

Billiard-AI

Ein intelligenter Billardtisch

BSc Thesis

Studienrichtung:

Autor:

Dozent:

Experte:

Datum:

Informatik - Computer Perception and Virtual Reality

Lukas Seglias, Luca Ritz

Markus Hudritsch

Andreas Dürsteler

27. Oktober 2021

Inhaltsverzeichnis

1 Zusammenfassung	1
2 Einführung	3
3 Ziele	5
3.1 Planung	6
3.1.1 Meilensteine	6
4 Billiard-AI	9
4.1 Vorarbeiten	9
4.2 Risikoanalyse	10
4.3 Klassifikation	11
4.4 Suchalgorithmus	12
4.4.1 Kandidatensuche	12
4.4.1.1 Bewertungsfunktion	13
4.4.2 Berechnung der Initialgeschwindigkeit	16
4.4.2.1 Reibungsverlust über Bahn	16
4.4.2.2 Elastischer Stoss bei Kugelkollision	16
4.4.3 Modellierung physikalisches System	18
4.4.3.1 Ereignisse und ihre Repräsentation	19
4.4.3.2 Kantenfunktion	20
4.4.3.3 Layer	20
4.4.3.4 Beispiel eines Graphen	20
4.4.4 Simulation	21
4.4.4.1 Reibungsverlust über die Zeit	22
4.4.4.2 Elastischer Stoss zweier Kugeln	22
4.4.4.3 Bandenreflektion	23
4.4.4.4 Kollisionsprüfung	23
4.4.4.5 Ereignis - Energie-Transfer über Kugelkollision	24
4.4.4.6 Ereignis - Energie-Transfer über Bandenkollision	26
4.4.4.7 Ereignis - Totaler Energieverlust	27
4.4.4.8 Ereignis - Verlassen des Systems	27
4.4.4.9 Bewertungsfunktion	28
4.4.5 Animation	29
4.5 Berechnungsprozess	31
5 Resultate	35
5.1 Klassifikation	35
5.2 Bestimmen des Rollreibungskoeffizienten	36
5.3 Energieverlust	37
5.3.1 Energieverlust bei Kugelkollision	37
5.3.2 Energieverlust bei Bandenkollision	37
6 Weitere Arbeiten	39
7 Fazit	41
8 Anhang	51
8.1 Herleitung Startgeschwindigkeit auf Basis bekannter Endgeschwindigkeit unter Einbezug von Reibung	51
8.2 Herleitung Initialgeschwindigkeit bei Kugel-Kollision	52
8.3 Herleitung Ereignis - Totaler Energieverlust	53
8.4 Herleitung Ereignis - Kollision dynamischer Objekte	55
8.5 Herleitung Ereignis - Kollision mit Bande	57
8.6 Herleitung Ereignis - Kollision mit Ziel	59

8.7	Herleitung Beschleunigung	61
8.8	Bestimmung Reibungskoeffizient	61
8.9	Herleitung Linie-Linie-Schnittpunkt	64

1 Zusammenfassung

TODO: Zusammenfassung

2 Einführung

Wie vieles andere ist auch das Erlernen des Billardspiels eine schwierige Sache. Es stellen sich Fragen wie „Welche Kugel soll man anspielen?“, „Wie soll man die Kugel anspielen?“ oder „Wie hält man den Queue richtig?“. Daraufgehend gibt es noch diverse weitere Überlegungen, welche den Profi vom Anfänger unterscheiden. Wie in anderen Spielen auch, ist hier Weitsicht gefragt. Es geht also nicht nur darum, eine Kugel zu versenken, sondern auch den Spielstand so zu verändern, dass optimal weitergespielt werden kann. Das Stichwort ist im Billard vor allem die Platzierung der weissen Kugel.

Diese Arbeit widmet sich nun einem Assistenten, der einen in die schwierige Kunst des Billardspiels einführen soll.

3 Ziele

Es wird ein Billardassistent entwickelt, welcher aufgrund des aktuellen Spielstands einen Stoss berechnet, der einerseits einfach durchzuführen ist und andererseits eine optimale Ausgangslage für den nächsten Stoss bieten soll. Es gab bereits einige Vorarbeiten[Luk21a], wie aus dem Kapitel 4.1 zu entnehmen ist.

Die Ziele für diese Arbeit sind nun die Klassifikation der Kugeln, die Entwicklung einer effizienten gerichteten Suche nach einem Stoss, die darauffolgende, möglichst realitätsnahe Simulation sowie einige Verbesserungen bei der Darstellung.

Es werden zwei verschiedene Spielmodi angestrebt. Beim ersten Modus kann der Spieler eine Kugel oder einen Kugeltypen (Farbe) wählen, nach dem gesucht wird. Daraufhin werden ihm alle gefundenen Resultate in geordneter Reihenfolge nach Schwierigkeit dargestellt. Die Suche kann über mehrere Stösse gehen und dabei werden auch die Regeln des Spiels beachtet. Dies ist wichtig, um die optimale Platzierung der weissen Kugel zu eruieren. Nach einem Stoss kann der Spieler wiederum eine Wahl treffen. Der zweite Modus ist der sogenannte „Infinity-Mode“. Die Idee dahinter ist die vereinfachte Einführung des Spielers. So soll das System bei stabilem Zustand automatisch eine Suche nach allen Kugeln starten und das beste Ergebnis anzeigen. Sobald der Spieler den Stoss durchgeführt hat, wird das System wiederum automatisch eine Suche starten. Dadurch kann ein Spieler optimal lernen, welche Stösse einfach sind und wo eine Kugel mit welcher Geschwindigkeit getroffen werden muss, um das gewünschte Ergebnis zu erzielen. Die Regeln spielen in diesem Fall eine untergeordnete Rolle und werden daher ignoriert.

Es ist weiterhin anzumerken, dass es in erster Linie um Snooker-Billard geht. Dies hat mehrere Gründe. Einerseits soll in dieser Arbeit nicht die Klassifikation der Kugeln im Zentrum stehen, sondern die Suche nach einem optimalen Stoss. Es wird angenommen, dass dies mit Snooker-Kugeln einfacher ist als mit Pool-Billard-Kugeln. Andererseits wird das Projekt zusammen mit einem Unternehmen durchgeführt, welches eventuell auch einen kommerziellen Ansatz verfolgen will. Da grössere Turniere wie Weltmeisterschaften in Snooker ausgetragen werden, kam schnell der Wunsch auf, das Hauptaugenmerk darauf zu legen. Nichtsdestotrotz wird die Anwendung so abstrakt gehalten, dass sie mit wenig Aufwand auf Pool-Billard portiert werden könnte. Dies bildet jedoch kein Ziel der Bachelor-Thesis.

3.1 Planung

Die initiale Planung beinhaltet eine Auflistung der Tätigkeiten, deren zugewiesenen Meilensteine sowie deren Schätzung in PT (Personen-Tage). Jedem Arbeitspaket wird eine ID zugewiesen, welche bei der Zeiterfassung verlinkt wird. Das Total der zu vergebenden PT beträgt 90.

ID	Name	Meilenstein	Schätzung in PT
T-1	Klassifikation der Kugeln	M-1	6
T-2	Aufsetzen Dokumentation	M-1	1
T-3	Beschreibung Suchalgorithmus	M-1	3
T-4	Implementation Suchalgorithmus	M-1	5
T-5	Beschreibung der physikalischen Eigenschaften für die einfache Suche	M-1	6
T-6	Implementation der einfachen Suche und deren Bewertungsfunktion	M-1	14
T-7	Beschreibung der physikalischen Eigenschaften für die erweiterte Suche	M-2	6
T-8	Implementation der erweiterten Suche und deren Bewertungsfunktion	M-2	8
T-9	Überprüfen/Verbessern der Detektionsgenauigkeit	M-1	6
T-10	Video erstellen	M-3	2
T-11	Plakat schreiben	M-3	2
T-12	Booklet-Eintrag schreiben	M-3	1
T-13	Präsentation des Finaltags vorbereiten	M-3	2
T-14	Präsentation der Verteidigung vorbereiten	M-3	2
T-15	Finalisieren Dokumentation andere Arbeiten	M-3	4
T-16	Projektmanagement	Kein	4
T-17	Effizienz Erfassung und Steigerung der einfachen Suche	M-1	0
T-18	Effizienz Erfassung und Steigerung der erweiterten Suche	M-2	0
T-19	Dokumentation der Resultate der einfachen Suche	M-1	6
T-20	Dokumentation der Resultate der erweiterten Suche	M-2	2
T-21	Umbau in Unity	M-1	4
T-22	Infinity-Modus	M-1	5
O-1	Suche über mehrere Spielstände	M-2	
O-2	Detektion des Queues in 2D	M-2	
O-3	Detektion des Queues in 3D	M-2	
O-4	Stossberechnung anhand detektiertem Queue in 2D	M-2	
O-5	Stossberechnung anhand detektiertem Queue in 3D	M-2	
O-6	Spielerabhängige Heuristik	M-2	
O-7	Live-Verfolgung und Darstellung der Kugeln	M-2	
Total			90

Tabelle 3.1: Ziele

3.1.1 Meilensteine

Es werden drei Meilensteine definiert, welche auch aus optionalen Zielen bestehen können. Die Deadlines ergeben sich aus den Schätzungen der zugewiesenen Arbeitspakete.

Meilenstein 1 - 15.11.2021 Das Ziel ist eine sehr einfache simple Suche. Darunter zu verstehen ist eine Lösung, welche einen direkten Treffer findet (Weiss -> Kugel -> Loch).

Code Deliverables:

Klassifikation - T-1

Alle Kugeln können entsprechend ihrer Farbe klassifiziert werden.

Suchalgorithmus für einfache Suche - T-4, T-6, T-17

Ein direkter Stoss wird in akzeptabler Zeit gefunden.

Unity-Umbau - T-21

Unity ist bereit für den Einsatz. Zum Umbau gehören insbesondere die Farbe der Markierung der Kugeln und deren Bahnen. Weiterhin muss Unity mehrere Suchergebnisse anzeigen können.

Infinity-Modus - T-22

Der Spieler erhält automatisch einen Vorschlag für einen Stoss, wenn die Kugeln auf dem Tisch stillstehen. Sobald der Spieler einen Stoss ausgeführt hat und die Kugeln wieder stillstehen, wird der nächste Stoss vorgeschlagen. Vorgeschlagen werden Stösse, welche eine beliebige Kugel ins Loch spielen.

Dokumentation Deliverables:

Klassifikation - T-1

Das Vorgehen der Klassifikation wie deren Resultate und Genauigkeit sind dokumentiert.

Suchalgorithmus für Suche - T-3

Der Algorithmus der Suche ist theoretisch und mit Pseudocode beschrieben. Die theoretische Beschreibung muss nicht gänzlich mit der effektiven Implementation übereinstimmen, da diese auf Performance optimiert wird.

Resultate der einfachen Suche - T-19

In den Resultaten ist die Genauigkeit und Performance des einfachen Suchvorgangs beschrieben.

Physik der einfachen Suche - T-5

Die benötigte Physik der einfachen Suche ist beschrieben.

Bewertungsfunktion - T-6

Die Bewertungsfunktion der einfachen Suche ist dokumentiert.

Meilenstein 2 - 13.12.2021 Das Ziel ist eine erweiterte Suche, die auch indirekte Stösse über weitere Kugeln oder Banden finden kann. Optional sollen auch mehrere Stösse berücksichtigt werden.

Code Deliverables:

Suchalgorithmus für erweiterte Suche - T-8, T-18

Ein indirekter Stoss wird in akzeptabler Zeit gefunden.

Suchalgorithmus über mehrere Stösse - O-1

Es werden mehrere Spielstände bei der Suche berücksichtigt.

Queue in 2D detektieren - O-2

Der Queue wird als 2D-Objekt detektiert.

Queue in 3D detektieren - O-3

Der Queue wird mittels Tiefeninformationen der Kamera als 3D-Objekt detektiert.

Stossberechnung anhand detektiertem 2D-Queue - O-4

Der Stoss wird je nach Haltung des Queues in 2D berechnet. Es wird angenommen, dass der Queue zentral auf die weisse Kugel gerichtet ist.

Stossberechnung anhand detektiertem 3D-Queue - O-5

Der Stoss wird je nach Haltung des Queues in 3D berechnet. Der Queue muss nicht mehr zentral auf die weisse Kugel gerichtet sein.

Spielerabhängige Heuristik - O-6

Je nach Spieler kann eine andere Heuristik zur Bewertung der Stösse eingestellt werden. Durch die Unterscheidung können für professionelle Spieler erfolgsversprechendere schwerer durchzuführende und für Anfänger eher leichtere Stösse gefunden werden.

Live-Verfolgung und Darstellung der Kugeln - O-7

Die Kugeln werden ohne Benutzereingabe getrackt und deren Position über den Projektor dargestellt.

Dokumentation Deliverables:

Physik der erweiterten Suche - T-7

Die benötigte Physik der erweiterten Suche ist beschrieben.

Resultate der erweiterten Suche - T-20

In den Resultaten ist die Genauigkeit und Performance des erweiterten Suchvorgangs beschrieben.

Bewertungsfunktion - T-8

Die Bewertungsfunktion der erweiterten Suche ist dokumentiert.

Meilenstein 3 - 17.01.2022 Das Ziel ist der Abschluss aller Arbeiten zu denen auch Plakat, Booklet oder Video gehören.
Deliverables:

Video - T-10

Das finale Video ist erstellt.

Plakat - T-11

Das Plakat ist erstellt.

Booklet - T-12

Der Booklet-Eintrag ist erstellt.

Präsentation für Finaltag - T-13

Die Präsentation/Ausstellung für den Finaltag ist vorbereitet.

Präsentation für Verteidigung - T-14

Die Präsentation für die Verteidigung ist vorbereitet.

Finalisieren der Arbeiten - T-15

Die Dokumentation wie auch der Code sind abgeschlossen.

4 Billiard-AI

4.1 Vorarbeiten

Zu dieser Bachelor-Thesis gab es bereits eine Vorarbeit im letzten Semester. Diese hatte zum Ziel, einige grundlegende Funktionalitäten bereitzustellen. In Abbildung 4.1 ist der Zyklus angegeben, welcher alle Aufgaben zusammenfasst. In einem ersten Schritt wird der aktuelle Spielstand über eine Kamera detektiert. Daraus kann eine textuelle Beschreibung abgeleitet werden. Diese dient wiederum als Input für das Kernstück, die Suche nach einem optimalen Stoss. Als Ausgabe dieses Schrittes erfolgt eine detaillierte Animation des auszuführenden Stosses. Die Animation wird über einen Projektor auf dem Billardtisch dem Spieler zugänglich gemacht. Dieser kann den Stoss ausführen, was zu einer neuen Situation führt und der Zyklus beginnt von vorne.

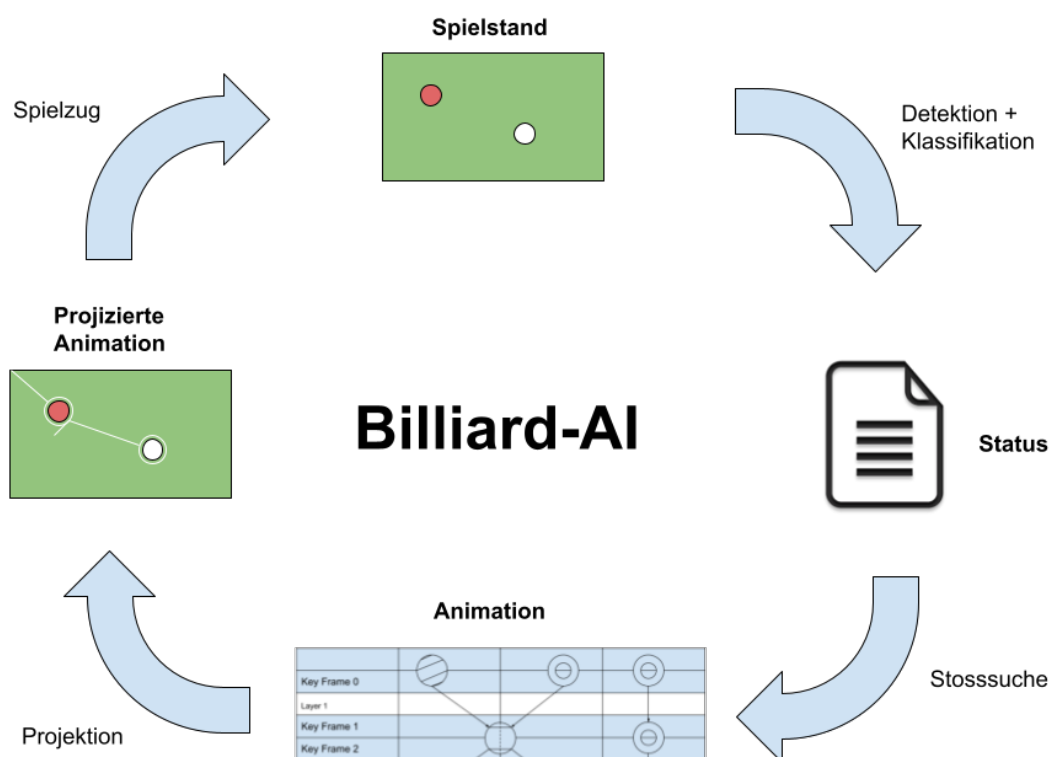


Abbildung 4.1: Billard-AI-Cycle

Es sind nun die Schritte der Detektion des Spielstands wie auch die Projektion und Darstellung der Animation auf dem Spieltisch bereits umgesetzt. Dabei ging es in erster Linie um die Frage der Repräsentation in geeignetem Koordinatensystem sowie die Messung der Genauigkeit dieser Übersetzung und der darauf basierenden Anzeige über den Projektor.

Die Bachelor-Thesis widmet sich zuerst der Frage nach der Klassifikation in Schritt eins, wie auch der Suche nach dem optimalen Stoss in Schritt zwei.

4.2 Risikoanalyse

Es gibt diverse Risiken, die während dieser Arbeit eintreten können. Um das Bewusstsein dafür zu stärken, wird vorgängig eine Risikoanalyse durchgeführt, wobei es um die Identifikation wie auch die Zuordnung deren Auftretenswahrscheinlichkeit und Auswirkungen geht. Weiterhin werden geeignete Massnahmen definiert, die entweder die Eintrittswahrscheinlichkeit reduziert oder bei Auftreten angegangen werden können.

Die Risiken werden im Detail aufgelistet. Die Spalte „WK“ steht für die Eintrittswahrscheinlichkeit, welche in den Wahrscheinlichkeiten „Gering“, „Möglich“, „Wahrscheinlich“, „Sehr Wahrscheinlich“ angegeben wird. Die Spalte „AW“ steht für die Auswirkungen, welche in den Grössen „Klein“, „Mittel“, „Gross“ angegeben wird.

ID	Risiko	Massnahme	WK	AW
R-1	Verlust von Programm-Sourcen	Führen eines Repositories auf GIT, welches das Wiederherstellen eines bestimmten Standes erlaubt. Zudem wird jeden Freitag ein Backup des GIT-Standes auf eine externe Festplatte geschrieben, sollte der unwahrscheinliche Fall eintreten, dass GIT nicht mehr verfügbar sein sollte oder seine Bestände verliert.	Möglich	Mittel
R-2	Ausfall der Arbeitsgeräte	Es stehen Ersatzgeräte bereit, welche sofort zum Einsatz kommen könnten.	Möglich	Klein
R-3	Krankheitsausfall der Teammitglieder	Es wird wenn möglich von Zuhause aus gearbeitet, um das Risiko einer Ansteckung zu vermindern.	Möglich	Gross
R-4	Parallele Entwicklung derselben Funktionen	Durch eine anfängliche Planung der Arbeitspakete, wie auch den ständigen Austausch und Einsatz von Pair-Programming an geeigneten Stellen, wird das Risiko stark reduziert. Sollte es trotzdem Eintreten, sind die Auswirkungen marginal, da die ständige Kommunikation dies sofort aufdecken würde.	Gering	Klein
R-5	Unterschätzen der Komplexität	Um zumindest ein brauchbares Resultat vorweisen zu können, wurde der erste Meilenstein möglichst simpel gehalten.	Wahrscheinlich	Gross
R-6	Verpassen wichtiger Termine	Es wird ein Kalender mit allen Terminen geführt, welcher mehrmals eine Erinnerung anzeigt.	Gering	Gross

Tabelle 4.1: Risiken

Die identifizierten Risiken werden in der Abbildung 4.2 für eine bessere Übersicht eingetragen.



