der Simulation aus. Der Befehl, der die Geschwindigkeit zuordnet, lautet:

```
D.vx=Ab1/5;
```

Die Geschwindigkeit in x-Richtung wird definiert, indem ihr der Abstand des Schiebereglers zugeordnet wird. Dieser Abstand wird hier noch durch 5 dividiert, damit die Flugbahn der Kugelmasse nicht über das Simulationsfenster hinausgeht.

```
if(Text0.pressed,m=m++[D.xy];connect(m,color->(0.1,0.1,0.1)),m=[]);
```

Diese `if`-Schleife legt fest, dass eine Spur wie in 6.4.1 für die Kugelmasse gezeichnet werden soll, sobald der zugehörige Knopf gedrückt wird.

```
if(Text1.pressed,
drawcurves(H,[D.fy,D.v,D.y],height->175,width->300,
back->[0.9,0.9,0.9],border->false,range->"peek"););
```

Die `if`-Schleife für den zweiten Knopf besagt, dass Graphen gezeichnet werden sollen, sobald der Knopf gedrückt ist. Hier werden Graphen an den Punkt `H` angebunden, und es werden die Kraft in `y`-Richtung, die Geschwindigkeit und der `y`-Wert gezeichnet. Die Optionen besagen, dass jeder Graph eine Höhe von 175 und eine Breite von 300 hat. Die Hintergrundfarbe ist fast weiß, es werden keine Grenzen gezeichnet und der Zeichenbereich wird so gewählt, dass der maximale Ausschlag die Skalierung des Graphen bestimmt.

```
if(Text2.pressed,  
SpitzeD:=[D.x+3*D.vx,D.y+3*D.vy];  
SpitzeD2:=SpitzeD-D.v/|D,SpitzeD|;  
SeitenpktD1:=SpitzeD2+0.6*[D.vy,-D.vx]/|D,SpitzeD|;  
SeitenpktD2:=SpitzeD2-0.6*[D.vy,-D.vx]/|D,SpitzeD|;  
draw(D,SpitzeD,color->(0,0,0));  
drawpoly([SpitzeD,SeitenpktD1,SeitenpktD2],color->(0,0,0)););
```

Dem dritten Knopf wird bei der Betätigung die Zeichnung eines Geschwindigkeitsvektors zugeordnet, wie es in 6.2.1 beschrieben wurde. Analog wurde dem vierten Knopf ein Beschleunigungsvektor zugeordnet:

```
if(Text3.pressed,  
SpitzeDf:=[D.x+10*D.fx,D.y+10*D.fy];  
SpitzeDf2:=SpitzeDf-3.5*D.f/|D,SpitzeDf|;  
SeitenpktDf1:=SpitzeDf2+2*[D.fy,-D.fx]/|D,SpitzeDf|;  
SeitenpktDf2:=SpitzeDf2-2*[D.fy,-D.fx]/|D,SpitzeDf|;  
draw(D,SpitzeDf,color->(0,0,0));  
drawpoly([SpitzeDf,SeitenpktDf1,SeitenpktDf2],color->(0,0,0)););
```

Dieser Beschleunigungsvektor müsste korrekterweise als Kraftvektor angegeben werden, da man die auf die Kugelmasse wirkende Kraft nutzt, um diesen Vektor zu erstellen. Da aber die Kraft proportional zu der Beschleunigung ist, kann auch die Bezeichnung Beschleunigungsvektor verwendet werden. Die verschiedenen Faktoren in den Beschreibungen der Vektoren wurden individuell an diese Simulation angepasst, damit die Länge des Vektors angemessen erscheint.

In der Durchführung ist positiv aufgefallen, dass bei einer langsamen Ablaufgeschwindigkeit die Geschwindigkeitsänderung anhand des Vektors gut zu erkennen ist. Bei angeschalteter Spur lässt sich ebenfalls gut erkennen, dass die Bahn, die die Kugelmasse zurücklegt, eine Parabelbahn ist. Leider tritt auch hier wieder das Problem auf, dass bei einem Neustart der Simulation die Kugelmasse springt und die Spur diesen Sprung nachzeichnet.

Es wurde versucht, die Kugelmasse waagrecht auf der Wand laufen zu lassen ehe sie hinter der

Kante herunterfällt. Leider war dies nicht möglich. Wenn die Masse auf der Wand konstruiert wurde, blieb sie an der Kante stehen und fiel nicht herunter. Wenn sie über der Wand konstruiert wurde, kam es zu Reflexionen mit der Wand, wodurch ein waagerechter Wurf nicht mehr möglich war. Bei den Einstellungen der Wand gibt es die Option y- bzw. x-Dämpfung. Bei komplett eingeschalteter y-Dämpfung sprang die Masse zwar nicht mehr auf und ab, blieb aber stehen, obwohl es in x-Richtung keine Dämpfung hätte geben dürfen. Dies scheint ein Programmierfehler in Cinderella 2.0 zu sein.

Die Darstellung der Graphen sollte das Verständnis der Bewegungsgleichungen fördern, aber leider ist es mit CindyScript nicht möglich, die Graphen individuell zu gestalten. Beispielsweise ist der Startpunkt des Graphen immer der Nullpunkt. Es wäre wünschenswert, die Ober- bzw. Untergrenze eines Graphen selbst bestimmen zu können. Hierfür gibt es nur die Optionen, dass der jeweils höchste bzw. tiefste Wert im Graphenfenster die Grenzen beschreibt. Die andere Option besteht darin, dass der maximale Wert der Simulation die Obergrenze des Graphenfenster bestimmt. Wünschenswert wäre zusätzlich die Möglichkeit Achsen in die Graphen einzufügen, um ein besseres Verständnis zu ermöglichen.

6.3.2. Der schräge Wurf

In dieser Simulation ist es möglich, eine Kugelmasse in einem beliebigen Winkel zu werfen. Ein homogenes Kraftfeld wirkt nach unten, so dass ein schräger Wurf entsteht. Es wurden mehrere Werkzeuge eingebaut, um möglichst viele Einstellungen vornehmen zu können. Der Abwurfwinkel wird durch einen verstellbaren Winkel in der linken oberen Ecke eingestellt. Direkt darunter befindet sich ein Schieberegler, mit dem die Abwurfgeschwindigkeit festgelegt wird. Außerdem wurden vier Knöpfe integriert, mit denen man sich die Spur, den Geschwindigkeits- bzw. Beschleunigungsvektor oder Graphen anzeigen lassen kann. Die Graphen stellen jeweils die Beschleunigung, die Geschwindigkeit und den Ort in y-Richtung dar.

Umgebung:

1. Füge einen Boden ein.
2. Erstelle eine Masse auf dem Boden.

Die Winkeleinstellung:

1. Erstelle einen Kreis mit „Kreis um einen Punkt“.
2. Erstelle zwei Punkte auf dem Kreis, einen waagrecht zum Mittelpunkt auf der rechten Seite und einen beliebig.
3. Verbinde den Mittelpunkt und die beiden Punkte auf dem Kreis mit Strecken.
4. Messe den Winkel mit „Winkel messen“, indem beide Strecken markiert werden.
5. Kennzeichne den Winkel mit „Winkel markieren“, indem die drei Punkte angeklickt werden.

6.3. Wurfbewegungen

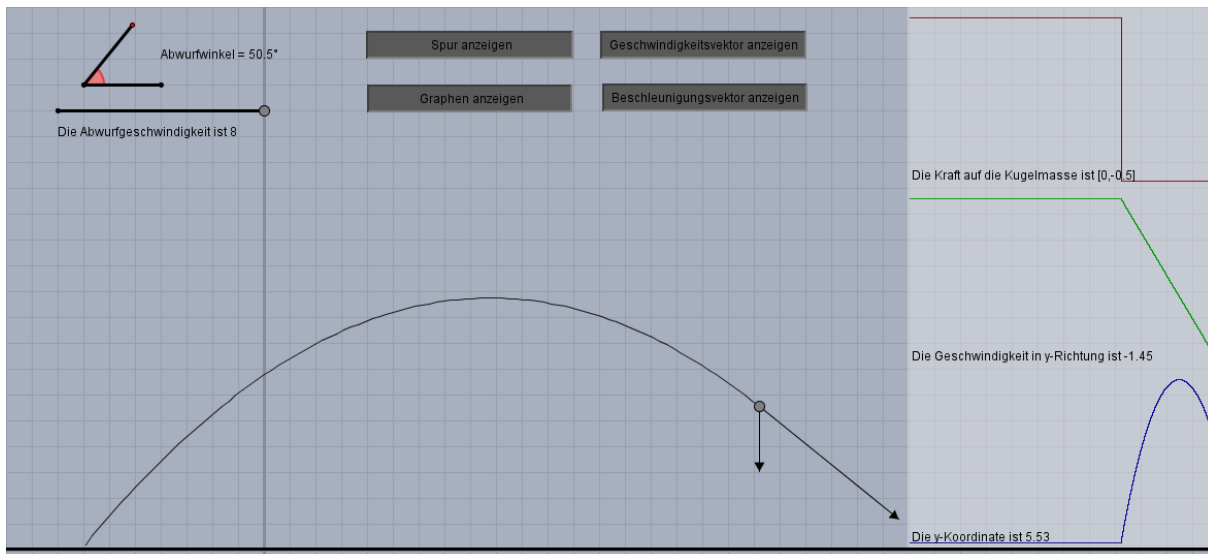


Abbildung 6.8.: Ein schräger Wurf

Schieberegler:

1. Erstelle eine Strecke.
2. Erstelle einen Punkt auf der Strecke.

Knöpfe:

1. Erstelle ein Textfeld mit passendem Text.
2. Aktiviere im *Informationsfenster* die Option „als Knopf benutzen“.

Die CindyScript Befehle sind bis auf eine Ausnahme analog zu denen beim waagerechten Wurf auf Seite 72. Der Unterschied besteht nur in der Bestimmung der Anfangsgeschwindigkeit und dem Abwurfwinkel. Dazu werden im CindyScript-Eingabefenster im Unterordner *Zeichnen* die Zeilen

```
Faktor:=|G,K|;  
Gx:=Faktor*cos(C.angle);  
Gy:=Faktor*sin(C.angle);
```

eingefügt. Der **Faktor** ist der Abstand des bewegten Punktes auf dem Schieberegler zum linken Eckpunkt. **Faktor** deshalb, weil ein normierter Geschwindigkeitsvektor genommen wird, der mit diesem **Faktor** multipliziert wird, um die gewählte Geschwindigkeit zu erhalten. Mit **Gx** und **Gy** werden die Geschwindigkeitskomponenten in x- und y-Richtung definiert. Hierzu wird der **Faktor** mit dem Sinus bzw. Kosinus multipliziert. Im Unterordner *Simulationsstart* muss noch die Anfangsgeschwindigkeit mit

A.vx=Gx; A.vy=Gy;

zugeordnet werden. Eine Änderung der Geschwindigkeit wirkt sich also erst nach einem Neustart der Simulation aus.

