Deutsche JuniorAkademien

JuniorAkademie Adelsheim

11. Science Academy Baden-Württemberg 2013

Astronomie

Chemie

Germanistik/Geschichte

Mathematik

Physik/Informatik

TheoPrax

H.W. & J. Hector Stiftung

Regierungspräsidium Karlsruhe
Abteilung 7 – Schule und Bildung

Dokumentation der JuniorAkademie Adelsheim 2013

11. Science Academy Baden-Württemberg

Träger und Veranstalter der JuniorAkademie Adelsheim 2013:
Regierungspräsidium Karlsruhe
Abteilung 7 –Schule und Bildung–
Hebelstr. 2
76133 Karlsruhe
Tel.: (0721) 926 4454
Fax.: (0721) 933 40270
E-Mail: georg.wilke@scienceacademy.de
petra.zachmann@scienceacademy.de
www.scienceacademy.de

Die in dieser Dokumentation enthaltenen Texte wurden von den Kurs- und Akademieleitern sowie den Teilnehmern der 11. JuniorAkademie Adelsheim 2013 erstellt. Anschließend wurde das Dokument mit Hilfe von LaTeX gesetzt.

Gesamtredaktion und Layout: Jörg Richter
Druck und Bindung: RTB Reprotechnik Bensheim

Vorwort

Mittlerweile sind es schon 11 Jahre, in welchen sich rund 70 Schülerinnen und Schüler aus ganz Baden-Württemberg gemeinsam mit dem 30-köpfigen Leitungsteam im Rahmen der Science Academy Baden-Württemberg in Adelsheim treffen. Sie verbringen dort auf dem Eckenberg am Landesschulzentrum für Umwelterziehung das Eröffnungswochenende, die 14-tägige Akademie im Sommer und das Dokumentationswochenende.

In dieser Zeit wächst aus den einzelnen Teilnehmern eine große Gemeinschaft, die auch über die Zeit der Akademie hinaus bestehen bleibt. Jeder Einzelne beschäftigt sich während dieser Zeit nicht nur mit den naturwissenschaftlichen Inhalten der Kurse, sondern entwickelt sich auch persönlich weiter.

Um der Akademie über die Kursarbeit hinaus einen Rahmen zu geben, steht sie in jedem Jahr unter einem übergeordneten Motto.

In diesem Jahr war dieses Motto das Thema „Licht“. Natürlich ist Licht mit unglaublich vielen Assoziationen verbunden, die in sehr verschiedene Richtungen gehen. Für uns war Licht während der Akademie mit Erkenntnissen und mit Lichtblicken verbunden. Wir alle haben viel Neues gelernt und hatten oft wunderbare Erlebnisse, die sich in diesen Lichtblicken wiederfinden. Ein anderer Aspekt für uns war das Licht in Form einer Flamme, die uns während der gesamten Akademie begleitete, und die hoffentlich auch nach der Akademie in uns allen weiter brennen wird. Jeder der Teilnehmer weiß, dass es nicht die Kurse allein sind, die die einzigartige Akademieatmosphäre schaffen. Auch diesen Aspekt spiegelt das Licht wider. Kommt wie im Bild Licht verschiedener Farben zusammen, so entsteht etwas Neues: helles, weißes Licht.

Wir alle zusammen lassen das Akademielicht entstehen, und auch wenn wir dieses Licht nun wieder in seine Farben getrennt haben, so können wir euch mit Sicherheit garantieren: es findet auch wieder zusammen. Ihr habt während unserer Zeit in der Akademie Freundschaften geschlossen, Erlebnisse gehabt und Erkenntnisse gewonnen, die euch keiner mehr nehmen kann. Das Akademie-„Licht“ wird euch von nun an begleiten und euch vielleicht auch auf das ein oder andere Projekt aufmerksam machen. Geht mit offenen Augen durchs Leben und achtet auf neue Möglichkeiten, die sich euch auftun, und vor allem habt den Mut, diese auch wahrzunehmen.

Wir wünschen euch alles Liebe und Gute für das, was als nächstes auf euch zukommt, und wir freuen uns darauf, euch – in egal welchem Zusammenhang – wieder zu sehen, vielleicht ja sogar in zwei oder drei Jahren hier in Adelsheim.

Und nun wünschen wir euch viel Spaß beim Lesen und Schmökern!

Eure/Ihre Akademieleitung

Patricia Keppler	Wendelin Wiedemer
Patricia Keppler (Assistenz)	Wendelin Wiedemer (Assistenz)
Georg Wilke	Petra Zachmann
Georg Wilke	Dr. Petra Zachmann

Inhaltsverzeichnis

Kurs 5 – Physik/Informatik: Billard – Auf der Suche nach dem perfekten Stoß

Vorwort

Die Kursleiter

Man nehme 13 Schüler aus ganz Baden-Württemberg. Man füge zwei kompetente Kursleiter und eine verrückte Schülermentorin hinzu. Man gebe weiterhin ein zu lösendes Problem, Computer und – ganz wichtig – viele Gummibärchen und andere Süßigkeiten dazu. Man rühre dann einen guten Schuss Fantasie und Motivation darunter. Nun fülle man das Ganze mit Spaß und Information und lasse es zwei Wochen gehen. Schon hat man unseren Physik-Informatik-Kurs 2013 und damit verbunden: ein fertiges Billardspiel für den Computer!

Anstatt faul am Strand zu liegen, programmierten unsere Kursteilnehmer in ihren wohlverdienten Sommerferien ein realistisches Billardspiel samt Queue, Kugeln und Tisch. Bis zum fertigen Spiel war es ein langer Weg, da erst eine grundlegende, aber auch fortführende Basis geschaffen werden musste. Trotz intensiver Arbeit in den Bereichen Physik, Mathematik und Informatik war die Stimmung die ganze Zeit über klasse. Neben einem starken Gruppenzusammenhalt ist am Ende ein Spiel entstanden, das sich wirklich sehen lassen kann.

Wir Kursleiter möchten euch an dieser Stelle herzlich für eure tolle Mitarbeit im Sommer danken.

Unser Kurs

Luisa Feifel

Anne wirkte auf den ersten Blick eher ruhig und zurückhaltend. Wir haben aber sehr schnell festgestellt, dass sie das definitiv nicht ist. Wenn man sie besser kennenlernt, merkt man, dass sie es „faustdick hinter den Ohren hat“ und gern mal den einen oder anderen frechen Kommentar von sich gibt. Außerdem ist sie eine Spitzen-Klarinettenspielerin, was jeder, der sie beim Hausmusikabend oder am Abschlussabend gehört hat, nur bezeugen kann.

Antonia verschmähte die von uns allen so heiß geliebten Gummibärchen ebenso wie die Schokolade und überhaupt alles, wovon andere gar nicht genug bekommen konnten. Warum, war uns anderen ein Rätsel. Außerdem hat sie sich als Chef-Designerin des „Bleistift-Queues“ für immer in unserem Spiel verewigt.

Axel kam manchmal aus unerfindlichen Gründen zu spät, war dafür im Kurs aber meist doppelt so gut gelaunt. Man konnte immer bestens mit ihm zusammenarbeiten, solange genügend Vorrat an Gummibärchen vorhanden war. Er schaffte es immer wieder, uns durch seine lustigen Kommentare zum Lachen zu bringen. Dies ging so lange, bis ihn seine treuen Groupies zum Mittagessen abholten.

Corinna – auch bekannt als Coco – war einfach immer gut gelaunt, auch wenn andere morgens gedanklich noch im Bett lagen und sich fragten, wie sie das eigentlich anstellte. Aber wenn die anderen dann irgendwann aufgestanden waren, stellten sie fest, dass Coco eine total coole Freundin ist, und man mit ihr garantiert immer genügend Spaß hat.

Daniela spielte Klavier und Gitarre in der Akademieband. Außerdem designte sie mit viel Kreativität eine laufende Billardkugel für unser Kurs-T-Shirt und programmierte zusammen mit Luisa und Antonia die Punkteanzeige mit einrollenden Kugeln. Die Zusammenarbeit mir ihr funktionierte immer einwandfrei, da sie als Freundin einfach unschlagbar ist. Mit ihren vielen tollen Ideen war sie aus unserem Kurs nicht wegzudenken.

Elias fand jeden noch so kleinen Fehler und ließ nie etwas außer Acht. Er war für jede Teamarbeit immer eine Bereicherung und half den anderen Teilnehmern gerne mit zusätzlichen Erklärungen, die immer „ganz einfach“ waren. Er hatte für jedes Problem eine Idee, um es zu lösen. Als tapferer (manchmal auch einziger) Zeitungs-KüA-Teilnehmer hielt er uns immer über Wetter, Politik und Wirtschaft auf dem Laufenden.

Eugen war am Eröffnungswochenende leider noch nicht da und so merkten wir erst im Sommer, was uns eigentlich die ganze Zeit gefehlt hatte. Nämlich Eugens gute Laune, seine freundliche Art und sein Fachwissen, von dem er gleich eine ganze Menge mitbrachte. Durch das Programmieren der Zielhilfe machte er unser Billardspiel auch für blutige Anfänger spielbar.

Fabian war, wenn Werwolf gespielt wurde, eigentlich immer anzutreffen. Auch wenn es anfangs keiner vermutete, so stellte sich doch nach einiger Zeit heraus, dass Fabian einer der hinterlistigsten Werwölfe sein konnte und, wahrscheinlich auf Grund seines unschuldigen Aussehens und seiner ruhigen Art, nur selten als einer entlarvt wurde. Leider hatte er das Pech, während der Akademie von einer Hornisse, die sich fieser Weise in seinem T-Shirt versteckt hatte, in die Hand gestochen zu werden, was ihn dann für einige Tage am Cello Spielen hinderte.

Luca widmete sich hingebungsvoll dem Suchen und Herstellen von Texturen. Daher haben wir ihm schlussendlich zu verdanken, dass unser Billardtisch jetzt so richtig cool aussieht. Er war immer sehr engagiert, egal ob bei der Organisation des Bergfests oder seiner Comic-KüA. Seine Angewohnheit barfuß herumzulaufen begründete er zwar damit, seine Flip Flops seien kaputt gegangen, aber wir kamen zu dem Schluss, dass er einfach nur ein großer Fan von Mathematik-Kursleiter Damián ist.

Lucas war unser „Physik-King“ und hat es immer wieder geschafft, uns durch sein charmantes „Mr. Burns-Lächeln“, seine Vorliebe für Hipp-Babybrei und seine einzigartig lustige Art zum Lachen zu bringen. Ohne ihn wären unsere Kursstunden sicherlich nur halb so lustig gewesen, und das Wort „floaten“ wäre wohl nie erfunden worden.

Luisa bestand zwar auf ein einfaches „unbeschreiblich“ in ihrer Beschreibung, wir finden aber, dass es über sie noch viel mehr zu sagen gibt. Mit ihrer lockeren, kumpelhaften Art und ihren schlagfertigen Sprüchen zauberte sie uns immer ein Grinsen ins Gesicht. Da sie sonst einen sehr sportlichen Look hat, der ihre Begeisterung für Fußball unterstreicht, überraschte sie uns mit ihrem „Abschlussabendkleid“. In diesem und mit ihren geflochtenen Zöpfen sah sie einfach bezaubernd aus.

Marvin hat uns durch sein Schwäbisch jeden Probevortrag versüßt und war auch sonst als Teamkollege immer total zuverlässig. Dass er mit einem „Stecken“ Billard spielt, wird jetzt wohl so schnell keiner mehr vergessen. Trotz seiner überdurchschnittlichen Billardskills reichte es dank Sophie nicht für den Sieg beim Billardturnier. Da er aber über eine gehörige Portion Selbstironie verfügt, versaute ihm auch das nicht die Stimmung.

Timo versorgte uns bestens mit seinem Geburtstagskuchen und war ebenfalls einer der Top-Programmierer in unserem Kurs. Aber auch mit seiner E-Gitarre in der Akademieband oder als Tänzer hatte er einiges zu bieten. Egal was passierte, Timo war immer gut drauf. Dabei konnte es schon mal passieren, dass Billardkugeln personifiziert wurden und dann plötzlich laufen, springen oder sogar hüpfen konnten.