Abbildung 4.2: Risikoanalyse

4.3 Klassifikation

4.4 Suchalgorithmus

Die Suche wird über zwei Schritte durchgeführt. Der erste Schritt besteht aus der Suche nach einem Lösungskandidaten, wobei vom Ziel aus ein Stoss gesucht wird, welcher das Potenzial hat, eine Kugel in diesem Ziel zu versenken. Das Resultat dieses Schrittes ist lediglich der Geschwindigkeitsvektor der weissen Kugel. Ob dieser Stoss tatsächlich das Resultat zur Folge hat, welches er voraussagt, wird im zweiten Schritt geprüft. Der Geschwindigkeitsvektor der weissen Kugel kann mit unterschiedlichem Betrag in einen Simulationsschritt eingegeben werden. Der Simulationsschritt wird als Resultat ein physikalisches System wie in Kapitel 4.4.3 ergeben.

Im Nachfolgenden wird auf die verschiedenen Schritte und deren Funktionsweise sowie die optimale parallele Durchführung der Berechnungen eingegangen.

4.4.1 Kandidatensuche

Die Kandidatensuche, im Folgenden Suche genannt, wird über eine klassische Graphensuche durchgeführt, wobei der vollständige Graph alle möglichen Stösse enthält. Für die Suche gibt es zwei Möglichkeiten, entweder wird bei der weissen Kugel gestartet und von dort ein Stoss gesucht, welcher eine andere Kugel ins Loch spielt, oder es wird bei einem oder mehreren Löchern gestartet und von dort ein Stoss gesucht, welcher von der weissen Kugel ausgehend eine andere Kugel ins Loch spielt.

Nachfolgend wird die Suche beschrieben, welche beim Loch, dem Ziel, startet, eine einzulochende Kugel findet und anschließend den Stoss bis zur weissen Kugel zurück sucht. Dementsprechend ist der Root-Knoten des Suchbaumes das zu treffende Ziel (Loch). Da ein handelsüblicher Billardtisch mehrere Löcher hat, muss pro Loch eine separate Suche durchgeführt werden.

Bei der Durchführung eines Expansionsschrittes werden ausgehend von einem Knoten im Suchbaum dessen Nachfolger-Knoten ermittelt. Diese stellen im Fall vom Root-Knoten Kugeln dar, welche in dieses Loch gespielt werden könnten. Aus diesen Kugeln werden Kugel-Knoten gebildet, welche diese Kugeln entweder auf direktem Wege oder indirekt über die Bande in das Loch spielen lassen sollen. Ausgehend von diesen Kugel-Knoten, werden deren Nachfolger-Knoten in weiteren Expansionsschritten ermittelt, welche wiederum Kugel-Knoten darstellen. Diese Kugel-Knoten stellen dann Kugeln dar, welche die Kugel des Vorgänger-Kugel-Knotens entweder auf direktem Wege oder indirekt über die Bande treffen sollen. Sofern ein Kugel-Knoten die weisse Kugel darstellt, so ist dieser Kugel-Knoten ein Endzustand und damit ist der Stoss über die Kette von Nachfolger- zu Vorgänger-Kugel-Knoten definiert.

Zur Veranschaulichung des Prinzips folgt ein Beispiel. Es wird vereinfacht angenommen, dass der Tisch nur ein Loch hat. Für mehrere Ziele ergeben sich mehrere Suchbäume, einen pro Loch. In Abbildung 4.3 erfolgt die Eingabe des Suchalgorithmus in Form des Root-Knotens. Es wird nur das zu treffende Ziel definiert. Auf der rechten Seite des Tisches ist der Suchbaum dargestellt.

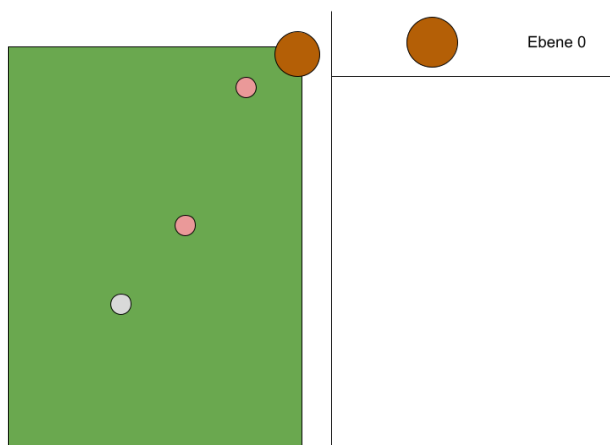


Abbildung 4.3: Kandidatensuche 1

In einem zweiten Schritt wird die einzulochende Kugel definiert. Es kommen lediglich die beiden roten Kugeln in Frage. Nachfolgend wird der Pfad weiter betrachtet, bei dem die rote Kugel, welche näher beim Loch ist, gewählt wurde. Abbildung 4.4 zeigt, dass der Suchbaum um einen Knoten erweitert wurde.

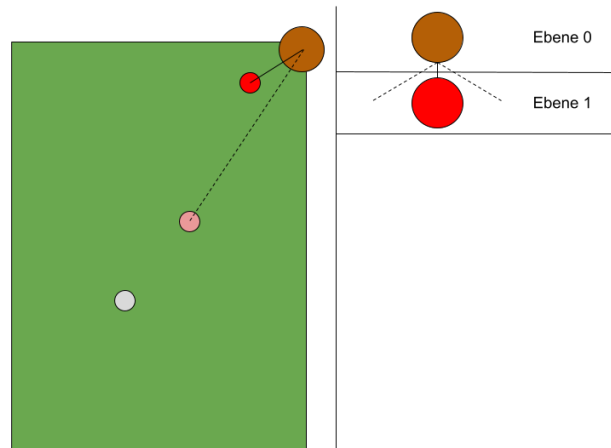


Abbildung 4.4: Kandidatensuche 2

In Abbildung 4.5 erfolgt der letzte Schritt. Hier sind verschiedene Optionen möglich, bspw. könnte die weiße Kugel direkt oder über die Bande an die zuvor gewählte rote Kugel gespielt werden. Es könnte aber auch die andere rote Kugel an die zuvor gewählte rote Kugel gespielt werden. Hier wird der Fall betrachtet, dass die weiße Kugel indirekt über eine Bande an die zuvor gewählte rote Kugel gespielt wird.

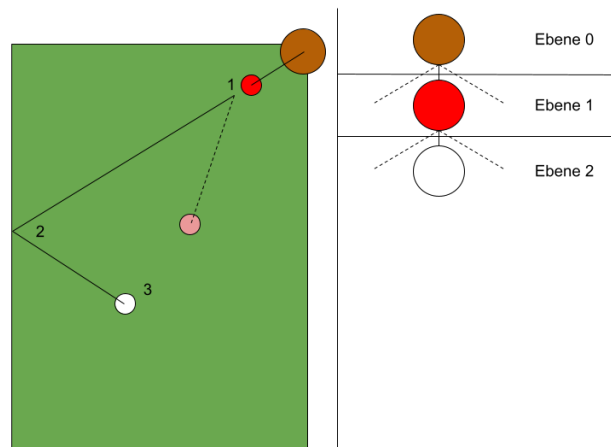


Abbildung 4.5: Kandidatensuche 3

Algorithmus 1 verdeutlicht den Ablauf der „Expand-Funktion“. Zuerst wird eine leere Liste namens „nodes“ angelegt. Diese wird danach mit Nodes gefüllt, welche entweder durch einen Stoß über eine weitere Kugel oder indirekt über die Bande zustande kommen. Die Nodes bilden das Ergebnis der Funktion.

```

Function expand(node: Node, constantObjects: list) → list[Node]
    nodes ← list()
    nodes ← append(expandBalls(node, constantObjects), nodes)
    nodes ← append(expandBank(node, constantObjects), nodes)
    return nodes

```

Algorithm 1: Algorithmus zur Durchführung eines Expansionsschritts bei der Kandidatensuche

4.4.1.1 Bewertungsfunktion

Um die Suche zu vereinfachen und in eine spezifische Richtung zu lenken, wo die besten Resultate zu erwarten sind, ist es unerlässlich eine Bewertung des Stoßes durchzuführen. Die Heuristik wurde so definiert, dass sie sich auf den aktuell expandierten Knoten beschränkt. Die Kosten für eine Expansion werden über die Pfade aufsummiert. Anhand der Summen kann jeweils der kostengünstigste Pfad evaluiert und verfolgt werden. Das Prinzip wird in Abbildung 4.6 veranschaulicht. Die Knoten werden je nach Bedeutung mit unterschiedlichen Farben markiert. Blau sind sie, wenn sie bereits expandiert wurden. Grün, wenn der Knoten im nächsten Schritt expandiert wird, da er die geringsten Pfadkosten aufweist und rot, wenn der Knoten

nicht in Frage kommt aufgrund zu hoher Kosten. Dieses Vorgehen entspricht demjenigen des Dijkstra-Algorithmus [Unk21c].

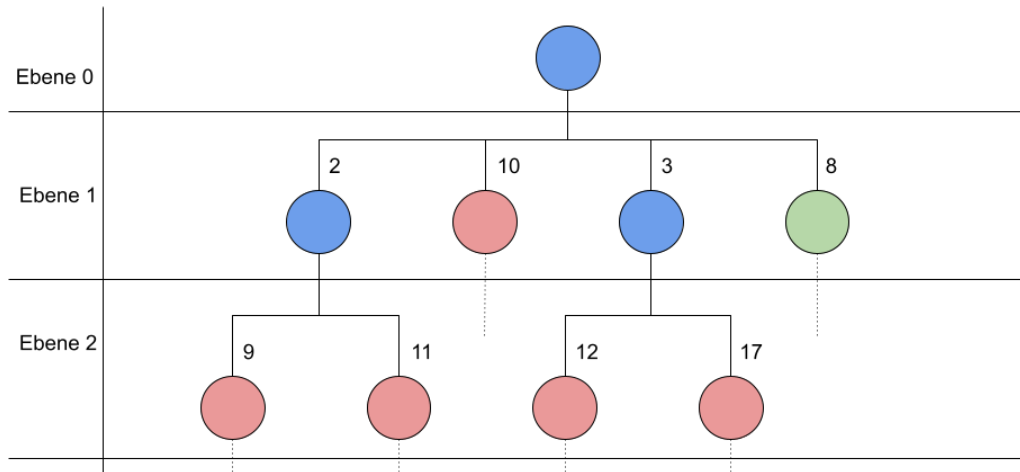


Abbildung 4.6: Bewertung eines Suchbaums

Die Kosten werden auf Basis dreier Kriterien¹ gebildet. Das Erste bildet die Berücksichtigung der Distanz, welche eine Kugel zurücklegt. Das Zweite behandelt den Winkel, in welcher ein Zielpunkt getroffen werden muss. Das Dritte fügt den Kosten einen Wert für jede Indirektion hinzu. Jeder dieser Werte liegt zwischen 0 und 1. Die ersten beiden Kriterien sind in Abbildung 4.7 veranschaulicht. Es werden zwei Expansionen gezeigt. Die relevanten Informationen d für die Distanz sowie α für den Winkel weisen einen entsprechenden Index auf, welcher den Expansionsschritt markiert. Beim ersten Expansionsschritt bildet der Zielpunkt den Elternknoten. Der Winkel α_1 ist definiert durch die Rollrichtung der Kugel und einer Normalen auf den Zielpunkt. Die Normalen zeigen jeweils zum Ursprung in der Mitte des Tisches. Die Distanz d_1 ist definiert über die Länge des zurückzulegenden Weges. Im zweiten Fall wird der Winkel α_2 durch die Rollrichtung der ersten und der Rollrichtung der zweiten Kugel definiert.

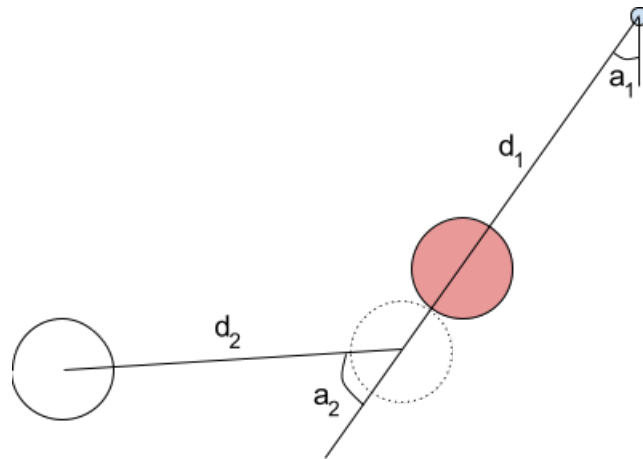


Abbildung 4.7: Expansionskosten eines Knotens

Um die beiden Größen vergleichen zu können, müssen sie in dieselbe Größenordnung gebracht werden. Aus diesem Grund werden sie durch die maximal möglichen Werte normiert[Mül21]. Für die Distanz ist dies die Diagonale über den Tisch, für den Winkel wird ein maximal möglicher Wert von 90° gewählt, dies ist bereits bei der Suche berücksichtigt. Daher kann der Winkel nur noch umgerechnet werden. Die resultierende 0 beim Cosinus bei 90° wird auf 1 gemappt, die 1 bei 0° auf 1. Nun gilt die Annahme, dass je kürzer der Weg und je kleiner der Winkel, desto einfacher der Stoss.

$$d_{krit} = \frac{d_i}{d_{max}} \quad (4.1)$$

$$\alpha'_{krit} = \cos \alpha * -1 + 1 \quad (4.2)$$

¹ Die Kriterien betreffend der Distanz wie auch des Winkels werden wie in Publikation [Smi06] verwendet.

Aktuell fließt der Winkel α_{krit} zu stark in die Bewertung ein. Daher wird dieser durch eine kubische Bézier-Kurve[Unk21b] gewichtet. Die Parameter lauten wie folgt.

$$P_0 = \begin{pmatrix} 0 & 0 \end{pmatrix} \quad (4.3)$$

$$P_1 = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad (4.4)$$

$$P_2 = \begin{pmatrix} 0.5 & 1 \end{pmatrix} \quad (4.5)$$

$$P_3 = \begin{pmatrix} 1 & 1 \end{pmatrix} \quad (4.6)$$

$$P = \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} \quad (4.7)$$

$$T = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \quad (4.8)$$

$$M = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (4.9)$$

$$\alpha_{krit} = f(t = \alpha'_{krit})[1] = T \cdot M \cdot P \quad (4.10)$$

Nach dieser Gewichtung wird die y-Komponente als α_{krit} verwendet. Die Kurve ist in Abbildung 4.8 visualisiert. Es wird deutlich, dass kleinere Winkel einen eher geringen Einfluss auf die Kosten haben. Ab einem Winkel von 50° , welche auf der Grafik 4.8 als Punkt D_1 markiert ist, beginnt die Kurve bis etwa 80° stark zu steigen. Ab dort flacht sie wiederum ab, bis die Kosten von 90° schliesslich den Wert 1 erreichen.

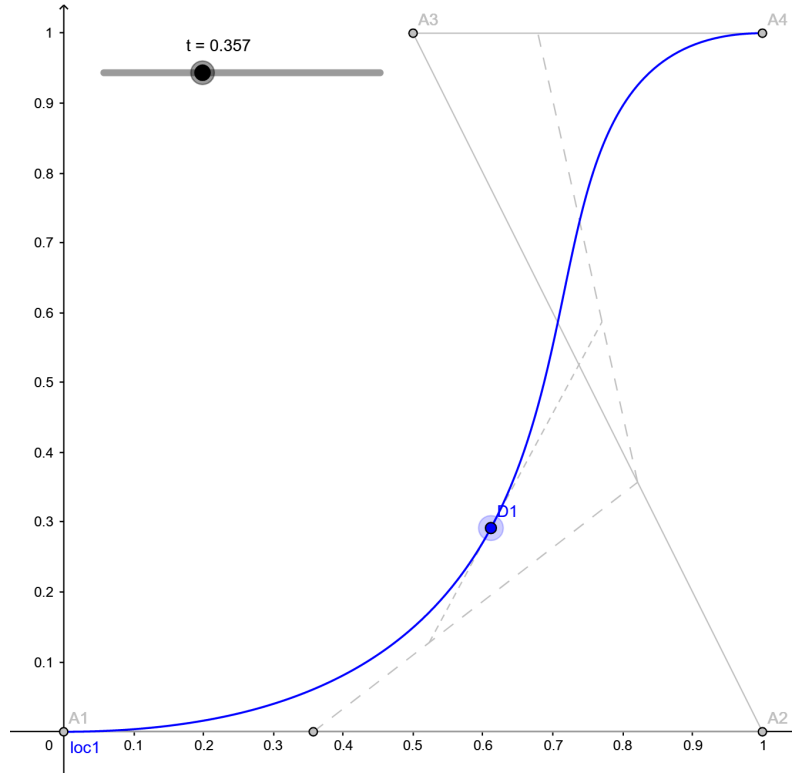


Abbildung 4.8: Gewichtung der Winkelkosten

Das Kriterium der Indirektion über Kugeln wird wiederum über einen maximal möglichen Wert gelöst. Es wird eine maximale Indirektion $K_{I,max}$ angegeben und die Anzahl der Vorkommnisse $K_{I,n}$ wird durch die Konstante dividiert. Dadurch werden die Kosten erhöht, je grösser der Indirektionsgrad ist.

$$K_{I,krit} = \frac{K_{I,n}}{K_{I,max}} \quad (4.11)$$

Die endgültigen Kosten werden über die Addition aller Kriterien gebildet.

$$K = d_{krit} + \alpha_{krit} + K_{I,krit} \quad (4.12)$$

4.4.2 Berechnung der Initialgeschwindigkeit

Sofern ein Stosskandidat wie in Abschnitt 4.4.1 beschrieben gefunden wurde, muss berechnet werden, welche Geschwindigkeit die weisse Kugel erhalten muss, damit die gewünschte Kugel ins Loch gespielt wird. Relevant dafür ist der Reibungsverlust wenn die Kugel rollt und die Kollision mit anderen Kugeln.

Wenn eine Kugel in ein Loch rollen soll, kann eine minimale Endgeschwindigkeit angenommen werden und aufgrund der zurückzulegenden Strecke die Startgeschwindigkeit bestimmt werden, dies wird in Abschnitt 4.4.2.1 beschrieben. Sofern diese Kugel über den Spielball ins Loch gespielt werden soll, dann muss der Spielball mit einer gewissen Geschwindigkeit die Kugel treffen, damit diese die gewünschte Startgeschwindigkeit erhält. Dies wird in Abschnitt 4.4.2.2 beschrieben.

4.4.2.1 Reibungsverlust über Bahn

Das Ziel eines Billiardstosses ist es, eine Kugel von einem Punkt A zu einem Punkt B rollen zu lassen. Dabei findet Rollreibung statt, welche das Abbremsen der Kugel verursacht. Es ist relevant zu wissen, welche Startgeschwindigkeit v_1 eine Kugel am Punkt A haben muss, um mit der Endgeschwindigkeit v_2 am Punkt B anzukommen. Beispielsweise wenn eine Kugel X in eines der Löcher gespielt werden soll, wobei die Kugel eine gewisse minimale Endgeschwindigkeit haben soll, damit sie tatsächlich ins Loch rollt.

Es kann die nachfolgende Formel angewendet werden²:

$$\vec{v}_1 = \sqrt{|\vec{v}_2|^2 + 2 \cdot g \cdot c_R \cdot \Delta s} \cdot \frac{\vec{v}_2}{|\vec{v}_2|} \quad (4.13)$$

Sei S die Startposition der Kugel, E die Endposition der Kugel und V der gewünschte Betrag der Endgeschwindigkeit, dann lässt sich die Endgeschwindigkeit \vec{v}_2 wie folgt berechnen:

$$\vec{\Delta s} = E - S \quad (4.14)$$

$$\hat{\Delta s} = \frac{\vec{\Delta s}}{|\vec{\Delta s}|} \quad (4.15)$$

$$\vec{v}_2 = V \cdot \hat{\Delta s} \quad (4.16)$$

4.4.2.2 Elastischer Stoss bei Kugelkollision

Um eine Kugel T in eine gewünschte Richtung, wie etwa zum Loch, zu rollen, muss sie von einer anderen Kugel A, evtl. vom Spielball, angestossen werden. Es gilt herauszufinden, wo die Kugel T getroffen werden muss und welche Geschwindigkeit die Kugel A zum Kollisionszeitpunkt haben muss, um die Kugel T in die gewünschte Richtung mit der geforderten Geschwindigkeit rollen zu lassen.

Die Situation mit der Kugel T, an der Position C und der Kugel A, an der Position A ist in Abbildung 4.9 dargestellt. Die Kugel A muss zum Punkt B rollen, wo sie auf Kugel T prallt und ein elastischer Stoss[Unk21e] stattfindet. Während die Kugel die Distanz $|\vec{d}|$ zurücklegt, verliert sie an Geschwindigkeit aufgrund von Reibung, diese wird in Abschnitt 4.4.2.1 behandelt. Hier ist lediglich relevant, welche Geschwindigkeit die Kugel A am Punkt B haben muss.

