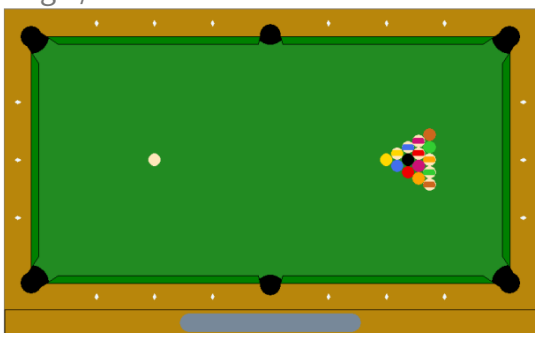
1 Einführung

1.1 Aufgabenstellung

> Aufgabe: Erstellen Sie ein initiales Spielfeld wie angegeben. Durch Mouseevents soll zu Spielbeginn die Stoßrichtung und Geschwindigkeit der weißen Kugel vorgegeben werden. Dazu ist mit der Mouse ein Queue anzutragen. Im weiteren Spielverlauf sollen alle Kugeln wie beim klassischen Poolbillard per Queue bewegt werden. Nutzen Sie die in der Vorlesung gegebenen Grundlagen zur Darstellung und Simulation von Stoßereignissen. Beachten Sie die unterschiedlichen Bewegungsgleichungen für Kugel/Kugel bzw. Kugel/Banden-Kontakt. Zusätzlich soll



Billiard ist ein Geschicklichkeitsspiel bei dem Kugeln mit Hilfe eines Stabes, der Queue genannt wird, in Löcher gestossen werden. Dabei werden die 15 farbigen Kugeln nicht direkt mit dem Stab angespielt. Vielmehr spielt man mit dem Queue den weißen Spielball, der dann die anderen Kugeln so anstossen muss, dass diese in die Löcher fallen. Bei der von uns betrachteten Variante dem Poolbillard muss die schwarze 8 zuletzt eingelocht werden.

1.2 Stoss

Der wichtigste physikalische Vorgang bei der Simulation eines Billardspiels ist der Stoss. Dieser ist das Zusammentreffen zwischen zwei Körpern, bei dem die Kinetische Energie konstant bleibt. Bei der Kollision wird die kinetische Energie zunächst in potentielle Energie umgewandelt. Dies geschieht durch eine Verformung der Körper, welche die gespeicherte Energie dann wieder als kinetische Energie abgeben.

Grundsätzlich lassen sich Stossvorgänge in elastisch und inelastisch einteilen. Beim elastischen Stoss wird die gesamte kinetische Energie bei der Verformung der Körper in potentielle Energie umgewandelt. Beim inelastischen Stoss wird hingegen ein Teil der Energie in andere Energieformen konvertiert – vor allem Wärme, Schall und die plastische Verformung von einem der Körper.

Im makroskopischen Bereich ist der elastische Stoss nur als Idealisierung zu betrachten, da immer Wärme erzeugt wird. Wirklich elastische Kollisionen sind nur auf der Ebene von Atomen oder Quantenobjekten möglich. Zur Vereinfachung kann man den Verlust von kinetischer Energie vernachlässigen, unter der Voraussetzung, dass er gegenüber der Bewegungsenergie sehr klein ist. Bei der Kollision von zwei Billardkugeln – welche oft als Beispiel für den elastischen Stoss verwendet wird

ist dies der Fall (Cross 2014).

Im gegensatz zu den Kugeln, welche aus Phenoplast hergestellt werden, besteht der Bandenspiegel um den Tisch herum aus vulkanisierten Elastomeren, welche dazu dienen den Energieverlust des Balles gering zu halten. Je nach den gewählten Materialien unterscheidet sich das Effet verhalten des Balles. In unserem Fall gehen wir von einer harten und dichten Bande aus. Der Stoss kann so als rein elastisch betrachtet werden.

1.3 Rollreibung

Das zweite physikalische Phänomen, welches beim Billardtisch auftaucht ist die Rollreibung der Kugel auf dem Tisch beziehungsweise Tuch. Die Rollreibung lässt sich durch die nachfolgende Gleichung bestimmen.

$$F_R = c_R \cdot F_N$$

F_R : Rollwiderstand

F_N : Normalkraft

c_R : Rollwiderstandskoeffizient

Die Rollreibung entsteht durch die Verformung des Untergrundes durch das rollende Objekt, sowie an durch die Verformung des Objektes selbst. Je weicher der Untergrund oder Objekt desto größer die Verformung und dementsprechend die Reibung.