Unsere Kursleiter

Alex begrüßte uns jeden Morgen mit perfekt gestylter Frisur und den Worten „Hallooo erst mal“ und hatte dann meistens, wenn es nicht gerade verschollen war, sein Nutellaglas vom Frühstück dabei. Er war im Kurs für die Physik zuständig und kann richtig gut erklären, auch wenn er danach immer meinte: „Jetzt habe ich euch wohl mehr verwirrt als was erklärt.“ Dass er nicht nur Ultimate Frisbee spielen und die besten PowerPoint-Präsentationen machen, sondern auch singen kann, hat er uns als „Pool-Brother“ bewiesen. Seine abendlichen Gute-Nacht-Geschichten sorgten zwar nicht dafür, dass danach alle brav ins Bett gingen, aber viel Gelächter und ein randvolles Treppenhaus waren ihm immer gewiss.

Daniel sorgte immer dafür, dass die Informatik, ob im Kurs oder in den Überschriften unserer Präsentationen, auf keinen Fall zu kurz kam. Er war immer hilfsbereit, und wenn wir gerade keine Variablen „vergewaltigt“ hatten, auch immer gut gelaunt. Beim „Wildhüten“ schaffte er es, ohne eine Miene zu verziehen, alle Teilnehmer seriös und qualifiziert auszubilden. Außerdem hat er auf der Akademie sein Talent für das Gitarrespielen entdeckt und als „Pool-Brother“ sein Können unter Beweis gestellt. Jeder aus unserem Kurs ist wohl auf die Schüler eifersüchtig, die sich weiterhin jeden Morgen aufs Neue auf sein „Mathe macht glücklich“ T-Shirt, zu dem jetzt auch noch ein „Mathe macht schön“ T-Shirt hinzugekommen ist, und seinen Mathe-Unterricht freuen dürfen.

Sophie brachte uns einige der mathematischen Grundlagen bei und war für das „Gutaussehen“ und die Beschaffung von Gummibärchen zuständig. Sie hatte für alle immer ein offenes Ohr und für jedes Problem eine Lösung. Man konnte wirklich alles mit ihr machen, sei es über die Kursleiter zu lästern oder neue Verkupplungspläne zu schmieden. Mit ihren frechen Sprüchen, die sie immer und überall parat hatte, und ihrer liebevollen Art Alex zu mobben, war sie der perfekte Ausgleich zu den zwei Männern und sorgte dafür, dass die Stimmung im Kurs dauerhaft gut war.

Unser Spiel

Eugen Dizer

Die aktuelle Version unseres Billardspiels besteht aus insgesamt 16 Kugeln, einem Billardtisch, mehreren Queue-Designs, einem Stärkebalken und vielem mehr. Im Folgenden wird unser Spiel beschrieben und erläutert.

Bei unserer Billardvariante, auch Pool-Billard genannt, gibt es sieben halbe und sieben volle Kugeln sowie die schwarze Acht. Außerdem enthält das Programm eine weiße Kugel, mit der man die anderen Kugeln anspielt. Ziel des Spiels ist es, die eigenen Kugeln und anschließend die schwarze Acht einzulochen. Das heißt man muss acht Kugeln lochen, um zu gewinnen – daher der Name „8-Ball-Spiel“. Realitätsgetreu sind die 15 farbigen Kugeln am Anfang des Spiels in einem Dreieck angeordnet. Nach dem Anstoßen der Kugeln mit dem Queue rollen sie über den Tisch und prallen physikalisch korrekt an den Banden ab.

Unser Billardspiel vor dem ersten Anstoß.

Zu Beginn der zwei Wochen im Sommer beschäftigten wir uns mit dem Verschwinden der Kugeln in den Löchern. Durch geschickte Befehle und dem Satz des Pythagoras, der zur „Berührungserkennung“ dient, konnten wir die Kugeln dazu bringen, in den Löchern zu verschwinden. Für ein realistischeres Verschwinden ließen wir die Kugeln später in den Löchern langsam kleiner werden. So sieht es aus, als würden die Kugeln wirklich in die Löcher fallen. Natürlich soll die weiße Kugel nicht verschwinden, sondern wieder auftauchen, wenn sie gelocht wurde. Deshalb setzt man diese nach dem Einlochen wieder auf den Tisch und kann sie entlang einer gedachten Linie positionieren.

Als Nächstes arbeiteten wir an der Kollision zweier Kugeln. Hierfür benötigten wir wieder den Satz des Pythagoras, um zu erkennen, ob sich zwei Kugeln berühren. Mit den an zwei Tagen erlernten physikalischen und mathematischen Grundlagen gelang es uns, die Kollision zweier Kugeln in das Programm einzubauen.

Nun machten wir uns zur Aufgabe, einen Queue zu programmieren, mit dem man die weiße Kugel anstoßen kann. Wir haben insgesamt drei Queues erstellt (einen „echten“ Queue, einen Designer- und einen Bleistiftqueue), die man mit den Tasten 1, 2 und 3 auswählen kann. Der Queue erscheint nur, wenn alle Kugeln ruhen. Er ist an der weißen Kugel ausgerichtet und man kann ihn durch Mausbewegung steuern und durch einen Mausklick „aufziehen“. Je länger man die Maustaste gedrückt hält, desto stärker stößt er die weiße Kugel an. Zur Visualisierung der Stoßkraft dient der Stärkebalken an der rechten Seite des Spielfeldes.

Damit die Kugeln nach dem Stoß nicht ewig weiterrollen, werden sie durch entsprechende Reibungskräfte gebremst. Mit Hilfe eines von uns selbst entworfenen Experiments haben wir einen realistischen Wert für die negative Beschleunigung der Rollreibung ermitteln können. Anschließend setzten wir den experimentell ermittelten Wert in unser Programm ein und erhielten eine realitätsnahe Verlangsamung der Kugeln. Des Weiteren werden die Kugeln auf realistische Weise gebremst, wenn sie an der Bande anstoßen. Ein weiteres Feature unseres Spiels ist die Zielhilfe. Ungeübte Anfänger können diese einschalten, um damit das Spiel zu vereinfachen. Die Zielhilfe zeigt den Weg der weißen Kugel über eine Bande an.

Gegen Ende der Akademie wollten wir dem monotonen Spiel ein realistischeres Aussehen verleihen. Der Billardtisch und die Kugeln sind dem echten Billard nachempfunden. Hierbei haben wir sowohl die Maße des Tischs als auch die Position der Löcher im Internet recherchiert. Das Tischtuch haben wir von unserem eigenen Billardtisch abfotografiert und den Tisch sowie die Kugeln mit unseren eigenen Texturen „tapeziert“. Zudem haben wir allen Objekten Schatten verliehen, um einen räumlichen Eindruck hervorzurufen.

Fortgeschrittener Spielverlauf mit Bleistift-Queue.

Natürlich gibt es bei unserem Programm auch Regeln – wie bei einem echten Spiel. Es spielen immer zwei Spieler gegeneinander. Man versucht, als Erster die eigenen Kugeln zu versenken. Dabei wechseln sich die Spieler nach jedem Stoß ab. Um einen besseren Überblick zu erhalten, werden die gelochten Kugeln oben angezeigt und mitgezählt. Wenn man aus Versehen die weiße Kugel locht, so wird sie auf den Tisch zurückgesetzt, und der Nächste ist dran. Locht man am Ende des Spiels – oder unglücklicherweise schon davor – die schwarze Kugel ein, so ist das Spiel vorbei und es erscheint „Game Over“. Tritt dieser Fall ein, so kann man nichts mehr machen, außer das Spiel neu zu starten.

Die schwarze Kugel wurde zu früh im Spiel gelocht.

Bevor wir allerdings ein solch komplexes Billardspiel entwickeln konnten, mussten wir zuerst einige Programmiergrundlagen erlernen.

Programmierung in C++

Elias Hofmann

Beim Programmieren bringt man einem Computer bei, was er machen soll. Der Zweck eines Computerprogramms kann ganz unterschiedlicher Art sein. Zum Beispiel könnte man ein Computerspiel programmieren, oder ein Programm, das Temperaturen von Grad Fahrenheit in Grad Celsius umrechnen kann. Man verwendet dazu eine Programmiersprache. Mit Hilfe dieser Programmiersprache lässt sich ein für uns verständlicher Code schreiben, der vom Computer in eine Folge von Nullen und Einsen – den Maschinencode – übersetzt wird. Diesen Vorgang nennt man *kompilieren*. Wir arbeiteten mit der Programmiersprache C++, die zu den am weitesten verbreiteten Programmiersprachen zählt. Außerdem unterstützt C++ das sogenannte *objektorientierte Programmieren*, aber dazu später mehr.

Ausgabe

Damit das Ergebnis einer Berechnung des Computers von Nutzen sein kann, muss dieses Ergebnis ausgegeben und so dem Nutzer zugänglich gemacht werden können. Diese Ausgabe kann grafisch (z. B.: Bild eines Billardtisches) oder eine einfache Textausgabe sein. Hier ein Beispiel für eine Ausgabe von Text:

```
cout << "Physik" << endl;
```

Dieser Befehl lässt den Computer das Wort „Physik“ ausgeben. `cout` bedeutet hierbei ausgeben. `Physik` ist das, was ausgegeben wird und `endl` sorgt dafür, dass eine neue Zeile begonnen wird. Am Ende eines Befehls steht ein Semikolon.

Variablen

Oft ist es notwendig, bestimmte Informationen (Zahlen oder Texte) speichern zu können, um sie später an einer anderen Stelle des Programms wieder zu verwenden. Hierzu verwendet man sogenannte *Variablen*. Das kann man sich vorstellen wie einen Schrank mit Schubladen: Man gibt den Schubladen (den Variablen) Namen und kann anschließend Zahlen oder Texte hinein packen (der Variable einen Wert zuweisen), um sie später wieder zu verwenden.

```
int Zahl = 8;
```

Dieser Befehl lässt den Computer eine Variable mit Namen `Zahl` anlegen, der anschließend der Wert 8 zugewiesen wird. Hierbei bedeutet `int` (kurz für Integer), dass in der Variable ganze Zahlen gespeichert werden können.

Eingabe

Der Computer kann aber nicht nur Dinge ausgeben, sondern auch über die Eingabe Informationen vom Benutzer erhalten, um mit diesem zu interagieren:

```
int Alter = 0;
cin >> Alter;
```

Mit diesem Befehl fragt der Computer das Alter des Nutzers ab. Zunächst legt der Computer die Variable `Alter` an und belegt sie mit dem Wert `0`. Durch den Befehl `cin » Alter;` kann der Benutzer sein Alter eingeben und der Computer kann Berechnungen damit durchführen.

If-Abfragen

Damit ein Computerprogramm je nach „Situation“ unterschiedlich ablaufen und so auf bestimmte Sachverhalte reagieren kann, gibt es sogenannte *If-Abfragen*:

```
if(Alter < 18) {
  ...
}
```

Falls die Bedingung in den runden Klammern erfüllt ist, wird der Befehl in den geschweiften Klammern ausgeführt, sonst nicht.

Zählschleife

Manchmal muss ein Befehl oder ein Block von Befehlen mehrmals hintereinander mit einer nur geringen oder auch überhaupt keiner Abweichung ausgeführt werden. Es wäre sehr viel Arbeit, diese Befehle immer und immer wieder zu programmieren, damit sie entsprechend oft ausgeführt werden. Aus diesem Grund gibt es *Schleifen*, z. B. die *Zählschleife*, auch *for-Schleife* genannt:

```
for (int i = 1; i <= 10; i++) {
  cout << i << endl;
}
```

Zu Beginn der Schleife wird die *Zählvariable* `i` auf den Wert `1` gesetzt. Die Schleife wird ausgeführt, solange die Bedingung `i <= 10` erfüllt ist, und bei jedem Schleifendurchlauf wird die Zählvariable um eins erhöht (`i++`). In diesem Fall wird die Schleife demnach zehnmal durchlaufen.