Sei \vec{u} , die gewünschte Richtung und Geschwindigkeit der Kugel T nach der Kollision, dann ist der Zielpunkt B der Kugel A wie folgt definiert³:

$$B = C - 2 \cdot r \cdot \hat{u} \quad (4.17)$$

Sei \hat{d} der Richtungsvektor der Länge 1 zwischen den Punkten A und B , E_v die Energieverlustkonstante in Prozent, dann gilt für die Geschwindigkeit \vec{v}_1 der Kugel A bei der Kollision⁴:

$$\hat{u} = \frac{\vec{v}_1}{|\vec{v}_1|} \quad (4.18)$$

²Die Herleitung findet sich im Anhang 8.1

³Die Herleitung findet sich im Anhang 8.2

⁴Die Herleitung findet sich im Anhang 8.2

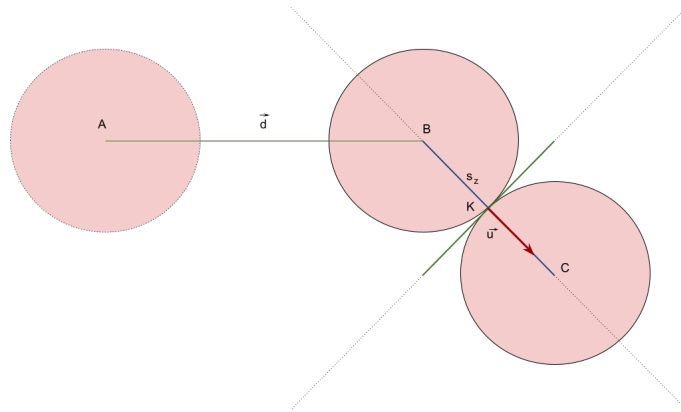


Abbildung 4.9: Kollisionspunkt zweier Kugeln

$$\vec{v}_1 = \frac{\vec{u} \cdot \vec{u}}{\hat{d} \cdot \vec{u}} \cdot \frac{1}{1 - E_v} \cdot \hat{d} \quad (4.19)$$

Anschliessend kann die Startgeschwindigkeit der Kugel A an der Position A berechnet werden, weil die gewünschte Endgeschwindigkeit und der zurückzulegende Weg bekannt ist⁵.

⁵siehe Kapitel 4.4.2.1

4.4.3 Modellierung physikalisches System

Ein Stoss kann für die Simulation mithilfe eines Modells beschrieben werden. Dieses Modell beschreibt den Ablauf eines Stosses durch Ereignisse, welche den beteiligten Objekten widerfahren. Die Kugeln, Banden und Löcher sind die Objekte dieses Modells. Jedes Objekt dieses Modells hat einen veränderlichen Energiewert, bspw. sind Kugeln in Bewegung oder sie sind ruhend. Eine Bande oder ein Loch bewegen sich nie, d.h. deren Energiewert ist konstant 0. Kollisionen zweier Kugeln oder einer Kugel und der Bande sind Beispiele von Ereignissen, welche Veränderungen der Energiezustände der Objekte bewirken.

Objekte, die einen Energiewert besitzen, welcher sich über die Zeit oder durch Interaktionen mit anderen Objekten verändert, werden nachfolgend variabel genannt. Sofern ein variables Objekt einen Energiewert grösser 0 hat, dann wird es weiter als dynamisch bezeichnet. Sobald es den Energiewert 0 erreicht, ist es als statisches variables Objekt kategorisiert. Banden und Löcher, welche einen fixen Energiewert besitzen, werden als konstant beschrieben.

Zwischen den Ereignissen kann eine Energieabnahme stattfinden. Beispielsweise unterliegt eine rollende Kugel der Rollreibung und verliert dadurch bis zum nächsten Ereignis Energie. Diese Energieabnahme erfolgt durch die Anwendung der Kantenfunktion auf der Kante zwischen zwei Ereignissen.

Diese Nomenklatur ist in Abbildung 4.10 ersichtlich.

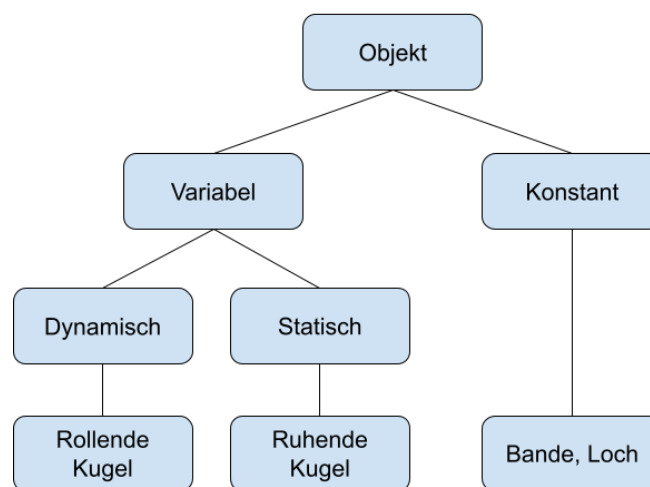


Abbildung 4.10: Typen von Objekten

4.4.3.1 Ereignisse und ihre Repräsentation

Das Ziel ist der Aufbau eines graphenähnlichen Konstrukts, welches aus Layern besteht und Zustandsübergänge variabler Objekte durch Knoten repräsentiert. Einige dieser Knoten treten bei Ereignissen wie einer Kollision oder wenn eine Kugel in ein Loch rollt auf. Andere Knoten dienen der reinen Abbildung des variablen Objekts innerhalb des Layers. Nachfolgend werden die Knoten definiert.



Energy-Input-Node: Dieser Node beschreibt das Auftreten eines Energie-Inputs von Aussen. Beispielsweise wenn die weisse Kugel mit dem Queue gestossen wird.



Energy-Transfer-Node (Collision-Node): Dieser Node beschreibt die Übergabe von Energie auf die beteiligten Objekte. Ein Beispiel ist die Kollision zweier Kugeln, wobei diese Energie austauschen.

Der Energy-Transfer-Node wird in drei Schichten aufgeteilt. Bei der Input-Schicht wird die Kantenfunktion (siehe S. 20) angewendet. Diese berücksichtigt den Energieverlust bis zum Auftreten des Ereignisses. Die mittlere Schicht beschreibt die Übergabefunktion der beteiligten Objekte. Die dritte Schicht repräsentiert das Resultat, also den Status der Objekte nach der Energieübergabe.

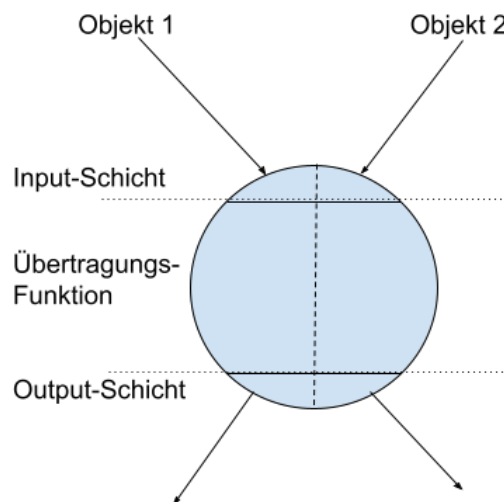


Abbildung 4.11: Der Energy-Transfer-Node



No-Energy-Node: Dieser Node wird eingesetzt, sobald der Energiewert eines variablen Objekts auf 0 sinkt und somit auch den Übergang von dynamisch zu statisch repräsentiert.



Out-Of-System-Node: Dieser Node wird eingesetzt, sobald ein variables Objekt das System verlässt, indem beispielsweise eine Kugel in ein Loch rollt.



Cutting-Node: Dieser Node ist ein spezieller Energy-Transfer-Node. Er wird bei variablen Objekten eingesetzt, die nicht an einem Ereignis beteiligt sind, wenn ein solches auftritt. Zum Ereigniszeitpunkt wird ein neuer Layer (siehe S. 20) geschaffen, welcher die Objekte, die am Ereignis beteiligt sind, in einem Energy-Transfer-Node festhält. Weiterhin wird der Status aller dynamischen Objekte ebenfalls zu diesem Zeitpunkt festgehalten. Der Cutting-Node beschränkt sich auf ein Input- sowie Output-Objekt und beinhaltet als Energieübertragungsfunktion die Identitätsfunktion.

4.4.3.2 Kantenfunktion

Die Kantenfunktion beschreibt eine Energieabnahme über die Zeit oder den Weg.

4.4.3.3 Layer

Ein Layer beinhaltet sämtliche Veränderungen der variablen Objekte und beschreibt den Status aller Objekte zu diesem Zeitpunkt. Sobald ein Ereignis auftritt wird ein neuer Layer im Modell eingefügt.

4.4.3.4 Beispiel eines Graphen

Das beschriebene Datenmodell kann als Graph⁶ visualisiert werden. Als Beispiel wird die Idee eines Billardstosses in Abbildung 4.12 hinzugezogen. Auf dem Tisch liegt eine weisse wie auch zwei rote Kugeln. Die zweite rote Kugel wird an keiner Interaktion beteiligt sein, weswegen sie ihren Zustand nie verändert. Es ist ersichtlich, dass bei jedem Ereignis für diese Kugel ein „Out-of-energy-Node“ eingefügt wird. Das erste Event beschreibt die Kollision des dynamischen Objekts „weisse Kugel“ sowie des statischen Objekts „rote Kugel“. Danach verliert die weisse Kugel sämtliche Energie und wechselt in den Status „Out-of-energy“. Da die rote Kugel zu diesem Zeitpunkt dynamisch ist, wird für sie ein „Cutting-Node“ eingefügt. Zuletzt rollt die rote Kugel in das Loch, für sie wird ein „Out-of-system-Node“ erstellt und das System hat sämtliche Energie verloren. Ein Endzustand wurde erreicht, da alle Kugeln nun statisch sind.

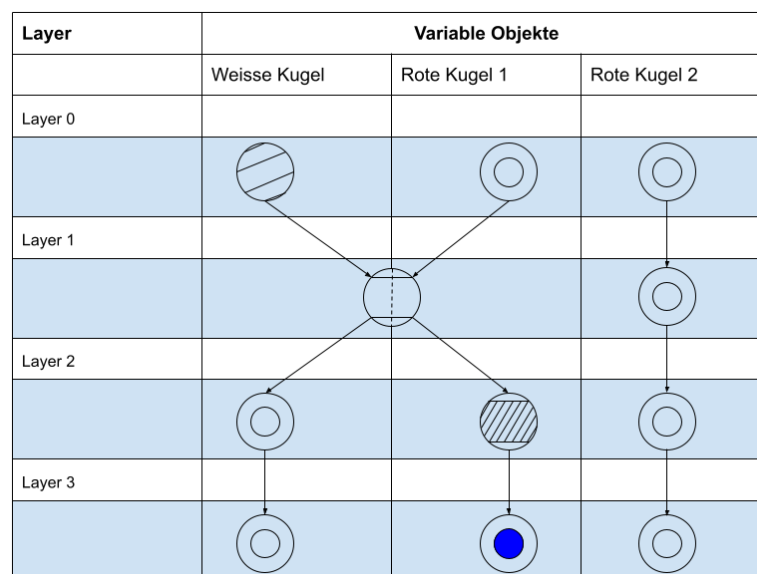


Abbildung 4.12: Beispiel für ein Resultat des Algorithmus 2

In Algorithmus 2 wird die Grundidee erläutert. Als Input für die Funktion „simulate“ dient der erste Layer, welcher mehrere variable Objekte beinhalten kann, wobei mindestens ein Objekt dynamisch (energiereich) sein sollte. Dieser Layer wird direkt dem erzeugten System hinzugefügt und dieses wird solange bearbeitet, bis alle variablen Objekte statisch sind (das System hat keine Energie mehr). In jedem Schleifendurchlauf wird das nächste Event berechnet. Die Events können diverser Natur sein. Es werden Kollisionen mit dynamischen, statischen wie auch konstanten Objekten geprüft. Bei der Kollision mit konstanten Objekten können die Ereignisse „Energy transfer“ oder „Out of System“ auftreten. Weiterhin wird geprüft, ob ein dynamisches Objekt seine Energie durch die Kantenfunktion verliert. Aufgrund eines Ereignisses wird zunächst ein neuer Layer generiert, dann für die am Ereignis beteiligten Objekte den entsprechenden Node und zuletzt für die anderen variablen Objekte entweder

⁶Aus effizienzgründen wird auf einen Graphen in der Implementation verzichtet. Das Modell setzt sich aus verschiedenen Layern zusammen, bei welchen die Informationen der Events für jedes variable Objekt separat und mit Zugriffszeit $O(1)$ abgelegt werden.

ein „Cutting-Node“ oder ein „Out-of-energy-Node“ eingefügt.

```

struct {
    timestamp: datetime
    me: string
    partner: string
    partnerType: [BALL, TARGET, RAIL]
    event: [OutOfEnergy, Collision]
} Event

Function simulate(start: Layer, constantObjects: list)  $\longrightarrow$  System
    system  $\leftarrow$  System()
    system  $\leftarrow$  appendLayer(system, start)
    while ! system.isStatic() do
        nextEvent  $\leftarrow$  nextEvent(system.lastLayer(), constantObjects)
        layer  $\leftarrow$  atMoment(system.lastLayer(), nextEvent)
        system  $\leftarrow$  appendLayer(system, start)
    end
    return system

```

```

Function nextEvent(layer: Layer, constantObjects: list)  $\longrightarrow$  Event
    nextEvent: Event  $\leftarrow$  none
    for object in layer.dynamicObjects() do
        nextEvent  $\leftarrow$  min(nextEvent, outOfEnergy(object))
        nextEvent  $\leftarrow$  min(nextEvent, collision(object, layer.dynamicObjects()))
        nextEvent  $\leftarrow$  min(nextEvent, collision(object, layer.staticObjects()))
        nextEvent  $\leftarrow$  min(nextEvent, collision(object, constantObjects))
    end
    return nextEvent

```

```

Function atMoment(layer: Layer, event: Event)  $\longrightarrow$  Event
    nextLayer  $\leftarrow$  layer() for node in layer.objects() do
        if node.first == event.me or node.first == event.partner then
            | nextLayer  $\leftarrow$  append(nodeForEvent(node, event))
        end
        else
            if node.second.hasEnergy() then
                | nextLayer  $\leftarrow$  append(cuttingNode(node, event.timestamp))
            end
            else
                | nextLayer  $\leftarrow$  append(node, nextLayer)
            end
        end
    end
    return nextLayer

```

Algorithm 2: Algorithmus zum Aufbau eines physikalischen Systems

4.4.4 Simulation

Sobald ein möglicher Lösungskandidat anhand der in Abschnitt 4.4.1 beschriebenen Suche gefunden und dessen Initialgeschwindigkeit nach Abschnitt 4.4.2 berechnet wurde, wird eine Simulation durchgeführt, um die Lösung definitiv zu bestätigen. Durch das Anwenden verschiedener Anfangsgeschwindigkeiten der weissen Kugel können in diesem Schritt mehrere Situationen evaluiert werden.

Die Simulation wird durch die Definition eines physikalischen Systems wie in Kapitel 4.4.3 durchgeführt. Hierbei gelten die Zuordnungen wie sie nachfolgend beschrieben werden.

Ereignisse

Energy-Input-Node Wird modelliert über die Eingabe der Energie der weissen Kugel. Ein spezifischer Node zur Modellierung wird nicht implementiert, es wird der Energy-Transfer-Node verwendet, wobei nur der Output-Wert relevant ist.

Energy-Transfer-Node Tritt bei der Kollision zwischen zweier Kugeln oder einer Kugel mit der Bande auf.

No-Energy-Node Tritt auf, wenn eine Kugel vom dynamischen in den statischen Zustand wechselt (ausrollt). In jedem Layer, wo eine Kugel statisch ist, wird sie durch diesen Node modelliert.

Out-of-System-Node Sobald eine Kugel mit dem Zielkreis kollidiert, tritt dieses Ereignis auf. Dem System wird die Energie entzogen und die Kugel ist nicht mehr verfügbar.

Kantenfunktion Die Kantenfunktion zwischen den Übergängen innerhalb des Layers bildet der Reibungsverlust der Kugel über eine bestimmte Zeit oder einen bestimmten Ort.

Dynamische/Statische Objekte Im Billiard gibt es nur die Kugeln als statische und/oder dynamische Objekte.

Konstante Objekte Die konstanten Objekte bilden die Banden wie auch die Ziele.

Es wird ungefähr der Pseudoalgorithmus wie in 2 angewendet, optimal auf das Problem „Billiard“ abgestimmt. Es folgen die physikalischen Berechnungen zur Durchführung der Simulation.

4.4.4.1 Reibungsverlust über die Zeit

Die Geschwindigkeit einer Kugel wird durch die Reibung über die Zeit reduziert. Dazu wird die Formel der gleichförmig beschleunigten Bewegung verwendet, wobei \vec{v}_0 , \vec{a} sowie t gegeben sind.

$$\vec{v} = \vec{a} \cdot t + \vec{v}_0 \quad (4.20)$$

4.4.4.2 Elastischer Stoss zweier Kugeln

Die Kollision zweier Kugeln wird im folgenden als zweidimensionaler elastischer Stoss angesehen. Dabei ist es wichtig, zwei Komponenten beim Zusammenprall zu betrachten. Dies ist einerseits das Liniensegment s_z zwischen den Mittelpunkten der Kugeln sowie die orthogonal dazu stehende Gerade s_t . Die Kugel, welche Energie mitbringt, übergibt der anderen Kugel Energie in Richtung von s_z . Die übrig gebliebene Energie zeigt in Richtung von s_t [Unk21e]. In Abbildung 4.13 ist ein solcher Stoss dargestellt. Kugel eins bringt dabei die Geschwindigkeit \vec{v}_1 und Kugel zwei die Geschwindigkeit \vec{v}_2 mit. Ein Anteil der Geschwindigkeit von Kugel eins wird der Kugel zwei in Richtung von s_z übergeben. Dasselbe gilt für die Kugel zwei. Sie übergibt einen Teil ihrer Geschwindigkeit an Kugel eins ebenfalls in Richtung von s_z . Die verbleibende Energie der Kugeln zeigt jeweils in Richtung von s_t .

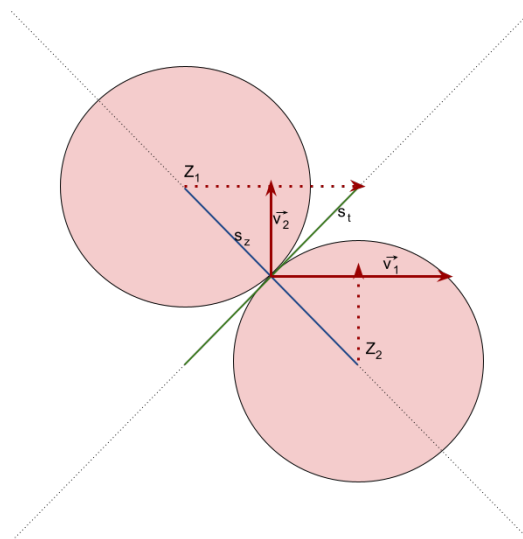


Abbildung 4.13: Elastischer Stoss zweier Kugeln

Die initialen Geschwindigkeiten \vec{v}_1^i und \vec{v}_2^i werden aufgrund eines natürlichen Energieverlusts um eine Konstante E_v , welche in Prozent angegeben wird, reduziert.

$$\vec{v}_1 = \vec{v}_1^i \cdot (1 - E_v) \quad (4.21)$$

$$\vec{v}_2 = \vec{v}_2^i \cdot (1 - E_v) \quad (4.22)$$

(4.23)

Um nun die neuen Geschwindigkeiten \vec{u}_1 und \vec{u}_2 zu berechnen, müssen die initialen Geschwindigkeiten \vec{v}_1 sowie \vec{v}_2 auf die Komponenten in Richtung von s_z und s_t aufgeteilt werden.

$$\vec{z} = \vec{Z}_2 - \vec{Z}_1 \quad (4.24)$$

$$\hat{z} = \frac{\vec{z}}{|\vec{z}|} \quad (4.25)$$

Von Interesse ist dabei nur \hat{z} . Mittels diesem Vektor kann die Komponente in Richtung von s_z beider Geschwindigkeiten ausgerechnet werden.

$$v_{1,z} = (\vec{v}_1 \cdot \hat{z}) \cdot \hat{z} \quad (4.26)$$

$$v_{2,z} = (\vec{v}_2 \cdot \hat{z}) \cdot \hat{z} \quad (4.27)$$

Mittels dieses neuen Vektors in s_z Richtung kann der Vektor in s_t Richtung berechnet werden.

$$\vec{v} = \vec{v}_t + \vec{v}_z \quad (4.28)$$

$$\vec{v}_t = \vec{v} - \vec{v}_z \quad (4.29)$$

Daraus folgt für die jeweiligen Kugeln:

$$\vec{v}_{1,t} = \vec{v}_1 - \vec{v}_{1,z} \quad (4.30)$$

$$\vec{v}_{2,t} = \vec{v}_2 - \vec{v}_{2,z} \quad (4.31)$$

Die neuen Geschwindigkeitsvektoren setzen sich wie beschrieben aus den jeweiligen Komponenten zusammen. Das Resultat lautet wie folgt:

$$\vec{u}_1 = \vec{v}_{2,z} + \vec{v}_{1,t} \quad (4.32)$$

$$\vec{u}_2 = \vec{v}_{1,z} + \vec{v}_{2,t} \quad (4.33)$$

Somit können bei gegebener Kollision zweier Kugeln die Geschwindigkeiten nach der Kollision bestimmt werden.