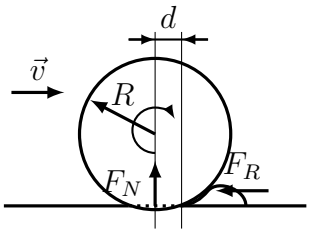


Abbildung 1: Schematische Darstellung der Kräfte und Verformungen beim Rollvorgang.

Die Verformung des Untergrundes besteht aus einem Einsinken des Körpers und einer Anhäufung von Material vor dem Körper selbst. Dies ist Schematisch in der obigen Skizze dargestellt. Ein Weicherer untergrund führt zu mehr Einsinken und damit auch zu mehr Material in der Bahn des Rollkörpers. Wenn die tiefe des Einsinkens bekannt ist lässt sich hieraus auch der Rollwiderstandskoeffizient zu $c_R = \frac{d}{R}$ bestimmen. Hierbei ist d die Strecke zwischen der Rotationsachse des Rollkörpers und der Stelle, an der sich dessen Außenradius mit der weitergedachten Höhe des Untergrundes schneidet. Da beim Billard entscheidend ist, dass die Rollreibung möglichst gering ist, werden Billardtische in der Regel aus Schiefer gefertigt, da dieser Stein besonders spröde ist. Auch Tische mit Holz oder Stahlplatten werden gefertigt. Der Tisch wird dann mit einem Schweren Tuch bespannt

2 Grundlagen

In diesem Kapitel werden die physikalischen Gesetzmäßigkeiten die zur Simulation des

Spiels nötig sind erklärt, so wie die Schritte die Nötig sind um sie in einer numerischen Simulation einzusetzen.

2.1 Physikalisch

2.1.1 Geradlinige Bewegung

Bevor es überhaupt zu den in der Einleitung angesprochenen Stössen kommen kann, muss zunächst die geradlinige Bewegung der Kugeln realisiert werden. Da die Ausgabe als 2D Grafik erfolgt, ist es naheliegend auch ein 2-Dimensionales Koordinatensystem für die Positionen und die Bewegungen der Kugeln zu wählen. Demnach hat jede Kugeln eine Position, eine Geschwindigkeit und eine Beschleunigung als Eigenschaften.

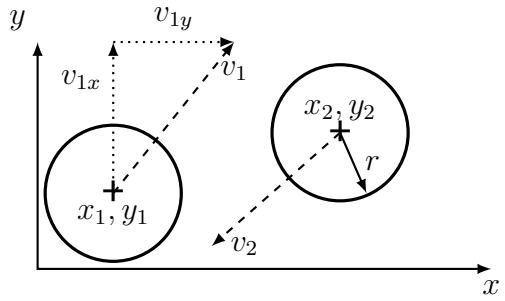


Abbildung 2: Koordinaten und Geschwindigkeiten von Kugeln

Die aktuelle Position der Kugel lässt sich in x, bzw y Richtung lässt sich also durch $x = v * t + x_0$ berechnen. Da in unserem die einzige wirkende Beschleunigung die konstante Rollreibung ist, lässt sich diese einfach als abnehmende Geschwindigkeit darstellen. Durch die iterative und diskrete Natur von Computerprogrammen macht es Sinn die aktuelle Position $\vec{p}$ anhand der zuletzt errechneten zu bestimmen.

$$\vec{p}_t = \vec{p}_{t-1} + \vec{v}_t$$

Hier wird dann noch in jedem Iterationsschritt die Geschwindigkeit um den durch die Rollreibung vorgegebenen Faktor verringert.

$$\vec{v}_t = \vec{v}_{t-1} \cdot (1 - \mu_R)$$

2.1.2 Kollision mit der Bande

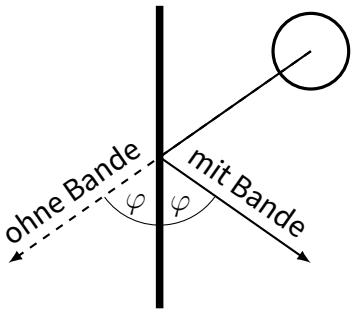


Abbildung 3: Kollision mit der Bande

Das nächste physikalische Problem ist die Kollision von Kugeln und Bande. Hier handelt es sich wie eingangs erwähnt um einen einen elastischen Stoss. Da die Bande unbeweglich und gerade ist gilt, dass sich nur der Anteil der Kraft im Vorzeichen ändert, welcher Senkrecht zur Wand steht. Demnach wäre die Geschwindigkeit nach der Kollision in Abbildung