Die Vor- bzw. Nachteile dieser Simulation entsprechen denen beim waagerechten Wurf, weshalb sie hier nicht noch einmal gesondert aufgelistet werden. Allerdings könnte diese Simulation noch insofern erweitert werden, dass eine Coulomb-Reibung eingebaut werden könnte. Hierdurch könnte der reale Luftwiderstand dargestellt werden. Die Schüler könnten in diesem Zusammenhang lernen, dass ein Abwurfwinkel von 45° nur in der Theorie die größte Weite bewirkt.

6.4. Planetenbewegungen

Die keplerschen Gesetze werden in der 10. Klassenstufe im Zusammenhang mit den astronomischen Weltbildern eingeführt. Die Schüler sollen die Gesetze kennenlernen und auf die Bewegungen von Himmelskörpern anwenden können. Im Unterricht kann mit Hilfe einer solchen Konstruktion leicht klargemacht werden, wie Bewegungen von Himmelskörpern ablaufen. Die Schwierigkeit eine solche Beobachtung in der Realität zu machen, liegt zum einen daran, dass solche Planetenumläufe mehrere Jahre benötigen können und dass sie am besten nachts mit Hilfe von Teleskopen beobachtbar sind. Dem Lehrer steht also nur die Möglichkeit offen, die Gesetze anhand von Modellen, Bildern oder Simulationen zu erklären, wobei die Schüler bei einer Simulation ein Gefühl für den Ablauf einer Planetenbewegung bekommen. Nachfolgend werden zwei Simulationen vorgestellt, wobei anhand der ersten die Gesetzmäßigkeiten des ersten keplerschen Gesetzes, anhand der zweiten die des zweiten keplerschen Gesetzes verdeutlicht werden sollen. Die zweite Simulation wurde von Martin Gagern entwickelt und auf <http://www-m10.ma.tum.de/bin/view/MathVital/PhysicsExperiments/PhysikC1> zur Verfügung gestellt.

6.4.1. Ellipsen als Planetenbahnen (Erstes keplersches Gesetz)

In dieser Simulation soll dargestellt werden, dass Trabanten sich in einer elliptischen Bahn um ihren Schwerpunkt bewegen, wobei dieser Schwerpunkt einer der Brennpunkte ist. Die Konstruktion wurde als eine Art Baukasten entworfen. Der Nutzer soll die Möglichkeit haben sich „sein eigenes kleines Sonnensystem“ zu erstellen und zu beobachten, wie sich die Planeten verhalten und auf welchen Bahnen sie sich bewegen.

Es wurden drei Planeten (Kugelmassen) und zwei Sonnen (zentrale Kraftfelder) integriert. Sie werden von der Simulation nur dann berücksichtigt, wenn sie sich links von den eingefärbten Flächen befinden. Die Planeten wurden so erzeugt, dass ihnen eine Anfangsgeschwindigkeit zugeordnet worden ist. Diese kann vor Beginn der Simulation verändert werden. Je nach Massenverhältnis von Sonne zu Planet ist dies sogar nötig, da sonst die Umlaufbahn zu groß wird.

Für die Planeten und die Sonnen wurden Schieberegler zum Einstellen ihrer Masse eingefügt. Die Planeten wurden zusätzlich mit einem Knopf ausgestattet, der es ermöglicht, eine Spur zu zeichnen. Diese soll dazu dienen, die elliptische Laufbahn zu verdeutlichen.

Die Konstruktionsbeschreibung beschränkt sich auf die Beschreibung der Konstruktionsumgebung sowie die Erstellung einer Kugelmasse und eines zentralen Kraftfeldes. Die Einstellungen wie Farbe und Größe der Objekte können über zugehörige Informationsfenster eingestellt werden.

Umgebung:

1. Konstruiere in der Nähe des rechten Randes eine senkrechte Strecke vom oberen bis zum unteren Bildrand.
2. Zeichne eine waagerechte Strecke von dieser Geraden nach rechts bis zum Bildrand.
3. Erstelle Punkte in der oberen bzw. unteren rechten Ecke des Bildes.
4. Färbe die obere und untere Hälfte, indem ein „*Polygon definiert*“ wird.
5. Zeichne jeweils für eine Kugelmasse und jede Sonne eine waagerechte Strecke in das untere Polygon.
6. Setze auf jede Strecke einen Punkt.
7. Erstelle hinter den für Kugelmassen erstellten Strecken einen Text, z. B. „Bahnspur“ und aktiviere „*als Knopf benutzen*“ in den Eigenschaften.

Sonne und Planet:

1. Erstelle eine Sonne (zentrales Kraftfeld).
2. Erstelle eine Kugelmasse mit einer Geschwindigkeit.

Die restlichen Einstellungen werden mit CindyScript vorgenommen. Die Schieberegler werden wie in 6.2.1 festgelegt, wobei der Abstand bei den zentralen Kraftfeldern mit dem Faktor zehn multipliziert wurde. Die Spur wird durch folgende Befehlszeile erstellt:

```
if(Text0.pressed, l=l++[B.xy]; connect(1, color->(0,0.8,0)), l=[]);
```

Das bedeutet, dass nach jeder Neuberechnung des Punktes **B** ein neuer Punkt der Liste **l** zugefügt wird. Der Befehl `connect` verbindet alle so erzeugten Punkte **l** miteinander, wodurch eine Spur gezeichnet wird. Diese Spur ist nicht vergleichbar mit der Spur die unter 6.1.2 erzeugt werden sollte. Hier wird die Spur eines Punkt nachgezeichnet, bei der stehenden Welle sollten bewegte Strecken eine Spur bekommen und nicht nur deren Endpunkte.

In manchen Situationen kann es vorkommen, dass es so aussieht, als wenn ein Planet an einer Sonne reflektiert wird. Dies liegt daran, dass die Planeten bzw. die Sonnen zur Berechnung als Punktmassen aufgefasst werden. Es erscheint dem Nutzer so, als würde der Planet reflektiert,

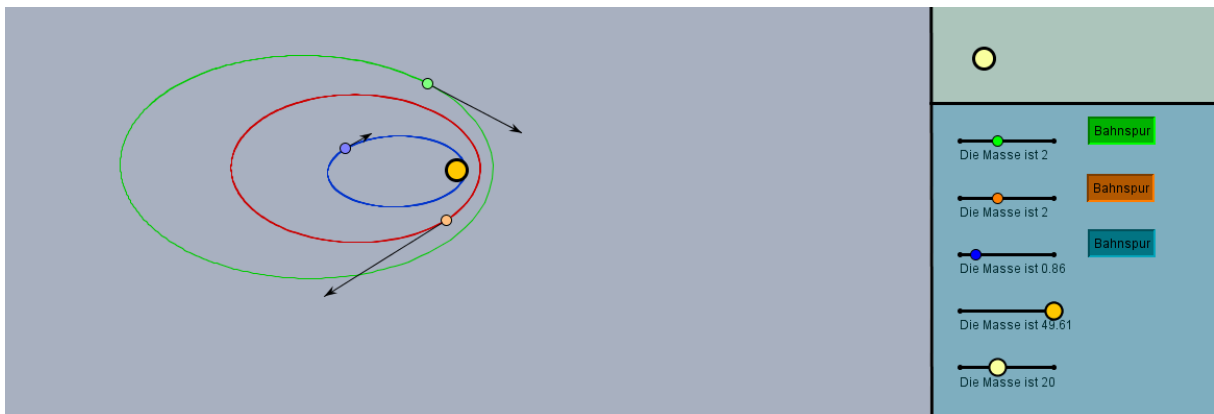


Abbildung 6.9.: Simulation von Planetenbewegungen

allerdings ist dies nur eine sehr extreme Ellipsenbahn, auf der sich der Planet befindet.

In der Simulation wurde bewusst darauf verzichtet, dass sich die Planeten untereinander anziehen. Wäre die Option „Anziehung zwischen Massen“ aktiviert, wäre es kaum möglich stabile Planetenbahnen zu erzeugen.

Weiterhin ist aufgefallen, dass mit dem Betätigen der Stop-Taste alle Punkte auf ihre ursprünglichen Koordinaten springen. Das Problem ist, dass die Spur der Kugelmassen auch diesen Sprung aufzeichnet. Pausiert man die Simulation, kann man diesen Sprung umgehen und die Simulation danach weiterlaufen lassen. Wenn die Simulation beendet oder verändert wird, sollte der Knopf „Bahnspur“ deaktiviert und danach wieder aktiviert werden, um die neue Bahnspur zu zeichnen. Die gezeichneten Spuren lassen allerdings erkennen, dass die Umlaufbahnen Ellipsen sind. Leider war es nicht möglich, Ellipsen geometrisch so zu definieren, dass sie die Bahnen der Kugelmassen vorraussagen. Man benötigt zum Zeichnen einer Ellipse beide Brennpunkte und einen Punkt auf der Ellipse, es können aber nur ein Brennpunkt und ein Punkt auf der Ellipse angegeben werden. Die farbliche Abstimmung von Schieberegler, Knöpfen und Kugelmassen erleichtert den Schülern den Umgang mit der Simulation und schafft so freie kognitive Ressourcen.

