Technische Realisierung einer Computer-Simulation für die Reflexion von Lichtstrahlen in einem zweidimensionalem Raum

Ben Weber

November 2021

Inhaltsverzeichnis

1 Einleitung

Die physikalischen Grundlagen für die Reflexion von Lichtstrahlen ist für das menschliche Gehirn einfach zu verstehen. So kann der Mensch die Bahn von reflektierten Lichtwellen an einfachen geometrischen Formen fast intuitiv vorhersagen. In einem komplexeren Umfeld mit einer erhöhten Anzahl an physikalisch relevanten Entitäten wird die Prognose der Gesamtheit aller stattfindenden Einwirkungen auf den Lichtweg jedoch für Menschen zunehmend schwierig zu prognostizieren und vor allem zeitaufwendig. Wenn auch der initiale Aufwand für die digitale Implementierung dieser physikalischen optischen Grundlagen der Reflexion größer sein mag, ist dies vor allem in erwähnten umfangreicheren Situationen praktikabel und ermöglicht schnelle Iterationen bezüglich der Veränderung des Resultats im Zusammenhang mit der Ausgangssituation.

Die vorliegende Arbeit beschäftigt sich mit der Frage, wie eine solche digitale Simulation umgesetzt werden kann. Hierbei wird nicht nur auf die Umsetzung der tatsächlichen Reflexion eingegangen, sondern auch auf den Aufbau einer solchen Computer-Simulation, sodass eine Erweiterung hinsichtlich weiterer strahlenoptischer Phänomene möglich ist.
Zur Vereinfachung wird von einer idealisierten, zweidimensionalen, evakuierten Umgebung ausgegangen, in der Lichtstrahlen nicht an Intensität verlieren und die Oberflächen von Hindernissen störungsfrei sind.

Die Quellenlage bezüglich der hier verwendeten Grundlagen der geome-

trischen Optik ist als sehr sicher zu bewerten. Der Aufbau und die technische Umsetzung der Simulation entstammen fast ausschließlich aus eigenen Überlegungen.

2 Aufbau

Im Folgenden wird der Aufbau der Simulation zunächst theoretisch und abstrahiert betrachtet. Folgende Notation soll jedoch eine Verknüpfung mit dem Quellcode vereinfachen: (`Name im Quellcode`)

2.1 Welt

Die Welt (`World`) ist der Grundstein für die Simulation. Zu diesem virtuellen zweidimensionalen Raum können sämtliche Entitäten hinzugefügt werden. Hier werden diese verwaltet, verarbeitet und zur Visualisierung ausgegeben.

2.2 Lichtstrahl

In der geometrischen Optik geht man bei einem Lichtstrahl (`Ray`) von einem Strahl aus, der sich von einem Ursprung A geradlinig in eine Richtung mit dem Winkel α ausbreitet. (vgl. Tipler 2015, S. 1041) mathematisch wird dieser als Halbgerade angesehen $\overrightarrow{AB}$. In der Simulation ist es sinnvoll lediglich den Startpunkt als Ortsvektor $\vec{O}$ (`origin`) innerhalb der Welt und den Winkel (`angle`), in den der Strahl ausgeht direkt zu speichern.

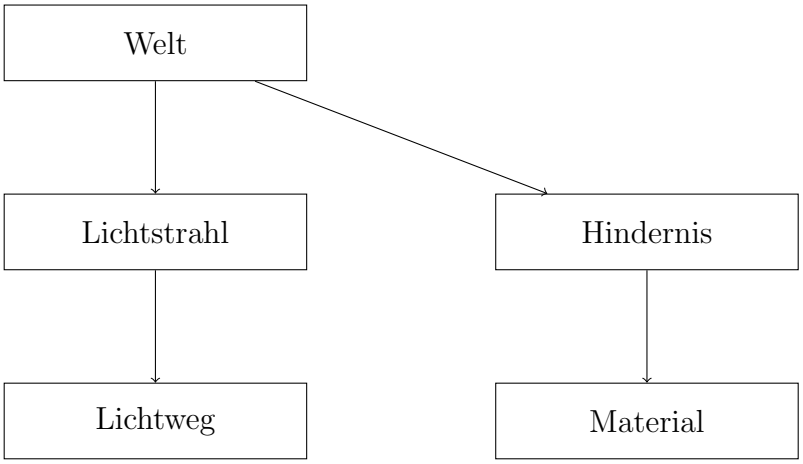

Abbildung 2.1: Schematischer Aufbau der Simulation
Quelle: Eigene Darstellung

2.3 Lichtweg

Es muss davon ausgegangen werden, dass ein Lichtstrahl in der die Kollision mit anderen Objekten seine Richtung verändert. Ab einer Richtungsänderung wird der bisherige Lichtstrahl als Lichtweg (`Line`) gespeichert, um in später zu visualisieren. Hierbei besteht ein Lichtstrahl aus den Ortsvektoren $\vec{O}$ (`origin`) dem Ursprung des Lichtstrahls und $\vec{E}$ (`end`) dem Punkt der Kollision.

2.4 Hindernis

Bei allen Entitäten in der Welt, die nicht Strahl oder Lichtweg sind, handelt es sich um Hindernisse. Dieser definieren sich durch ihre geometrische Form und Position in der Welt und können von Lichtstrahlen getroffen werden. Es sind verschiedene Variationen möglich. Jedes Hindernis (`Obstacle`) hat jedoch einen Startpunkt als Ortsvektor $\vec{A}$ (`start`) in der Welt. Und ein Material (`material`) (vgl. 2.5). Folgende Typen wurden für diese Arbeit ausgewählt,

eine Erweiterung ist jedoch möglich:

2.4.1 Kreis

Der Kreis (`type: 'cirlce'`) hat eine zusätzliche Angabe über den Radius r (`radius`).