4.4.4.3 Bandenreflektion

Sofern eine Kugel an eine Bande stösst, so wird diese abgelenkt. In dem hier beschriebenen Modell wird der Drall[Unk21d], welcher die Bahn einer Kugel nach Kollision mit der Bande ablenken würde, ignoriert. Das bedeutet, es wird davon ausgegangen, dass der Ausfallswinkel nach der Bandenreflektion gleich dem Eifallswinkel sei. Dazu kann die folgende Formel[Bou00] verwendet werden, wobei I der einfallende und R der ausgehende Weg der Kugel und N der Normalenvektor der Bande sind:

$$R = I - 2 \cdot N \cdot (I \cdot N) \quad (4.34)$$

Da bei einer Bandenkollision Energie verloren geht, wird dieses Verhalten mit der Konstanten E_v modelliert. Diese gibt den Verlust in Prozent an, bezogen auf den minimal berechneten Wert.

$$R = (I - 2 \cdot N \cdot (I \cdot N)) \cdot \frac{1}{1 + E_v} \quad (4.35)$$

4.4.4.4 Kollisionsprüfung

Während der Simulation ist es notwendig zu prüfen, welche Kugeln mit welchen kollidieren könnten. Um zu wissen, ob eine Kugel eine Strecke zurücklegen kann, ohne mit einer anderen Kugel zu kollidieren, muss für jede andere Kugel geprüft werden, ob diese im Weg liegt. Daher sollte dieser Test effizient sein. Für den Test wird die zurückzulegende Strecke als Liniensegment zwischen Punkt A und Punkt B und die Position einer zu testenden Kugel C als Punkt verstanden. Anschliessend wird der Abstand zwischen dem Punkt C und dem Liniensegment AB geprüft, ob dieser kleiner als der Kugeldurchmesser ist. Sofern dies der Fall ist, liegt die Kugel an der Position C im Weg und es würde eine Kollision stattfinden, sofern die Ausgangskugel die Strecke AB rollt. Dies wird für jede Kugel geprüft.

4.4.4.5 Ereignis - Energie-Transfer über Kugelkollision

Dieses Ereignis beschreibt die Kollision mit einer anderen Kugel zum Zeitpunkt t . Bekannt sind dabei die Beschleunigung \vec{a}^7 , die Geschwindigkeit \vec{v} und der Ort \vec{s} beider Kugeln. Diese werden entsprechend indexiert.

Weiterhin sind folgende Abstraktionen bekannt:

$$\vec{\Delta a} = \vec{\Delta a}_1 - \vec{\Delta a}_2 \quad (4.36)$$

$$\vec{\Delta v} = \vec{\Delta v}_1 - \vec{\Delta v}_2 \quad (4.37)$$

$$\vec{\Delta s} = \vec{\Delta s}_1 - \vec{\Delta s}_2 \quad (4.38)$$

Es entsteht ein Polynom vierten Grades⁸. Um diese Gleichung zu lösen bedarf es der entsprechenden Lösungsformel [Unk21f]:

$$ax^4 + bx^3 + cx^2 + dx + e = 0 \quad (4.39)$$

$$x_{1,2} = -\frac{b}{4a} - S \pm \frac{1}{2} \sqrt{-4S^2 - 2p + \frac{q}{S}} \quad (4.40)$$

$$x_{3,4} = -\frac{b}{4a} + S \pm \frac{1}{2} \sqrt{-4S^2 - 2p - \frac{q}{S}} \quad (4.41)$$

$$p = \frac{8ac - 3b^2}{8a^2} \quad (4.42)$$

$$q = \frac{b^3 - 4abc + 8a^2d}{8a^3} \quad (4.43)$$

$$S = \frac{1}{2} \sqrt{-\frac{2}{3}p + \frac{1}{3a} \left(Q + \frac{\Delta_0}{Q} \right)} \quad (4.44)$$

$$Q = \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} \quad (4.45)$$

$$\Delta_0 = c^2 - 3bd + 12ae \quad (4.46)$$

$$\Delta_1 = 2c^3 - 9bcd + 27b^2e + 27ad^2 - 72ace \quad (4.47)$$

Wobei die Koeffizienten folgendermassen lauten:

$$a = \frac{1}{4} \cdot (\vec{\Delta a} \cdot \vec{\Delta a}) \quad (4.48)$$

$$b = \vec{\Delta a} \cdot \vec{\Delta v} \quad (4.49)$$

$$c = \vec{\Delta a} \cdot \vec{\Delta s} + \vec{\Delta v} \cdot \vec{\Delta v} \quad (4.50)$$

$$d = 2 \cdot (\vec{\Delta v} \cdot \vec{\Delta s}) \quad (4.51)$$

$$e = \vec{\Delta s} \cdot \vec{\Delta s} - D^2 \quad (4.52)$$

Durch Lösen des Polynoms werden alle Zeitpunkte $x_{1-4} = t_{1-4}$ bestimmt, wobei der relevante Zeitpunkt t der Erste ist.

$$t = \min(t_1, t_2, t_3, t_4) \quad (4.53)$$

⁷Die Beschleunigung \vec{a} ist durch die Herleitung in Kapitel 8.7 gegeben.

⁸Für Herleitung, siehe Anhang 8.4

Das Lösen dieses Gleichungssystems erfordert viel Rechenzeit, was die Frage nach Performanceverbesserungen aufwirft. Diese werden durch Vorbedingungen erzielt, welche geprüft werden. Es findet zu Beginn eine Fallunterscheidung statt. Wenn die zweite Kugel statisch ist, so wird geprüft, ob die Distanz d zwischen der Position P_2 der zweiten Kugel K_2 und der halben Geraden g definiert durch die Position P_1 der ersten Kugel K_1 wie auch deren Geschwindigkeitsvektor \vec{v} kleiner oder gleich dem Kugeldurchmesser ist. Nur in diesem Fall kann es eine Kollision geben. Sämtliche Angaben beziehen sich auf die Grafik 4.14.

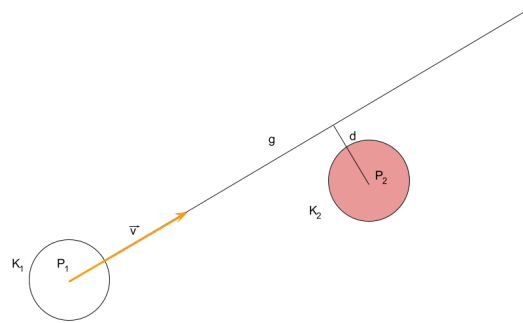


Abbildung 4.14: Vorbedingung einer Prüfung auf Kollision zwischen Kugeln bei statischer Beteiligung

Sobald beide Kugeln dynamisch sind, wird es ein wenig komplizierter. Da beide in Bewegung sind, reicht eine einfache Prüfung auf die Distanz nicht mehr. Es muss also ein Schnittpunkttest der beiden halben Geraden, definiert durch die beiden Geschwindigkeitsvektoren, durchgeführt werden. Hierbei ist der Durchmesser der Kugel mitzubeachten, was drei Spezialfälle eröffnet. Abbildung 4.15 zeigt das Vorgehen zur Behandlung des ersten Falls. Würde nur ein Schnittpunkttest der beiden halben Geraden durchgeführt, so ergäbe diese Situation keine Lösung.

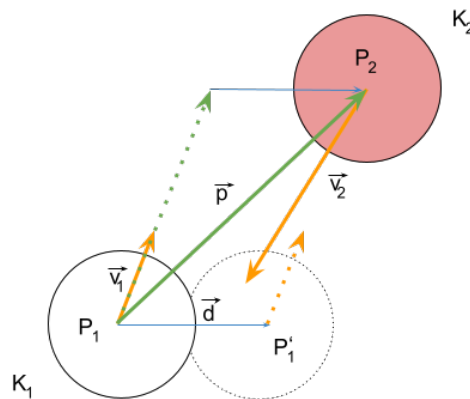


Abbildung 4.15: Vorbedingung einer Prüfung auf Kollision zwischen dynamischen Kugeln

Daher wird der Vektor \vec{p} zwischen den Positionen P_1 und P_2 gebildet (grün).

$$\vec{p} = P_2 - P_1 \quad (4.54)$$

Der erhaltene Vektor wird auf den Geschwindigkeitsvektor \vec{v}_1 projiziert (grün gestrichelt).

$$\vec{p}' = (\vec{p} \cdot \hat{v}_1) \cdot \vec{v}_1 \quad (4.55)$$

Danach wird der Zwischenvektor \vec{d} gebildet (blau).

$$\vec{d} = \vec{p} - \vec{p}' \quad (4.56)$$

Die Kugel K_1 wird um diesen Vektor verschoben, es resultiert P'_1 .

$$P'_1 = P_1 + \vec{d} \quad (4.57)$$

Von dieser Position aus kann nun ein Schnittpunkttest zwischen den beiden halben Geraden, durch die Geschwindigkeitsvektoren definiert, durchgeführt werden.

Ein weiterer Spezialfall tritt auf, wenn sich die Kugeln parallel fortbewegen. Da dies ein sehr seltener Fall ist und wohl kaum auftreten wird, wird die Berechnung zugelassen, wenn der Vektor \vec{d} kleiner oder gleich gross wie der Durchmesser ist. Die Situation wird in Abbildung 4.16 erläutert, die Berechnungen für den Vektor \vec{d} sind äquivalent zum Spezialfall 1.

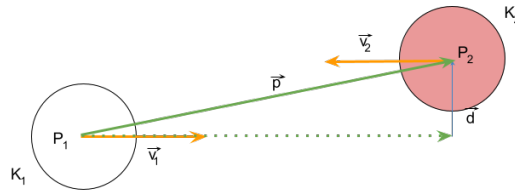


Abbildung 4.16: Vorbedingung einer Prüfung auf Kollision zwischen dynamischen parallel verlaufenden Kugeln

Der letzte Spezialfall befasst sich mit der Tatsache, dass ein Schnittpunkttest, so wie er unter Berücksichtigung der obigen Fälle durchgeführt wird, kein Ergebnis findet, wenn der Schnittpunkt hinter einer der Geraden liegt. Diese Situation kann in Abbildung 4.17 entnommen werden. Es ist ersichtlich, dass der Geschwindigkeitsvektor \vec{v}_1 einen grösseren Betrag aufweist als der Geschwindigkeitsvektor \vec{v}_2 . Um diesem Umstand Rechnung zu tragen, wird die Startposition der Geraden um den Durchmesser nach hinten verschoben.

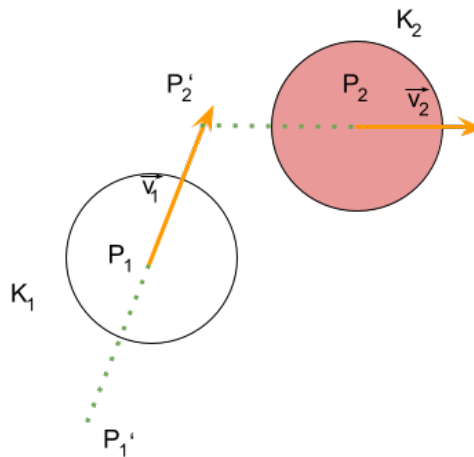


Abbildung 4.17: Vorbedingung einer Prüfung auf Kollision zwischen dynamischen hintereinander verlaufenden Kugeln

4.4.4.6 Ereignis - Energie-Transfer über Bandenkollision

Dieses Event beschreibt die Kollision mit der Bande. Es soll der Zeitpunkt t der Kollision mit der Bande festgestellt werden. Der Algorithmus funktioniert so, dass zuerst geprüft wird, ob eine Kollision stattfinden kann. Dies erfolgt über einen Schnittpunkt-Test zwischen einer Linie und einem Liniensegment. Eine Bande kann als Liniensegment zwischen dem Startpunkt R_1 und R_2 betrachtet werden. Diese Punkte müssen demnach bekannt sein. Weiterhin ist der Geschwindigkeitsvektor \vec{v} und die Position der Kugel C bekannt. Aufgrund dieser Informationen kann eine Linie definiert werden.

Die Punkte R_1 und R_2 werden um den Kugelradius zur Tischmitte verschoben, damit dem Kugelradius Rechnung getragen wird und dieser nicht weiter betrachtet werden muss. Daraus ergeben sich R'_1 und R'_2 .⁹

Erster Schritt - Prüfe, ob Kollision möglich:

Dazu wird festgestellt, ob ein Schnittpunkt zwischen der Lauflinie der Kugel und dem Liniensegment der Bande existiert¹⁰. Sofern kein Schnittpunkt vorliegt, können weitere Tests abgebrochen werden, da keine Kollision mit der Bande stattfinden wird.

⁹Für Herleitung, siehe Anhang 8.5

¹⁰Siehe Anhang 8.5

Zweiter Schritt - Ort der Kollision bestimmen:

Betrachte dazu eine der Gleichungen mit zugehörigem λ :

$$s(\vec{\lambda}_1) = \vec{R}'_1 + \lambda_1 \cdot \Delta \vec{R}' \quad (4.58)$$

$$s(\vec{\lambda}_2) = \vec{C} + \lambda_2 \cdot \vec{v} \quad (4.59)$$

Dritter Schritt - Zeitpunkt der Kollision bestimmen:

Dies kann über die folgenden Gleichungen gelöst werden:

$$\Delta t_1 = \frac{-v + \sqrt{v^2 + 2 \cdot a \cdot \Delta s}}{a} \quad (4.60)$$

$$\Delta t_2 = \frac{-v - \sqrt{v^2 + 2 \cdot a \cdot \Delta s}}{a} \quad (4.61)$$

Bevor die Lösung zu t_1 und t_2 berechnet wird, muss jedoch der Radikand auf folgende Eigenschaft geprüft werden:

$$0 \leq v^2 + 2 \cdot a \cdot \Delta s \quad (4.62)$$

Nur in diesem Fall gibt es eine Lösung und somit an dieser Position eine Kollision. Ansonsten hat die Kugel vorher schon ihre Gesamtenergie verloren und steht still. Es wird das Minimum der beiden Zeiten Δt_1 und Δt_2 verwendet.

4.4.4.7 Ereignis - Totaler Energieverlust

Dieses Ereignis trifft ein, sobald eine Kugel stillsteht. Der Zeitpunkt wird über die Formel 4.63 beschrieben¹¹.

$$t = \max\left(\frac{-v_{0,x}}{a_x}, \frac{-v_{0,y}}{a_y}\right) \quad (4.63)$$

4.4.4.8 Ereignis - Verlassen des Systems

Der Ereigniszeitpunkt erfolgt über zwei Schritte. Im ersten Schritt wird der Kollisionspunkt der Kugel mit dem Zielkreis bestimmt. Anhand dieser Information kann die zurückgelegte Distanz der Kugel bestimmt werden. Es gelten die folgenden Definitionen:

\vec{C} = Zielmittelpunkt

\vec{p} = Position der Kugel

\hat{v} = Normierte Geschwindigkeit der Kugel

Die nachfolgende Formel definiert, mit welchem Faktor der Geschwindigkeitsvektor multipliziert werden muss, damit er die Länge der zurückzulegenden Distanz zum Kollisionsort mit dem Ziel erreicht¹². Zu beachten ist, dass l nur gültig ist, wenn es grösser als 0 ist.

$$l_1 = \frac{-(2 \cdot \hat{v} \cdot (\vec{p} - \vec{C})) + \sqrt{4 \cdot (\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) \cdot r^2}}{2} \quad (4.64)$$

$$l_2 = \frac{-(2 \cdot \hat{v} \cdot (\vec{p} - \vec{C})) - \sqrt{4 \cdot (\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) \cdot r^2}}{2} \quad (4.65)$$

$$l = \max(\min(l_1, l_2), 0) \quad (4.66)$$

$$d(\vec{l}) = l \cdot \hat{v} \quad (4.67)$$

¹¹Die Herleitung erfolgt in Kapitel 8.3. Die Beschleunigung \vec{a} ist durch die Herleitung in Kapitel 8.7 gegeben.

¹²Für Herleitung, siehe Anhang 8.6.

Nachdem die Distanz bestimmt wurde, kann der Kollisionszeitpunkt berechnet werden¹³. Zu beachten ist, dass keine Lösung existiert, wenn die Diskriminante kleiner denn 0 ist. Die Kugel rollt zu langsam und der Reibungsverlust ist zu gross, als dass sie den Zielkreis erreichen könnte.

$$t_1 = \frac{-|\vec{v}| + \sqrt{|\vec{v}|^2 + 2 \cdot |\vec{a}| |d(\vec{l})|}}{|\vec{a}|} \quad (4.68)$$

$$t_2 = \frac{-|\vec{v}| - \sqrt{|\vec{v}|^2 + 2 \cdot |\vec{a}| |d(\vec{l})|}}{|\vec{a}|} \quad (4.69)$$

$$t = \min(t_1, t_2) \quad (4.70)$$

4.4.4.9 Bewertungsfunktion

Wie auch die Suche nach einem guten Lösungskandidaten muss das Endresultat der Simulation bewertet werden. In dem Fall geht es um die Bestimmung der Belohnung eines Stosses oder umgekehrt die Bestrafung in Form von Kosten für eine geringe Ausbeute. In einem ersten Schritt wird die Belohnung berechnet. Diese richtet sich an der Anzahl und dem Typen der eingelochten Kugeln. Es gibt die Schwarze (7), Pinke (6), Blaue (5), Braune (4), Grüne (3), Gelbe (2) und Rote (1) Kugel. Die Punkte sind jeweils hinter dem Typen aufgeführt. In einem Stoss wird eine maximal mögliche Punktzahl von 7 definiert. Sollten mehr Punkte erzielt werden, wird dies im Weiteren keinen Einfluss mehr haben, dazu später mehr. Dieser Score S wird durch eine Normierung der erzielten Punkte p anhand der maximal möglichen Punkte p_{max} verkleinert. Zumeist wird er zwischen 0 und 1 liegen, er kann aber auch grösser werden.

$$S = \frac{p}{p_{max}} \quad (4.71)$$

Anschliessend wird der Score auf eine Bestrafung/Kosten umgerechnet. Hierbei wird ignoriert, dass mehr Punkte wie die maximal möglichen erzielt wurden. Dies geschieht, in dem das Minimum von 1 und dem Score S genommen wird.

$$K = 1 - \min(S, 1) \quad (4.72)$$

Dieser so erhaltene Wert ist noch nicht aussagekräftig genug. Er soll in Relation zu der Schwierigkeit des Stosses, also den Suchkosten, angegeben werden können. Die Belohnung soll zum Beispiel nur 50% Gewicht haben. Um dies zu erreichen, werden die definitiven Kosten über eine solche Normierung erreicht, wobei K die nicht normierten Kosten zwischen 0 und 1, g die Gewichtung in Prozent gegenüber der Schwierigkeit und K_s die Suchkosten sind.

$$K_{norm} = K \cdot g \cdot K_s \quad (4.73)$$

¹³Für Herleitung, siehe Anhang 8.6. Die Beschleunigung \vec{a} ist durch die Herleitung in Kapitel 8.7 gegeben.

4.4.5 Animation

Um eine möglichst optimale Benutzerfreundlichkeit zu gewährleisten, soll der berechnete Stoss über eine Animation auf dem Tisch dargestellt werden. Dies wird durch eine Übersetzung des physikalischen Simulationsmodells, wie in Kapitel 4.4.3 beschrieben, erzielt.

Die Animation besteht aus Keyframes, zwischen welchen interpoliert werden kann. Das grundlegende Prinzip wird in Abbildung 4.18 visualisiert. Auf der linken Seite wird der Zeitstrahl angegeben. Die Zeit schreitet in vertikaler Richtung nach unten voran. Der rote Pfeil markiert den aktuellen Zeitpunkt, dieser liegt zwischen Keyframe 1 und Keyframe 2. Der Status zu diesem Zeitpunkt kann durch Interpolation zwischen den genannten Keyframes berechnet werden.