6.4.2. Zweites keplersches Gesetz

Diese Simulation verdeutlicht das zweite keplersche Gesetz. Hierbei wird der Umlauf eines Planeten um eine Sonne simuliert. Ein Schieberegler legt ein Zeitintervall fest, für das die Fläche zwischen der Umlaufbahn und dem Brennpunkt der Ellipse gezeichnet wird. Weiterhin eingefügt wurden Textzeilen, die die kinetische Energie, die potentielle Energie, die Summe aus beiden, den Flächeninhalt der gezeichneten Fläche und das Zeitintervall ausgeben.

Die Konstruktion an sich ist simpel, allerdings sind die Befehle in CindyScript zum Zeichnen und Erstellen der Fläche sehr komplex und erfordern ein tieferes Verständnis der Scriptsprache.

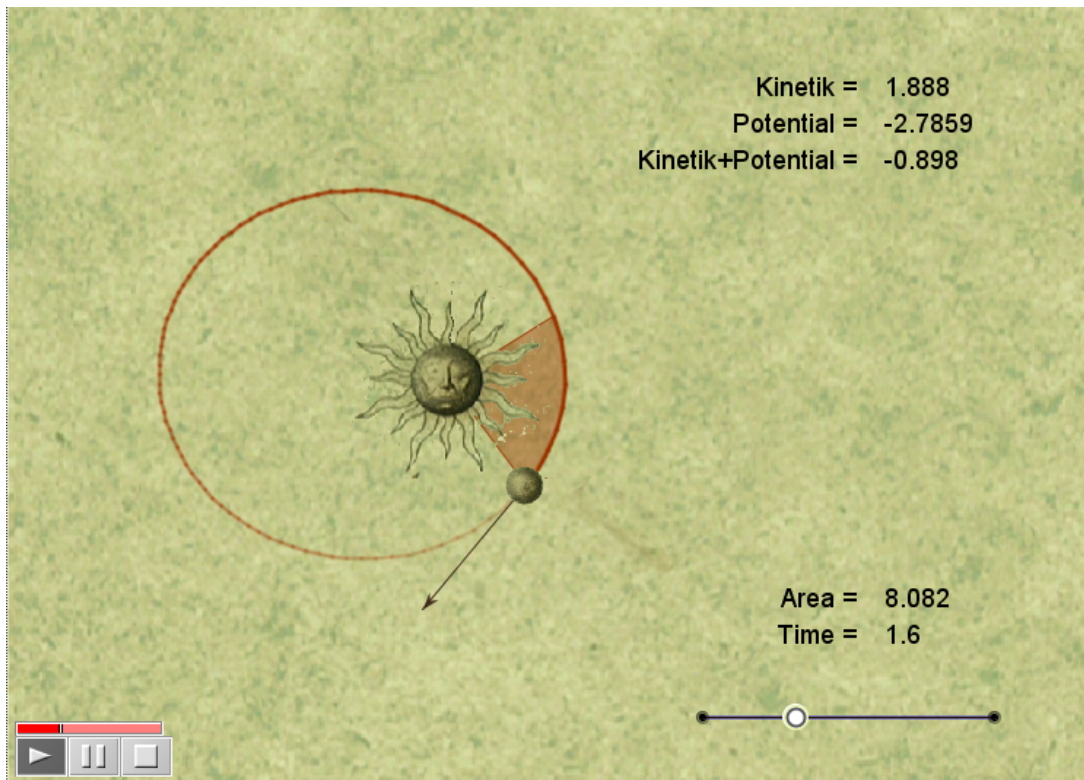


Abbildung 6.10.: Simulation des zweiten keplerschen Gesetzes

1. Konstruiere ein zentrales Kraftfeld (Sonne).
2. Konstruiere eine Kugelmasse mit Geschwindigkeitsvektor.
3. Erstelle eine waagerechte Strecke.
4. Zeichne einen Punkt auf diese Strecke.

In dieser Simulation sind Eingaben in verschiedenen Unterordnern des CindyScript-Eingabefensters nötig. Auf diese Weise werden die wichtigen Vorgänge, z. B. die Erstellung der Fläche, beschrieben und schon bekannte, z. B. die Ausgabe eines Textes, vernachlässigt. In dem Unterordner *Timerschritt* wird die Momentansituation mit den notwendigen Variablen und Einstellungen initialisiert.

```
li=[B.xy]++li;if(length(li)>n1,li=apply(1..n1,li_#));
```

Es wird in jedem Rechenschritt eine Liste *li* definiert, der die Koordinaten von *B* zugeordnet werden. Die *if*-Schleife besagt, dass jedem Element dieser Liste ein eigener Name *li_#* zugeordnet wird, sobald die Anzahl der Einträge größer wird als eine Variable *n1* (wird im Unterordner *Zeichnen* definiert). *#* steht hierbei für Zahlen von 1 bis *n1*. Man erhält also für jeden Rechenschritt eine Liste von Punkten, die sich ständig erneuert.

Im Unterordner Zeichnen werden die Spur der Kugelmasse sowie die gefärbte Fläche definiert.

```
l_pos=B.xy;
nl=round(200*(|D,G|/|D,F|))+1;
if(length(li)<=nl,
drawpolygon(li++[A.xy],color->(0.6,0.2,0),alpha->0.4);
connect([A.xy]++li++[A.xy],color->(0.6,0.2,0),alpha->0.8);
ss=sum(1..(nl-1),area(li_#,li_(#+1),A.xy)););
```

Durch den obigen Quelltext wird die Fläche definiert und gezeichnet. Der Wert `l_pos` wird dem Punkt `B` zugeordnet. Die Variable `nl` wird ausgedrückt als Verhältnis des Abstandes des Punktes auf dem Schieberegler vom linken Ende des Schiebereglers zu der Gesamtlänge des Schiebereglers. Das Ergebnis des Verhältnisses wird gerundet und eins hinzu addiert, damit `nl` nicht null sein kann.

In der `if`-Schleife findet der eigentliche Vorgang statt. Die Befehle innerhalb der Schleife werden aktiviert, wenn die Anzahl der Elemente in `li` kleiner oder gleich der Variablen `nl` sind. Dies ist grundsätzlich immer der Fall, es sei denn der Schieberegler ist auf der äußersten linken Einstellung. Je weiter er vom linken Ende entfernt ist, desto größer wird `nl` und somit auch die Fläche. Der Befehl `drawpolygon` zeichnet ein Polygon, aus allen Punkten, die in der Liste `li` enthalten sind, und dem Punkt `A.xy` (der Sonne). Mit den Befehlen `color` und `alpha` wird der Fläche eine Farbe sowie eine Durchlässigkeit zugeteilt. In der fünften Zeile wird mit dem Befehl `connect` der Rahmen der Fläche gezeichnet. Der Startpunkt ist `A.xy`, der mit dem ersten Punkt aus `li` verbunden wird. Anschließend werden alle Punkte aus `li` und zum Schluss der letzte Punkt aus `li` wieder mit dem Startpunkt `A.xy` verbunden, wodurch sich der Rahmen schließt. In der letzten Zeile der `if`-Schleife wird der Flächeninhalt mit dem `area`-Befehl bestimmt. Dieser Befehl benötigt die Eckpunkte eines Polygons und kann damit die Fläche berechnen. Es wird eine Summe aus Dreiecken berechnet. Die Eckpunkte des ersten Dreiecks wären `A.xy`, der erste Punkt aus `li` und der zweite Punkt aus `li`. Die Eckpunkte des zweiten Dreiecks `A.xy`, der zweite Punkt aus `li` und der dritte Punkt aus `li` usw. Je kleiner die Abstände der Punkte in `li` sind, desto genauer wird das Ergebnis der Fläche werden.

Um eine Spur zu erzeugen, die nach einer festgelegten Anzahl von weiteren Schritten verblasst, wird dieser Quellcode benötigt:

```
if(running, apply(2..n-2,i,
draw(l_(mod(pos-i,n)+1),
l_(mod(pos-i-1,n)+1),
alpha->1-(i/n), color->(0.6,0.2,0),size->2));
apply(2..n-2,i,
draw(l_(mod(pos-i,n)+1),
```

```
l_(mod(pos-i-1,n)+1),  
alpha->(1-(i/n))*0.4, color->(0.6,0.2,0),size->4));  
pos=pos+1;  
if(pos==n+1,pos=1););
```

Wenn das Programm läuft, werden in der `if`-Schleife durch den `draw`-Befehl Strecken zwischen zwei Punkten der Liste `l` gezeichnet. Der `apply`-Operator ordnet jedem `i` die Zahlen 2 bis `n-2` zu. Und die Variable `pos` wird am Ende jedes Schleifendurchlaufs um eins erhöht bis `pos` gleich `n+1` wird. Dann wird `pos` auf den Wert 1 zurückgesetzt. Der `mod`⁶ im `draw`-Befehl bewirkt, dass die Spur gleichmäßig verläuft und nicht periodisch ein Stück der Spur gezeichnet wird. Die Befehlsoption `alpha` ordnet den Spurpunkten eine abstandsabhängige Durchlässigkeit zu.

Positiv aufgefallen ist, dass die Gesetzmäßigkeiten des zweiten keplerschen Gesetzes sehr gut sichtbar gemacht werden und zugleich die Größe des Flächeninhalts frei variiert werden kann. Negativ hingegen fällt auf, dass die potentielle Energie einen negativen Wert hat. Da die potentielle Energie im Grunde nur eine Energiedifferenz auf einen Bezugspunkt darstellt, ist dieses Ergebnis zwar möglich, kann aber für den Schüler verwirrend sein. Schließlich betrachten sie die potentielle Energie oft nicht als Energiedifferenz sondern als „Höhenenergie“ und ihnen ist nicht bewusst, dass sie auch dort einen Bezugspunkt haben. Außerdem kann es bei einer schnellen Ablaufgeschwindigkeit der Simulation dazu kommen, dass die Fläche sich minimal verändert. Dies liegt aber daran, dass bei schnellen Bewegungen die Punkte der Liste `li` weiter auseinander liegen, wodurch die Berechnung des Flächeninhalts durch die Summe der Dreiecke vom ursprünglichen Wert abweichen kann.