2.2

$$v_0 = \begin{bmatrix} -v_{x,0} \\ v_{y,0} \end{bmatrix}, \quad v_1 = \begin{bmatrix} -v_{x,0} \\ v_{y,0} \end{bmatrix}$$

2.1.3 Kollision von zwei Kugeln

Wie oben bereits angeführt, ist das Zusammentreffen von zwei Kugeln ein elastischer Stoss. Für diesen gilt für dein eindimensionalen Fall folgende Gleichung.

$$v_{a1}m_{a1} + v_{b1}m_{b1} = v_{a2}m_{a2} + v_{b2}m_{b2}$$

In betrachteten Fall vereinfacht sich die Gleichung weiter, da alle Kugeln die selbe Masse haben, damit ergibt sich.

$$v_{a1} + v_{b1} = v_{a2} + v_{b2}$$

Dieser Zusammenhang sich allerdings nicht direkt auf den Zweidimensionalen Fall Anwenden. Dieser kann jedoch so vereinfacht werden, dass die obige Formel zumindest indirekt zur Berechnung verwendet werden kann. Um eine effiziente Berechnung im Computer zu gewährleisten, sollte auch möglichst auf den Einsatz von trigonometrischen Funktionen verzichtet werden, da diese nach eigenen Tests mit einem mehr an Rechenaufwand verbunden sind. Eine überprüfung von Verschiedenen Rechenoperationen ergab folgendes Ergebnis:[1]

Operation	Zeit für 10e8 Wiederholungen in s
plus	0.1681851
minus	0.1604299
mult	0.1584505
div	0.3010148
sqrt	0.2361831
sin	1.1046720
cos	0.9211685
tan	2.6617312
atan	2.5101218
exp	0.5926420

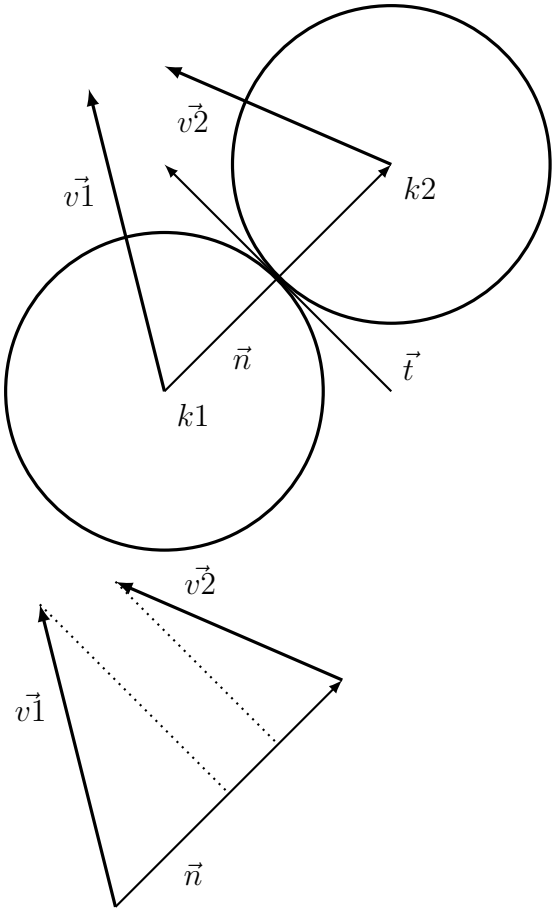
Demnach braucht eine $\sin()$ Operation fast 7x mehr Zeit, als eine Multiplikation. Da die Geschwindigkeiten und Positionen der Kugeln bereits Vektoren sind, ist es Sinnvoll auch die Berechnung der Geschwindigkeiten nach dem Stoss durchzuführen. Hierzu wird zunächst ein Vektor bestimmt und normiert, welcher vom Mittelpunkt von Kugel 1 zu Kugeln 2 Zeigt.

$$\vec{n} = \begin{bmatrix} k2_x \\ k2_y \end{bmatrix} - \begin{bmatrix} k1_x \\ k1_y \end{bmatrix}$$
$$\hat{n} = \frac{\vec{n}}{||\vec{n}||}$$

Die Geschwindigkeiten der Kugeln werden dann auf den Normalenvektor projiziert. Hieraus wird dann die Differenz der Geschwindigkeiten normal zum Differenzvektor der Kugeln ermittelt.

$$\Delta v = \hat{n} \bullet \vec{k2}_v - \hat{n} \bullet \vec{k1}_v$$

[1] Der für den Test verwendete Code findet sich im Anhang. Als Compiler wurde gcc, verwendet, da kein ersatz für die C++-crono library in Borland gefunden werden konnte.