Kommentare

Große Programme werden oftmals unübersichtlich. Um den Überblick zu behalten, kann man Kommentare einfügen. Diese enthalten Erläuterungen zu den Befehlen des Programms.

```
// Ich bin ein Kommentar!
```

Kommentare werden durch zwei Schrägstriche eingeleitet und beim Kompilieren des Programms einfach übersprungen.

Wir hatten diese Grundlagen während des Eröffnungswochenendes gelernt und in den folgenden Wochen mehrere Hausaufgaben bearbeitet, sodass wir zu Beginn der Akademie im Sommer mit den nötigen Programmierkenntnissen starten konnten.

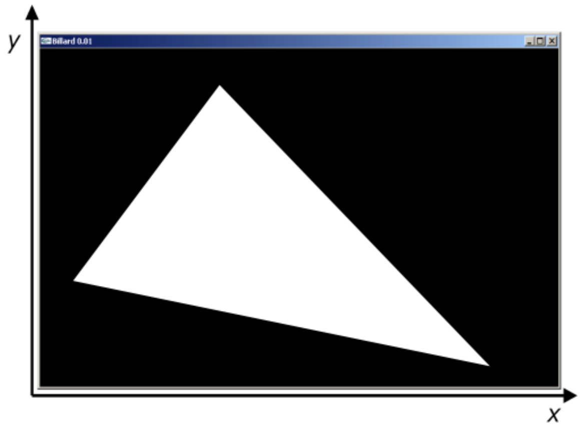
Grafisches Programmieren

Fabian Kolar

Neben der Ausgabe eines Textes mussten wir zudem einige Grundlagen der grafischen Programmierung erlernen und verstehen. Am Eröffnungswochenende hatten wir bereits gelernt, wie man ein Programm beispielsweise Dreiecke oder Kreise zeichnen lassen kann. Hierfür hatten wir die Grafikbibliothek *OpenGL* verwendet, die viele Befehle beinhaltet und für eine effiziente Darstellung der Grafiken sorgt, da sie eng mit der Grafikkarte zusammenarbeitet. Die Befehle

```
glBegin(GL_TRIANGLES); {
  glVertex2f(50.0, 150.0);
  glVertex2f(600.0, 50.0);
  glVertex2f(300.0, 420.0);
} glEnd();
```

ermöglichen beispielsweise das Zeichnen eines Dreiecks.



Dreieck im Koordinatensystem.

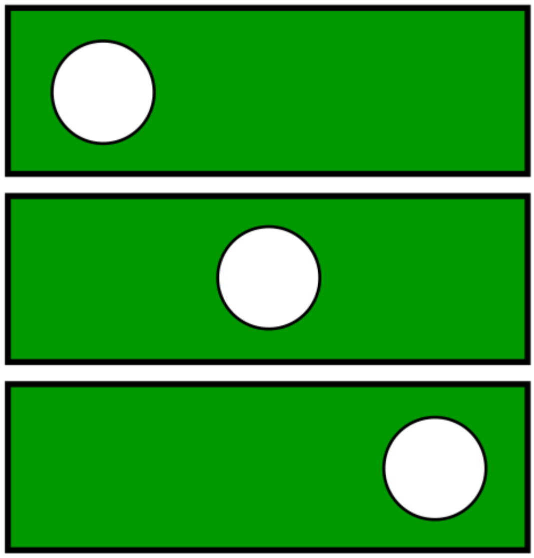
Mit dem Befehl `glVertex2f(x, y);` werden die Koordinaten der Eckpunkte des Dreiecks im entsprechenden Koordinatensystem festgelegt.

Bewegte Bilder

Im Fall unseres Billardspiels sollen sich die Kugeln natürlich bewegen. Das funktioniert mit der sogenannten *idle-Funktion*, die immer aufgerufen wird, wenn das aktuelle Bild fertig gezeichnet wurde. Dann wird die neue Position der Kugel berechnet:

```
x = x + vx * t;
```

Mathematisch sieht die „Formel“ zwar inkorrekt aus, aber in C++ symbolisiert das „=“ eine Zuweisung. Dies bedeutet, dass der Wert rechts vom „=“-Zeichen ausgerechnet und der Variable links vom „=“-Zeichen zugewiesen wird. Wenn das Bild also neu gezeichnet wird, hat sich der Mittelpunkt $(x \mid y)$ der Kugel verändert. Ähnlich wie bei einem Daumenkino wird eine Bewegung der Kugel sichtbar.



Bilderfolge einer sich bewegenden Kugel.

Wenn man das entsprechend mit dem y-Wert macht, kann sich die Kugel in beliebige Richtungen bewegen.

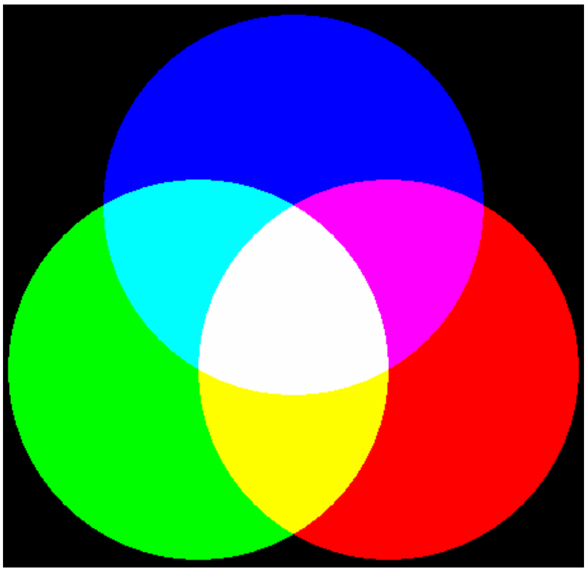
Farben und Texturen

Luca Schweizer

Wie man ein Dreieck zeichnen kann, wurde bereits im vorigen Abschnitt beschrieben. Dieses bekommt jetzt noch eine Farbe. Genauer gesagt, jeder Eckpunkt des Dreiecks bekommt eine Farbe, die jeweils durch den Befehl

```
glColor3f(1.0, 0.3, 0.0);
```

festgelegt wird. Dabei bedeutet `gl`, dass der Befehl aus der *Graphics Library* ist. `Color` ist der Befehl, der entsprechend einfärbt. `3f` heißt, dass dieser Befehl drei Fließkommawerte (`float`) als Parameter erhält, nämlich den Anteil der roten, grünen und blauen Farbe. `(1.0, 0.3, 0.0)` gibt demnach den Farben Rot, Grün und Blau ihren Wert. Dabei kann man einen Wert zwischen 0 und 1 wählen. In diesem Beispiel wird viel Rot mit etwas Grün gemischt – es entsteht Orange. Wegen der Grundfarben bezeichnet man das Ganze als *RGB-System*. Der Computer setzt alle Farben, die auf dem Monitor ausgegeben werden, aus diesen Grundfarben zusammen. In jedem Pixel des Bildschirms leuchtet dabei ein rotes, ein grünes und ein blaues Lämpchen entsprechend stark, da unsere Augen zur Farbwahrnehmung unterschiedliche Zäpfchen benutzen, die rotes, grünes oder blaues Licht erkennen können.



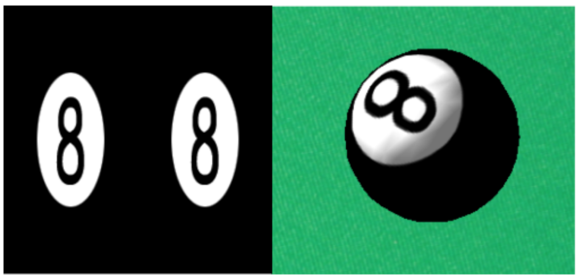
Der Farbkreis der additiven Farbmischung.

Die Farben auf unseren Bildschirmen entstehen, indem die Lämpchen unterschiedlich hell leuchten. Schaltet man in einem Pixel alle Farben mit voller Intensität an, erscheint dieser weiß. Da die einzelnen Farben durch Addition der Grundfarben entstehen, nennt sich das System *additive Farbmischung*. Dabei werden die einzelnen Lichter entsprechend addiert – klar, drei Taschenlampen derselben Sorte sind heller als nur eine. Wir haben im Kunstunterricht allerdings etwas anderes gelernt, nämlich, dass alle Farben aus Cyan, Magenta und Gelb gemischt werden – je mehr Farbe man verwendet, desto dunkler wird das Resultat. Hier wird aber mit Farbpigmenten und Papier gearbeitet. Man spricht daher von *subtraktiver Farbmischung*.

Allerdings besteht unsere Farbwelt nicht nur aus „klaren“ Farben, sondern auch aus Übergängen. Will man solche programmieren, ordnet man jedem Eckpunkt eines Dreiecks (oder eines anderen Vielecks) eine separate Farbe zu. Der Computer berechnet dann automatisch die Übergänge.

Der Farbverlauf wird automatisch berechnet, wenn jedem Eckpunkt eine separate Farbe zugewiesen wird.

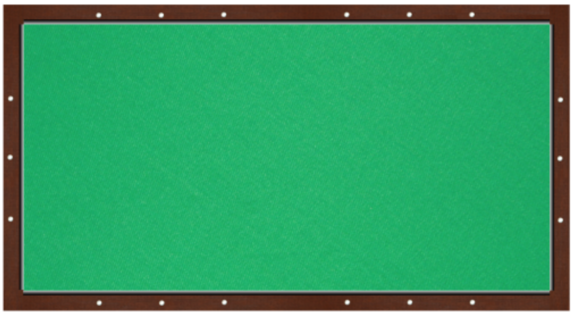
Solche Farbverläufe sehen aber bald langweilig und unrealistisch aus. Wenn beispielsweise ein „richtiger“ Baum gezeichnet werden soll, sieht ein echtes Foto sicherlich besser aus, als eine grüne Kugel auf einem braunen Rechteck. Dafür verwendet man *Texturen* – Bilder, die man in das Programm einfügt. Diese Texturen werden wie eine Art Tapete über die Rechtecke oder auch unsere Billardkugeln gelegt.



Textur der schwarzen Kugel und texturierte Kugel.

Zunächst wird die Fläche definiert, auf die die Textur gezeichnet werden soll. Wir wählen der Einfachheit halber ein Rechteck. Jedem Eckpunkt des Vierecks wird dann eine sogenannte *Texturkoordinate* zugewiesen, damit klar ist, welche Ecke des Bildes auf welche Ecke des Rechtecks „geklebt“ werden soll. Hierzu dient der Befehl

```
glTexCoord2f(1.0, 1.0);
```



Mit Hilfe der entsprechenden Texturkoordinaten, kann das Bild auf eine Fläche „aufgeklebt“ werden.

Dabei entspricht (0 | 0) der linken unteren Ecke des Bildes und (1 | 1) der rechten oberen. Gibt man statt (1 | 1) beispielsweise (0.5 | 1) ein, so wird nur die Hälfte des Bildes auf das Viereck „geklebt“.

Objektorientierte Programmierung

Antonia Seifert

Mit diesen Kenntnissen könnte man sicher fast ein ganzes Billardspiel programmieren, aber es wäre bald so komplex, dass man es kaum noch verstehen würde. Abhilfe schafft die *Objektorientierte Programmierung*.