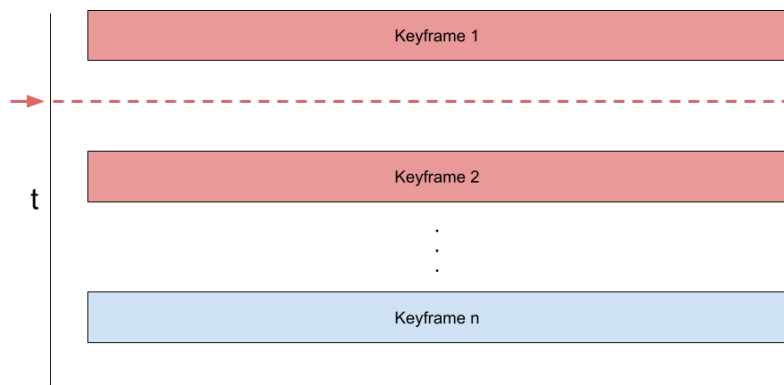


Abbildung 4.18: Keyframe Animation

Das Animationsmodell wird anhand des Simulationsmodells abgeleitet. Dazu kann Abbildung 4.19 betrachtet werden. Wie in Kapitel 4.4.3 erklärt, besteht das physikalische Simulationsmodell aus in Layern gruppierte Knoten. Diese Knoten haben jeweils eine Eingabe- wie auch eine Ausgabeschicht. Eine Gruppierung dieser Knoten in einem Layer weist dadurch dieselbe Eigenschaft auf.

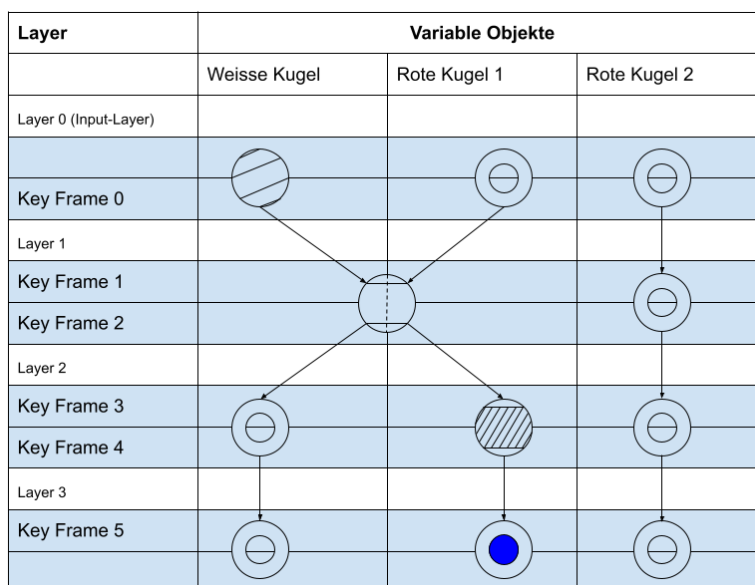


Abbildung 4.19: Simulationsmodell mit Keyframes

Jede Eingabe- wie auch Ausgabeschicht bildet so im Animationsmodell ein Keyframe. Die Besonderheit liegt darin, dass nicht wie bei einer nach Lehrbuch aufgebauten Keyframe-Animation mit Fortlaufen der Zeit die Keyframes durchiteriert¹⁴ werden. Die Keyframes, welche aufgrund des Simulationsmodells generiert werden, werden paarweise als Fenster betrachtet. Dieses Paarfenster wird bei Erreichen des Endkeyframes auf das nächste Paar verschoben. Die Funktionsweise ist in Abbildung 4.20 beschrieben. Dort werden die verschiedenen Keyframefensterpaare unterschiedlich rot markiert. Die Zeit schreitet wiederum

¹⁴Durchiterieren bedeutet, dass wenn ein Keyframe vorher das Ende markierte, wie es das Keyframe 2 in Abbildung 4.18 tut, dann wird das Keyframe 2 mit Fortlaufen der Zeit zum Startkeyframe und das nächstfolgende zum neuen Ende.

in vertikaler Richtung nach unten voran, der rote Pfeil auf der linken Seite zeigt eine mögliche aktuelle Position innerhalb einer Animation. Demnach wäre das zweite Keyframefenster mit den Keyframes 2 und 3 aktiv.

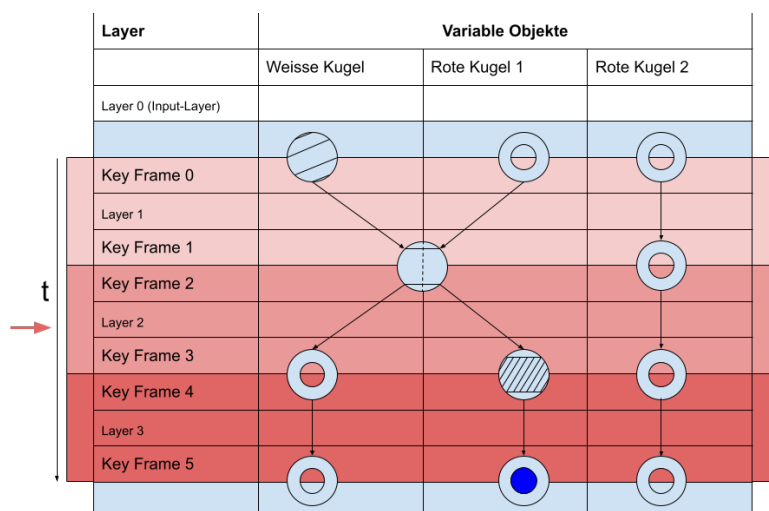


Abbildung 4.20: Simulationsmodell mit paarweisen Keyframefenstern

Die Begründung der etwas spezielleren Handhabung liegt an der Tatsache, dass bei einem physikalischen Ereignis wie einer Kugelkollision eine Veränderung des Status geschieht. So werden im Wesentlichen die Magnituden wie auch die Richtungen der Geschwindigkeitsvektoren verändert. Würde nur ein Keyframe bei einem solchen Ereignis generiert werden, dann könnte nicht mehr zwischen den Geschwindigkeitsvektoren aufgrund der geänderten Richtung interpoliert werden.

Die so generierte Animation kann als Gruppierung von einzelnen Keyframeanimationen, wie in Abbildung 4.18 angegeben, mit jeweils nur zwei Keyframes angesehen werden.

4.5 Berechnungsprozess

Die Berechnung eines optimalen Stosses wird aufgrund der vielen Möglichkeiten sehr zeitintensiv, weswegen die expandierten Teilschritte parallel gerechnet werden.

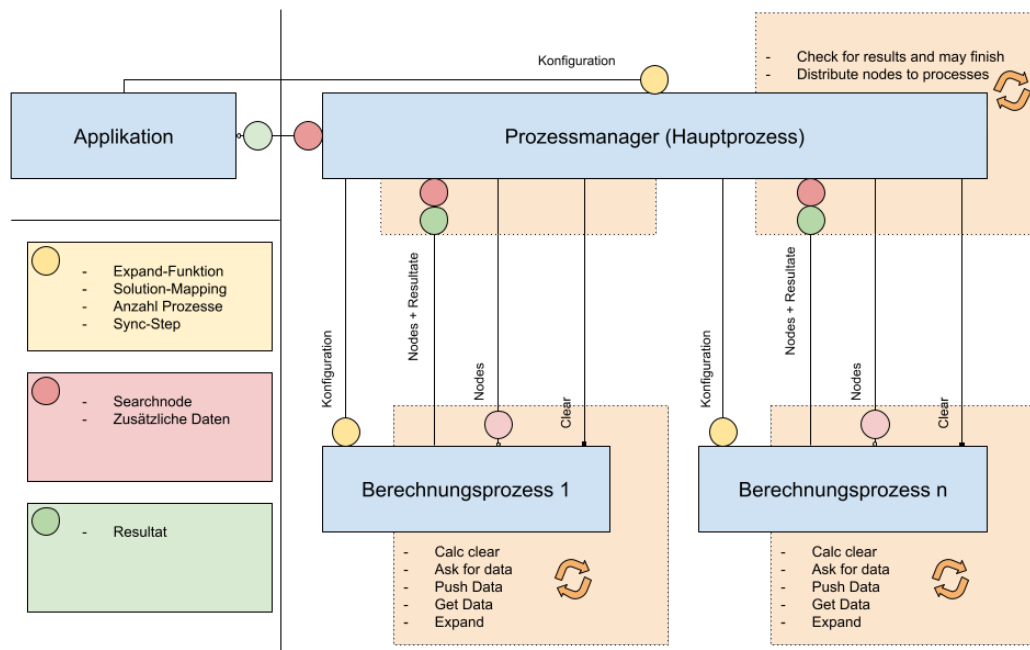


Abbildung 4.21: Berechnungsprozess

Abbildung 4.21 erläutert die Durchführung dieser Expansionsschritte. Die Applikation auf der linken Seite liefert die konkreten Aufgaben, welche von einem Pool an Berechnungsprozessen bearbeitet werden sollen. Sie beinhaltet die Logik, um eine solche Aufgabe zu lösen, da die Berechnungsprozesse selbst nicht über dieses Wissen verfügen. Der Pool von Berechnungsprozessen wird vom Prozessmanager verwaltet und dieser ist die zentrale Ansprechperson der Applikation.

Datenaustausche sind über gefärbte Kreise repräsentiert. Handelt es sich um einen asynchronen Datenaustausch, dann ist der entsprechende Kreis heller eingefärbt und die Verbindungslinie ist durch einen nicht ausgefüllten Punkt gekennzeichnet. Es werden die Farben Rot den zu bearbeitenden Daten, Gelb den Konfigurationsdaten und Grün den Resultaten zugeordnet. Signale sind durch ein ausgefülltes Quadrat an der Verbindungslinie gekennzeichnet. Das Signal oder die Daten sind jeweils beim Empfänger angegeben. Repetitive Aufgaben sind durch ein hinterlegtes oranges Rechteck markiert.

Bei der Erstellung des Prozessmanagers werden alle Konfigurationen mitgeliefert. Der Prozessmanager erzeugt während seiner Instanziierung die Prozesse, welche ebenfalls direkt konfiguriert werden. Die Prozesse laufen nun im Hintergrund und warten auf Daten.

Erhält die Applikation eine Suchanfrage, ruft sie in einem ersten Schritt den Prozessmanager auf. Dieser nimmt noch zu berechnende Daten entgegen. Im Fall von Billard sind dies die Root-Nodes des Suchbaums. Jeder Root-Node repräsentiert ein Loch. Der Prozessmanager gibt in dem Fall eine Datenstruktur zurück, welche es erlaubt, in Zukunft auf die Anfrage zu antworten. In dieser Antwort werden die Resultate geliefert. Der Hauptprozess prüft regelmässig, ob die Bearbeitung der gestellten Aufgaben beendet werden soll. Dies ist der Fall, wenn entweder genügend Lösungen gefunden wurden, die maximal zur Verfügung stehende Zeit abgelaufen ist oder keine noch zu expandierenden Nodes mehr vorhanden sind. Sollte die Berechnung nicht abgebrochen werden, dann werden die offenen Anfragen der Berechnungsprozesse beantwortet. Der Prozessmanager verteilt alle ihm gemeldeten noch zu bearbeitenden Daten an die einzelnen Berechnungsprozesse.

In Abbildung 4.22 wird das Ziel verdeutlicht, welches der Prozessmanager bei einer Verteilung dieser Daten verfolgt. So sollen die jeweils n besten Kandidaten, wobei n für die Anzahl der Berechnungsprozesse steht, auf die verschiedenen Berechnungsprozesse verteilt werden. Danach erhalten die Berechnungsprozesse die nächstbesten n -Kandidaten. Dadurch wird sichergestellt, dass immer an den vielversprechendsten Arbeitspaketen weitergemacht wird.

Der Algorithmus 3 zeigt, wie die offenen Arbeitspakete auf die Berechnungsprozesse verteilt werden, damit das Ziel in Abbildung 4.22 erreicht werden kann. So werden zuerst alle Antworten auf die offenen Anfragen erstellt. Dies geschieht in einer „While-Loop“, welche solange läuft, wie es noch offene Arbeitspakete hat. In jedem Schritt der Schleife

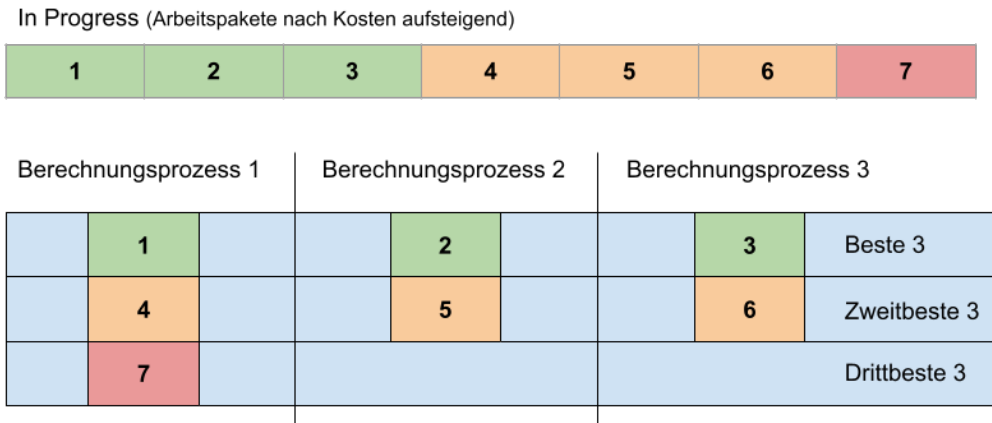


Abbildung 4.22: Berechnungsprozessverteilung

fe wird ein Arbeitspaket einer Antwort für eine Anfrage hinzugefügt. Am Ende der Schleife wird eine Index-Variable hochgezählt. Sollte diese die Anzahl der offenen Anfragen erreichen, beginnt sie wieder bei 0. Dies ist z.B. in Abbildung 4.22 beim Wechsel von Arbeitspakete 3 zu 4 ersichtlich. Sobald keine offenen Arbeitspakete mehr existieren, werden sämtliche Anfragen beantwortet, zu welchen Antworten erstellt werden konnten. Wenn z.B. mehr Anfragen als Arbeitspakete existieren, bleiben diese erhalten und werden im nächsten Synchronisationszyklus beantwortet¹⁵.

Function `distribute(requests: List[future], inProgress: Queue[Node]) → List[future]`

```

answers: List[List[Node]] ← list()
forRequest ← 0
while ! inProgress.empty() do
  if answers[forRequest] is null then
    | answers[forRequest] = list()
  end
  node ← pop(inProgress)
  answers[forRequest] ← append(node, answers[forRequest])
  forRequest ← (forRequest + 1) % size(requests)
end
while ! answers.empty() do
  request ← pop(requests)
  fulfill(request, pop(answers))
end
return requests

```

Algorithm 3: Algorithmus zur Durchführung eines Expansionsschritts im Berechnungsprozess

In einem Zyklus eines Berechnungsprozesses wird zuerst geprüft, ob eine Berechnung abgebrochen werden soll. Dies ist der Fall, wenn der Prozessmanager ein „Clear-Signal“ gesendet hat. Dieses Signal tritt auf, wenn eine neue Berechnung gestartet oder wenn eine aktuell Laufende erfolgreich beendet oder abgebrochen wird.

Der Berechnungsprozess stellt in einem nächsten Schritt eine Datenanfragen an den Prozessmanager, sollte keine offene oder noch nicht bearbeitete Anfrage existieren. Danach erfolgt die Prüfung, ob die Zeit zwischen einem Synchronisationsschritt abgelaufen ist. In dem Fall liefert der Berechnungsprozess seine besten weiterzuführenden Berechnungen wie auch Resultate dem Prozessmanager. Er wird demnach für eine Weile an weniger erfolgsversprechenden Resultaten weiterarbeiten. In einem weiteren Schritt wird geprüft, ob eine Datenanfrage bereits beantwortet wurde. Ist dies der Fall, so werden die Daten den zu bearbeitenden Daten des Berechnungsprozesses hinzugefügt. Zuletzt erfolgt der Expansionsschritt, wobei der erfolgsversprechendste Kandidat expandiert wird. Der Algorithmus 4 zeigt, wie dies in etwa funktionieren kann. Als Eingabe dient eine priorisierte Queue von offenen Arbeitspaketen sowie die Implementation der „Expand-Funktion“. Wenn es offene Arbeitspakete gibt, dann wird das Beste genommen und über die Implementation der „Expand-Funktion“ expandiert. Das Resultat dieses Schrittes wird danach entweder den Lösungen oder den offenen Arbeitspaketen hinzugefügt. Am Schluss werden die offenen

¹⁵Man stelle sich bei der Abbildung 4.22 nur zwei offene Arbeitspakete vor. Demnach würde Paket 1 auf den Berechnungsprozess 1 und Paket 2 auf den Berechnungsprozess 2 verteilt werden. Die Anfrage des Berechnungsprozesses 3 bliebe offen.

Arbeitspakete wie auch die Lösungen zurückgegeben.

```

Function expand(inProgress: Queue[Node], expandImpl: functional) → Queue[Node], Queue[Node]
  solutions ← priority_queue()
  if !inProgress.empty() then
    node ← pop(inProgress)
    expandedNodes ← expandImpl(node)
    for expandedNode in expandedNodes do
      if expandedNode.isSolution() then
        | solutions ← append(expandedNode, solutions)
      end
    else
      | inProgress ← append(expandedNode, inProgress)
    end
  end
end
return inProgress, solutions

```

Algorithm 4: Algorithmus zur Durchführung eines Expansionsschritts im Berechnungsprozess

Die Synchronisation zwischen den Prozessen wird in Abbildung 4.23 erläutert. Sie findet nach einer konfigurierten Zeit k statt. Diese wird in Millisekunden $[ms]$ angegeben. Da die Synchronisation ein exklusives Verwenden einer geteilten Ressource erfordert, um die erledigte Arbeit dem Prozessmanager mitzuteilen, erhalten alle laufenden Prozesse (dies umfasst den Hauptprozess des Prozessmanagers wie auch alle Berechnungsprozesse) ein Zeitfenster zugeteilt, welches nach k Millisekunden startet. Danach kann jeder Prozess nach $k + i \cdot n$, wobei i für die Prozessid beginnend bei 0 und n für das Zeitfenster einer Synchronisation steht, mit der Synchronisation beginnen. Es wird so verhindert, dass zu viele Prozesse auf die Ressource des Prozessmanagers warten müssen und untätig bleiben. Ein weiterer Vorteil dabei ist die Tatsache, dass in der Zeit der Synchronisation einige Prozesse Zeit erhalten, um eher schlechter bewertete Kandidaten weiterzuverfolgen. Dadurch ist es möglich, dass ein sehr gutes Ergebnis gefunden werden kann, obwohl dessen Kandidat anfänglich als schlecht beurteilt wurde.

Prozess\Zeit [ms]	k	k + 0n	k + n	k + 2n	2k	2k + 0n	2k + n	2k + 2n
Hauptprozess								
Berechnungspr. 1								
Berechnungspr. 2								

Abbildung 4.23: Berechnungsprozesssynchronisation

Die Abbildung 4.23 zeigt den Hauptprozess wie auch zwei Berechnungsprozesse, welche eine Synchronisation in einem Intervall von k Millisekunden durchführen. Grüne Spalten stehen hierbei für die Zeit, welche für die Berechnung einer Lösung verwendet wird, also als produktiv bezeichnet werden kann. Rote Spalten hingegen signalisieren den unproduktiven Overhead, der bei der Synchronisation entsteht. Der Prozessmanager macht nebst seinem Synchronisationsfenster nichts, weswegen seine Spalten grau markiert sind. Eine mögliche Verbesserung wäre die Auslastung des Prozessmanagers mit zusätzlicher Berechnungsarbeit, wie sie die Berechnungsprozesse durchführen.

5 Resultate

Dieses Kapitel beinhaltet die Aufführung aller Ergebnisse der Arbeit. Dies betrifft insbesondere die Fortsetzung der Genauigkeitsanalyse der Kugeldetektion aus der Vorarbeit[Luk21b], die Genauigkeitsanalyse der Klassifikation wie auch eine Aufstellung einiger Spielstände, deren Suchresultate und verwendete Berechnungszeiten für die einfache direkte wie auch erweiterte Suche. Zudem wird die Berechnung des Rollreibungskoeffizienten und der Energieverluste bei Ereignissen behandelt.