6.5. Elektrische Feldlinienbilder

Der Lehrplan der 11. Klassenstufe sieht vor, dass die Schüler das statische elektrische Feld kennenlernen. Hierzu soll das elektrische Feld an Feldlinienbildern veranschaulicht werden. Dabei lernen die Schüler, „wie sich das bisher nur qualitativ betrachtete elektrische Feld über die Kraftwirkung auf einen geladenen Probekörper quantitativ und in seiner räumlichen Struktur genauer erfassen lässt.“⁷

Das Darstellen von elektrischen Feldlinien kann experimentell durchgeführt werden. Dies geschieht üblicherweise entweder mit Hilfe von Rizinusöl und Grießkörnern oder mit Plastikfasern auf Karton. Für die erste Variante wird eine Petrischale auf einen Tageslichtprojektor gestellt. Es werden Rizinusöl und Grießkörner in die Schale gegossen. Außerdem werden zwei Elektroden in der Schale platziert, die an eine Hochspannungsquelle angeschlossen werden. Die Grießkörner richten sich entsprechend des elektrischen Feldes zwischen den Elektroden aus, da sie im elektrischen Feld zu einem Dipol werden. Es bilden sich Ketten aus den Grießkörnern, welche

⁶Mit dem `mod`-Befehl ist eine modulo Rechnung gemeint, also eine Division bei der nur der Rest betrachtet wird.

⁷STAATSWISSENSCHAFTLICHES INSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrpläne Gymnasium G8*, <http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=27147>.

das elektrische Feld darstellen sollen. Die Variante mit den Plastikfasern basiert auch auf der Ausrichtung der Plastikfasern durch ihre Dipoleigenschaften im elektrischen Feld.

Der Nachteil der ersten Variante ist, dass es mit einem enormen Aufwand verbunden ist, die Gerätschaften wieder zu säubern. Außerdem muss das Verhältnis von Rizinusöl zu Grießkörner stimmen, um gute Ergebnisse zu erhalten. Die Schüler selbst können nur unter sehr strenger Aufsicht die Versuche durchführen, da beide Experimente mit Hochspannung durchgeführt werden müssen. Ein weiterer Nachteil ist, dass die Stärke des elektrischen Feldes nicht dargestellt werden kann, da die Anzahl der Grießkörner bzw. Plastikfasern nicht beeinflusst werden kann.

6.5.1. Elektrische Feldlinien

Diese Simulation stellt eine Art Baukasten für elektrische Feldlinienbilder dar. Die verschiedenen „Elektroden“ (Körper) können mit der Maus angeordnet werden, um bspw. die Versuche mit dem Rizinusöl darzustellen. Die Darstellung der Feldlinienbilder erfolgt über Nadeln die zufällig auf den Feldlinien entstehen und wieder verschwinden. Es wurden vier Körper in die Simulation integriert. Zwei Punktladungen und zwei Stabladungen. Alle Ladungsträger können mit einem Schieberegler eingestellt werden. In der rechten oberen Ecke befindet sich ein Knopf, der die Darstellung der Feldlinienbilder aktiviert.

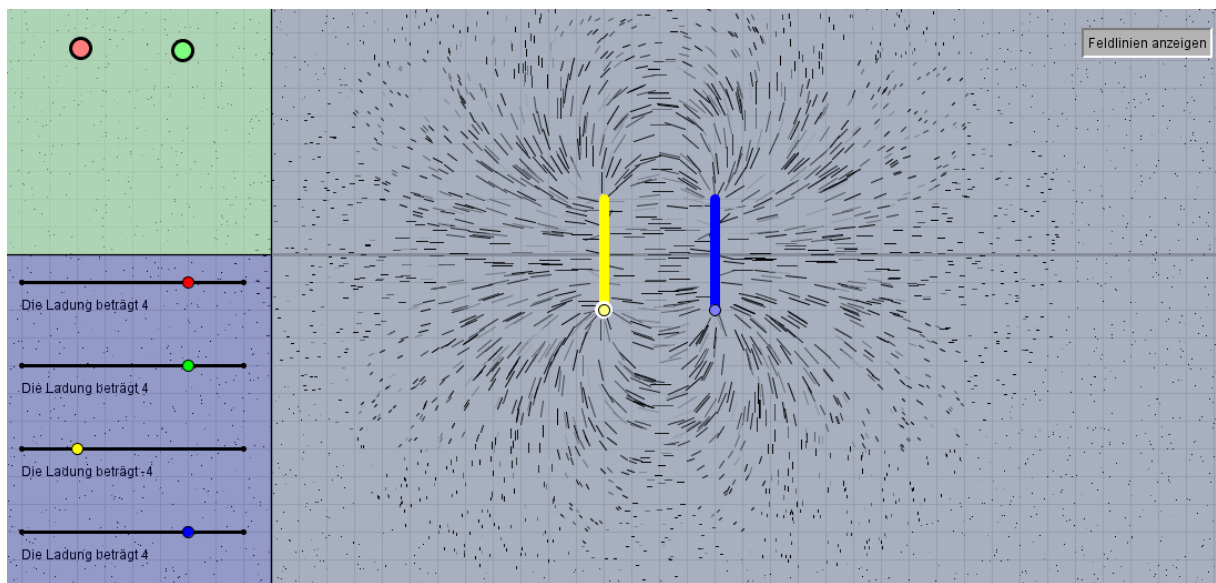


Abbildung 6.11.: Darstellung von Feldlinien mit Cinderella 2.0

Die Konstruktionsbeschreibung wird nur für jeweils eine Punkt- bzw. Stabladung dargestellt. Die Umgebung sowie farbliche Gestaltung der Simulation wird hier nicht erläutert.

Punktladung:

1. Erstelle einen Kugelmasse.

Stablading:

1. Erstelle eine Kugelmasse.
2. Zeichne Kreise um diese Masse, die jeweils den gleichen Abstand voneinander haben.
3. Zeichne eine Gerade durch den Punkt.
4. Erstelle Kugelmassen auf den Schnittpunkten der Kreise und der Geraden.
5. Blende die Kreise und die Gerade aus, indem im Informationsfenster der Haken bei sichtbar entfernt wird.
6. Verbinde die äußeren Kugelmassen mit einer Strecke.

Schieberegler:

1. Erstelle für jeden Ladungsträger eine Strecke.
2. Erstelle auf jeder Strecke einen Punkt.

Knopf:

1. Erstelle ein Textfeld.
2. Aktiviere im Informationsfenster „Als Knopf benutzen“.

Mit Cindyscript werden den Ladungen, Schieberegler und dem Knopf Funktionen zugewiesen. Auch die CindyScript Befehle werden jeweils nur einmal erläutert und können danach analog angewendet werden.

```
S1:=(|P0,P2|-4);  
if(G.x>0,G.charge=round(2*S1),G.charge=0);  
drawtext([P0.x,P0.y-1],"Die Ladung beträgt "+round(2*S1));
```

In der ersten Zeile wird eine Variable $S1$ definiert, die sich als Abstand von $P0$ zu $P2$ berechnet und von diesem Wert 4 subtrahiert. Hier wird ein Wert für einen Schieberegler berechnet. Die 4 werden abgezogen, um in der Mitte des Schiebereglers den Wert 0 zu erhalten. Die zweite Zeile legt für eine Punktladung fest, dass die Ladung zweimal so groß sein soll wie der Wert von $S1$, wenn der x-Wert ihrer Koordinaten größer als null ist und dieser Wert durch den `round`-Befehl gerundet wird, so dass es nur ganzzahlige Ladungen geben kann. Ist der x-Wert kleiner als null, so wird dem Körper keine Ladung zugeteilt. Der `drawtext`-Befehl dient hier dazu, den Wert der Ladung ablesen zu können. Er erzeugt eine Ausgabe unterhalb des Schiebereglers eines jedes Körpers.

```
L1:=[P,Q,R,S,T];
```

```
if(P.x>0,apply(L1,#.charge=round(2*S4)),apply(L1,#.charge=0));
```

In der dritten Zeile wird eine Liste von Punkten definiert. Diese Punkte sind die Körper, die die Stabladung darstellen sollen. In der `if`-Schleife wird, sobald der `x`-Wert des untersten Punktes größer als 0 ist, mit dem `apply`-Befehl jedem Körper innerhalb dieser Liste eine Ladung zugeteilt, die doppelt so groß ist wie der Abstand des Schiebereglers. Der Abstand wird wie oben beschrieben definiert und gerundet. Wenn der unterste Punkt der Stabladung kleiner als 0 ist, wird der ganzen Liste die Ladung 0, also keine Ladung zugeteilt.

```
if(Text0.pressed,
drawforces(stream->>true,move->0,jitter->100,resolution->12,
color->[0,0,0],color2->[0,0,0],factor->2));
```

Diese `if`-Schleife weist dem Knopf eine Bedeutung zu. Wenn der Knopf aktiviert ist, werden die Kräfte innerhalb der Simulation gezeichnet. In den Simulationseinstellungen muss „*Ladungen erzeugen Kräfte*“ aktiviert und bei allen in der Simulation vorkommenden Massen muss im Informationsfenster „*fixiert*“ angekreuzt sein. Die Optionen sind in 5.4.2 genauer erklärt.

Positiv ist bei der Simulation aufgefallen, dass die Stärke des Feldes dargestellt wird, indem die Nadeln länger bzw. kürzer erscheinen. Außerdem sind in weit entfernten Punkten kaum Nadeln, da hier das Feld immer schwächer wird. Die Tatsache, dass die Nadeln zufällig entstehen und wieder verschwinden, kann den Schüler anfangs irritieren. Dennoch spiegelt die Simulation eine korrekte Darstellung der Feldlinien wider. Noch positiv ist die Tatsache zu bewerten, dass Schüler die Simulation ohne Gefahr durchführen können, wobei beim Realversuch aufgrund der Hochspannung hohe Sicherheitsvorkehrungen getroffen werden müssen. Außerdem entsteht durch den Einsatz dieser Simulation nicht der Aufwand, die Elektroden säubern und das Öl von allem entfernen zu müssen.