Der Computer kennt bereits verschiedene einfache Variablentypen, wie zum Beispiel `int` (ganze Zahlen) und `float` (Dezimalzahlen). Das Objektorientierte Programmieren ermöglicht es, weitere Variablentypen zu „erschaffen“ – sogenannte Klassen (`class`). Diese werden üblicherweise in separaten Dateien programmiert. Eine solche Klasse könnte beispielsweise so aussehen:

```
class Kugel {

// Eigenschaften:
private:
    // Koordinaten:
    float x;
    float y;

    // Geschwindigkeiten:
    float vx;
    float vy;

    float radius;

    // Farbe:
    float r;
    float g;
    float b;

// Methoden:
public:
    void init(...);
    void bewegen(float t);
    void zeichnen();
};
```

Diese Klasse stellt den Bauplan des Variablentyps `Kugel` dar. Sie besteht, wie alle Klassen, aus zwei Teilen: `private` und `public`. Vereinfacht gesagt werden unter `private` alle Eigenschaften als Variablen deklariert, unter `public` stehen sämtliche Methoden der Klasse als entsprechende Funktionen. Die erste Funktion ist dabei üblicherweise `void init (...)`. In dieser Funktion werden die Eigenschaften des Objekts zu Programmstart mit Werten initialisiert. Die Methoden programmiert man dann in einer weiteren Datei. Dort programmiert man beispielsweise, wie sich die Kugel bewegen soll:

```
void Kugel::bewegen(float t) {

    x = x + vx * t;
    y = y + vy * t;

}
```

Im Hauptprogramm wird dann mit dem Befehl

```
Kugel kugel;
```

ein Objekt der Klasse `Kugel` mit dem Namen `kugel` erzeugt. Dann muss man nur noch die richtige Funktion an der richtigen Stelle aufrufen, z. B.

```
kugel.bewegen(t);
```

oder

```
kugel.init(200, 100, 10, 0.5,
           0.2, 1.0, 0.0, 1.0);
```

Man schreibt dazu erst den Namen, den man der Variable (nicht der Klasse!) gegeben hat, dann – nach einem Punkt – den Namen der Funktion. In die Klammer dahinter werden gegebenenfalls die Parameter der Funktion notiert.

Auf diese Weise wird das Hauptprogramm bedeutend kürzer und verständlicher. Das Objektorientierte Programmieren erleichtert zudem die gemeinsame Arbeit an einem Programm, da man parallel verschiedene Klassen entwickeln kann. Des Weiteren kann man so auf einfache Art beliebig viele Kugeln – oder mit weiteren Klassen den Queue oder den Tisch – programmieren.

Wann berühren sich zwei Billardkugeln?

Timo Kandra

Da wir nun mit allen wichtigen Grundlagen der Programmierung vertraut waren, konnten wir uns dem Billardspiel zuwenden. Als erstes brachten wir dem Computer bei, wann sich zwei Billardkugeln berühren. Doch wie funktioniert das?

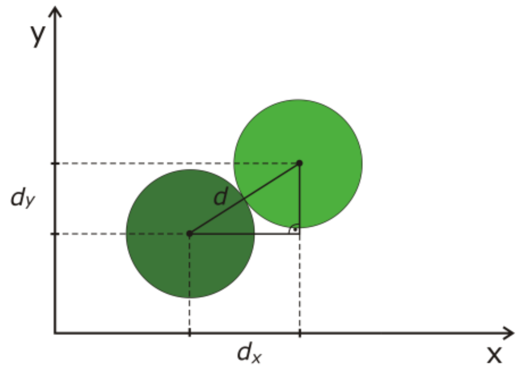
Als Ausgangsinformationen sind nur die Koordinaten der Mittelpunkte und die Radien der beiden Kugeln bekannt. Nun müssen wir aus diesen Werten den Abstand d der zwei Kugeln ausrechnen. Die Mittelpunkte der beiden Kugeln haben die Koordinaten $(x_1 \,|\, y_1)$ und $(x_2 \,|\, y_2)$. Die Abstände in den jeweiligen Koordinatenrichtungen errechnen sich durch

$$d_x = x_1 - x_2 \tag{1}$$

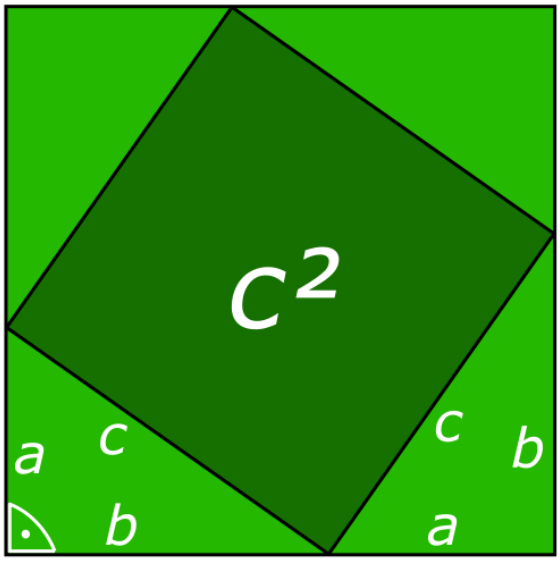
und

$$d_y = y_1 - y_2. \tag{2}$$

Da d_x und d_y orthogonal aufeinander stehen, kann der gesuchte Abstand d mit Hilfe des Satzes von Pythagoras berechnet werden:



Der Satz des Pythagoras besagt, dass in einem rechtwinkligen Dreieck die Summe der Quadrate der beiden Katheten a^2 und b^2 das Quadrat der Hypotenuse c^2 ergibt.



Beweis:

Das in der Abbildung zu sehende Quadrat hat einen Flächeninhalt von $(a+b)^2$. Zieht man die Flächen der kleinen Dreiecke am Rand ab, so erhält man für die Fläche des mittleren Quadrates

$$(a+b)^2 - 4 \cdot \frac{ab}{2} = c^2. \tag{3}$$

Durch Auflösen der Klammer mit Hilfe der ersten binomischen Formel und entsprechendem Zusammenfassen der Terme auf der linken Seite ergibt sich

$$a^2 + b^2 = c^2. \tag{4}$$

Somit ergibt sich für unseren gesuchten Abstand

$$d = \sqrt{d_x^2 + d_y^2}. \quad (5)$$

Diesen vergleichen wir dann mit der Summe der beiden Radien. Wenn die Werte gleich sind, berühren sich die Kugeln. In unserem Programm verwenden wir jedoch nicht den Gleichheits-, sondern den Kleiner-Gleich-Operator. Dafür stelle man sich die Bewegung der Kugeln als viele kleine, schnelle, aufeinander folgende Schritte vor. Es kommt dann fast nie vor, dass der Abstand der Kugeln nach einem Schritt exakt der Summe der Radien entspricht – die Kugeln überlappen immer unmerklich. Daher wird durch die Verwendung des Kleiner-Gleich-Operators jede Kollision entsprechend erkannt. Mit einer if-Abfrage können wir also überprüfen, ob zwei Kugeln mit den Radien r_1 und r_2 im letzten Rechenschritt zusammengestoßen sind:

```
if(d <= r1 + r2) {
  // Die Kugeln kollidieren!
}
```

Diese if-Abfrage wird jedes Mal ausgeführt, wenn die Kugeln ihre Position geändert haben – und zwar in jeder möglichen Kombination:

Kollidiert Kugel 1 mit Kugel 2?
Kollidiert Kugel 1 mit Kugel 3?
...
Kollidiert Kugel 2 mit Kugel 3?
...
Kollidiert Kugel 15 mit Kugel 16?

Vektorrechnung

Lucas Wollenhaupt

Um ein physikalisch korrektes Billardspiel zu entwickeln, mussten wir uns nicht nur mit Programmierung und Physik auseinandersetzen. Wir lernten auch mathematische Grundlagen kennen, um die beim Billard auftretenden komplexen physikalischen Phänomene möglichst einfach beschreiben zu können. Für den zentralen Stoß waren unsere Schulkenntnisse noch ausreichend. Um jedoch den allgemeinen Stoß zweier Billardkugeln realistisch darstellen zu können, mussten wir erst lernen, was es mit der Vektorrechnung auf sich hat.

In der Schule lernt man bis zur 10. Klasse das Rechnen mit reellen Zahlen – also das Rechnen mit festen Werten, wie 2 oder -4. Die reellen Zahlen nennt man in der Mathematik auch *Skalare*. Es handelt sich dabei um Größen ohne Richtung. Es gibt aber Größen, die eine Richtung im Raum haben, die also in einem Koordinatensystem z. B. nach rechts oben zeigen. Solche Größen nennt man in der Mathematik *Vektoren*. Es handelt sich dabei im Prinzip um Pfeile im Raum, die ähnlich wie Punkte aus zwei Zahlen (im zweidimensionalen Koordinatensystem) bestehen. Mathematisch schreibt man sie so:

$$\vec{v} = \begin{pmatrix} x \\ y \end{pmatrix} \quad (6)$$

Dieser Vektor wäre ein Pfeil, der vom Ursprung zu dem Punkt mit den Koordinaten $(x \mid y)$ zeigt. Ein solcher Vektor besitzt zwei wichtige Eigenschaften: die Richtung und die Länge $|\vec{v}|$, auch Betrag genannt. Der Betrag kann mit dem Satz des Pythagoras ausgerechnet werden:

$$|\vec{v}| = \sqrt{x^2 + y^2} \quad (7)$$

Um jedoch mit Hilfe von Vektoren, wie in unserem Beispiel, die Geschwindigkeit von zusammenstoßenden Kugeln zu berechnen, muss man mit Vektoren rechnen können. Es gibt für Vektoren ähnliche Rechenregeln wie für Skalare, das heißt man kann sie addieren, subtrahieren und multiplizieren. Diese Rechenregeln werden im Folgenden kurz dargestellt.

Zuerst betrachten wir die Addition bzw. Subtraktion zweier Vektoren. Es sind zwei Vektoren gegeben, z. B. die Vektoren

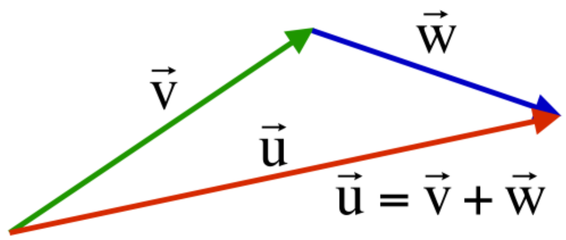
$$\vec{v} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{8}$$

und

$$\vec{w} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}. \tag{9}$$

Die Summe der beiden Vektoren entspricht dabei dem Vektor, der direkt vom Anfang des ersten Vektors bis zum Ende des zweiten führt. Man kann sich diese beiden Vektoren als zwei Wegabschnitte entlang einer Hauswand vorstellen: Man läuft jeweils an der Hausseite entlang, die Strecke in Luftlinie entspricht jedoch dem direkten Weg. Genauso entsprechen die beiden Vektoren $\vec{v}$ und $\vec{w}$ dem Weg an der Hauswand und deren Summe $\vec{u} = \vec{v} + \vec{w}$ der Luftlinie. Um die Summe auszurechnen, muss man jeweils die beiden x-Koordinaten und die beiden y-Koordinaten addieren:

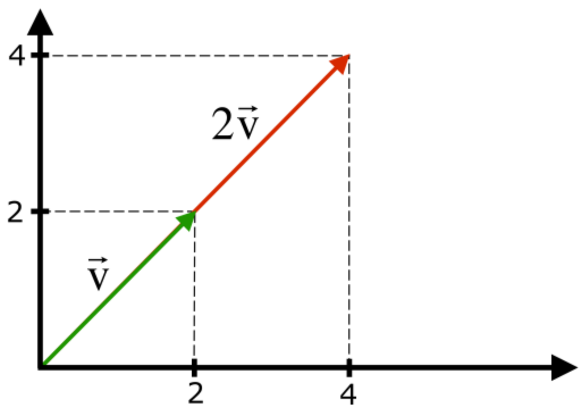
$$\vec{u} = \vec{v} + \vec{w} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \end{pmatrix} \tag{10}$$



Vektoraddition: Die Summe der Wege $\vec{v}$ und $\vec{w}$ entspricht der „Luftlinie“ $\vec{u}$.

Die Subtraktion von zwei Vektoren erfolgt nach demselben Prinzip. Des Weiteren kann man einen Vektor mit einem Skalar multiplizieren. Dabei wird der Vektor um den Faktor gestreckt, den der Skalar vorgibt. Auch hier ist die Rechenregel denkbar einfach. Man muss jede Koordinate des Vektors mit dem Skalar multiplizieren und erhält das entsprechende Produkt. Das Dividieren funktioniert ebenfalls nach demselben Prinzip. Man multipliziert mit dem Kehrwert des Skalars, durch den dividiert werden soll. Die Rechenregel lautet demnach

$$r \cdot \vec{v} = r \cdot \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} r \cdot x_1 \\ r \cdot y_1 \end{pmatrix}. \tag{11}$$



Streckung des Vektors $\vec{v}$ um den Faktor $r = 2$.