Die so erhaltene Differenz der Geschwindigkeiten kann verwendet werden um die Finale Geschwindigkeit der beiden Kugeln zu errechnen. Diese ergibt sich dann zu

$$k1_{v2} = k1_{v1} + \Delta v \circ \hat{n}$$
$$k2_{v2} = k2_{v1} - \Delta v \circ \hat{n}$$

Dadurch, dass der Anteil in der normalen Richtung direkt zur vorherigen Geschwindigkeit addiert, bzw subtrahiert wird, ist eine separate Berechnung der Geschwindigkeiten in tangentiale Richtung nicht mehr notwendig. Auch unsere anfängliche Bedingung, dass die Summe der Geschwindigkeiten konstant bleiben muss ist somit erfüllt.

2.2 Informatik

Nachdem im vorherigen Kapitel die physikalischen Grundlagen diskutiert wurden die Notwendig sind, um ein zweidimensionales Billardspiel zu Modellieren, soll es nun um einige Ideen gehen die zur Praktischen Umsetzung notwendig sind.

2.2.1 Representation der Kugeln

In diesem Fall ist es recht naheliegend eine Klasse für die Kugeln zu setzen, wobei die Klasse folgende Attribute und Methoden hat.

Kugel		
n	:	int
r	:	double
pos	:	TVektor
next	:	TVektor
v	:	TVektor
inGame	:	bool
color	:	TColor
init()	:	void
move()	:	void
draw()	:	void

Abbildung 4: Kugelklasse UML-Diagram

Masse Auffällig ist hier, dass die Kugel, kein Attribut für ihr Masse besitzt. Da angenommen

wird, dass alle Kugeln die gleiche Masse besitzen spielt diese bei den Stossereignissen zwischen den Kugeln keine Rolle. Lediglich bei der beim abbremsen durch die Rollreibung würde das Gewicht eine Rolle spielen. Andererseits kann die Verlangsamung aber auch durch eine simple Konstante abgebildet werden und die Kugeln benötigt somit kein Attribut der Masse.

2.2.2 Kollisionserkennung

Um nun zu erkennen, ob eine Kugel mit einer anderen Kollidiert kann der Hypothenusensatz angewendet werden. Hierzu kann man die Positionen pos von zwei Kugeln subtrahieren und erhält so einen Vektor für den Abstand. Eine Kollision zwischen zwei Kugeln liegt also vor, wenn gilt:

$$r \geq \sqrt{(\mathsf{k1}_{\mathsf{pos,x}} - \mathsf{k2}_{\mathsf{pos,x}})^2 + (\mathsf{k1}_{\mathsf{pos,y}} - \mathsf{k2}_{\mathsf{pos,y}})^2}$$

Hierbei bezeichnen k1 und k2 jeweils eine Instanz der Kugel-Klasse.

Die Erkennung von Kollisionen mit der Bande gestaltet sich rechnerisch noch simpler.

$$r \geq |\mathsf{k}_{\mathsf{pos,x}} - \mathsf{Bande}_{\mathsf{x}}|$$

Allerdings kann es hier zu einem Optische störenden Phänomen kommen, bei dem die Kugel für einen Frame außerhalb der Spielfeldes gezeichnet wird. Da die Berechnung für die einzelnen Frames diskret verläuft, wird erst wieder im nächsten Durchlauf überprüft, ob die Bedingung für die Kollision erfüllt ist. Besonders bei hohen Geschwindigkeiten der Kugel, kann es also dazu kommen, dass diese merklich außerhalb der Spielgrenzen dargestellt wird.