2.4.2 Linie

Neben einen Anfangspunkt hat die Linie (`type: 'line'`) auch einen Ortsvektor $\vec{B}$ als Endpunkt.

2.4.3 Kurve

Zusätzlich kann ein Hindernis auch eine Kurve (`type: 'curve'`) sein. Das ermöglicht die Konstruktion verschiedener besonderer Reflektoren (z. B. Parabolischer Reflektor). Für den Verlauf der Kurve besitzt dieser Hindernis-Typ eine Funktion f. Um eine Kurve beliebig in der Welt zu positionieren, kann ihr zusätzlich eine Winkel γ (`rotation`) übergeben werden, der alle Punkte auf Kurve relative zum Ursprung rotiert.
Im Umfang dieser Arbeit wird bewusst auf eine vollständige und genaue Darstellung eines Graphen verzichtet. Daher werden nur Punkte in einem Interval $[-40, 40]$ im Abstand von einer übergebene Skalierung k (`scale`) errechnet, als Linien verbunden und als diese Fortlaufend verarbeitet. Diese Annäherung ist ausreichend, während gleichzeitig der Umfang stark reduziert wird.

2.5 Material

Die Existenz eines Materials (`Material`) in der Welt besteht nur in der Verknüpfung zu einem Hindernis. Ein Material ist für die Verarbeitung eines einfallenden, mit dem Hindernis kollidierenden, Lichtstrahls verantwortlich und kann eine beliebig Anzahl an neuen Lichtstrahlen zurückgeben.
In dieser Arbeit wird nur die Reflexion bei dem Material Spiegel (`mirror`) behandelt. Eine Trennung von Material und Hindernis ermöglicht jedoch eine einfache Erweiterung. Dabei können trotz Veränderung der strahlenoptischer Funktion, die geometrischen Eigenschaften beibehalten werden. So könnte beispielsweise ein Kreis einerseits als Spiegel die Lichtstrahlen reflektieren und anderseits als Glas die Lichtstrahlen brechen. Ohne das dafür die Veränderung des Hindernisses von Nöten wäre.

3 Technische Voraussetzungen

Um die Zugänglichkeit der Simulation zu erhöhen, wird diese als Web-App programmiert. Dazu wird die Programmiersprache TypeScript benutzt und die Applikation wird mit dem Framework „SvelteKit“[1] kompiliert. Zur Visualisierung wird die Library „Pts“[2] verwendet. Diese ermöglicht eine einfache Verwendung der Canvas-API. Zusätzlich werden aber auch einige mathematisch Hilfsfunktionen von dieser Library verwendet.

Der gesamte Quellcode für die Simulation ist auf GitHub aufzufinden[3]. Im Folgenden werden nur kleine, unvollständige Ausschnitte aus dem Code eingebunden.

[1] https://kit.svelte.dev

[2] https://ptsjs.org

[3] https://github.com/b3ngg/ray-optics-simulation

4 Simulationsprozess

Im Folgenden wird nun der Ablauf der Simulation vom erstellen der Szene bis hin zum Visualisieren des Resultats beschrieben.

4.1 Erstellen der Welt

Zunächst muss für die Simulation eine Umgebung errichtet werden. Dazu wird eine Welt erstellt. Zu dieser werden in dieser Szene ein von links oben (man beachte, dass das Koordinatensystem den Ursprung links oben hat und die y-Achse umgekehrt zum normalen Koordinatensystem ist) beginnender und diagonal nach rechts unten strahlender Lichtstrahl hinzugefügt.
Zudem wird auch ein Hindernis in der Form einer vertikalen Linie in der Mitte der Szene hinzugefügt. Diese hat das Material Spiegel (`mirror`).
Nun wird mit `world.update()` die Simulation aktualisiert und die Lichtwege berechnet. Dieser werden zum Schluss visualisiert.

Bei den Objekt `space` handelt es sich um ein von Pts bereitgestellte und zur Visualisierung benötigte Objekt. Die Klasse `Pt` ist ebenfalls Teil von Pts und wird zur Vektormathematik verwendet.

```
/**
 * Simple reflection of a ray on a vertical line
 */
export const lineReflection: Scene = (space) => {
    const form = space.getForm();
    const world = createWorld();

    world.addSource('ray', createRay(new Pt(), Math.PI * 2.25));
    world.addObstacle(
        'line',
        createLine(new Pt(800, 0), { end: new Pt(800, 5000), material: mirror })
    );

    world.update();
    space.add(() => {
        world.draw(form);
    });

    space.playOnce();
};
```

5 Fazit

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum

Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache

gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Test (vgl. Hering 2017, S. 15)

Literatur

Hering, Ekbert (2017). *Optik für Ingenieure und Naturwissenschaftler Grundlagen und Anwendungen*. München: Fachbuchverlag Leipzig im Carl Hanser Verlag. ISBN: 978-3-446-44281-8.

Tipler, Paul (2015). *Physik für Wissenschaftler und Ingenieure [der Begleiter bis zum Bachelor*. BerlinHeidelberg: Springer Spektrum. ISBN: 978-3-642-54165-0.