Man kann nicht nur Vektoren mit Skalaren multiplizieren, sondern auch Vektoren miteinander – das Ergebnis ist dann ein Skalar. Das Produkt zweier Vektoren nennt man *Skalarprodukt* und berechnet sich nach

$$\vec{v} \cdot \vec{w} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = x_1 \cdot x_2 + y_1 \cdot y_2. \tag{12}$$

Beim Skalarprodukt gibt es zwei wichtige Spezialfälle. Der erste ist die Multiplikation eines Vektors mit sich selbst:

$$\vec{v} \cdot \vec{v} = x_1 \cdot x_1 + y_1 \cdot y_1 = |\vec{v}|^2 \tag{13}$$

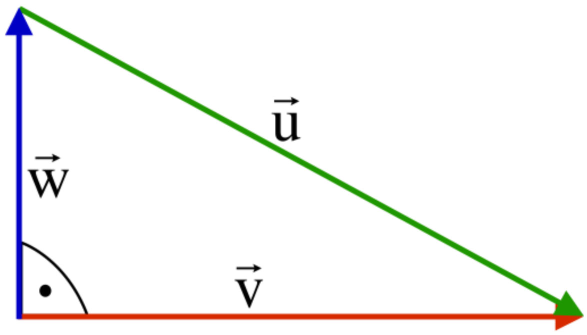
Der zweite Spezialfall tritt auf, wenn das Skalarprodukt zweier Vektoren gleich Null ist, also

$$\vec{v} \cdot \vec{w} = 0 \tag{14}$$

gilt. In diesem Fall stehen die Vektoren orthogonal aufeinander, bzw.: Wenn zwei Vektoren orthogonal zueinander sind, ist ihr Skalarprodukt gleich Null. Diesen Satz werden wir im Folgenden beweisen:

Wir nehmen an, dass die Vektoren $\vec{v}$ und $\vec{w}$ orthogonal zueinander sind. Der Vektor $\vec{u}$ verbindet die beiden Vektoren zu einem Dreieck. Dann gilt

$$\vec{w} + \vec{u} = \vec{v}. \tag{15}$$



Nach dem Satz des Pythagoras und (15) gilt

$$|\vec{v}|^2 + |\vec{w}|^2 = |\vec{v} - \vec{w}|^2. \tag{16}$$

Hieraus folgt

$$x_1^2 + y_1^2 + x_2^2 + y_2^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 \tag{17}$$

und nach Anwendung der zweiten binomischen Formel und Subtraktion der quadratischen Terme

$$0 = x_1 \cdot x_2 + y_1 \cdot y_2 = \vec{v} \cdot \vec{w}. \tag{18}$$

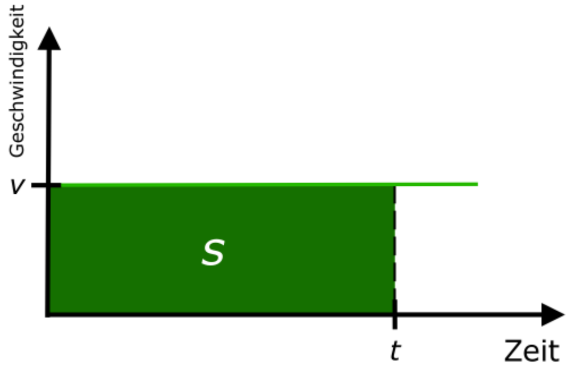
Da man den Beweis auch von unten nach oben lesen kann, ist der obige Satz in beiden Richtungen bewiesen.

Damit haben wir die für unsere Zwecke notwendigen mathematischen Grundlagen, um Kugelkollisionen im zweidimensionalen Raum zu berechnen. Wie dies genau funktioniert, wird in einem späteren Kapitel erläutert.

Kinematik und Dynamik

Daniela Winter

Bevor wir uns mit dem genaueren Verhalten der Kugeln im Billardspiel beschäftigten, schauten wir uns erst einmal die grundlegenden Regeln der *Kinematik* (Lehre der Bewegung) und der *Dynamik* (Lehre vom Einfluss der Kräfte) an. Dazu stellten wir uns als erstes die Frage, welche Formen der Bewegung es überhaupt gibt. Die wichtigsten sind neben dem Stillstand, die *gleichförmige Bewegung* und die *gleichmäßig beschleunigte Bewegung*. Bei der gleichförmigen Bewegung hat der bewegte Körper immer eine konstante Geschwindigkeit v. Das heißt pro



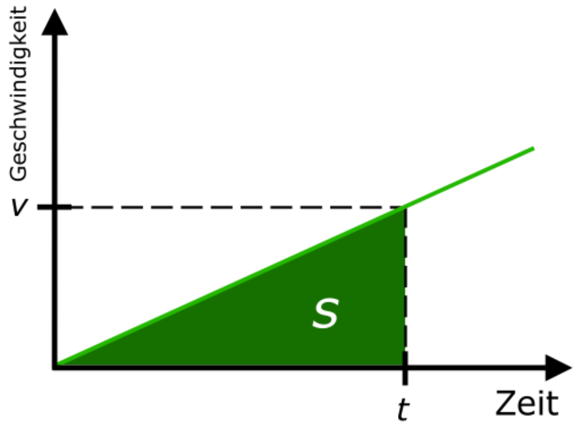
v-t-Diagramm der gleichförmigen Bewegung.

Zeiteinheit t legt der Körper eine bestimmte Strecke zurück, sodass gilt

$$v = \frac{s}{t}. \tag{19}$$

In diesem Diagramm wird die Geschwindigkeit gegen die Zeit aufgetragen. Man nennt es daher *v-t-Diagramm*. Die Fläche unter dem Graphen entspricht der zurückgelegten Strecke, da $s = v \cdot t$ gilt. Bei der gleichmäßig beschleunigten Bewegung ändert sich hingegen die Geschwindigkeit v pro Zeiteinheit um einen konstanten Wert a (die Beschleunigung). Es gilt

$$v = a \cdot t. \tag{20}$$



v-t-Diagramm der gleichmäßig beschleunigten Bewegung.

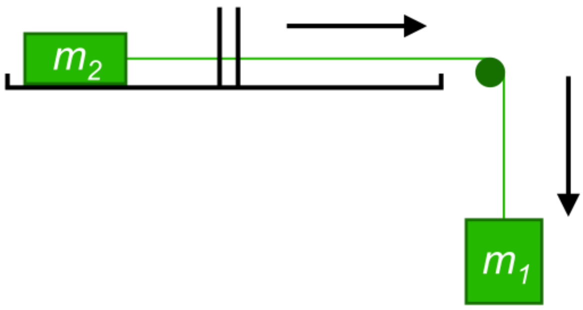
Betrachtet man die gleichmäßig beschleunigte Bewegung im v-t-Diagramm, so sieht man, dass die zurückgelegte Strecke unter dem Graphen ein Dreieck beschreibt. Den Flächeninhalt dieses Dreiecks – und somit die Strecke s – errechnet sich durch die Formel

$$s = \frac{1}{2} \cdot v \cdot t. \tag{21}$$

Zusammen mit Gleichung (20) ergibt sich

$$s = \frac{1}{2} \cdot a \cdot t \cdot t = \frac{1}{2} \cdot a \cdot t^2. \tag{22}$$

Als nächstes untersuchten wir, wie Objekte beschleunigt werden. Anhand einer Luftkissenbahn konnten wir eindeutig sehen, dass ein sich bewegendes Objekt sich immer weiterbewegt, wenn keine Kräfte darauf wirken. Werden auf ein Objekt jedoch Kräfte ausgeübt, ändert sich der Bewegungszustand – das Objekt wird beschleunigt. Um herauszufinden, welche Kraft zu welcher Beschleunigung führt, machten wir zwei unterschiedliche Versuche.



Skizze des Versuchsaufbaus.

Beim ersten Versuch erhöhten wir bei jedem Durchlauf m_1 um 10g und verkleinerten m_2 um 10g, sodass die Gesamtmasse konstant blieb. Dabei konnten wir beobachten, dass sich die Beschleunigung des Wagens stets gleichmäßig vergrößerte. Somit ist die Beschleunigung a des Wagens proportional zu der Kraft F, die an dem Wagen zieht. Beim zweiten Versuch wurde nach jeder Messung m_2 um 100g erhöht, m_1 wurde nicht verändert. Das heißt, die Gesamtmasse wurde nun immer größer, während die Kraft konstant blieb. Dieses Mal bemerkten wir, dass die Beschleunigung regelmäßig abnahm, und zwar proportional zur Gesamtmasse. Also ist die Beschleunigung umgekehrt proportional zur Masse. Aus den zwei Ergebnissen

$$F \sim a \tag{23}$$

und

$$a \sim \frac{1}{m} \tag{24}$$

konnten wir folgern, dass

$$F \sim m \cdot a \tag{25}$$

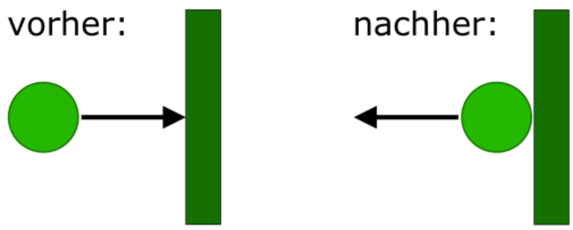
sein muss. Die Einheit Newton einer Kraft F ist schließlich so definiert, dass gilt

$$F = m \cdot a. \tag{26}$$

Kollision an der Bande

Daniela Winter

Inzwischen kann sich die Kugel bewegen. Unser nächstes Ziel war aber, dass die Kugel an den Banden des Billardtisches abgestoßen wird und nicht einfach weiterrollt, bis sie vom Bildschirm verschwunden ist.



Kollision an der Bande.

Man muss also dem Computer sagen, dass die Geschwindigkeit v_x bzw. v_y der Kugel umgedreht werden soll, wenn der x- bzw. y-Wert der Kugel zu groß oder zu klein ist:

```
if (x < 20)
  vx = -vx;
if (x > 620)
  vx = -vx;
if (y < 20)
  vy = -vy;
if (y > 460)
  vy = -vy;
```

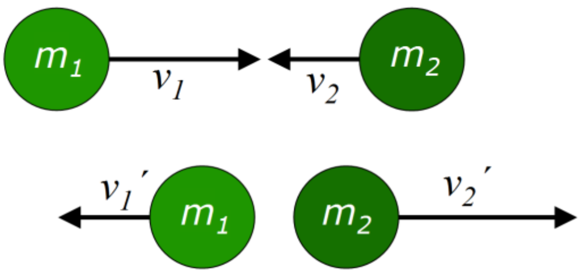
So kann der Computer die Richtungsänderung der Kugel an einer Bande physikalisch korrekt darstellen. Sie wird nach dem Gesetz „Einfallswinkel = Ausfallswinkel“ reflektiert.

Zentraler Stoß

Anne Baier

Mittlerweile weiß der Computer, wann sich zwei Kugeln berühren, allerdings ist noch unklar, was bei der Kollision passiert. Wir wollen nun herausfinden, wie sich die Geschwindigkeiten der Kugeln durch den Stoß ändern. Dazu betrachten wir zunächst den Spezialfall des zentralen Stoßes.

Treffen die beiden Kugeln mit ihren jeweiligen Massen m_1 und m_2 und ihren jeweiligen Geschwindigkeiten v_1 und v_2 zentral aufeinander, so bewegen sie sich nach der Kollision mit anderen Geschwindigkeiten v_1' und v_2' auseinander. Wir kennen die Massen m_1 und m_2, sowie die Anfangsgeschwindigkeiten v_1 und v_2 der Kugeln. Gesucht sind die Endgeschwindigkeiten v_1' und v_2'.