Negativ ist die Tatsache zu bewerten, dass sich mit Cinderella 2.0 keine korrekten Stabelektroden, Kreiselektroden usw. erstellen lassen. Will man eine solche Elektrode erstellen, muss durch möglichst äquidistante Punktladungen versucht werden, die gewünschte Form möglichst genau nachzustellen. Außerdem ist es in dem Fall nicht möglich, eine Ringelektrode zu erstellen, in deren Mitte kein elektrisches Feld vorherrscht, da die einzelnen Punktladungen ein Feld in alle Richtungen besitzen.

Wünschenswert wäre es, beliebige Formen erstellen und eine Spannung an diesen Formen anlegen zu können. Außerdem wäre eine Möglichkeit, elektrische Schaltungen bzw. ein homogenes elektrisches Feld zu erzeugen, wie es mit dem homogenen Magnetfeld möglich ist, von Vorteil.

6.6. Das homogene Magnetfeld

In der 11. Jahrgangsstufe wird das Verhalten von geladenen Teilchen in elektrischen und magnetischen Feldern untersucht. Dabei werden die in der 10. Klassenstufe erarbeiteten newton'schen Gesetze als Grundlage genommen, um die Bewegung der Teilchen qualitativ und quantitativ zu beschreiben. Dabei lernen die Schüler Effekte kennen, z. B. die Massenzunahme bei sehr schnellen Teilchen, die mit dieser Theorie nicht mehr zu erklären sind, aber durch die Relativitätstheorie erklärt werden können.

Ein Teil dieser Unterrichtseinheit befasst sich mit der Bewegung von geladenen Teilchen in einem homogenen Magnetfeld. Die Schüler lernen die Kräfte kennen die innerhalb eines Magnetfeldes wirken und verstehen, warum Ladungen im homogenen Magnetfeld auf eine Kreisbahn abgelenkt werden.

Im Unterricht selbst wird diese Kreisbahn häufig an einem Fadenstrahlrohr verdeutlicht. Der Lehrplan gibt als Beispiel für die Anwendung in der Wissenschaft das Beispiel ein einfachen Massenspektrographen an. Das Prinzip des Massenspektrographen basiert darauf, dass geladene Teilchen so in ein homogenes Magnetfeld gelenkt werden, dass nur Teilchen mit einer bestimmten Geschwindigkeit das Magnetfeld erreichen. Dies wird durch einen Wien-Filter erreicht. Dort werden sie auf eine Kreisbahn abgelenkt. Nach einer Halbkreisbahn verlassen sie das Magnetfeld wieder. Es wird ein Detektor am Rand des Magnetfeldes angebracht, der das Auftreffen von geladenen Teilchen registrieren kann. Anhand des Auftreffpunktes kann die spezifische Masse $\frac{q}{m}$ bestimmt werden.

Die folgende Simulation kommt ohne einen Wien-Filter aus, da die Teilchen, die in das Magnetfeld eintreten immer die gleiche Geschwindigkeit und Richtung haben.

6.6.1. Einfacher Massenspektrograph

Die Simulation stellt ein Teilchen dar, dass sich auf ein homogenes Magnetfeld zubewegt. Die Geschwindigkeit des Teilchens muss vor dem Start der Simulation durch eine Verlängerung oder Kürzung des Vektors eingestellt werden. Man hat die Möglichkeit, die Masse, die Ladung und die magnetische Feldstärke mit Schieberegler einzustellen. Die Beschriftung der Schieberegler und damit die genauen Werte der Einstellungen können durch Knöpfe ein- bzw. ausgestellt werden. In der linken oberen Hälfte werden außerdem noch zwei Texte ausgegeben. Einer gibt die Geschwindigkeit des Teilchens an, der andere den Radius, den das Teilchen im Magnetfeld benötigt, bevor es das Magnetfeld wieder verlässt. Die folgende Konstruktionsbeschreibung beschränkt sich auf einen Schieberegler inklusive Knopf, die anderen können analog konstruiert werden.

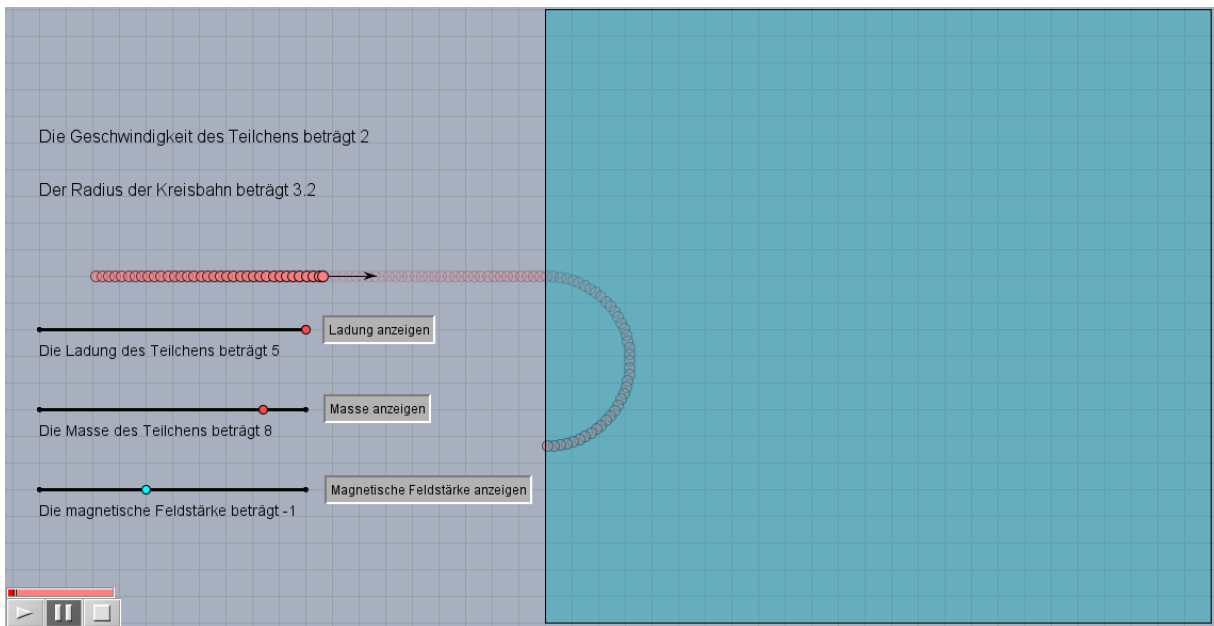


Abbildung 6.12.: Einfacher Massenspektrograph

Umgebung:

1. Erstelle eine Kugelmasse mit Geschwindigkeitsvektor.
2. Zeichne vier Punkte, die die Ecken des homogenen Magnetfeldes darstellen sollen.
3. Erstelle ein homogenes Magnetfeld.
4. Erstelle zwei Punkte und deaktiviere den Haken „sichtbar“ im *Informationsfenster*

Schieberegler und Knopf:

1. Erstelle eine Strecke.
2. Setze einen Punkt auf diese Strecke.
3. Erstelle ein Textfeld mit dem zugehörigen Text, z. B. „Masse anzeigen“.
4. Aktiviere im *Informationsfenster* des Textfeldes „Als Knopf benutzen“.

Die beiden erstellten Punkte, deren Sichtbarkeit entfernt wurde, dienen dazu später den Radius der Kreisbahn bestimmen zu können. Die restlichen Einstellungen werden mit CindyScript vorgenommen.

```
trigger(G.x>=11,moveto(E,G));
trigger(G.x<=11,moveto(F,G));
```

Den beiden nicht sichtbaren Punkten wird zugeordnet, dass sie die Koordinaten des Teilchens übernehmen sollen sobald es die Grenze des Magnetfeldes passiert und dort bleiben, bis das Teilchen wieder die Grenze überschreitet. Der `trigger`-Operator wirkt nur in eine Richtung, d. h. er wirkt sich nur aus, wenn das Teilchen bspw. die angegebene Grenze von links nach rechts

überschreitet.

```
trigger(G.x<11,moveto(G,[-6,2]));  
trigger(G.x<11,G.vx=|G.v|);  
trigger(G.x<11,G.vy=0);
```

Die drei Operatoren legen fest, dass die Kugelmasse nach dem Verlassen des Magnetfeldes, also in der Bewegung von rechts nach links, zurück auf ihren Startpunkt gesetzt wird und dabei den gleichen Geschwindigkeitsbetrag behält wie zum Start der Simulation. Die Geschwindigkeit in y-Richtung wird dabei 0 gesetzt, damit die Ladung beim nächsten Durchlauf wieder senkrecht auf das Magnetfeld trifft. Die Einstellungen der Schieberegler werden hier nur an einem Beispiel erklärt, da der Rest analog programmiert werden kann.

```
w=|Q,S|;  
G.charge=(round(w)-5);  
if(Text0.pressed,  
drawtext(U,"Die Ladung des Teilchens beträgt "+(round(w)-5),size->14));
```

In der ersten Zeile wird der Abstand von linken Punkt der Strecke zu dem verschiebbaren Punkt auf der Strecke definiert. Die zweite Zeile legt die Ladung der Kugelmasse fest, indem der Abstand gerundet und der Nullpunkt des Schiebereglers in die Mitte verlegt wird. Die `if`-Schleife beschreibt: wenn der Knopf gedrückt ist, dann schreibe unter den Schieberegler die Ladung des Teilchens.

```
drawtext([T.x,T.y+2],"Die Geschwindigkeit des Teilchens beträgt "+(|G.v|));  
if(G.x<11,  
drawtext(T,"Der Radius der Kreisbahn beträgt "+format((|E,F|/2),2),size->16),  
drawtext(T,"Der Radius der Kreisbahn beträgt ",size->16));
```

Die erste Zeile erstellt eine Ausgabe der Geschwindigkeit. Die `if`-Schleife lässt den Radius der Kreisbahn dann ausgeben, wenn das Teilchen sich außerhalb des Magnetfeldes befindet. Der Radius wird berechnet, indem der Abstand der unsichtbaren Punkte halbiert und auf zwei Stellen nach dem Komma gerundet wird. Somit wird für jeden Durchlauf erneut ein Ergebnis berechnet, falls die Einstellungen während der ablaufenden Simulation geändert werden.