5.1 Klassifikation

5.2 Bestimmen des Rollreibungskoeffizienten

Rollt eine Kugel einen bestimmten Weg über den Billardtisch, so entsteht eine der Geschwindigkeit entgegengesetzte Richtung wirkende Kraft. Die Kraft kann als negative Beschleunigung angegeben werden, wobei diese von einem Rollreibungskoeffizienten c_R abhängig ist. Das Finden dieses Koeffizienten wird in diesem Kapitel beschrieben. Die Herleitung und theoretischen Überlegungen finden sich in Kapitel 8.8.



Abbildung 5.1: Versuchsaufbau zur Bestimmung des Reibungskoeffizienten

Abbildung 5.1 veranschaulicht den Versuchsaufbau zur Bestimmung des Rollreibungskoeffizienten. Wie in Kapitel 8.8 erläutert, muss für die Rampe ein Material mit möglichst geringer Reibung verwendet werden, weswegen die Wahl auf Glas fiel. Weiterhin wird der Weg der Kugel geführt, da sie eine möglichst gerade Bahn rollen muss. Die so entstehende Reibung an der Bahn wird ebenfalls vernachlässigt.

Es wurden zwei Versuche durchgeführt mit unterschiedlicher Höhe der Rampe. Die Ergebnisse dieser Messungen wie deren Median sind in der Tabelle 5.1 aufgeführt.

Distanz [mm] \ Höhe [mm]	13	19
	957	1228
	949	1253
	926	1263
	914	1280
	931	1281
	932	1296
	941	1256
	926	1295
	931	1295
	941	1304
	943	1314
	937	1316
	947	1284
	949	1303
	947	1319
	-	1320
Median	941	1295

Tabelle 5.1: Ergebnisse Distanzmessung einer rollenden Kugel

Anhand des Ablaufs aus Kapitel 8.8 können die Reibungskoeffizienten aus Tabelle 5.2 bestimmt werden. Daraus wird der Mittelwert berechnet, welcher das Resultat bildet.

Strecke [mm]	Reibungskoeffizient
941	0.0138151
1295	0.0146718
Mittelwert	0.0142435

Tabelle 5.2: Reibungskoeffizienten über Strecke

5.3 Energieverlust

In den Formeln für die Berechnung der Initialgeschwindigkeit wie auch die Auswirkung bei einer auftretenden Kugelkollision wurde ein Faktor E_v , angegeben in Prozent, berücksichtigt, welcher den Energieverlust modelliert. Es gilt durch ein experimentelles Verfahren diesen Wert für die jeweilige Art zu bestimmen.

5.3.1 Energieverlust bei Kugelkollision

TODO: $E_v = 0.05$

5.3.2 Energieverlust bei Bandenkollision

Es wird angenommen, dass der Energieverlust bei einer Kollision einer Kugel mit der Bande ca. 2% beträgt[MJP10].

6 Weitere Arbeiten

TODO: Weitere Arbeiten

7 Fazit

TODO: Fazit

Abbildungsverzeichnis

4.1	Billard-AI-Cycle	9
4.2	Risikoanalyse	11
4.3	Kandidatensuche 1	12
4.4	Kandidatensuche 2	13
4.5	Kandidatensuche 3	13
4.6	Bewertung eines Suchbaums	14
4.7	Expansionskosten eines Knotens	14
4.8	Gewichtung der Winkelkosten	15
4.9	Kollisionspunkt zweier Kugeln	17
4.10	Typen von Objekten	18
4.11	Der Energy-Transfer-Node	19
4.12	Beispiel für ein Resultat des Algorithmus 2	20
4.13	Elastischer Stoss zweier Kugeln	22
4.14	Vorbedingung einer Prüfung auf Kollision zwischen Kugeln bei statischer Beteiligung	25
4.15	Vorbedingung einer Prüfung auf Kollision zwischen dynamischen Kugeln	25
4.16	Vorbedingung einer Prüfung auf Kollision zwischen dynamischen parallel verlaufenden Kugeln	26
4.17	Vorbedingung einer Prüfung auf Kollision zwischen dynamischen hintereinander verlaufenden Kugeln	26
4.18	Keyframe Animation	29
4.19	Simulationsmodell mit Keyframes	29
4.20	Simulationsmodell mit paarweisen Keyframefenstern	30
4.21	Berechnungsprozess	31
4.22	Berechnungsprozessverteilung	32
4.23	Berechnungsprozesssynchronisation	33
5.1	Versuchsaufbau zur Bestimmung des Reibungskoeffizienten	36
8.1	Kollisionspunkt zweier Kugeln	52
8.2	Elastischer Stoss zweier Kugeln	53
8.3	Modell zur Berechnung der Startgeschwindigkeit auf dem Tisch	62
8.4	Modell zur Berechnung des Reibungskoeffizienten	63

Tabellenverzeichnis

3.1	Ziele	6
4.1	Risiken	10
5.1	Ergebnisse Distanzmessung einer rollenden Kugel	36
5.2	Reibungskoeffizienten über Strecke	37

Literatur

- [Bou00] Paul Bourke. *Calculating reflected ray*. [Online; accessed 05.10.2021]. 2000. URL: <http://paulbourke.net/geometry/reflected/>.
- [Luk21a] Luca Ritz Lukas Seglias. "Billiard-AI". In: (2021), S. 5–6.
- [Luk21b] Luca Ritz Lukas Seglias. "Billiard-AI". In: (2021), S. 21–23.
- [MJP10] Senthana Mathavan, M Jackson und Robert Parkin. "A theoretical analysis of billiard ball dynamics under cushion impacts". In: *Proceedings of The Institution of Mechanical Engineers Part C-journal of Mechanical Engineering Science - PROC INST MECH ENG C-J MECH E* 1 (Sep. 2010), S. 1–10. doi: 10.1243/09544062JMES1964.
- [Mül21] Arnd Müller. *Ein Billardroboter - Praktische Realisierung von ausgewählten Konzepten der Robotik*. [Online; accessed 28.10.2021]. 2021. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-164739>.
- [Smi06] Michael Smith. "Running the Table: An AI for Computer Billiards." In: Bd. 1. Jan. 2006, S. 996.
- [Unk21a] Unknown. *Arbeit (Physik)*. [Online; accessed 04.10.2021]. 2021. URL: [https://de.wikipedia.org/wiki/Arbeit_\(Physik\)](https://de.wikipedia.org/wiki/Arbeit_(Physik)).
- [Unk21b] Unknown. *Bézier curve*. [Online; accessed 28.10.2021]. 2021. URL: https://en.wikipedia.org/wiki/B%C3%A9zier_curve.
- [Unk21c] Unknown. *Dijkstra's algorithm*. [Online; accessed 28.10.2021]. 2021. URL: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm.
- [Unk21d] Unknown. *Effet*. [Online; accessed 05.10.2021]. 2021. URL: <https://de.wikipedia.org/wiki/Effet>.
- [Unk21e] Unknown. *Elastischer Stoss (Physik)*. [Online; accessed 05.10.2021]. 2021. URL: [https://de.wikipedia.org/wiki/Sto%C3%9F_\(Physik\)#Elastischer_Sto%C3%9F](https://de.wikipedia.org/wiki/Sto%C3%9F_(Physik)#Elastischer_Sto%C3%9F).
- [Unk21f] Unknown. *Polynom vierten Grades*. [Online; accessed 04.10.2021]. 2021. URL: https://de.wikipedia.org/wiki/Polynom_vierten_Grades.
- [Unk21g] Unknown. *Quadratische Gleichung - Lösungsformel*. [Online; accessed 04.10.2021]. 2021. URL: https://de.wikipedia.org/wiki/Quadratische_Gleichung#Allgemeine_L%C3%B6sungsformeln.
- [Unk21h] Unknown. *Rollwiderstand*. [Online; accessed 04.10.2021]. 2021. URL: <https://de.wikipedia.org/wiki/Rollwiderstand>.



Erklärung der Diplomandinnen und Diplomanden *Déclaration des diplômé-e-s*

Selbständige Arbeit / *Travail autonome*

Ich bestätige mit meiner Unterschrift, dass ich meine vorliegende Bachelor-These selbständig durchgeführt habe. Alle Informationsquellen (Fachliteratur, Besprechungen mit Fachleuten, usw.) und anderen Hilfsmittel, die wesentlich zu meiner Arbeit beigetragen haben, sind in meinem Arbeitsbericht im Anhang vollständig aufgeführt. Sämtliche Inhalte, die nicht von mir stammen, sind mit dem genauen Hinweis auf ihre Quelle gekennzeichnet.

Par ma signature, je confirme avoir effectué ma présente thèse de bachelor de manière autonome. Toutes les sources d'information (littérature spécialisée, discussions avec spécialistes etc.) et autres ressources qui m'ont fortement aidé-e dans mon travail sont intégralement mentionnées dans l'annexe de ma thèse. Tous les contenus non rédigés par mes soins sont dûment référencés avec indication précise de leur provenance.

Name/*Nom*, Vorname/*Prénom*

Datum/*Date*

Unterschrift/*Signature*

Dieses Formular ist dem Bericht zur Bachelor-These beizulegen.
Ce formulaire doit être joint au rapport de la thèse de bachelor.



Erklärung der Diplomandinnen und Diplomanden *Déclaration des diplômé-e-s*

Selbständige Arbeit / *Travail autonome*

Ich bestätige mit meiner Unterschrift, dass ich meine vorliegende Bachelor-These selbständig durchgeführt habe. Alle Informationsquellen (Fachliteratur, Besprechungen mit Fachleuten, usw.) und anderen Hilfsmittel, die wesentlich zu meiner Arbeit beigetragen haben, sind in meinem Arbeitsbericht im Anhang vollständig aufgeführt. Sämtliche Inhalte, die nicht von mir stammen, sind mit dem genauen Hinweis auf ihre Quelle gekennzeichnet.

Par ma signature, je confirme avoir effectué ma présente thèse de bachelor de manière autonome. Toutes les sources d'information (littérature spécialisée, discussions avec spécialistes etc.) et autres ressources qui m'ont fortement aidé-e dans mon travail sont intégralement mentionnées dans l'annexe de ma thèse. Tous les contenus non rédigés par mes soins sont dûment référencés avec indication précise de leur provenance.

Name/*Nom*, Vorname/*Prénom*

Datum/*Date*

Unterschrift/*Signature*

Dieses Formular ist dem Bericht zur Bachelor-These beizulegen.
Ce formulaire doit être joint au rapport de la thèse de bachelor.

8 Anhang

8.1 Herleitung Startgeschwindigkeit auf Basis bekannter Endgeschwindigkeit unter Einbezug von Reibung

Dies erfordert den Energieerhaltungssatz. Dazu wird die kinetische Energie vor sowie nachher betrachtet.

$$E_{kin} = \frac{1}{2} \cdot m \cdot |\vec{v}|^2 \quad (8.1)$$

$$E_{vorher} = \frac{1}{2} \cdot m \cdot |\vec{v}_1|^2 \quad (8.2)$$

$$E_{nachher} = \frac{1}{2} \cdot m \cdot |\vec{v}_2|^2 \quad (8.3)$$

Die Rollreibung F_R wird mithilfe der Normalkraft F_N und dem Rollwiderstandskoeffizienten c_R definiert[Unk21h]. Mithilfe der Masse m der Kugel und der Schwerebeschleunigung g kann die Normalkraft F_N berechnet werden.

$$F_R = c_R \cdot F_N \quad (8.4)$$

$$F_N = m \cdot g \quad (8.5)$$

Die Rollreibung F_R wird über eine bestimmte Strecke Δs , welche die Kugel zurücklegt, angewendet. Dadurch entsteht eine Arbeit $E_{Reibung}$ [Unk21a]:

$$E_{Reibung} = F_R \cdot \Delta s \quad (8.6)$$

Nun kann der Energieerhaltungssatz angewendet werden:

$$E_{vorher} = E_{nachher} + E_{Reibung} \quad (8.7)$$

$$\frac{1}{2} \cdot m \cdot |\vec{v}_1|^2 = \frac{1}{2} \cdot m \cdot |\vec{v}_2|^2 + F_R \cdot \Delta s \quad (8.8)$$

$$m \cdot |\vec{v}_1|^2 = m \cdot |\vec{v}_2|^2 + 2 \cdot F_R \cdot \Delta s \quad (8.9)$$

$$|\vec{v}_1|^2 = \frac{m \cdot |\vec{v}_2|^2 + 2 \cdot F_R \cdot \Delta s}{m} \quad (8.10)$$

$$|\vec{v}_1|^2 = \frac{m \cdot |\vec{v}_2|^2 + 2 \cdot m \cdot g \cdot c_R \cdot \Delta s}{m} \quad (8.11)$$

$$|\vec{v}_1|^2 = |\vec{v}_2|^2 + 2 \cdot g \cdot c_R \cdot \Delta s \quad (8.12)$$

$$|\vec{v}_1| = \sqrt{|\vec{v}_2|^2 + 2 \cdot g \cdot c_R \cdot \Delta s} \quad (8.13)$$

Damit ist die Startgeschwindigkeit bestimmt. Nun stellt sich noch die Frage nach der Richtung. Diese zeigt in dieselbe, wie die Endgeschwindigkeit. Daher kann nun die bekannte Länge mit dem normalisierten Vektor \vec{v}_2 multipliziert werden:

$$\vec{v}_1 = |\vec{v}_1| \cdot \frac{\vec{v}_2}{|\vec{v}_2|} \quad (8.14)$$

Daraus folgt die Formel:

$$\vec{v}_1 = \sqrt{|\vec{v}_2|^2 + 2 \cdot g \cdot c_R \cdot \Delta s} \cdot \frac{\vec{v}_2}{|\vec{v}_2|} \quad (8.15)$$

8.2 Herleitung Initialgeschwindigkeit bei Kugel-Kollision

Eine Kugel T soll in eine gewünschte Richtung rollen. Dazu soll sie von einer anderen Kugel A angestossen werden. Gesucht ist der Punkt, wohin Kugel A rollen muss, um mit Kugel T zusammenzustossen. Die Kugel T soll mit einer bestimmten Geschwindigkeit von der Kollision ausgehen. Dazu ist die Geschwindigkeit zu bestimmen, die Kugel A zum Zeitpunkt der Kollision haben muss.

Die Situation mit der Kugel T, an der Position C und der Kugel A, an der Position A ist in Abbildung 8.1 dargestellt. Die Kugel A muss zum Punkt B rollen, wo sie auf Kugel T prallt und ein elastischer Stoss[Unk21e] stattfindet. Während die Kugel die Distanz $|\vec{d}|$ zurücklegt, verliert sie an Geschwindigkeit aufgrund von Reibung, diese wird in Abschnitt 4.4.2.1 behandelt. Hier ist lediglich relevant, welche Geschwindigkeit die Kugel A am Punkt B haben muss.

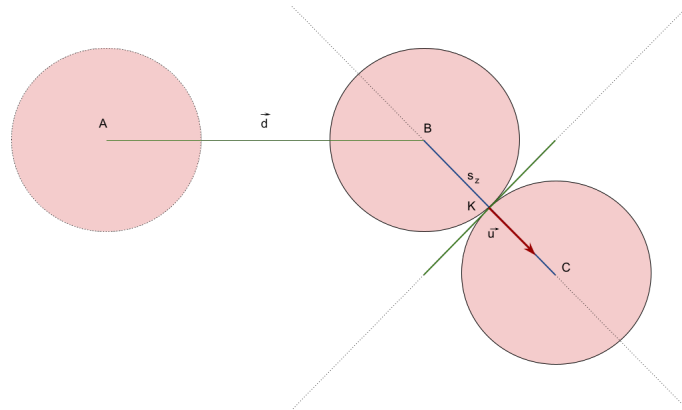


Abbildung 8.1: Kollisionspunkt zweier Kugeln

Sei die gewünschte Richtung, in die Kugel T nach der Kollision rollen soll, \hat{u} und der Kugelradius r , dann kann die Position B bestimmt werden:

$$s_z = 2 \cdot r \quad (8.16)$$

$$B = C - s_z \cdot \hat{u} \quad (8.17)$$

$$B = C - 2 \cdot r \cdot \hat{u} \quad (8.18)$$

Nun gilt es die Geschwindigkeit $\vec{v}_{1,z}$ der Kugel A zum Zeitpunkt der Kollision zu bestimmen. Nachfolgend wird angenommen, dass die zu treffende Kugel T stillsteht und somit die Geschwindigkeit $\vec{v}_2 = \vec{0}$ hat. Der Geschwindigkeitsvektor \vec{u} ist durch die gewünschte Richtung und Geschwindigkeit, welche die Kugel T nach dem Zusammenstoß haben soll, gegeben.

$$\vec{u} = \vec{v}_{1,z} + \vec{v}_{2,t} \quad (8.19)$$

$$\vec{u} = \vec{v}_{1,z} + \vec{0} \quad (8.20)$$

$$\vec{u} = \vec{v}_{1,z} \quad (8.21)$$

Die Situation ist in Abbildung 8.2 dargestellt.

Gegeben sind die nachfolgenden Informationen:

- A: Startposition der Kugel A.
- C: Position der Kugel T.
- \vec{u} : Gewünschte Geschwindigkeit und Richtung der Kugel T nach dem Zusammenstoß.

Weiterhin ist die Richtung von \vec{v}_1 gegeben, da dieser parallel zum Vektor \vec{d} ist. Allerdings muss die Länge des Vektors noch bestimmt werden:

$$\vec{d} = B - A \quad (8.22)$$

$$\hat{d} = \frac{\vec{d}}{|\vec{d}|} \quad (8.23)$$

$$\vec{v}_1 = \alpha \cdot \hat{d} \quad (8.24)$$

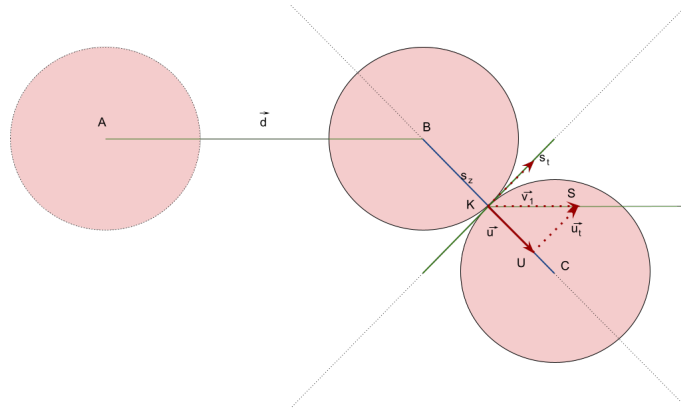


Abbildung 8.2: Elastischer Stoss zweier Kugeln

Die Länge des Vektors \vec{v}_1 muss nach Projektion auf den Einheitsvektor \hat{u} in Richtung von \vec{u} der Länge des Vektors \vec{u} entsprechen. Dazu muss ein Skalierungsfaktor α bestimmt werden, mit dem der Einheitsvektor \hat{d} skaliert werden kann:

$$\hat{u} = \frac{\vec{u}}{|\vec{u}|} \quad (8.25)$$

$$\vec{v}_1 \cdot \hat{u} = |\vec{u}| \quad (8.26)$$

$$(\alpha \cdot \hat{d}) \cdot \hat{u} = |\vec{u}| \quad (8.27)$$

$$\alpha \cdot (\hat{d} \cdot \hat{u}) = |\vec{u}| \quad (8.28)$$

$$\alpha = \frac{|\vec{u}|}{\hat{d} \cdot \hat{u}} \quad (8.29)$$

$$\alpha = \frac{|\vec{u}|}{\hat{d} \cdot \frac{\vec{u}}{|\vec{u}|}} \quad (8.30)$$

$$\alpha = \frac{|\vec{u}|^2}{\hat{d} \cdot \vec{u}} \quad (8.31)$$

$$\alpha = \frac{\vec{u} \cdot \vec{u}}{\hat{d} \cdot \vec{u}} \quad (8.32)$$

$$\vec{v}_1 = \alpha \cdot \hat{d} \quad (8.33)$$

$$\vec{v}_1 = \frac{\vec{u} \cdot \vec{u}}{\hat{d} \cdot \vec{u}} \cdot \hat{d} \quad (8.34)$$

Somit ist die Geschwindigkeit, welche die Kugel A an Position B haben muss bestimmt. Der Energieverlust wird über eine Konstante E_v definiert, welche in Prozent angegeben wird. Es resultiert eine Verlängerung des berechneten Vektors.

$$\vec{v}_1 = \frac{\vec{u} \cdot \vec{u}}{\hat{d} \cdot \vec{u}} \cdot \frac{1}{1 - E_v} \cdot \hat{d} \quad (8.35)$$

8.3 Herleitung Ereignis - Totaler Energieverlust

Dieses Ereignis tritt ein, sobald eine Kugel ausrollt. Der Zeitpunkt wird über die Geschwindigkeitsgleichung der gleichmässig beschleunigten Bewegung bestimmt.

$$\vec{v}(t) = \vec{a} \cdot t + \vec{v}_0 \quad (8.36)$$

Bekannt ist, dass der Betrag des Geschwindigkeitsvektors 0 sein muss.

$$|\vec{v}(t)| = 0 \quad (8.37)$$

$$\vec{0} = \vec{a} \cdot t + \vec{v}_0 \quad (8.38)$$

Um t anhand von bekannten Werten \vec{a} und \vec{v}_0 zu ermitteln, gibt es zwei Möglichkeiten. Bei der Ersten kann die Gleichung 8.38 Komponentenweise aufgestellt werden, wobei auch t in zwei verschiedene Richtungen betrachtet werden muss.

$$0 = a_x \cdot t_x + v_{0,x} \quad (8.39)$$

$$0 = a_y \cdot t_y + v_{0,y} \quad (8.40)$$

Damit ergeben sich zwei unabhängige Gleichungen mit zwei unbekannten t_x und t_y , die Zeitpunkte, an denen die Kugel in der entsprechenden Richtung stillsteht. Angenommen, eine Kugel rollt nur in X-Richtung, dann ist der Zeitpunkt t_x , zu dem die Kugel in X-Richtung stillsteht und der Zeitpunkt t_y , zu dem die Kugel in Y-Richtung stillsteht, unterschiedlich. Angenommen, eine Kugel rollt mit einer Geschwindigkeit in X- und Y-Richtung. Dann entspricht die Beschleunigung \vec{a} einem konstanten Faktor angewendet auf den Geschwindigkeitsvektor.

$$\vec{a} = -\mu \cdot \vec{v}_0 \quad (8.41)$$

Die beiden Variablen t_x und t_y sind in diesem Fall äquivalent. Eine Unterscheidung auf die beiden Komponenten wird also nur getroffen, da sich das Problem auf nur eine Dimension beschränken kann. Es ergeben sich die beiden nachfolgenden Formeln.

$$t_x = \frac{-v_{0,x}}{a_x} \quad (8.42)$$

$$t_y = \frac{-v_{0,y}}{a_y} \quad (8.43)$$

Der Zeitpunkt t zu dem die Kugel vollständig stillsteht, ist so wie folgt gegeben:

$$t = \max(t_x, t_y) \quad (8.44)$$

$$t = \max\left(\frac{-v_{0,x}}{a_x}, \frac{-v_{0,y}}{a_y}\right) \quad (8.45)$$

Die andere Möglichkeit ist die Betrachtung der Gleichung 8.38 über die Beträge der Vektoren \vec{a} sowie \vec{v}_0 .