Da die Menge an verlorener Bewegungsenergie während des Aufpralls sehr gering ist, gilt der Energieerhaltungssatz

$$E_1 + E_2 = E_1' + E_2', \tag{27}$$

der besagt, dass die Summe der Bewegungsenergien vor dem Stoß gleich der Summe der Bewegungsenergien nach dem Stoß ist. Wir nehmen also an, dass der Energieverlust gleich Null ist. Durch Einsetzen von

$$E = \frac{1}{2}mv^2 \tag{28}$$

in den Energieerhaltungssatz ergibt sich

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 = \frac{1}{2}m_1v_1'^2 + \frac{1}{2}m_2v_2'^2. \tag{29}$$

Außerdem gilt die Impulserhaltung

$$p_1 + p_2 = p_1' + p_2', \tag{30}$$

da die Kugeln näherungsweise nur wechselseitige Kräfte spüren, deren Summe Null ergibt.

Wenn wir für Impuls p ebenfalls seine Definition

$$p = m \cdot v \tag{31}$$

in den Impulserhaltungssatz einsetzen, erhalten wir die Formel

$$m_1 \cdot v_1 + m_2 \cdot v_2 = m_1 \cdot v_1' + m_2 \cdot v_2'. \tag{32}$$

Wir formen diese Gleichung nun so um, dass alle Werte mit demselben Index auf einer Seite stehen und klammern anschließend m_1 und m_2 aus:

$$m_1 \cdot (v_1 - v_1') = m_2 \cdot (v_2' - v_2) \tag{33}$$

Den mit 2 durchmultiplizierten Energieerhaltungssatz (29) stellen wir genauso um:

$$m_1 \cdot (v_1^2 - v_1'^2) = m_2 \cdot (v_2'^2 - v_2^2) \tag{34}$$

Mit Hilfe der dritten binomischen Formel ergibt sich daraus

$$m_1(v_1 - v_1')(v_1 + v_1') = m_2(v_2' - v_2)(v_2' + v_2). \tag{35}$$

Die Division von (35) durch (33) ergibt

$$v_1 + v_1' = v_2 + v_2'. \tag{36}$$

Mit den beiden linearen Gleichungen (33) und (36) für die beiden Unbekannten v_1' und v_2' ergeben sich die Lösungen

$$v_1' = \frac{2m_2v_2 + m_1v_1 - m_2v_2}{m_1 + m_2} \tag{37}$$

und

$$v_2' = \frac{2m_1v_1 + m_2v_2 - m_1v_1}{m_1 + m_2}. \tag{38}$$

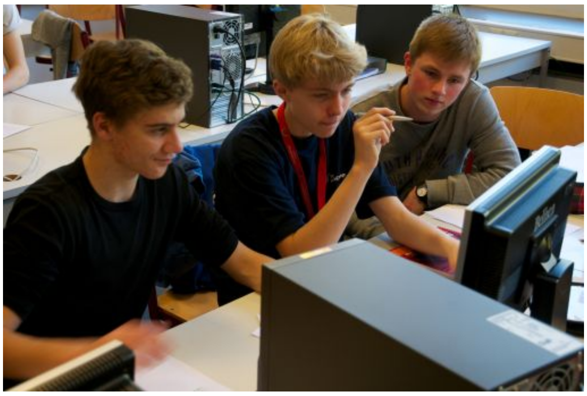
Damit können wir die Geschwindigkeiten nach dem Stoß recht einfach berechnen. Wenn, wie im Billard, der Spezialfall gilt, dass die Kugeln die gleichen Massen haben ($m_1 = m_2 = m$), dann vereinfachen sich (37) und (38) zu

$$v_1' = v_2 \tag{39}$$

und

$$v_2' = v_1. \tag{40}$$

Beim Stoß tauschen die Kugeln also ihre Geschwindigkeiten.



Stoß in zwei Dimensionen

Anne Baier

Bei einem richtigen Billardspiel können die Kugeln auch schräg zur Einlaufrichtung voneinander abprallen. Hierfür verwenden wir die Vektorschreibweise, da diese die Rechnung erheblich erleichtert. Es gilt weiterhin die Impulserhaltung und die Erhaltung der Bewegungsenergie, die wir diesmal aber mit Impulsen statt mit Geschwindigkeiten ausdrücken.

Impulserhaltung:

$$\vec{p_1} + \vec{p_2} = \vec{p_1}' + \vec{p_2}' \tag{41}$$

Energieerhaltung:

$$\frac{|\vec{p_1}|^2}{2m_1} + \frac{|\vec{p_2}|^2}{2m_2} = \frac{|\vec{p_1}'|^2}{2m_1} + \frac{|\vec{p_2}'|^2}{2m_2} \tag{42}$$

Wir drücken den Impuls der ersten Kugel nach dem Stoß durch die Impulsänderung $\Delta\vec{p}$ aus:

$$\vec{p_1}' = \vec{p_1} + \Delta\vec{p} \tag{43}$$

Der neue Impuls der Kugel ist also sein alter Impuls plus die Impulsänderung. Mit der Impulserhaltung ergibt sich damit für die zweite Kugel

$$\vec{p_2}' = \vec{p_2} - \Delta\vec{p}. \tag{44}$$

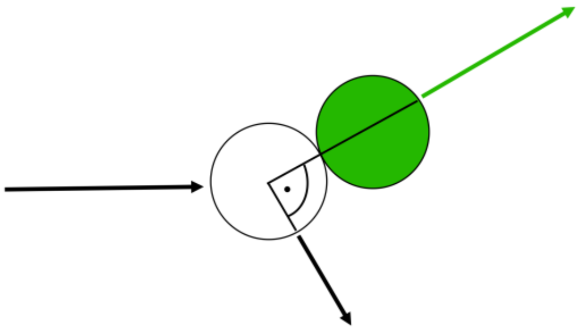
Die Impulsänderung zeigt in Kraftrichtung und damit in Richtung der Verbindungslinie der Kugelmittelpunkte, die wir mit $\vec{x_1} - \vec{x_2}$ bezeichnen. Um die Richtung der Impulsänderung anzugeben, verwenden wir einen Richtungsvektor der Länge 1, den wir $\vec{\delta}$ nennen:

$$\Delta\vec{p} = k \cdot \frac{\vec{x_1} - \vec{x_2}}{|\vec{x_1} - \vec{x_2}|} = k \cdot \vec{\delta} \tag{45}$$

Wenn wir den Faktor k kennen, dann kennen wir die Impulse und damit die Geschwindigkeiten nach dem Stoß. Wenn wir die Gleichung (45) in (43) und (44) einsetzen und diese wiederum in den Energieerhaltungssatz (42), so erhalten wir nach zahlreichen Umformungen

$$k = \frac{2 \cdot \vec{\delta} \cdot (\vec{v_2} - \vec{v_1})}{\frac{1}{m_1} + \frac{1}{m_2}}. \tag{46}$$

Seit wir das in unser Programm eingebaut haben, kollidieren die Kugeln im Programm realistisch miteinander. Ein wichtiger Spezialfall hilft auch beim richtigen Billardspielen weiter: Wenn sich zwei Kugeln mit gleichen Massen treffen, wobei eine davon in Ruhe ist, dann trennen sie sich im rechten Winkel.



Beim Stoß mit einer ruhenden Kugel gleicher Masse laufen die Kugeln im rechten Winkel auseinander.

Gleichung (42) vereinfacht sich dann mit $m = m_1 = m_2$ und $\vec{v_2} = 0$ zu

$$\frac{1}{2}m|\vec{v_1}|^2 = \frac{1}{2}m|\vec{v_1}'|^2 + \frac{1}{2}m|\vec{v_2}'|^2. \tag{47}$$

Multiplikation mit $\frac{2}{m}$ ergibt

$$|\vec{v_1}|^2 = |\vec{v_1}'|^2 + |\vec{v_2}'|^2. \tag{48}$$

Nach dem Satz des Pythagoras stehen daher die Vektoren $\vec{v_1}'$ und $\vec{v_2}'$ senkrecht aufeinander.

Reibung

Axel Printschler

Unsere Billardkugeln können nun über den Tisch rollen und miteinander kollidieren, doch stehenbleiben können sie noch nicht. Wir stellten uns daher die Frage, warum Gegenstände allgemein und speziell unsere Billardkugeln stehen bleiben, wie wir dieses Phänomen physikalisch und mathematisch beschreiben und anschließend ins Spiel einbauen können.

Um die theoretischen Grundlagen zu lernen, verbrachten wir einen gesamten Vormittag im Physiksaal. Zunächst ging es darum, welche verschiedenen Reibungsarten es überhaupt gibt. Als erstes tritt die *Haftreibung* in Kraft. Diese wirkt, weil alle Gegenstände, auch wenn sie glatt aussehen wie die Billardkugeln, mikroskopisch betrachtet gar nicht glatt sind. Jeder Gegenstand weist kleine Erhebungen auf, die sich mit den Erhebungen des anderen Gegenstandes, beispielsweise einer Billardkugel mit dem Tisch, verzahnen. Dadurch benötigt man anfangs eine höhere Kraft, um den Gegenstand in Bewegung zu bringen, als danach, wenn er sich schon bewegt. Man kennt dieses Prinzip, wenn man versucht einen schweren Schrank zu verschieben. Am Anfang braucht man viel Kraft, um den Schrank in Bewegung zu versetzen, doch wenn er einmal in Bewegung ist, geht es einfacher. Dann gleitet er über den Boden. Es wirkt die *Gleitreibung*. Diese ist nicht mehr so groß wie die Haftreibung, da die Erhebungen sich nicht mehr verzahnen, sondern übereinander hinweggleiten. Da Kugeln rollen, tritt diese Reibungsart bei ihnen nur selten in Kraft. Häufiger ist stattdessen die *Rollreibung*. Sie ist noch kleiner als die Gleitreibung. Außerdem gibt es noch weitere Effekte, die bewegte Objekte abbremsen, beispielsweise den Luftwiderstand oder den Wasserwiderstand. Der Luftwiderstand tritt beim Billardspielen auf, der Wasserwiderstand hingegen nicht, nicht einmal beim Poolbillard. Der Luftwiderstand ist allerdings zu vernachlässigen.

In unser Billardspiel haben wir nur eine Reibungsart eingebaut. Die Haftreibung ist nicht weiter von Belang, da jeder Billardspieler es schafft, mit dem Queue die Kugel so anzustoßen, dass der Haftwiderstand überwunden wird und sich die Kugel bewegt. Die Gleitreibung tritt beim Billard nur in Kraft, wenn die Kugel nach dem Stoß nicht sofort anfängt zu rollen, sondern am Anfang über den Tisch schlittert, was beim Spin wichtig wird. Doch da wir diesen nur theoretisch behandelt, und nicht in unser Spiel eingebaut haben, benötigen wir auch die Gleitreibung nicht. Daher haben wir uns damit befasst, die Rollreibung zu bestimmen.

Zuerst haben wir uns gefragt von welcher Kraft die Reibung abhängt, und was diese Kraft genau bewirkt. Die Reibungskraft F_r wirkt entgegen der Bewegungsrichtung und bremst so den Gegenstand ab. Sie hängt von der Schwerkraft F_g ab. Die Reibungskraft F_r ergibt sich als Produkt der Schwerkraft F_g und dem Reibungskoeffizienten μ:

$$F_r = F_g \cdot \mu \tag{49}$$

Pro Zeitabschnitt wird die Geschwindigkeit um einen konstanten Wert kleiner. Diesen haben wir als a_r definiert:

$$|v'| = |v| - a_r \cdot t \tag{50}$$

Doch nun wissen wir immer noch nicht, was genau a_r ist. Wir haben zwar die Formel, um die Reibung zu berechnen, doch es fehlt noch ein genauer Wert der negativen Beschleunigung a_r. Diesen Wert haben wir in einem selbst ausgedachten Versuch gemessen. In diesem Versuch ließen wir eine Kugel eine kleine Rampe herunter rollen und stoppten die Zeit t, die die Kugel brauchte, bis sie zum Stillstand kam. Danach maßen wir die zurückgelegte Strecke s. Mit diesen beiden Werten konnten wir nun ausrechnen, wie groß die negative Beschleunigung war, denn es gilt

$$a_r = 2\frac{s}{t^2}. \tag{51}$$

Das Ergebnis unserer Messung war

$$a_r = 6{,}8\frac{\text{cm}}{\text{s}^2}. \tag{52}$$

Das bedeutet, dass eine Kugel pro Sekunde um $6{,}8\frac{\text{cm}}{\text{s}}$ langsamer wird. Mit Hilfe dieses gemessen Wertes konnten wir eine realistische Verlangsamung in unser Programm einbauen.