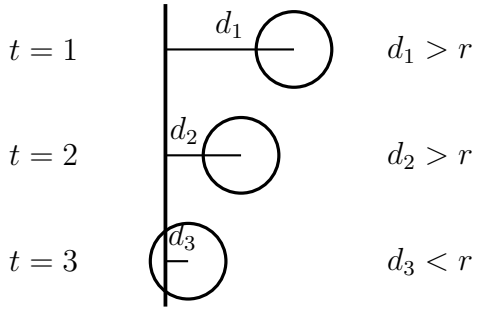


Abbildung 5: Erst für t_3 wird die Kollision erkannt.

Grundsätzlich gibt es drei Möglichkeiten dieses Problem zu lösen. Zum einen kann eine Höchstgeschwindigkeit eingeführt werden, um die Bewegung in Pixeln pro Iteration zu verringern. Eine Option ist es die Anzahl an Iterationen zu erhöhen um den selben Effekt zu erreichen. Die Lösung die jedoch nicht nur stochastische Sicherheit bietet ist die kontinuierliche Kollisionskontrolle CCD.

2.2.3 Continuous Collision Detection (CCD)

Um sicherzugehen, dass keine Kugeln außerhalb des Spielfeldes angezeigt werden können wurde an den Rändern eine CCD implementiert. Dabei wird berechnet, ob im kommenden Zeitschritt eine Kollision mit der Bande Auftritt, falls dies der Fall ist, wird Position

errechnet, welche die Kugel, nach der Richtungsänderung haben wird. Diese Position wird dann in das Attribut `next` der entsprechenden Kugel gespeichert und im nächsten Zeitschritt als Position der Kugel gesetzt.

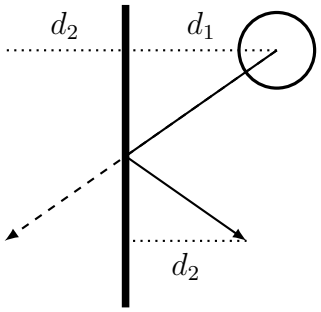


Abbildung 6: CCD Schaubild

Zur Berechnung ist vor allem der Anteil der Zeit wichtig, nach der die Kollision mit der Bande stattfinden würde. Um die kritische Zeit t_c zu erhalten, gilt:

$$t_c = \text{Bande}_x + r - \frac{\text{k}_{\text{next,x}}}{\text{k}_{\text{pos,x}} - \text{k}_{\text{next,x}}}$$

Hier ist t_c der Anteil der Zeit an einem Zeitschritt, nach dem die Kollision auftreten würde, wenn die Kugel die Bahn entlang der gestrichelten Linie in Abbildung 2.5 weiter mit Konstanter geschwindigkeit verfolgt. $\text{k}_{\text{next,x}}$ ist in diesem Szenario die Hypothetische Position, ohne CCD. Wenn man nun den die t_c kennt kann man die Strecken d_1 und d_2 bestimmen. Hier ergibt sich

$$\begin{aligned} d_1 &= t_c \text{k}_{\text{v,x}} \\ d_2 &= (1 - t_c)\text{k}_{\text{v,x}} \\ \text{k}_{\text{next,x}} &= \text{Bande}_\text{x} + r + (1 - t_c)\text{k}_{\text{v,x}}(1 - \mu_R) \end{aligned}$$

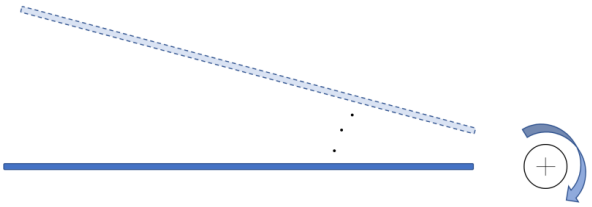
Der $(1 - \mu_R)$ Term, geht aus der Einbeziehung der Rollreibung hervor. Da in jedem Zeitschritt die Geschwindigkeit durch die Reibung verringert wird, muss dies auch bei der Berechnung der zukünftigen Positionen einbezogen werden.

Die so erhalten X-Koordinate wird als Position für die Kugel k im nächsten Zeitschritt gesetzt. Somit übertritt die Kugel niemals die Grenze des Spielfeldes und der oben genannte Darstellungsfehler wird behoben.