Werden die Beträge genommen, dann kann die Gleichung 8.38 als eindimensionales Problem aufgefasst und dementsprechend gelöst werden. Auch hier ist der Vektor \vec{a} abhängig von der Geschwindigkeit \vec{v}_0 über einen konstanten Faktor.

$$v(t) = |\vec{a}| \cdot t + |\vec{v}_0| \quad (8.46)$$

Es folgen einige Umformungen, um die Variable t auf eine Seite der Gleichung zu bringen.

$$0 = |\vec{a}| \cdot t + |\vec{v}_0| \quad (8.47)$$

$$-|\vec{v}_0| = |\vec{a}| \cdot t \quad (8.48)$$

$$-\sqrt{v_{0,x}^2 + v_{0,y}^2} = \sqrt{a_x^2 + a_y^2} \cdot t \quad (8.49)$$

$$v_{0,x}^2 + v_{0,y}^2 = a_x^2 + a_y^2 \cdot t^2 \quad (8.50)$$

$$\frac{v_{0,x}^2 + v_{0,y}^2}{a_x^2 + a_y^2} = t^2 \quad (8.51)$$

$$\sqrt{\frac{v_{0,x}^2 + v_{0,y}^2}{a_x^2 + a_y^2}} = t \quad (8.52)$$

Damit stehen zwei Möglichkeiten zur Verfügung.

$$t = \max\left(\frac{-v_{0,x}}{a_x}, \frac{-v_{0,y}}{a_y}\right) \quad (8.53)$$

$$t = \sqrt{\frac{v_{0,x}^2 + v_{0,y}^2}{a_x^2 + a_y^2}} \quad (8.54)$$

Verwendet wird die Gleichung ??, da angenommen wird, dass das Nehmen eines Maximalwerts zweier Divisionen effizienter ist, als die Berechnung einer Wurzel einer Division zweier Additionen.

8.4 Herleitung Ereignis - Kollision dynamischer Objekte

In diesem Kapitel erfolgt die Herleitung zur Bestimmung des Kollisionszeitpunkts zweier dynamischer Objekte. Beim Billard sind dies zwei Kugeln, welche in Bewegung sind.

Um die Formel herzuleiten, wird die Ortsgleichung der gleichförmig beschleunigten Bewegung betrachtet:

$$\vec{s}(t) = \frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v}_0 \cdot t + \vec{s}_0 \quad (8.55)$$

$$\begin{pmatrix} s(t)_x \\ s(t)_y \end{pmatrix} = \frac{1}{2} \cdot \begin{pmatrix} a_x \\ a_y \end{pmatrix} \cdot t^2 + \begin{pmatrix} v_{0,x} \\ v_{0,y} \end{pmatrix} \cdot t + \begin{pmatrix} s_{0,x} \\ s_{0,y} \end{pmatrix} \quad (8.56)$$

Diese ist für beide Kugeln unterschiedlich:

$$\vec{s}_1(t) = \frac{1}{2} \cdot \vec{a}_1 \cdot t^2 + \vec{v}_1 \cdot t + \vec{s}_1 \quad (8.57)$$

$$\vec{s}_2(t) = \frac{1}{2} \cdot \vec{a}_2 \cdot t^2 + \vec{v}_2 \cdot t + \vec{s}_2 \quad (8.58)$$

Da die Objekte zweidimensional sind, also eine bestimmte Fläche aufweisen, können die Gleichungen nicht einfach gleichgesetzt werden. Der Radius der Kugel muss mitbeachtet werden, weswegen die Gleichungen einen Punkt suchen, wo die Distanz zwischen ihnen dem Kugeldurchmesser entspricht. Dies ist der Ort, wo die Kugeln im zweidimensionalen (auch im dreidimensionalen) Raum kollidieren werden. Dazu wird die Parameterform nach \vec{d} bestimmt.

$$\vec{d} = \vec{s}_1(t) - \vec{s}_2(t) \quad (8.59)$$

$$\vec{d} = \frac{1}{2} \cdot \vec{a}_1 \cdot t^2 + \vec{v}_1 \cdot t + \vec{s}_1 - \left(\frac{1}{2} \cdot \vec{a}_2 \cdot t^2 + \vec{v}_2 \cdot t + \vec{s}_2 \right) \quad (8.60)$$

$$\vec{d} = \frac{1}{2} \cdot \vec{a}_1 \cdot t^2 + \vec{v}_1 \cdot t + \vec{s}_1 - \frac{1}{2} \cdot \vec{a}_2 \cdot t^2 - \vec{v}_2 \cdot t - \vec{s}_2 \quad (8.61)$$

$$\vec{d} = \frac{1}{2} \cdot (\vec{a}_1 - \vec{a}_2) \cdot t^2 + \vec{v}_1 \cdot t + \vec{s}_1 - \vec{v}_2 \cdot t - \vec{s}_2 \quad (8.62)$$

$$\vec{d} = \frac{1}{2} \cdot (\vec{a}_1 - \vec{a}_2) \cdot t^2 + \vec{v}_1 \cdot t - \vec{v}_2 \cdot t + \vec{s}_1 - \vec{s}_2 \quad (8.63)$$

$$\vec{d} = \frac{1}{2} \cdot (\vec{a}_1 - \vec{a}_2) \cdot t^2 + t \cdot (\vec{v}_1 - \vec{v}_2) + \vec{s}_1 - \vec{s}_2 \quad (8.64)$$

$$\vec{d} = t^2 \cdot \frac{1}{2} \cdot \begin{pmatrix} a_{1,x} - a_{2,x} \\ a_{1,y} - a_{2,y} \end{pmatrix} + t \cdot \begin{pmatrix} v_{1,x} - v_{2,x} \\ v_{1,y} - v_{2,y} \end{pmatrix} + \begin{pmatrix} s_{1,x} - s_{2,x} \\ s_{1,y} - s_{2,y} \end{pmatrix} \quad (8.65)$$

$$(8.66)$$

Die Parameterform wird nach Komponenten aufgeteilt:

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = t^2 \cdot \frac{1}{2} \cdot \begin{pmatrix} a_{1,x} - a_{2,x} \\ a_{1,y} - a_{2,y} \end{pmatrix} + t \cdot \begin{pmatrix} v_{1,x} - v_{2,x} \\ v_{1,y} - v_{2,y} \end{pmatrix} + \begin{pmatrix} s_{1,x} - s_{2,x} \\ s_{1,y} - s_{2,y} \end{pmatrix} \quad (8.67)$$

$$d_x = t^2 \cdot \frac{1}{2} \cdot (a_{1,x} - a_{2,x}) + t \cdot (v_{1,x} - v_{2,x}) + s_{1,x} - s_{2,x} \quad (8.68)$$

$$d_y = t^2 \cdot \frac{1}{2} \cdot (a_{1,y} - a_{2,y}) + t \cdot (v_{1,y} - v_{2,y}) + s_{1,y} - s_{2,y} \quad (8.69)$$

Wobei bekannt ist, dass \vec{d} die Länge D hat (entspricht dem Kugeldurchmesser):

$$|\vec{d}| = \sqrt{(d_x)^2 + (d_y)^2} = D \quad (8.70)$$

$$(d_x)^2 + (d_y)^2 = D^2 \quad (8.71)$$

Die einzelnen Komponenten werden wie folgt substituiert:

$$\Delta a_x = a_{1,x} - a_{2,x} \quad (8.72)$$

$$\Delta a_y = a_{1,y} - a_{2,y} \quad (8.73)$$

$$\Delta v_x = v_{1,x} - v_{2,x} \quad (8.74)$$

$$\Delta v_y = v_{1,y} - v_{2,y} \quad (8.75)$$

$$\Delta s_x = s_{1,x} - s_{2,x} \quad (8.76)$$

$$\Delta s_y = s_{1,y} - s_{2,y} \quad (8.77)$$

Daraus folgt:

$$d_x = t^2 \cdot \frac{1}{2} \cdot \Delta a_x + t \cdot \Delta v_x + \Delta s_x \quad (8.78)$$

$$d_y = t^2 \cdot \frac{1}{2} \cdot \Delta a_y + t \cdot \Delta v_y + \Delta s_y \quad (8.79)$$

$$d_x^2 = (t^2 \cdot \frac{1}{2} \cdot \Delta a_x + t \cdot \Delta v_x + \Delta s_x)^2 \quad (8.80)$$

$$d_x^2 = t^4 \cdot (\frac{1}{2})^2 \cdot \Delta a_x^2 + 2 \cdot t^3 \cdot \frac{1}{2} \cdot \Delta a_x \cdot \Delta v_x + 2 \cdot t^2 \cdot \frac{1}{2} \cdot \Delta a_x \cdot \Delta s_x + t^2 \cdot \Delta v_x^2 + 2 \cdot t \cdot \Delta v_x \cdot \Delta s_x + \Delta s_x^2 \quad (8.81)$$

$$d_x^2 = (\frac{1}{2})^2 \cdot \Delta a_x^2 \cdot t^4 + \Delta a_x \cdot \Delta v_x t^3 + (\Delta a_x \cdot \Delta s_x + \Delta v_x^2) t^2 + 2 \cdot \Delta v_x \cdot \Delta s_x \cdot t + \Delta s_x^2 \quad (8.82)$$

Aus Gleichung 8.82 folgt für d_y^2 :

$$d_y^2 = (\frac{1}{2})^2 \cdot \Delta a_y^2 \cdot t^4 + \Delta a_y \cdot \Delta v_y t^3 + (\Delta a_y \cdot \Delta s_y + \Delta v_y^2) t^2 + 2 \cdot \Delta v_y \cdot \Delta s_y \cdot t + \Delta s_y^2 \quad (8.83)$$

Aus Gleichung 8.71, 8.82 und 8.83 folgt:

$$\begin{aligned} D^2 = & (\frac{1}{2})^2 \cdot \Delta a_x^2 \cdot t^4 + \\ & \Delta a_x \cdot \Delta v_x t^3 + \\ & (\Delta a_x \cdot \Delta s_x + \Delta v_x^2) t^2 + \\ & 2 \cdot \Delta v_x \cdot \Delta s_x \cdot t + \\ & \Delta s_x^2 + \\ & (\frac{1}{2})^2 \cdot \Delta a_y^2 \cdot t^4 + \\ & \Delta a_y \cdot \Delta v_y t^3 + \\ & (\Delta a_y \cdot \Delta s_y + \Delta v_y^2) t^2 + \\ & 2 \cdot \Delta v_y \cdot \Delta s_y \cdot t + \\ & \Delta s_y^2 \end{aligned} \quad (8.84)$$

Daraus ergibt sich das folgende Polynom vierten Grades:

$$\begin{aligned} D^2 = & (\frac{1}{2})^2 \cdot (\Delta a_x^2 + \Delta a_y^2) \cdot t^4 + \\ & (\Delta a_x \cdot \Delta v_x + \Delta a_y \cdot \Delta v_y) \cdot t^3 + \\ & (\Delta a_x \cdot \Delta s_x + \Delta v_x^2 + \Delta a_y \cdot \Delta s_y + \Delta v_y^2) t^2 + \\ & 2 \cdot (\Delta v_x \cdot \Delta s_x + \Delta v_y \cdot \Delta s_y) \cdot t + \\ & \Delta s_x^2 + \Delta s_y^2 \end{aligned} \quad (8.85)$$

Die Koeffizienten können als Skalarprodukte von Vektoren repräsentiert werden. Dies geschieht durch Ausnutzung der Eigenschaft, dass die quadrierte Norm dem Skalarprodukt des Vektors mit sich selbst entspricht.

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad (8.86)$$

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2} \quad (8.87)$$

$$|\vec{a}|^2 = a_x^2 + a_y^2 \quad (8.88)$$

$$|\vec{a}|^2 = \vec{a} \cdot \vec{a} \quad (8.89)$$

$$a_x^2 + a_y^2 = \vec{a} \cdot \vec{a} \quad (8.90)$$

Wird 8.90 auf die Gleichung 8.85 angewendet, resultieren die nachfolgenden Eigenschaften.

$$\Delta a_x^2 + \Delta a_y^2 = \vec{\Delta a} \cdot \vec{\Delta a} \quad (8.91)$$

$$\Delta a_x \cdot \Delta v_x + \Delta a_y \cdot \Delta v_y = \vec{\Delta a} \cdot \vec{\Delta v} \quad (8.92)$$

$$\Delta a_x \cdot \Delta s_x + \Delta v_x^2 + \Delta a_y \cdot \Delta s_y + \Delta v_y^2 = \vec{\Delta a} \cdot \vec{\Delta s} + \vec{\Delta v} \cdot \vec{\Delta v} \quad (8.93)$$

$$2 \cdot (\Delta v_x \cdot \Delta s_x + \Delta v_y \cdot \Delta s_y) = 2 \cdot (\vec{\Delta v} \cdot \vec{\Delta s}) \quad (8.94)$$

$$\Delta s_x^2 + \Delta s_y^2 = \vec{\Delta s} \cdot \vec{\Delta s} \quad (8.95)$$

Eingesetzt in 8.85:

$$D^2 = \left(\frac{1}{2}\right)^2 \cdot (\vec{\Delta a} \cdot \vec{\Delta a}) \cdot t^4 + (\vec{\Delta a} \cdot \vec{\Delta v}) \cdot t^3 + (\vec{\Delta a} \cdot \vec{\Delta s} + \vec{\Delta v} \cdot \vec{\Delta v}) t^2 + 2 \cdot (\vec{\Delta v} \cdot \vec{\Delta s}) \cdot t + \vec{\Delta s} \cdot \vec{\Delta s} \quad (8.96)$$

8.5 Herleitung Ereignis - Kollision mit Bande

Das Ziel ist es, die Position einer Kugel zum Zeitpunkt einer Kollision mit einer Bande zu bestimmen.

Die Bande wird über zwei Punkte beschrieben R_1 und R_2 (für Rail).

Auf Basis dieser Punkte kann eine Geradengleichung formuliert werden:

$$\vec{\Delta R} = \vec{R}_2 - \vec{R}_1 \quad (8.97)$$

$$s(\vec{\lambda}_1) = \vec{R}_1 + \lambda_1 \cdot \vec{\Delta R} \quad (8.98)$$

Weiterhin kann aufgrund der Position C und der bekannten Geschwindigkeit \vec{v} der Kugel eine Linie definiert werden:

$$s(\vec{\lambda}_2) = \vec{C} + \lambda_2 \cdot \vec{v} \quad (8.99)$$

Der Kollisionspunkt mit der Bande ist damit der Schnittpunkt dieser Linie und dem Banden-Liniensegment. Nun kann der Schnittpunkt der beiden Geraden anhand der Formel in Anhang 8.9 bestimmt werden. Dabei sind einige Variablensubstitutionen zu tun:

$$P = \vec{R}_1 \quad (8.100)$$

$$D = \vec{\Delta R} \quad (8.101)$$

$$Q = \vec{C} \quad (8.102)$$

$$V = \vec{v} \quad (8.103)$$

$$(8.104)$$

Damit gilt dann:

$$\lambda_1 = \frac{V_x \cdot P_y - V_y \cdot P_x + Q_x \cdot V_y - Q_y \cdot V_x}{D_x \cdot V_y - D_y \cdot V_x} \quad (8.105)$$

$$\lambda_2 = \frac{P_x \cdot D_y - P_y \cdot D_x - Q_x \cdot D_y + Q_y \cdot D_x}{V_x \cdot D_y - V_y \cdot D_x} \quad (8.106)$$

Es findet nur dann eine Kollision statt, wenn der Nenner der λ ungleich 0 ist.

Da die Bande nur ein Liniensegment darstellt, muss λ_1 zwischen 0 und 1 liegen. Ausserdem muss λ_2 grösser oder gleich 0 sein, da der Weg der Kugel eine Half-Line darstellt und ansonsten Schnittpunkte hinter der Kugel möglich wären.

Sofern die beiden λ diese Bedingungen einhalten, dann findet eine Kollision statt.

Nun soll der Ort $s(t)$ und die Zeit t der Kollision festgestellt werden. Da der Kugelradius berücksichtigt werden muss, werden die beiden Punkte R_1 und R_2 um den Kugelradius zur Tischmitte verschoben. Diese verschobenen Punkte seien R'_1 und R'_2 .

Mithilfe dieser Punkte kann eine Linie definiert werden:

$$\Delta \vec{R}' = \vec{R}'_2 - \vec{R}'_1 \quad (8.107)$$

$$s(\vec{\lambda}_1) = \vec{R}'_1 + \lambda_1 \cdot \Delta \vec{R}' \quad (8.108)$$

Der Schnittpunkt dieser Linie mit der in 8.99 definierten Linie ist der Kollisionspunkt $s(t)$ der Kugel. Angewandt auf die in Anhang 8.9 definierte Formel gelten folgende Variablensubstitutionen:

$$P = \vec{R}'_1 \quad (8.109)$$

$$D = \Delta \vec{R}' \quad (8.110)$$

$$Q = \vec{C} \quad (8.111)$$

$$V = \vec{v} \quad (8.112)$$

$$(8.113)$$

Damit gilt dann:

$$\lambda_1 = \frac{V_x \cdot P_y - V_y \cdot P_x + Q_x \cdot V_y - Q_y \cdot V_x}{D_x \cdot V_y - D_y \cdot V_x} \quad (8.114)$$

$$\lambda_2 = \frac{P_x \cdot D_y - P_y \cdot D_x - Q_x \cdot D_y + Q_y \cdot D_x}{V_x \cdot D_y - V_y \cdot D_x} \quad (8.115)$$

Der Ort der Kollision $s(t)$ kann nun mit λ_1 berechnet werden.

Der Zeitpunkt t der Kollision wird mit der Formel der gleichmässig beschleunigten Bewegung berechnet.

$$s(\vec{t}) = \frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v} \cdot t + \vec{s}_0 \quad (8.116)$$

Die Beschleunigung \vec{a} , Geschwindigkeit \vec{v} , Startposition \vec{s}_0 zum Zeitpunkt t_0 und die Position \vec{s}_1 zum Zeitpunkt t_1 sind bekannt.