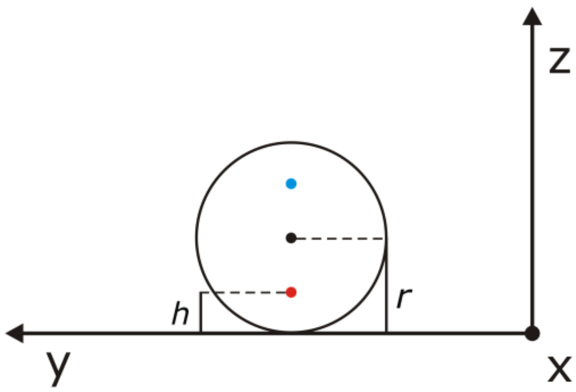
Spin – vom Laien zum Profi

Marvin Motzet

Bisher haben wir die Kollision zweier Kugeln, die Reflexion an der Bande, die Reibung sowie einige Regeln in das Billardspiel eingebaut. Das ähnelt beim ersten Blick fast einem realen Billardspiel, aber leider nur fast. Es fehlt jedoch etwas, mit dem das Billardspielen noch mehr Spaß macht: der Spin. Damit ist es u. a. möglich, die weiße Kugel als sogenannten *Nachläufer* oder als *Rückläufer* anzustoßen, indem der Spieler mit dem Queue die Kugel nicht mittig anspielt, sondern den Anstoßpunkt variiert. Wie wir solch einem Feature physikalisch auf den Grund gingen, stellen wir im Folgenden dar. Nebenbei kann der Leser noch seine Billardkenntnisse auffrischen oder erweitern.

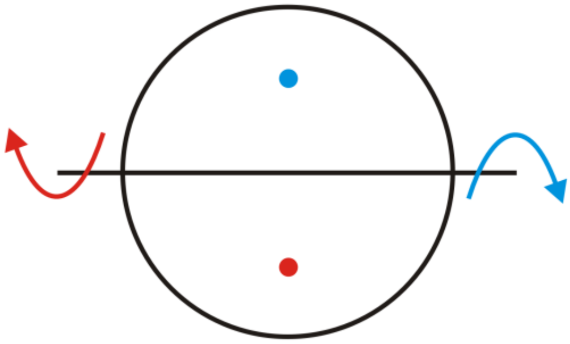
Wenn eine Billardkugel rollt, kann man die Drehgeschwindigkeit ω der Kugel (Radius r) mit der „normalen“ Geschwindigkeit v in Verbindung setzen:

$$\omega = \frac{v}{r} \tag{53}$$



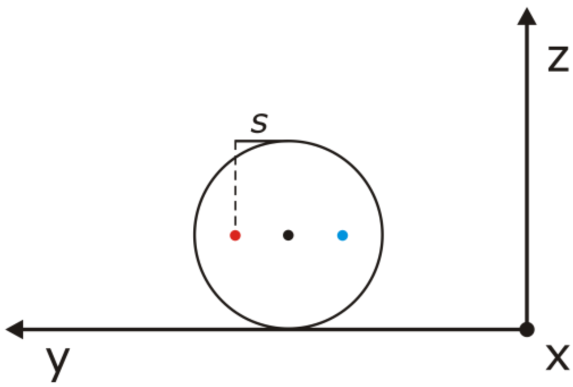
Verschiebung des Anstoßpunktes in z-Richtung – Drehung um y-Achse.

Wird die Kugel oberhalb bzw. unterhalb ihres Mittelpunkts angestoßen, kann es zu einem Nach- bzw. zu einem Rückläufer kommen: Die Kugel dreht sich dabei um die y-Achse. Beim Nachläufer (Topspin) dreht sich die Kugel mit hoher Geschwindigkeit um die y-Achse. Beim Zusammenstoß mit einer anderen Kugel läuft sie dieser hinterher, denn nach der Impulsübergabe dreht die Kugel hohl und gewinnt dadurch wieder an Geschwindigkeit. Beim Rückläufer (Backspin) läuft die Kugel nach dem Zusammenstoß mit einer anderen Kugel zurück, da sie sich entgegen der Rollrichtung um die y-Achse dreht.



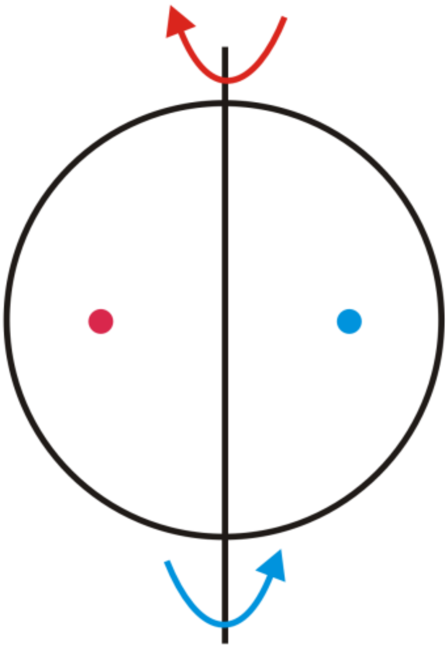
Blau: Topspin bzw. Nachläufer. Rot: Backspin bzw. Rückläufer.

Zudem haben wir die Drehung um die z-Achse untersucht, wenn der Anstoßpunkt in y-Richtung verschoben wird. Hier gilt, dass bei der Bandenreflexion der Ausfallswinkel nicht mehr dem Einfallswinkel entspricht. Die Kugel „rollt“ sich bei der Kollision ab.



Verschiebung des Anstoßpunktes in y-Richtung – Drehung um z-Achse.

Als weiteren Schritt haben wir Formeln für die Drehgeschwindigkeit hergeleitet. Wir nehmen dabei an, dass während des Stoßes eine konstante Kraft F auf die Kugel wirkt, mit der sie beschleunigt wird. Die Geschwindigkeit v der Kugel hängt dabei von der Stärke der Kraft ab. Je stärker die Kugel mit dem Queue angestoßen wird, desto schneller bewegt sie sich.



Blau: Drehung gegen den Uhrzeigersinn. Rot: Drehung im Uhrzeigersinn.

Diese Geschwindigkeit haben wir schließlich mit der Drehgeschwindigkeit in Verbindung gesetzt, wobei wir nach einer längeren Herleitung folgende Formeln für die Drehgeschwindigkeit ω_y (um die y-Achse) und ω_z (um die z-Achse) gefunden haben:

$$\omega_y = \frac{-5 \cdot F \cdot (r-h)}{2 \cdot r^2} \cdot v_x \tag{54}$$

$$\omega_z = \frac{-5 \cdot F \cdot s}{2 \cdot r^2} \cdot v_x \tag{55}$$

Die Längen s und h sind dabei den obigen Abbildungen zu entnehmen, r entspricht dem Kugelradius und v_x der Geschwindigkeit entlang der x-Achse. Bei beiden Formeln spielt die Kraft F, mit der die Kugel angestoßen wird, die Geschwindigkeit v und die jeweilige Anstoßposition eine wichtige Rolle. Zwar haben wir jetzt Formeln für die Drehbewegungen um die y- und z-Achse, aber während der Bewegung der Kugel geschieht noch mehr. So wirkt bei der Bewegung der Kugel Reibung, wodurch sie beschleunigt oder gebremst wird. Beim Topspin wird die Kugel durch die Gleitreibung beschleunigt. Für die Beschleunigung a gilt dann

$$a = \mu_{gleit} \cdot g. \tag{56}$$

Dabei ist μ_{gleit} der Gleitreibungskoeffizient, den wir vorher experimentell bestimmt haben. Beim Backspin wird die Kugel entsprechend mit

$$a = -\mu_{gleit} \cdot g \tag{57}$$

gebremst. Zeitgleich ändert sich die Winkelgeschwindigkeit ω_y. Für die Änderung der Winkelgeschwindigkeit pro Zeit, der Winkelbeschleunigung, gilt beim Topspin

$$a_\omega = -\frac{5}{2} \cdot \frac{\mu_{gleit} \cdot g}{r} \tag{58}$$

und beim Backspin

$$a_\omega = \frac{5}{2} \cdot \frac{\mu_{gleit} \cdot g}{r}. \tag{59}$$

Unser Billardturnier

Corinna Nowak

Während der Akademie haben wir nicht nur unser Spiel programmiert und uns mit physikalischen Aspekten zum Thema Billard beschäftigt, sondern auch einmal „richtiges“ Billard gespielt. An einem Nachmittag wurde ein kursinternes Billardturnier im Internat des Eckenberg Gymnasiums ausgetragen, sodass wir unsere Billardkenntnisse auch in der Praxis anwenden konnten.

Es gab mehrere Teams, die zufällig bestimmt und in zwei Gruppen eingeteilt wurden. Jedes Team musste gegen jedes andere Team aus der zugehörigen Gruppe spielen, während die Anderen gespannt mitfieberten. Zum Schluss traten die zwei Siegerteams gegeneinander an. Während die einzelnen Spiele stattfanden, spielten die anderen, wenn sie nicht gerade eines der Spiele verfolgten, Tischkicker, aßen Süßigkeiten, unterhielten sich oder diskutierten über mathematische Erkenntnisse.

Gewonnen haben schließlich Axel, Fabian und Elias im Finale gegen Daniela, Luisa und Lucas und sind so stolze Besitzer einer Packung Kekse geworden. Abschließend durften die Gewinner der Kursteilnehmer gegen unsere Kursleiter, Daniel und Alex, antreten, die schließlich als endgültige Sieger aus dem Billardturnier hervorgingen. Die beiden mussten jedoch nur ein Spiel gewinnen, im Gegensatz zu allen anderen Teams.

Auch wenn die Kugeln manchmal nicht so gerollt sind, wie wir es wollten, haben wir einen sehr lustigen und unterhaltsamen Nachmittag zusammen verbracht und konnten während des Turniers unsere Spielkenntnisse verbessern.

Rotation und Abschlusspräsentation

Corinna Nowak

Damit auch die anderen Teilnehmer der Akademie wussten, womit man sich in seinem Kurs beschäftigt hatte, fand nach der Hälfte der Sommerakademie die sogenannte Rotation statt. Jeder Kurs konnte den Teilnehmern der anderen Kurse präsentieren, welche Themen im Kurs bisher erarbeitet wurden. Dazu gab es vier verschiedene Gruppen, die aus Teilnehmern, Kursleitern und Schülermentoren bestanden und von Kurs zu Kurs rotierten.

Für eine Präsentation bedarf es natürlich an einigen Vorbereitungen. Zusammen haben wir uns im Kurs überlegt, welche Themen wir den anderen Teilnehmern präsentieren wollten. Wir erstellten eine einheitliche PowerPoint-Präsentation, bildeten vier Vortragsgruppen und teilten einzelne Themengebiete verschiedenen Personen zu. Diese reichten vom Programmieren über mathematische Formeln, wie zum Beispiel dem Satz des Pythagoras, bis hin zu physikalischen Themen, wie der Kollision zweier Kugeln. Natürlich durfte das Üben nicht fehlen. Also hielt jede Gruppe mindestens einmal einen Probevortrag. Das war sehr hilfreich, da man dabei merkte, an welchen Stellen Schwierigkeiten auftreten konnten und man lernte, wie ein Laserpointer richtig benutzt wird. Am Abend vor der Rotation waren dann alle gut für die Vorträge vorbereitet. Die Rotationsvorträge verliefen bei allen Gruppen erfolgreich. Zu unserem Billardspiel, das die Teilnehmer ausprobieren durften, obwohl es noch nicht ganz fertig war, haben wir einige hilfreiche Anregungen und Verbesserungsvorschläge bekommen.

Bei der Abschlusspräsentation war der Ablauf ähnlich. An diesem Tag präsentierte jeder Kurs den Eltern und Verwandten der Kursteilnehmer sowie anderen Interessierten, wie beispielsweise ehemaligen Teilnehmern, was wir in den zwei Wochen der Sommerakademie erarbeitet hatten.