In der Anwendung der Simulation ist aufgefallen, dass bei vielen Einstellungen die Ladung innerhalb des Magnetfeldes keine komplette Halbkreis Bewegung macht. Dies kann zum einen verhindert werden, indem die Fläche des Magnetfeldes größer gemacht wird. Andererseits dadurch, dass der Bereich der Schieberegler so an die Simulation angepasst wird, dass es keine Einstellung mehr gibt, in der das Magnetfeld verlassen wird.

Außerdem wäre es hilfreich, wenn zusätzlich zu der schriftlichen Ausgabe der Magnetfeldstärke angezeigt würde, in welche Richtung das Magnetfeld zeigt. Dies ist leider mit Cinderella 2.0 noch nicht möglich. Es gäbe zwar die Möglichkeit, diesen Effekt mit CindyScript zu programmieren, dies wäre aber äußerst aufwändig. Man müsste sich Punkte innerhalb des Magnetfeldes aussuchen, an denen zu bestimmten Einstellungen die Richtung des Magnetfeldes durch einen „Kreis mit Kreuz“ bzw. einen „Kreis mit Punkt“ eingezeichnet würde.

Hilfreich wäre es zudem, die Spur des Teilchen beliebig an- und ausschalten zu können. Allerdings müsste man dazu eine selbsterstellte Spur wie in 6.4 benutzen. Dabei tritt jedoch wieder das Problem auf, dass beim Verschieben der Ladung an seinen Ursprungspunkt auch diese Spur eingezeichnet würde. Dies würde den Betrachter nur unnötig verwirren.

6.7. Strahlenoptik

Im Lehrplan der 7. Klassenstufe wird im Fachbereich Natur und Technik der Schwerpunkt auf Physik gelegt. Unter anderem wird das Thema Strahlenoptik behandelt. Die Schüler sollen die Brechung, Abbildung durch Sammellinsen und Entstehung reeller Bilder kennenlernen.

Im Unterricht werden hierzu oft Versuche mit einer optischen Bank vorgeführt. Um die Strahlengänge zu verstehen, werden Zeichnungen angefertigt, die den Strahlengang verdeutlichen sollen. Dies ist oft sehr langwierig, da die Schüler viel zeichnen müssen. Außerdem muss für jede Einstellung eine neue Zeichnung erstellt werden. Die Anwendung einer Simulation scheint deshalb sinnvoll, um wertvolle Unterrichtszeit einzusparen und weil die Schüler die Konstruktion dynamisch verändern können, damit sie ein Gefühl für das Phänomen der Brechung bzw. der Abbildung bekommen.

Die Strahlenoptik war das erste Einsatzgebiet von DGS in Bezug auf die Physik. Die Strahlenoptik ist komplett geometrisch konstruierbar und deshalb nicht nur mit Cinderella 2.0, sondern auch mit jedem anderen DGS möglich. Aus diesem Grund wird an dieser Stelle nur ein Beispiel gezeigt. Dieses Beispiel ist deshalb interessant, weil es nicht nur geometrisch konstruiert, sondern auch umfangreich mit CindyScript bearbeitet wurde.

6.7.1. Die optische Bank

Diese von Martin Gagern erstellte Simulation wurde auf <http://www-m10.ma.tum.de/bin/view/MatheVital/PhysicsExperiments/PhysikG7> zur Verfügung gestellt. Sie beinhaltet eine Art Baukasten für Versuche mit der Strahlenoptik. Wie bei einer optischen Bank kann man zwischen verschiedenen Lichtquellen wählen. Es gibt einen Einzelstrahl, parallele Strahlen und eine punktförmige Lichtquelle, die aufgefächerte Lichtstrahlen aussendet. Drei Linsen wurden eingefügt, eine Konkavlinse sowie zwei unterschiedlich große Konvexlinsen, sowie zwei Spiegel, ein ebener Spiegel und ein kreisförmiger Spiegel, der sowohl außen als auch innen die Strahlen reflektiert. Mit einem Knopf „Virtuelle Strahlen“ kann man die virtuellen Strahlen anzeigen lassen, um bspw. den Strahlengang an einer Konkavlinse nachvollziehen zu können.

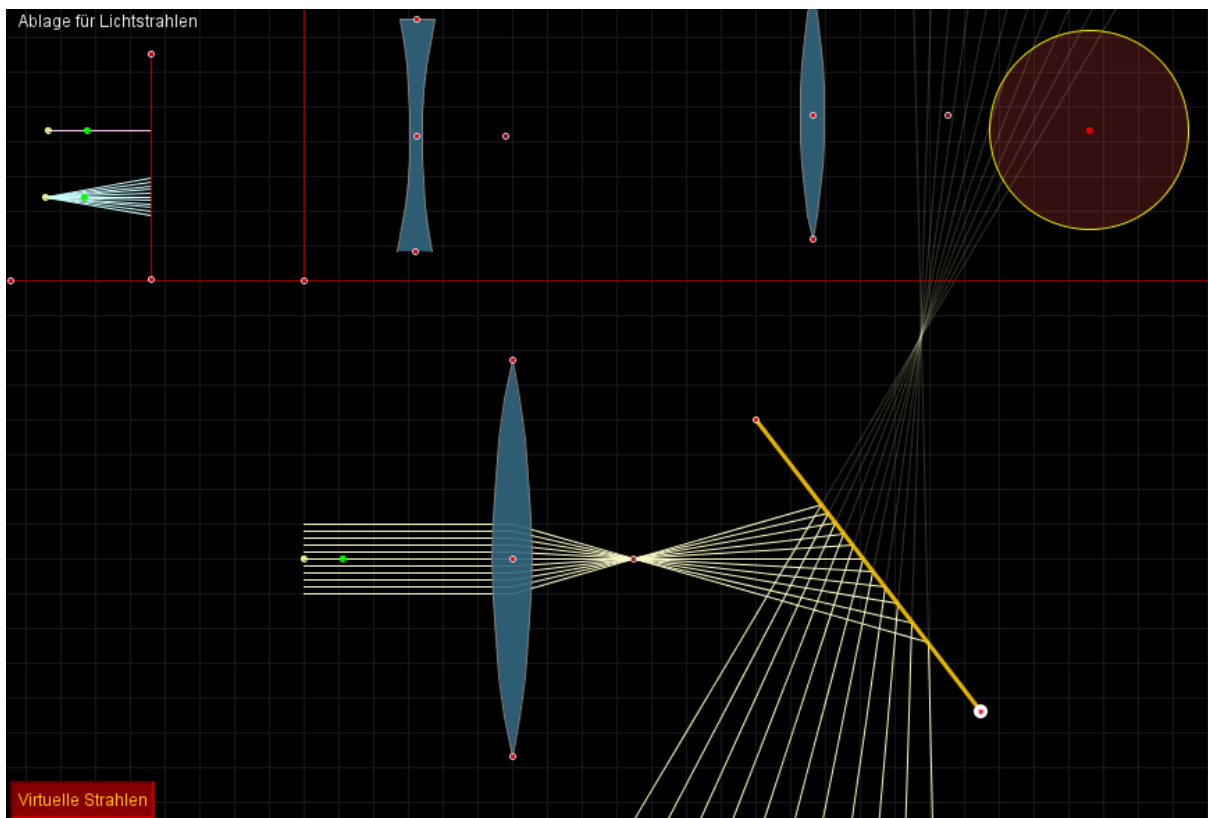


Abbildung 6.13.: Simulation einer optischen Bank

Da diese Simulation zum Download bereitgestellt ist, wird an dieser Stelle auf eine genauere Konstruktionsbeschreibung verzichtet. Eine genaue Konstruktionsbeschreibung kann mit Hilfe von Cinderella 2.0 unter „Ansichten -> Konstruktionsbeschreibung“ oder mit der Tastenkombination *Strg+4* geöffnet werden.

Der CindyScript-Quelltext ist in diesem Beispiel sehr lang und komplex, so dass es zu umfangreich wäre, ihn hier zu erläutern. Trotzdem sollte diese Konstruktion hier kurz vorgestellt werden, um zu verdeutlichen, dass durch die Scriptsprache CindyScript viel mehr Simulationen hergestellt werden können, als es mit einem reinen DGS möglich ist.

Bei der Nutzung dieser Simulation ist aufgefallen, dass sie viele Anwendungsoptionen zur Verfügung stellt und detailgenau ist. Diese Tatsache kann aber auch dazu führen, dass der Nutzer nicht weiß was hier untersucht werden soll, und sich deshalb überfordert fühlt. Außerdem wäre es sinnvoll, sich die Brennweiten sowie Bild- und Gegenstandsweiten anzeigen lassen zu können.

6.8. Möglichkeiten und Grenzen von Cinderella 2.0

Im Umgang mit Cinderella 2.0 sind etliche Vor-, aber auch einige Nachteile aufgefallen. Das Programm deckt große Teile der Schulphysik ab. Man kann z. B. homogene magnetische Felder erzeugen, homogene elektrische Felder aber nicht. Für den Physikunterricht hat dies zur

Konsequenz, dass man bspw. bewegte Ladungen in einem Magnetfeld simulieren kann (siehe 6.6). Es ist aber nicht möglich, die Bewegung von Teilchen innerhalb eines Plattenkondensators (homogenes elektrisches Feld) zu simulieren. Außerdem ist es störend, dass alle mit CindyScript ausgelesenen Daten ohne Einheiten angegeben wurden. Man kann zwar Verhältnisse bestimmen, aber in welcher Größenordnung sie sich befinden, wird daraus nicht ersichtlich. Der Nutzer hat nur die Möglichkeit die Größenordnung selbst festzulegen, indem eine Textausgabe erstellt wird, in der zusätzlich zu den ausgelesenen Werten auch Einheiten angegeben werden.

Im Ablauf der Simulationen ist immer wieder aufgefallen, dass nach dem Betätigen der Pausetaste die Starteinstellungen mit den neuen Pauseneinstellungen überschrieben wurden. Um danach die Simulation wieder in den Ursprungszustand zu versetzen, muss das Programm neu gestartet werden. Da CindyLab aber erst seit Cinderella 2.0 integriert ist, werden sicher einige dieser Probleme bei einem späteren Update beseitigt. Im Hilfe-Forum wird angekündigt, dass es in der nächsten Version möglich ist, Pfeile mit CindyScript zu erstellen.