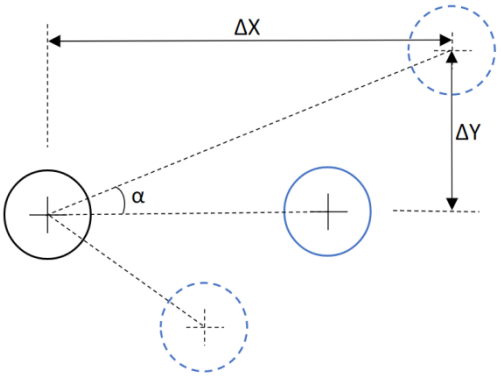
3 Erklärung der Graphischen Oberfläche

3.1 Queue

Die Zentralpunkt der Weißkugel wird nur berücksichtigt, damit der Schlagen zwischen Billardqueue und Weißkugel vereinfacht wird. Deshalb werden nur die Richtung und die Kraft des Billardqueue im Programm kontrolliert.



Wie in der Abbildung gezeigt, ist es notwendig, dass das Billardqueue rund um die Weißkugel $360°$ drehen kann. Außerdem muss es erfüllt werden, dass das Billardqueue sich verändert, wenn die Richtung der Zielkugel wechselt wird. Dieses stellt sicher, dass das Billardqueue in der Richtung bewegt. Deshalb wird die drehende Winkel des Billardqueue bestimmt, wenn die drehende Winkel der Zielkugel festgelegt wird. Im Folgend wird das Festlegen der Winkel von der Zielkugel beschrieben.



Grundlegende Änderung der Zielkugel wird in der Abbildung darstellt. Da es einstellt wird, dass die Zielkugel der Bewegung der Maus entspricht, werden die Länge und der Winkel der Zielkugel im Gegensatz zur Weißkugel geändert. Die Position der Zielkugel wird bestimmt und danach die Differentwerte zwischen Weißkugel und Zielkugel werden auch bekommen.

$$\tan(\alpha) = \frac{\text{Gegenkathete}}{\text{Ankathete}}$$

$$\alpha = \arccos \frac{\text{Ankathete}}{\text{Hypotenuse}}$$

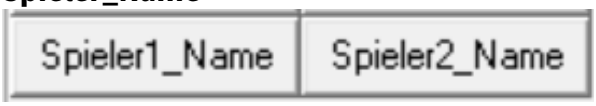
Nach Formel wird der Winkel durch die beiden Differentwerte berechnet. Aber es ist zu beachten, dass es Problem mit $\tan(\alpha)$ bei $\alpha = 90°$ gibt. Wenn α als $90°$ gleich, wird Ankathete gleich 0. Nach dem Prinzip, dass der Nenner nicht gleich 0 ist, geht das Programm schief. Deswegen wird der Winkel durch Formel 2 berechnet. Die Richtungen des Billardqueue und der Zielkugel können daher eingestellt und berechnet werden.

Die Kraft des Billardsqueue wird durch die Buttons in zwei Formen zeigt, wie in der Abbildung dargestellt.

Wenn der Button “Distanzenkraft” gedrückt wird, wird die Größe der Kraft proportional entsprechend der Distanzen zwischen Weißkugel und Zielkugel geändert. Wenn das Button “Zeitenkraft” gedrückt wird, wird die Größe der Kraft mit der Zeit proportional hin und her geändert.

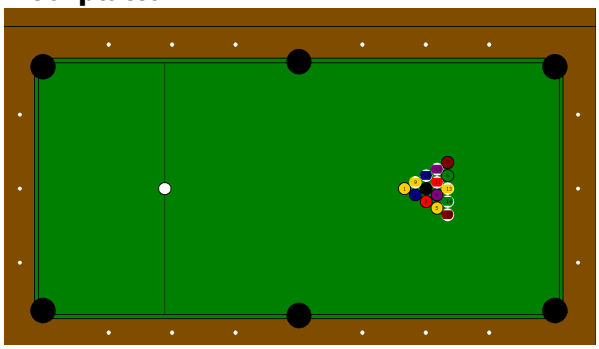
Im Programm ist es ersichtlich, dass die Größe der Kraft in eingestellter Umfang durch die rote Balkenänderung intuitiv dargestellt wird. Die niedrigste Grenze wird gestellt, um Gleitreibung zu vermeiden, und die hochste Grenze wird gestellt, um übermäßige Kollision zu vermeiden.