Da die Startposition der Kugel zum Zeitpunkt t_0 bekannt ist, kann die Formel auch mit der zurückzulegenden Strecke $\Delta \vec{s}$ definiert werden:

$$t = t_1 - t_0 \quad (8.117)$$

$$\Delta \vec{s} = \vec{s}_1 - \vec{s}_0 \quad (8.118)$$

$$\Delta \vec{s} = \frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v} \cdot t + \vec{s}_0 - \vec{s}_0 \quad (8.119)$$

$$\Delta \vec{s} = \frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v} \cdot t \quad (8.120)$$

Das zweidimensionale Problem kann auf eine Dimension reduziert werden, indem die Beträge der Vektoren verwendet werden, weil die Vektoren $\Delta \vec{s}$, \vec{a} und \vec{v} kollinear sind:

$$\Delta s = |\Delta \vec{s}| \quad (8.121)$$

$$a = -|\vec{a}| \quad (8.122)$$

$$v = |\vec{v}| \quad (8.123)$$

$$|\Delta \vec{s}| = \frac{1}{2} \cdot (-|\vec{a}|) \cdot t^2 + |\vec{v}| \cdot t \quad (8.124)$$

$$\Delta s = \frac{1}{2} \cdot a \cdot t^2 + v \cdot t \quad (8.125)$$

Umgeformt zu einer quadratischen Gleichung folgt:

$$0 = \frac{1}{2} \cdot a \cdot t^2 + v \cdot t - \Delta s \quad (8.126)$$

Diese Gleichung kann nun mit der allgemeinen Lösungsformel für quadratische Gleichungen nach t gelöst werden [Unk21g]:

$$t_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (8.127)$$

Diese beiden t_1 und t_2 sind nicht zu verwechseln mit den gleichnamigen Variablen zuvor, es handelt sich um Lösungen für t .

Wobei für die Koeffizienten gilt:

$$a = \frac{1}{2} \cdot a \quad (8.128)$$

$$b = v \quad (8.129)$$

$$c = -\Delta s \quad (8.130)$$

$$(8.131)$$

Die Lösungen lauten daher:

$$\Delta t_1 = \frac{-v + \sqrt{v^2 + 2 \cdot a \cdot \Delta s}}{a} \quad (8.132)$$

$$\Delta t_2 = \frac{-v - \sqrt{v^2 + 2 \cdot a \cdot \Delta s}}{a} \quad (8.133)$$

Die minimale Zeit repräsentiert den Ereigniszeitpunkt, weswegen nur diese verwendet wird.

$$t = \min(t_1, t_2) \quad (8.134)$$

Da es sich bei t nur um die vergangene Zeit von dem Startzeitpunkt t_0 bis zum Kollisionszeitpunkt t_1 handelt, muss der Startzeitpunkt wieder aufaddiert werden, sofern der Absolute Zeitpunkt relevant ist:

$$t_1 = t_0 + t \quad (8.135)$$

8.6 Herleitung Ereignis - Kollision mit Ziel

Um den Kollisionszeitpunkt zu bestimmen, wird in einem ersten Schritt der Schnittpunkt des Weges der Kugel mit dem Zielkreis berechnet. Dazu wird die Kreisgleichung wie in 8.137 verwendet.

$$|X - C|^2 = r^2 \quad (8.136)$$

$$(x - C_x)^2 + (y - C_y)^2 = r^2 \quad (8.137)$$

Hierbei gelten die Definitionen aus ??.

$$\hat{d} = \frac{\vec{d}}{|\vec{d}|} \quad (8.138)$$

Die Parameterform der Geradengleichung ist gegeben in 8.140.

$$\vec{d} = \vec{p} + l \cdot \hat{v} \quad (8.139)$$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} + l \cdot \begin{pmatrix} v_x \\ v_y \end{pmatrix} \quad (8.140)$$

Durch Einsetzen der Gleichung 8.140 in 8.137 folgt 8.141.

$$(p_x + l \cdot v_x - C_x)^2 + (p_y + l \cdot v_y - C_y)^2 = r^2 \quad (8.141)$$

Im nächsten Schritt wird die Gleichung 8.141 nach t umgeformt.

$$(p_x + l \cdot v_x - C_x)^2 + (p_y + l \cdot v_y - C_y)^2 = r^2 \quad (8.142)$$

$$|\vec{p} + l \cdot \hat{v} - \vec{C}|^2 = r^2 \quad (8.143)$$

$$|l \cdot \hat{v} + \vec{p} - \vec{C}|^2 = r^2 \quad (8.144)$$

$$|l \cdot \hat{v} + (\vec{p} - \vec{C})|^2 = r^2 \quad (8.145)$$

$$(l \cdot \hat{v} + (\vec{p} - \vec{C})) \cdot (l \cdot \hat{v} + (\vec{p} - \vec{C})) = r^2 \quad (8.146)$$

$$l^2 \cdot (\hat{v} \cdot \hat{v}) + 2l \cdot (\hat{v}(\vec{p} - \vec{C})) + (\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) = r^2 \quad (8.147)$$

$$l^2 \cdot (\hat{v} \cdot \hat{v}) + 2l \cdot (\hat{v}(\vec{p} - \vec{C})) + (\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) - r^2 = 0 \quad (8.148)$$

Die Gleichung 8.141 ist von einer quadratischen Form und kann mithilfe der Mitternachtsformel gelöst werden.

$$a \cdot l^2 + b \cdot l + c = 0 \quad (8.149)$$

$$a = \hat{v} \cdot \hat{v} = |\hat{v}|^2 = 1 \quad (8.150)$$

$$b = 2 \cdot \hat{v} \cdot (\vec{p} - \vec{C}) \quad (8.151)$$

$$c = (\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) - r^2 \quad (8.152)$$

Die Lösungen lauten demnach:

$$l_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (8.153)$$

$$l_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (8.154)$$

Wobei $b^2 - 4ac$ die Diskriminante ist. Hat diese einen negativen Wert, existiert kein Schnittpunkt, beim Wert 0 gibt es einen Schnittpunkt und bei einem Wert grösser als 0 gibt es zwei Schnittpunkte.

Die Lösung lautet demnach:

$$l_1 = \frac{-(2 \cdot \hat{v} \cdot (\vec{p} - \vec{C})) + \sqrt{4 \cdot ((\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) - r^2)}}{2} \quad (8.155)$$

$$l_2 = \frac{-(2 \cdot \hat{v} \cdot (\vec{p} - \vec{C})) - \sqrt{4 \cdot ((\vec{p} - \vec{C}) \cdot (\vec{p} - \vec{C}) - r^2)}}{2} \quad (8.156)$$

$$l = \max(\min(l_1, l_2), 0) \quad (8.157)$$

Ein l ist valid, wenn es positiv ist und nicht innerhalb des Ziellochs $(-l_1, +l_2)$ liegt. Dieser Fall wird nicht vorkommen, da die Kugel bei der ersten Kollision aus dem System entfernt wird. Sollte ein valides l existieren, kann damit die Distanz d zwischen der Kugel und dem Schnittpunkt bestimmt werden.

$$d(\vec{l}) = l \cdot \hat{v} \quad (8.158)$$

Anhand der Distanz aus 8.158 kann der Zeitpunkt durch die Gleichung 8.160 bestimmt werden, wobei gilt $s(\vec{t}) = d(\vec{l})$

$$s(\vec{t}) = \frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v} \cdot t \quad (8.159)$$

$$\frac{1}{2} \cdot \vec{a} \cdot t^2 + \vec{v} \cdot t - d(\vec{l}) = 0 \quad (8.160)$$

Da es sich um die Berechnung der benötigten Zeit von der Kugel zum Ziel handelt und keine weiteren Faktoren relevant sind, kann die Berechnung der Zeit entweder auf eine Dimension reduziert werden oder die X- wie auch Y-Dimensionen werden separat berechnet. Dieselbe Sachlage wurde bereits in Kapitel 8.3 erläutert. Da es sich hier um eine quadratische Gleichung handelt, wäre hingegen der Mehraufwand der separaten Betrachtung der X- und Y-Dimensionen höher als die Reduktion auf eine Dimension durch die Betragsnahme der Vektoren, weswegen der zweitgenannte Weg verfolgt wird. Demnach ergibt sich die Situation in 8.162.

$$a \cdot t^2 + b \cdot t + c = 0 \quad (8.161)$$

$$a = \frac{1}{2} \cdot -|\vec{a}| \quad (8.162)$$

$$b = |\vec{v}| \quad (8.163)$$

$$c = -|d(\vec{l})| \quad (8.164)$$

Es ergeben sich wiederum zwei Lösungen, wobei nur die der erste Schnittpunkte (die Minimalzeit) relevant ist.

$$t_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (8.165)$$

$$t_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (8.166)$$

$$t = \min(t_1, t_2) \quad (8.167)$$

Sollte die Diskriminante kleiner denn 0 sein, bedeutet dies, dass die Geschwindigkeit der Kugel nicht ausreichend ist, um den Zielkreis zu erreichen. Es existiert demnach keine Lösung. Die Lösung lautet demnach:

$$t_1 = \frac{-|\vec{v}| + \sqrt{|\vec{v}|^2 + 2 \cdot -|\vec{a}||d(\vec{l})|}}{-|\vec{a}|} \quad (8.168)$$

$$t_2 = \frac{-|\vec{v}| - \sqrt{|\vec{v}|^2 + 2 \cdot -|\vec{a}||d(\vec{l})|}}{-|\vec{a}|} \quad (8.169)$$

$$t = \min(t_1, t_2) \quad (8.170)$$

8.7 Herleitung Beschleunigung

Die Beschleunigung a einer Kugel, welche durch die Rollreibung auf dem Tisch entsteht, kann über die wirkenden Kräfte ausgedrückt werden. Die Länge des Vektors \vec{a} kann folgendermassen ausgedrückt werden [Unk21h], wenn die Reibungskraft F_R sowie die Normalkraft F_N bekannt ist:

$$F_R = m \cdot a \quad (8.171)$$

$$a = \frac{F_R}{m} \quad (8.172)$$

$$F_N = m \cdot g \quad (8.173)$$

$$F_R = F_N \cdot c_R \quad (8.174)$$

$$a = g \cdot c_R \quad (8.175)$$

Die Richtung des Vektors \vec{a} zeigt in die entgegengesetzte Richtung von \vec{v} . Dazu wird \vec{v} normiert. Damit kann \vec{a} bestimmt werden:

$$\vec{a} = g \cdot c_R \cdot \frac{\vec{v}}{|\vec{v}|} \quad (8.176)$$

8.8 Bestimmung Reibungskoeffizient

Um den Reibungskoeffizienten c_R zu bestimmen, wird ein Versuchsaufbau in Form einer Rampe benötigt, wobei eine theoretische Reibung auf einer perfekt glatten Oberfläche der Rampe von 0 angenommen wird.

In Abbildung 8.3 werden die wirkenden Kräfte und ihr Einfluss veranschaulicht. Es gelten die nachfolgenden Beziehungen.

$$g = 9.81 \frac{m}{s^2} \quad (8.177)$$

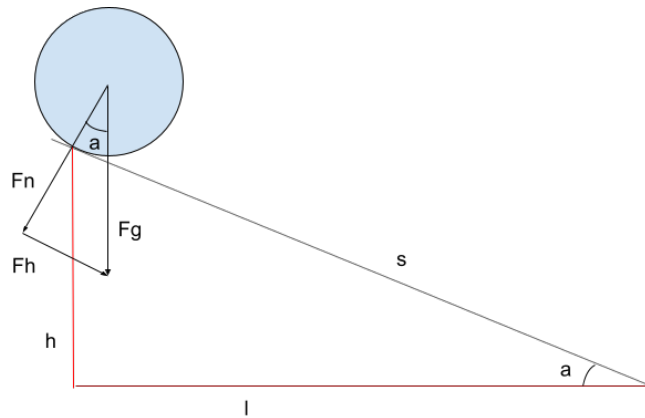


Abbildung 8.3: Modell zur Berechnung der Startgeschwindigkeit auf dem Tisch

$$\alpha = \arctan\left(\frac{h}{l}\right) \quad (8.178)$$

$$F_G = m \cdot g \quad (8.179)$$

$$F_N = \cos(\alpha) \cdot F_G \quad (8.180)$$

$$F_H = \sin(\alpha) \cdot F_G \quad (8.181)$$

$$s = \sqrt{l^2 + h^2} \quad (8.182)$$

$$(8.183)$$

Während die Kugel die Rampe herunterrollt, wird die Formel der gleichmässig beschleunigten Bewegung verwendet. Die Kugel wird auf der Rampe platziert und losgelassen. Das bedeutet, dass sie keine Initialgeschwindigkeit hat. Relevant ist am Ende die zurückgelegte Strecke wie auch die benötigte Zeit.

$$s(t) = s = \frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t + s_0 \quad (8.184)$$

$$v_0 = 0 \quad (8.185)$$

$$s_0 = 0 \quad (8.186)$$

$$s(t) = s = \frac{1}{2} \cdot a \cdot t^2 \quad (8.187)$$

$$F_H = m \cdot a \quad (8.188)$$

$$a = \frac{F_H}{m} \quad (8.189)$$

$$\frac{1}{2} \cdot \frac{F_H}{m} \cdot t^2 - s = 0 \quad (8.190)$$

Der Zeitpunkt, wo die Kugel das Ende der Rampe erreicht, kann mithilfe der allgemeinen Lösungsformel für quadratische Gleichungen nach t gelöst werden [Unk21g]:

$$t_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (8.191)$$

$$a = \frac{1}{2} \cdot \frac{F_H}{m} \quad (8.192)$$

$$b = 0 \quad (8.193)$$

$$c = -s \quad (8.194)$$

$$t_{1,2} = \frac{\pm \sqrt{-4ac}}{2a} \quad (8.195)$$

$$(8.196)$$

Es wird nur die positive Lösung in Betracht gezogen.

Die Geschwindigkeit der Kugel am Ende der Rampe ist somit:

$$v(t) = a \cdot t + v_0 \quad (8.197)$$

$$v_0 = 0 \quad (8.198)$$

$$v(t) = \frac{F_H}{m} \cdot t \quad (8.199)$$

Ab diesem Zeitpunkt wird angenommen, dass die Kugel auf dem Tisch rollt und nur noch von der Rollreibung zwischen Kugel und Tisch abgebremst wird. Es muss die von der Kugel bis zum Stillstand zurückgelegte Strecke s gemessen werden. Weiterhin gelten die Beziehungen in Abbildung 8.4.



Abbildung 8.4: Modell zur Berechnung des Reibungskoeffizienten

Anschließend kann die Beschleunigung berechnet werden:

$$s(t) = s = \frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t + s_0 \quad (8.200)$$

$$s_0 = 0 \quad (8.201)$$

$$v_0 = v(t_R) \quad (8.202)$$

$$\frac{1}{2} \cdot a \cdot t^2 + v_0 \cdot t - s = 0 \quad (8.203)$$

$$v(t) = 0 = a \cdot t + v_0 \quad (8.204)$$

$$a \cdot t = -v_0 \quad (8.205)$$

$$t = \frac{-v_0}{a} \quad (8.206)$$

$$\frac{1}{2} \cdot a \cdot \left(\frac{-v_0}{a}\right)^2 + v_0 \cdot \frac{-v_0}{a} - s = 0 \quad (8.207)$$

$$\frac{1}{2} \cdot a \cdot \frac{(-v_0)^2}{a^2} + v_0 \cdot \frac{-v_0}{a} - s = 0 \quad (8.208)$$

$$\frac{1}{2} \cdot \frac{(-v_0)^2}{a} + v_0 \cdot \frac{-v_0}{a} - s = 0 \quad (8.209)$$

$$\frac{1}{2} \cdot (-v_0)^2 \cdot \frac{1}{a} + v_0 \cdot (-v_0) \cdot \frac{1}{a} - s = 0 \quad (8.210)$$

$$\frac{1}{a} \cdot \left(\frac{1}{2} \cdot (-v_0)^2 + v_0 \cdot (-v_0)\right) - s = 0 \quad (8.211)$$

$$\frac{1}{2} \cdot (-v_0)^2 + v_0 \cdot (-v_0) = s \cdot a \quad (8.212)$$

$$a = \frac{\frac{1}{2} \cdot (-v_0)^2 + v_0 \cdot (-v_0)}{s} \quad (8.213)$$

Mit der Beschleunigung kann nun der Rollwiderstandskoeffizient berechnet werden. Da die Kugel auf dem Tisch liegt, ist die Normalkraft F_N gleich der Gewichtskraft F_G .

$$F_R = c_R \cdot F_N \quad (8.214)$$

$$F_R = m \cdot a \quad (8.215)$$

$$F_N = F_G \quad (8.216)$$

$$F_G = m \cdot g \quad (8.217)$$

$$m \cdot a = c_R \cdot m \cdot g \quad (8.218)$$

$$c_R = \frac{m \cdot a}{m \cdot g} \quad (8.219)$$

$$c_R = \frac{a}{g} \quad (8.220)$$

8.9 Herleitung Linie-Linie-Schnittpunkt

Der Schnittpunkt zweier Linien ist zu finden. Die Linien L_1 und L_2 seien anhand parametrischer Gleichungen wie folgt definiert:

$$L_1(\lambda_1) : P + \lambda_1 \cdot D \quad (8.221)$$

$$L_2(\lambda_2) : Q + \lambda_2 \cdot V \quad (8.222)$$

Wobei P und Q Punkte auf den Linien und D und V Richtungsvektoren der Linien sind.

Der Schnittpunkt kann bestimmt werden, indem beide Gleichungen gleichgesetzt werden:

$$L_1(\lambda_1) = L_2(\lambda_2) \quad (8.223)$$

$$P + \lambda_1 \cdot D = Q + \lambda_2 \cdot V \quad (8.224)$$

Komponentenweise formuliert folgt:

$$P_x + \lambda_1 \cdot D_x = Q_x + \lambda_2 \cdot V_x \quad (8.225)$$

$$P_y + \lambda_1 \cdot D_y = Q_y + \lambda_2 \cdot V_y \quad (8.226)$$

Nach Isolation einer Unbekannten auf eine Seite:

$$\lambda_2 = \frac{P_x + \lambda_1 \cdot D_x - Q_x}{V_x} \quad (8.227)$$

$$\lambda_2 = \frac{P_y + \lambda_1 \cdot D_y - Q_y}{V_y} \quad (8.228)$$

Nun können beide Gleichungen gleichgesetzt werden, um die Variable λ_2 zu entfernen:

$$\frac{P_x + \lambda_1 \cdot D_x - Q_x}{V_x} = \frac{P_y + \lambda_1 \cdot D_y - Q_y}{V_y} \quad (8.229)$$

Nun kann nach λ_1 umformuliert werden:

$$\frac{P_x + \lambda_1 \cdot D_x - Q_x}{V_x} = \frac{P_y + \lambda_1 \cdot D_y - Q_y}{V_y} \quad (8.230)$$

$$(P_x + \lambda_1 \cdot D_x - Q_x) \cdot V_y = (P_y + \lambda_1 \cdot D_y - Q_y) \cdot V_x \quad (8.231)$$

$$P_x \cdot V_y + \lambda_1 \cdot D_x \cdot V_y - Q_x \cdot V_y = P_y \cdot V_x + \lambda_1 \cdot D_y \cdot V_x - Q_y \cdot V_x \quad (8.232)$$

$$\lambda_1 \cdot D_x \cdot V_y - \lambda_1 \cdot D_y \cdot V_x = P_y \cdot V_x - P_x \cdot V_y - Q_y \cdot V_x + Q_x \cdot V_y \quad (8.233)$$

$$\lambda_1 \cdot (D_x \cdot V_y - D_y \cdot V_x) = P_y \cdot V_x - P_x \cdot V_y - Q_y \cdot V_x + Q_x \cdot V_y \quad (8.234)$$

$$\lambda_1 = \frac{P_y \cdot V_x - P_x \cdot V_y - Q_y \cdot V_x + Q_x \cdot V_y}{D_x \cdot V_y - D_y \cdot V_x} \quad (8.235)$$

$$\lambda_1 = \frac{V_x \cdot P_y - V_y \cdot P_x + Q_x \cdot V_y - Q_y \cdot V_x}{D_x \cdot V_y - D_y \cdot V_x} \quad (8.236)$$

Wenn der Nenner der Gleichung 8.236 null ist, dann gibt es keinen Schnittpunkt der beiden Linien und weitere Berechnungen können abgebrochen werden.

Die Variable λ_2 kann durch Einsetzen von λ_1 in Gleichung 8.227 oder 8.228 gefunden werden.

Die Lösung nach λ_2 kann auch analog λ_1 umgestellt werden:

$$\lambda_2 = \frac{P_x \cdot D_y - P_y \cdot D_x - Q_x \cdot D_y + Q_y \cdot D_x}{V_x \cdot D_y - V_y \cdot D_x} \quad (8.237)$$