Simulationen, die mit Cinderella 2.0 erstellt werden, können direkt als Java-Applet in eine html-Seite integriert werden. Deshalb können die Konstruktionen in jedem aktuellen Internetbrowser geöffnet und benutzt werden. Im Hinblick auf Hausaufgaben, die im Internet durchgeführt, oder Unterrichtsmaterialien, die von Schülern nochmals nachvollzogen werden sollen, ist dies von großem Nutzen.

Cinderella 2.0 kann für den Mathematik- und Physikunterricht genutzt und somit auch fächerübergreifend eingesetzt werden. Diese Arbeit hat gezeigt, dass das Programm didaktisch sinnvoll im Unterricht eingesetzt werden kann. Aber es ist noch nicht vollständig ausgereift, einige Probleme müssen überarbeitet werden. Cinderella 2.0 bietet als einziges DGS die Möglichkeit, physikalische Versuche zu simulieren. Die Umgebung CindyLab, die dafür zur Verfügung steht, ist sehr umfangreich, hat aber ihre beschriebenen Grenzen. Bspw. können nur Teilgebiete der Physik dargestellt werden. So wäre es hilfreich, elektrische Schaltungen konstruieren zu können. Teilweise fehlen wichtige Einstellungen auch ganz, wie etwa die Erstellung von Beschleunigungsvektoren. Vorteilhaft ist es bspw., dass man intuitiv einfache Versuche aufbauen kann, wie in 5.4.1 gezeigt wurde. Die Scriptsprache CindyScript erweitert diese Möglichkeiten noch deutlich, wobei in dieser Arbeit nicht untersucht wurde, ob CindyScript auch von Schülern genutzt werden kann. Ihre Komplexität spricht vorläufig eher dagegen. Für Lehrer hingegen bietet CindyScript viele Optionen zur individuellen Gestaltung ihrer Simulationen.

7. Fazit

Nachdem ich im letzten Abschnitt die Möglichkeiten und Grenzen von Cinderella 2.0 dargestellt habe, bleiben abschließend die Fragen zu klären, ob bei seinem Einsatz lerntheoretische Grundsätze berücksichtigt werden und inwiefern der Einsatz aus didaktisch-methodischer Sicht sinnvoll ist.

Im 1. Teil meiner Arbeit habe ich die lerntheoretischen Hintergründe beschrieben. Dabei sollte darauf geachtet werden, dass die Anforderungen der Lerntheorien bei dem Einsatz von Simulationen im Unterricht berücksichtigt werden. Nutzt man die Erkenntnisse der Cognitive-Load-Theorie (vgl. 4.1) kann mit einer durchdachten Aufarbeitung des Lernstoffs eine Senkung des Extraneous-Cognitive-Load erreicht werden. Durch die Förderung von Schemata kann die Kapazität für den Germane-Cognitive-Load möglichst groß gehalten werden, wodurch das Arbeitsgedächtnis die Vorgänge innerhalb einer Simulation effektiv aufnehmen kann. Dabei sollte bei allen Simulationen darauf geachtet werden, die beschriebenen Effekte anzuwenden bzw. zu vermeiden. In einer Simulation sollte kein langer Text stehen, der zum Verständnis benötigt wird, da sonst der *Split-Attention-Effekt* auftreten kann. Die Elementinteraktivität, also der Umfang und die Komplexität der Simulationen sollte an den Wissensstand der Schüler angepasst sein, damit der *Element-Interactivity-Effekt* nicht auftritt. Vielmehr sollte darauf geachtet werden, leicht zu verstehende bildliche Darstellungen einzusetzen, um den Schemaerwerb zu erleichtern. Der Lehrer wählt aus einer Vielzahl von Darstellungsformen diejenige aus, die ihm am geeignetsten erscheint, den Schemaaufbau bzw. den Modellaufbau möglichst effektiv zu gestalten. In der Mechanik ist es bspw. möglich, Bewegungen direkt mit Vektorpfeilen darzustellen (vgl. 3.2). Dadurch wird es für den Schüler leichter verständlich, diese Bewegungen nachzuvollziehen. Ein passendes Beispiel ist in 6.3 beschrieben. Die während eines Wurfes auf einen Körper wirkende Kraft wird durch einen Beschleunigungspfeil, dessen Richtung und Länge konstant nach unten zeigt, verdeutlicht.

Wie WILHELM¹ feststellte, nutzen 40% der Lehrer Simulationen zu Demonstrationszwecken. Hierbei sollten die Prinzipien der *kognitiven Theorie multimedialen Lernens* aus Kapitel 4.2 berücksichtigt werden. Dabei besagt z. B. das *zeitliche Kontiguitätsprinzip*, dass Simulationen mit gesprochenem Text unterlegt werden sollten, um besser verstanden werden zu können. Ebenso wichtig ist das *Kohärenzprinzip*, bei dem Unnötige Bilder, Optionen oder Beschriftungen in einer Simulation nicht vorhanden sein sollten, da sie den Schüler verwirren können.

Wie an den Beispielen gezeigt wurde, ist es mit Cinderella 2.0 möglich, lerntheoretische Erfordernisse zu berücksichtigen, um damit einen motivierenden Unterricht lerntheoretisch effektiv zu gestalten.

¹vgl. WILHELM und TREFZGER: *Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern*, S. 3.

An dieser Stelle kann eine klare Empfehlung für den Einsatz von Cinderella 2.0 abgegeben werden, wobei allerdings die beschriebenen Probleme nicht außer Acht gelassen werden dürfen. Cinderella 2.0 bietet aber jetzt schon die Möglichkeit, viele Simulation korrekt und ansprechend zu erstellen. Ein Programm wie Cinderella 2.0, das leicht verständlich ist und einen hohen Motivationsgrad besitzt, kann Schülern eine neue Sichtweise des Physikunterrichts vermitteln. Wenig motivierte Schüler können durch das Medium Computer vielleicht dazu angeregt werden, sich an einem interessant gestalteten Unterricht interessiert zu beteiligen.

8. Literaturverzeichnis

- BALLSTAEDT, STEFFEN-PETER (1997): *Wissensvermittlung: Die Gestaltung von Lernmaterial*. Weinheim: Psychologie Verlags Union - Beltz Verlag.
- BERGER, VEIT (2006): *Mit dem Computer im Unterricht modellieren*. In MIKELSKIS, HELMUT F. (HRSG.): *Praxishandbuch für die Sekundarstufe I und II*. Berlin: Cornelson Scriptor Verlag, S. 139–148.
- DEMTRÖDER, WOLFGANG (2003): *Experimentalphysik 1: Mechanik und Wärme*. Berlin, Heidelberg, New York: Springer-Verlag, S. 51–52.
- DEMTRÖDER, WOLFGANG (2006): *Experimentalphysik 2: Elektrizität und Optik*. Berlin, Heidelberg, New York: Springer-Verlag, S. 3.
- ELSCHENBROICH, HANS-JÜRGEN UND GAWLICK, THOMAS UND HENN, HANS-WOLFGANG UND HEINTZ, GABY UND RICHTER-GEBERT, JÜRGEN (2001): *Dynamische Geometrie-Software: Stand der Forschung und Perspektiven*. In ELSCHENBROICH, HANS-JÜRGEN UND GAWLICK, THOMAS UND HENN, HANS-WOLFGANG (HRSG.): *Zeichnung - Figur - Zugfigur*. Hildesheim Berlin: Verlag franzbecker, S. 13–20.
- GAGE, NATHANIEL L. UND BERLINER, DAVID C. (1996): *Pädagogische Psychologie*. 5. Auflage. Weinheim: Psychologie Verlags Union, S. 279–336.
- HASSELHORN, MARCUS (2009): *Pädagogische Psychologie*. Stuttgart: Kohlhammer, S. 49–60, S. 69–81.
- HECKER, ANDREAS: *Konstruktion an Sammellinsen*. [⟨URL: http://physik.psi-online.de/geogebra/Sammellinsen.html⟩](http://physik.psi-online.de/geogebra/Sammellinsen.html), online am 01.03.2010 um 16:00.
- HEUER, DIETER (2003): *Physikunterricht gestaltet mit Multimedia. Praxis der Naturwissenschaften - Physik in der Schule*, 52/3, S. 2–15.
- HOHENWARTER, MARKUS: *Geogebra*. [⟨URL: http://www.geogebra.org/de/wiki/index.php/GeoGebra⟩](http://www.geogebra.org/de/wiki/index.php/GeoGebra), online am 01.03.2010 um 15:30.
- HOHENWARTER, MARKUS: *GeoGebra - dynamische Mathematik für die Schule*. [⟨URL: http://www.geogebra.org/help/geogebra_flyer_de.pdf⟩](http://www.geogebra.org/help/geogebra_flyer_de.pdf), online am 01.03.2010 um 15:45.
- HOHENWARTER, MARKUS: *Was ist GeoGebra?* [⟨URL: http://www.geogebra.org/cms/index.php?option=com_content&task=blogcategory&id=67&Itemid=63⟩](http://www.geogebra.org/cms/index.php?option=com_content&task=blogcategory&id=67&Itemid=63), online am 01.03.2010 um 15:15.
- KIRCHER, ERNST UND GIRWIDZ, RAIMUND UND HÄUSSLER, PETER (2007): *Physikdidaktik - Theorie und Praxis*. Berlin: Springer.
- KOEPSSELL, ANDREAS UND TÖNNIES, DIRK (2007): *Dynamische Geometrie im Mathematikunterricht der Sekundarstufe 1*. Köln: Aulis Verlag Deubner.