Für diesen Tag haben wir unsere erste Präsentation ergänzt und verbessert. Außerdem haben wir eine Pinnwand erstellt, an der wir unseren Kurs vorstellten. Unsere Vorträge am Abschlusstag waren gut besucht. Auch diesmal war das Highlight unserer Präsentation der Schluss. Die Begeisterung der Zuhörer war immer groß, als wir ihnen erzählten, dass sie unser Spiel nun testen durften. Unsere Kursleiter und wir waren mit den Präsentationen sehr zufrieden. Den Zuhörern wurde außerdem das Thema Billard und die physikalischen Aspekte dazu ein Stück näher gebracht und sie hatten die Ehre, ein super Billardspiel am Computer spielen zu dürfen.

Exkursion

Timo Kandra

Am Montag der zweiten Akademiewoche besuchte unser Kurs die Arbeitsgruppe „Visualisierung und numerische Geometrie“ am Interdisziplinären Zentrum für Wissenschaftliches Rechnen (IWR) der Universität Heidelberg.

Nach einer relativ langen Busfahrt, die uns durch fiese mathematische Rätsel von Daniel und Alex nicht allzu entspannend gestaltet wurde, kamen wir an der Uni an. Unser Kurs fühlte sich sehr wohl, als wir über das Uni-Gelände spazierten – vielen von uns konnte man die Vorfreude auf das Studieren schon regelrecht anmerken.

Zu Beginn hörten wir Vorträge über spannende Themen der Mathematik an: Der erste handelte von Kryptographie. Das Ziel war es, uns die moderne RSA-Verschlüsselung näher zu bringen. Zunächst erfuhren wir etwas über sehr alte Verschlüsselungen aus der Zeit Cäsars. Auch die Prinzipien der Verschlüsselung von Nachrichten während der beiden Weltkriege wurden uns erklärt. Im zweiten Vortrag ging es um Knoten. Dieses exotische Teilgebiet der Mathematik wird unter anderem zur Erforschung von DNA-Verknotungen eingesetzt. Das war insofern interessant, da sich eigentlich noch niemand von uns mit diesem Thema beschäftigt hatte und es erstaunlich war, was schon alles über Knoten entdeckt, definiert und bewiesen wurde.

Nach der Pause, die wir in einem sehr stilvollen Aufenthaltsraum verbrachten, wurde uns das eigentliche Forschungsgebiet der Arbeitsgruppe vorgestellt. Sie zeigten uns, dass man Computergrafik auch anders verwenden kann als in unserem Billardspiel: Objekte, hauptsächlich archäologische Fundstücke, werden mit einem 3D-Scanner gescannt, um sie dann mit den Möglichkeiten eines Computers erforschen zu können.

In weiteren Vorträgen erfuhren wir, wie solch ein 3D-Scanner funktioniert und welche Schwierigkeiten bei dieser Aufnahmetechnik auftreten können. Zum Beispiel hatte sich einer der Referenten damit beschäftigt, die beim Scannen entstehenden Glanzlichter auf Objekten automatisch zu entfernen.

Nach den Vorträgen kam es dann zum Höhepunkt unseres Besuches: Wir durften den 3D-Scanner in Aktion sehen und ein interaktives 3D-Cyberboard ausprobieren. Dieses ermöglicht es durch Kameras den Standpunkt des Benutzers zu orten und somit den Blickwinkel des Dargestellten zu verändern. Auf diese Weise entsteht ein realistischer, dreidimensionaler Eindruck von den visualisierten Objekten, in welchen man mit einem Joystick eingreifen kann. In unserem Fall handelte es sich um eine Statue, die aus eingescannten Bruchstücken bestand. Da diese Statue in der Realität mehrere Meter hoch ist, war es leichter, sie auf virtuelle Weise zusammenzubauen.

Als alle von uns ausgiebig mit dem Cyberboard gespielt und gebaut hatten und alle offenen Fragen beantwortet waren, verließ der mit vielen Informationen und T-Shirts mit witzigem Aufdruck positiv gestimmte Physikkurs die Uni und fuhr zurück nach Adelsheim.

Sportfest

Luisa Feifel

Weil Physiker und Informatiker nicht nur total cool und intelligent (und eventuell etwas eingebildet), sondern auch sportlich sind, war das Sportfest, das der Physikkurs in den letzten zwei Jahren gewinnen konnte, die perfekte Gelegenheit, den anderen Kursen zu beweisen, dass wir nicht nur programmieren und Billard spielen können. Unseren Schlachtruf „`cout « Schlachtruf`" hat zwar außer uns fast keiner verstanden, aber er hat im Gegensatz zu manch anderen Kursen immer auf Anhieb funktioniert.

Insgesamt gab es beim Sportfest sieben Stationen und auch wenn man sich bei manchen fragen musste, wer sich so etwas ausgedacht hatte, haben wir zusammen dann doch alles problemlos gemeistert. Unsere erste Station war der Schwimmflossenlauf, bei dem jeder auf mehr oder weniger elegante Weise mit Schwimmflossen einen Parcours durchlaufen musste. Beim Teebeutelweitwurf, bei dem wir Teebeutel soweit wie möglich mit dem Mund werfen sollten, haben wir sehr kreative Techniken erfunden. Es lässt sich aber darüber streiten, ob diese wirklich effektiv waren. Bewiesen ist nur, dass es noch Schlechtere gab. Beim Autoziehen waren wir jedoch absolute Rekordhalter. Auch bei der nächsten Station, bei der wir Jonglierbälle in unterschiedlich große Reifen mit unterschiedlich hoher Punktzahl werfen sollten, haben wir uns ganz gut geschlagen. Bei der nächsten Station haben wir, dank Lucas' Anweisungen, Patricia sicher ins Ziel gebracht. Sie stand in einem großen A aus Holz, das nur durch vier Seile, die am oberen Ende befestigt waren, bewegt werden durfte. Die nächste Station war der Gummistiefelweitwurf. Hier hatten wir zwar viel Spaß, aber wenig Erfolg. Bei unserer letzten Station war wieder einmal Teamwork gefragt, denn die Aufgabe war es, ein Tuch umzudrehen, auf dem wir alle standen, ohne dass jemand den Boden berührt. Und irgendwann standen wir dann wirklich auf dem umgedrehten Tuch, wenn auch nicht ganz ohne Verluste. Zum Schluss kam noch ein Staffellauf, bei dem alle Kurse gegeneinender antraten. Ziel war es, mit nur einem Schwamm möglichst viel Wasser in einen Eimer zu befördern. Aber auch mit dem Wasser, das auf unerklärliche Weise noch nach dem Abpfiff in unseren Eimer gewandert war, konnten wir uns nicht gegen den Astronomiekurs durchsetzten, der mit eineinhalb Eimern Wasser der eindeutige Sieger der Staffel war.

Zum Schluss hatte es dann leider doch nicht ganz für den Sieg des Sportfestes gereicht, was aber auch daran liegen könnte, dass wir an manche Aufgaben einfach mit zu viel Strategie und logischem Denken herangegangen sind. Aber mit unserem hart erkämpften dritten Platz sind wir sehr zufrieden und hatten dabei auf jeden Fall sehr viel Spaß!

Dokumentation

Die Kursteilnehmer

Zu guter Letzt mussten wir die vorliegende Dokumentation verfassen. Der Entwurf wurde von den Teilnehmern zu Hause erstellt, sodass wir am Dokumentationswochenende den Text „nur noch" überarbeiteten, wobei sich das „nur noch" als erstaunlich viel entpuppte. Dennoch hatten wir viel Spaß beim Diskutieren der Grammatik, der Tempi und der Kommata und merkten, dass wir eben doch in den Physik-Informatik-Kurs gehören.

Danksagung

Die Kursteilnehmer

Ihr wart zusammen echt die coolsten Kursleiter (und die coolste Schülermentorin) – das haben sogar die anderen Kurse irgendwann zugegeben. Danke, dass ihr immer so geduldig mit uns wart, uns jede noch so blöde Frage beantwortet habt und wir so viel Spaß zusammen haben durften. Wir werden euch und die unvergessliche Zeit auf jeden Fall total vermissen.

Nachwort

Die Kursleiter

In den zwei Wochen wurde viel Wissen über Physik, Mathematik und Informatik gelehrt und gelernt, doch dieses „Rezept“ wäre nicht gelungen, wenn der Kurs nicht so einzigartig harmoniert hätte. Im Kurs herrschte die richtige Mischung zwischen Ernsthaftigkeit und Lockerheit, was nicht zuletzt daran lag, dass sich alle nicht zu ernst nahmen und auch mal einen Witz über sich ergehen ließen. Somit wurde eine tolle Arbeitsatmosphäre geschaffen, in der die Ideen nur so sprudelten. Doch obwohl unser Spiel wirklich klasse geworden ist, war das nur ein kleines Fragment unserer geleisteten Arbeit, denn viel mehr werden wir uns alle an das angenehme Gefühl der Akademie erinnern. Es ist nicht einfach, unter so vielen Leuten, in so wenig Zeit ein Gefühl der Geborgenheit und Freundschaft entstehen zu lassen und doch haben wir es alle zusammen geschafft. Wir werden die gemeinsame Zeit mit euch nie vergessen!

Danksagung

Die JuniorAkademie Adelsheim / Science Academy Baden-Württemberg fand in diesem Jahr bereits zum 11. Mal statt. Daher möchten wir uns an dieser Stelle bei denjenigen bedanken, die ihr Stattfinden überhaupt möglich gemacht haben.

In diesem Jahr wurde die Akademie in erster Linie durch die H. W. & J. Hector Stiftung finanziell unterstützt. Einen weiteren Teil der Mittel trugen das Ministerium für Kultus, Jugend und Sport von Baden-Württemberg und der Förderverein der Science Academy e. V. bei. Dafür möchten wir uns an dieser Stelle bei allen Unterstützern ganz herzlich danken.

Die JuniorAkademie Adelsheim ist ein Projekt des Regierungspräsidiums Karlsruhe, das im Auftrag des Ministeriums für Kultus, Jugend und Sport, Baden-Württemberg und mit Unterstützung der Bildung & Begabung gGmbH Bonn für Jugendliche aus dem ganzen Bundesland realisiert wird. Wir danken daher dem Schulpräsidenten im Regierungspräsidium Karlsruhe, Herrn Prof. Dr. Werner Schnatterbeck, der Referatsleiterin des Referates 75 – Allgemein bildende Gymnasien, Frau Leitende Regierungsschuldirektorin Dagmar Ruder-Aichelin, Herrn Jurke und Frau Reinhard vom Ministerium für Kultus, Jugend und Sport sowie dem Koordinator der Deutschen Schüler- und JuniorAkademien in Bonn, Herrn Volker Brandt.

Wie in jedem Jahr fanden die etwas über einhundert Gäste sowohl während des Eröffnungswochenendes und des Dokumentationswochenendes als auch während der zwei Wochen im Sommer eine liebevolle Rundumversorung am Eckenberg-Gymnasium mit dem Landesschulzentrum für Umwelterziehung (LSZU) in Adelsheim. Stellvertretend für alle Mitarbeiter möchten wir uns für die Mühen, den freundlichen Empfang und den offenen Umgang mit allen bei Herrn Oberstudiendirektor Meinolf Stendebach, dem Schulleiter des Eckenberg-Gymnasiums, und Herrn Bürgermeister Klaus Gramlich besonders bedanken.

Zuletzt sind aber auch die Kurs- und KüA-Leiter gemeinsam mit den Schülermentoren und der Assistenz des Leitungsteams diejenigen, die mit ihrer hingebungsvollen Arbeit das Fundament der Akademie bilden. Ein besonderer Dank gilt an dieser Stelle Jörg Richter, der auch in diesem Jahr für die Gesamterstellung der Dokumentation verantwortlich war.

Diejenigen aber, die die Akademie in jedem Jahr einzigartig werden lassen und die sie zum Leben erwecken, sind die Teilnehmerinnen und Teilnehmer. Deshalb möchten wir uns bei ihnen und ihren Eltern für ihr Vertrauen ganz herzlich bedanken.