Spieler_Name



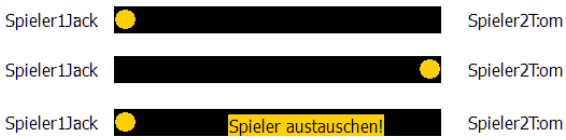
Wann das entsprechende Button in der Abbildung gedrückt wird, wird die Inputbox angezeigt, um den Name der Spieler, der man nennen möchte, einzugeben. Ansonsten wird der Standardname angezeigt.

Tischplatte



Auf der Tischplatte liegen 6 schwarzen Löcher und 16 Kugeln. Die Kugeln sind mit nummern und Farben gekennzeichnet. Die statistische Daten bzw. die Maße und Positionen der Kugeln sind In einem bestimmten Verhältnis standardisiert.

Spielereihe

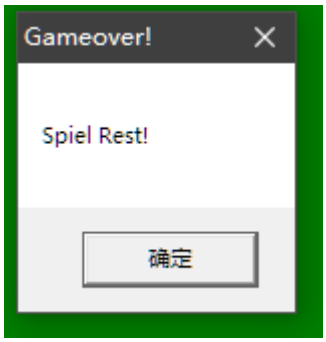


In jeder Runde sollte nur ein Spieler schlagen. Die Stelle des gelben Kugel zeigt den Spieler, der in dieser Runde schlagen sollte. Wenn die Reihe des Spielers gewechselt wird, entsteht der Hinweis „Spieler austaschen ”.

Zugehörigkeit Nachdem ein Spieler erst den Kugeln ins Loch gestossen, wird die Zugehörigkeit bestimmt. In einem Spiel sollte die bestimmte Zugehörigkeit nicht mehr verändert werden. Der Text auf gelbem Hintergrund zeigt, welche Kugeln(vollfarbe Kugeln oder Halbfarbe Kugeln) die unten gezeigten Spieler schlagen sollten.

Kugeln in löcher Das unter der Spielereihe stehenden Feld zeigt, welche Kugeln in diesem Augenblick in löcher sein, d.h. nicht in dem Game sind.

3.2 Gameover



Wenn ein Spieler den schwarzen Kugel zur falschen Zeit ins Loch geschlagen hat, wird der andere Spieler den Winner. In diesem Moment beendet das Spiel und taucht diesen Hinweis auf.

3.3 Winner

Wenn ein Spieler das Spiel gewinnt, tauscht bald diesen Hinweis auf.

4 Simulation

Nach dem in den vorangegangenen Kapiteln die zugrundeliegenden physikalischen Gesetze und ihre Umsetzung in ein Computerprogramm behandelt wurden, wird es im letzten Abschnitt um den Aspekt der Spielbarkeit gehen. Ein spiel, dass die physikalische Realität möglichst korrekt abbildet, sorgt nicht notwendigerweise für ein Angenehmes Spielerlebnis. In unserem Fall ist die Stellschraube an der wir drehen können vor allem die Reibung. Sie bestimmt, wie lange die Kugeln in Bewegung bleiben und damit ob ein Spieler lange warten muss, oder das Gefühl hat die Kugeln würden zu schnell liegen bleiben. Im folgenden werden die Beobachtungen aus einigen verschiedenen mathematischen Darstellungen der Verlangsamung der Kugel besprochen. Dabei soll untersucht werden, ob eine realitätsnahe Representation als angenehm empfunden wird, oder ob eine andere Darstellung als besser wahrgenommen wird.

4.1 Lineare Darstellung

Die einfachste Mathematische Funktion mit der die Reibung dargestellt werden kann. Hierbei wird die Reibung einfach für jede Iteration um einen konstanten Betrag reduziert, bis die Kugel zum Stillstand kommt.

$$v(t) = v_0 - c * t, \quad 0 \leq v$$

4.2 Quadratische Darstellung

Eine weitere Möglichkeit ist die Darstellung als Quadratische Funktion. Dabei werden die Kugeln zunächst nur wenig verlangsamt, bremsen aber mit ansteigenden Iterationen immer Schneller ab.

$$v(t) = v_0 - c * (c_1 t)^2, \quad 0 \leq v$$

Hierbei ist es angebracht, die Zeit mit einer konstante $0 < c_1 \ll 1$ zu multiplizieren um zu verhindern, dass die Kugeln bereits nach wenigen Durchlaufen stehen bleibt.

4.3 Exponentaldarstellung

5 Fazit

Quellen

Cross, R. 2014. „Ball collision experiments“. *Physics Education* 50 (1): 52–57. https://doi.org/10.1088/0031-9120/50/1/52.