- KORTENKAMP, ULRICH H. UND RICHTER-GEBERT, JÜRGEN (2006): *Cinderella.2 Math in Motion*. 2006 (URL: <http://cinderella.de/files/flyer-web-de.pdf>), online am 25.01.2010 um 12:05.
- LANDESINSTITUT FÜR SCHULENTWICKLUNG - LANDESBILDUNGSSERVER BADEN-WÜRTTEMBERG: *Bildungsstandards für Physik: Gymnasium - Klassen 6, 8, 10, Kursstufe*. (URL: http://www.bildung-staerkt-menschen.de/service/downloads/Bildungsstandards/Gym/Gym_Ph_bs.pdf), online am 05.02.2010 um 13:40.
- LUDWIG, MATTHIAS UND WEIGAND, HANS-GEORG (2009): *Konstruieren*. In WEIGAND, HANS-GEORG UND FILLER, ANDREAS UND SEBASTIAN KUNTZE, REINHARD HÖLZL ANS UND LUDWIG, MATTHIAS UND ROTH, JÜRGEN UND SCHMIDT-THIEME, BARBARA UND WITTMANN, GERALD (HRSG.): *Didaktik der Geometrie für die Sekundarstufe I*. Heidelberg: Spektrum Akademischer Verlag, S. 55–80.
- MAZARAKIS, ATHANASIOS (2007): *Cognitive Load Theory und der split-attention effect: Ein empirischer Test kognitionspsychologischer Erweiterungen*. Diplomarbeit Universität Mannheim, S. 13–17 (URL: http://madoc.bib.uni-mannheim.de/madoc/volltexte/2007/1612/pdf/DA_Mazarakis3.pdf).
- MECHLING, ROLAND: *EUKLID DynaGeo von innen*. (URL: <http://www.dynageo.de/scripte/discus/discus.pl>), online am 01.03.2010 um 16:15.
- NIEGEMANN, HELMUT UND DOMAGK, STEFFI UND HESSEL, SILVIA UND HEIN, ALEXANDRA UND HUPFER, MATTHIAS UND ZOBEL, ANNETT (2008): *Kompendium multimediales Lernen*. Berlin: Springer Verlag.
- RICHTER-GEBERT, JÜRGEN UND KORTENKAMP, ULRICH H.: *Cinderella.2 Documentation*. (URL: <http://doc.cinderella.de/tiki-index.php>), online am 05.02.2010 um 13:45.
- RICHTER-GEBERT, JÜRGEN UND KORTENKAMP, ULRICH H. (2001): *Benutzerhandbuch für die interaktive Geometrie-Software Cinderella*. Berlin, Heidelberg, New York: Springer-Verlag.
- RIEBER, LLOYD P. (2005): *Multimedia Learning in Games, Simulations and Microworlds*. In MAYER, RICHARD E. (HRSG.): *Cambridge Handbook of Multimedia Learning*. Cambridge: Cambridge University Press, S. 549–568.
- ROST, DETLEF H. (2006): *Handwörterbuch Pädagogische Psychologie*. Weinheim, Basel, Berlin: Beltz Verlag.
- ROTH, JÜRGEN (2008): *Dynamik von DGS – Wozu und wie sollte man sie nutzen?* In KORTENKAMP, ULRICH UND WEIGAND, HANS-GEORG UND WETH, THOMAS (HRSG.): *Informatische Ideen im Mathematikunterricht. Bericht über die 23. Arbeitstagung des Arbeitskreises „Mathematikunterricht und Informatik“ in der Gesellschaft für Didaktik der Mathematik e. V. vom 23. bis 25. September 2005 in Dillingen an der Donau*. Hildesheim: Verlag Franzbecker, S. 131–138.
- SCHNOTZ, WOLFGANG (1994): *Wissenserwerb mit logischen Bildern*. In WEIDENMANN, BERND

- (HRSG.): *Wissenserwerb mit Bildern. Instruktionale Bilder in Printmedien, Film/Video und Computerprogrammen*. Bern: Verlag Hans Huber, S. 95–147.
- SCHNOTZ, WOLFGANG (2005): *An integrated Model of Text and Picture Comprehension*. In MAYER, RICHARD E. (HRSG.): *Cambridge Handbook of Multimedia Learning*. Cambridge: Cambridge University Press, S. 49–69.
- SCHNOTZ, WOLFGANG UND BANNERT, MARIA (1999): *Einflüsse der Visualisierungsform auf die Konstruktion mentaler Modelle beim Text- und Bildverstehen*. *Zeitschrift für Experimentelle Psychologie*, Vol. 46, Nr. 3, (html-Ausgabe).
- STAATSIINSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrplan Physik 10*. [URL: http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=26439](http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=26439), online am 01.03.2010 um 17:00.
- STAATSIINSTITUT FÜR SCHULQUALITÄT UND BILDUNGSFORSCHUNG (ISB): *Lehrpläne Gymnasium G8*. [URL: http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=1](http://www.isb-gym8-lehrplan.de/contentserv/3.1.neu/g8.de/index.php?StoryID=1), online am 05.02.2010 um 13:35.
- WEIDENMANN, BERND (1991): *Lernen mit Bildmedien: Psychologische und didaktische Grundlagen*. Weinheim, Basel: Beltz-Verlag.
- WEIDENMANN, BERND (1994): *Informierende Bilder*. In WEIDENMANN, BERND (HRSG.): *Wissenserwerb mit Bildern. Instruktionale Bilder in Printmedien, Film/Video und Computerprogrammen*. Bern: Verlag Hans Huber, S. 9–58.
- WEIDENMANN, BERND (2005): *Multicodierung und Multimodalität im Lernprozess*. In ISSING, LUDWIG J. (HRSG.): *Information und Lernen mit Multimedia und Internet: Lehrbuch für Studium und Praxis*. Weinheim: Psychologie Verlags Union - Beltz Verlag, S. 45–64.
- WEIDENMANN, BERND UND KRAPP, ANDREAS (2001): *Pädagogische Psychologie*. 4. Auflage. Weinheim: Psychologie Verlags Union, S. 429.
- WILHELM, THOMAS (2005): *Konzeption und Evaluation eines Kinematik/Dynamik-Lehrgangs zur Veränderung von Schülervorstellungen mit Hilfe dynamisch ikonischer Repräsentationen und graphischer Modellbildung*. Dissertation an der Bayerischen Julius-Maximilians-Universität Würzburg, Würzburg, Deutschland, [URL: http://www.physik.uni-wuerzburg.de/~wilhelm/veroeffentlichung/Dissertation.pdf](http://www.physik.uni-wuerzburg.de/~wilhelm/veroeffentlichung/Dissertation.pdf).
- WILHELM, THOMAS (WS 2008/09): *Einführung in die Fachdidaktik I*. WS 2008/09 [URL: http://www.physik.uni-wuerzburg.de/~wilhelm/vorlesung/PPP_Kapitel15.pdf](http://www.physik.uni-wuerzburg.de/~wilhelm/vorlesung/PPP_Kapitel15.pdf).
- WILHELM, THOMAS UND TREFZGER, THOMAS (2010): *Erhebung zum Computereinsatz bei Physik-Gymnasiallehrern*. In GRÖTZEBAUCH, HELMUTH UND NORDMEIER, VOLKARD (HRSG.): *Didaktik der Physik - Hannover 2010, Frühjahrstagung der DPG*. Hannover.
- WISSENSCHAFTLICHER RAT DER DUDENREDAKTION (2000): *Duden, Das große Fremdwörterbuch*. Mannheim, Leipzig, Wien, Zürich.

A. Anhang

Angehängt ist eine CD auf der alle von mir erstellten Simulationen gespeichert sind, die in Kapitel 6 beschrieben wurden. Die aktuelle Version von Cinderella 2.0 kann als Shareware unter <http://cinderella.de/files/release/install.exe> heruntergeladen werden, um die Simulationen testen zu können. Der Aufbau der CD ist der Gleiche wie in Kapitel 6 der Arbeit. Es gibt Unterordner für jedes Unterkapitel in denen die einzelnen Simulationen zu finden sind. Zusätzlich ist jede Simulation auch als html-Seite in den jeweiligen Ordnern zu finden, um sie auch ohne eine Installation von Cinderella 2.0 starten zu können. Der Aufbau der CD ist wie folgt:

- Eigene Simulationen
 - Die Wellenmaschine
 - * Die Wellenmaschine für Transversalwellen
 - * Wellenmaschine für stehende Wellen
 - * Wellenmaschine für Longitudinalwellen
 - Elastische Stoßvorgänge
 - * Der elastische Stoß
 - * Impulserhaltung
 - Planetenbewegungen
 - * Ellipsen als Planetenbahnen (Erstes keplersches Gesetz)
 - * Zweites keplersches Gesetz
 - Elektrische Feldlinienbilder
 - * Elektrische Feldlinien
 - Das homogene Magnetfeld
 - * Einfacher Massenspektrograph
 - Wurfbewegungen
 - * Der waagerechte Wurf
 - * Der schräge Wurf
 - Strahlenoptik
 - * Die optische Bank

Danksagung

Bedanken möchte ich mich bei allen, die durch ihre Unterstützung dazu beigetragen haben, die Arbeit in dieser Form durchführen zu können.

Ein besonderer Dank richtet sich dabei an Johanna, die immer ein offenes Ohr für meine Ideen und Sorgen hatte und mir durch ihre Ermutigung ständig weitergeholfen hat.

Danke möchte auch Hans-Peter sagen, der mir gerade in den letzten Zügen dieser Arbeit eine große Unterstützung war.

Schließlich sei allen gedankt, die Teile der Arbeit Korrektur gelesen haben.

Ganz besonders möchte ich mich bei dem Betreuer meiner Arbeit AR Dr. Thomas Wilhelm bedanken. Seine Unterstützung bei der Realisierung meiner Ideen ging weit über das selbstverständliche Maß hinaus. So stand er mir immer mit Rat und Tat zur Seite und nahm sich auch Zeit wenn keine war, um meine Fragen zu beantworten.

Selbstständigkeitserklärung

Hiermit versichere ich,

Sebastian Krohne, Matrikel-Nummer: **1465359**,

dass ich diese Hausarbeit für die Staatsprüfung für das Lehramt am Gymnasium mit dem Thema

Physiksimulationen mit der dynamischen Geometriesoftware Cinderella 2.0

selbstständig angefertigt und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe. Die Zeichnungen und bildlichen Darstellungen habe ich – soweit nicht anders vermerkt – selbst erstellt.

(Ort, Datum)

(Unterschrift)