



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Bachelorarbeit

Interaktive Exploration gravitativer Orbits

Lucas Jillek

Matrikelnummer: 6070647

E-Mail: incanus@uni-paderborn.de

Paderborn, den 01.09.2009

vorgelegt bei: Dr. rer. nat. Martin Ziegler

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen oder anderen Quellen entnommen sind, sind als solche kenntlich gemacht.

Ort, Datum, Unterschrift

1	Einleitung.....	1
2	Grundlagen.....	2
2.1	Gravitation.....	2
2.2	Massenträgheit	3
2.3	Unterschiedliche Flugmodelle in Computerspielen.....	6
2.3.1	Lunar Lander	6
2.3.2	Asteroids	7
2.3.3	Elite.....	7
2.3.4	Wing Commander.....	8
2.3.5	Star Wars: X-Wing	9
3	Umsetzung	10
3.1	JLI Asteroids	10
3.2	MinSG.....	11
3.3	Config-Datei.....	12
3.4	Berechnung von Gravitation und Trägheit.....	13
3.5	Berechnung der Energie-Steuerung	15
4	Steuerungsarten	17
4.1	Steuerungsmodelle in 2D	17
4.1.1	„absolute Steuerung“	17
4.1.2	„direkte Steuerung“	18
4.1.3	„Fahrradsteuerung“	19
4.1.4	„realistische Steuerung“.....	19
4.1.5	Energie-Steuerung	20
4.2	Steuerungsmodelle in 3D	21
4.2.1	Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit.....	22
4.2.2	Steuerung ohne Geschwindigkeitsabstufungen, aber mit Trägheit	22
4.2.3	Steuerung mit Geschwindigkeitsabstufungen, aber ohne Trägheit	22
4.2.4	Steuerung mit Geschwindigkeitsabstufungen und mit Trägheit.....	23
5	Entwicklung des Fragebogens	24
5.1	Unterschiedliche Testszenarien.....	24
5.2	Subjektiver Abschnitt des Fragebogens	24
5.2.1	Vorkenntnisse	24
5.2.2	Einarbeitungszeit	25
5.2.3	Schwierigkeitsgrade der einzelnen Steuerungsmodelle.....	25
5.2.4	Besondere Schwierigkeiten beim Steuern	26

5.2.5	Realismus der einzelnen Steuerungsmodelle.....	26
5.3	Objektiver Abschnitt des Fragebogens	27
6	Auswertung der Fragebögen	29
6.1	Probleme bei den ersten Tests.....	29
6.2	Auswertung der Ergebnisse des subjektiven Testanteils	30
6.2.1	Erwartete Ergebnisse	30
6.2.2	Tatsächliche Ergebnisse.....	31
6.3	Auswertung der Testergebnisse der objektiven Tests.....	34
6.3.1	Erwartete Ergebnisse	34
6.3.2	Tatsächliche Ergebnisse.....	34
7	Fazit und Ausblick.....	37
8	Literaturverzeichnis	38
9	Anhang.....	40
9.1	Steuerung der potentiellen Energie separiert von der Navigation auf der Energiefläche.....	40
9.2	Fragebogen:.....	42
9.3	Datenträger:.....	43

1 Einleitung

Das Ziel dieser Bachelorarbeit ist es, unterschiedliche Steuerungsmodelle für ein Raumschiff im interplanetaren Raum zu entwickeln, umzusetzen und miteinander zu vergleichen. Die Simulation dieser Steuerungsmodelle soll gravitative Punkte berücksichtigen, welche die Flugbahn des Raumschiffs beeinflussen. Die verschiedenen Steuerungsmodelle werden auf unterschiedliche Kriterien wie Schwierigkeit, Realismus und Lernfähigkeit anhand eines Fragebogens und Testdurchläufen untersucht.

Um die verschiedenen Steuerungen zu entwickeln, wurden einzelne Steuerungsmodelle von unterschiedlichen Weltraumsimulationen und ähnlichen Computerspielen analysiert und teilweise umgesetzt und erweitert. Als Vorlage für die einzelnen Steuerungsmodelle dienten folgende Computerspiele:

- 2D-Spiele:
 - *Lunar Lander*
 - *Asteroids*
- 3D-Spiele
 - *Elite*
 - *Wing Commander*
 - *Star Wars – X-Wing*

Dabei stellte sich heraus, dass nur wenige der kommerziellen Weltraumspiele die physikalischen Prinzipien der Gravitation und der Trägheit umsetzen. Für diese Arbeit wurden die oben genannten Steuerungsmodelle um das Prinzip der Gravitation erweitert. Die Trägheit hingegen wurde nur bei einigen Steuerungsmodellen berücksichtigt. Dabei sollte untersucht werden, ob es größere Unterschiede bezüglich des Schwierigkeitsgrads bei der Handhabung von den einzelnen Steuerungsmodellen gibt. Neben den unterschiedlichen Schwierigkeitsgraden sollten auch die Kriterien Realismus und Lernfähigkeit anhand eines Fragebogens erörtert werden.

In Kapitel 2 werden kurz die Prinzipien der Gravitation und der Trägheit beschrieben, gefolgt von einer Vorstellung der einzelnen untersuchten Computerspiele. Kapitel 3 beschäftigt sich mit der Umsetzung der 2D- und 3D-Umgebung, sowie der Implementierung der Steuerungen. Im Anschluss daran befasst sich Kapitel 4 mit den umgesetzten Steuerungsmodellen im Einzelnen und zeigt die Unterschiede der verschiedenen Modelle auf. In Kapitel 5 wird die Entwicklung des Fragebogens beschrieben. Dieser dient zur Bewertung der Kriterien Schwierigkeitsgrad, Realismus und Lernfähigkeit der unterschiedlichen Steuerungen. Anschließend werden in Kapitel 6 die Ergebnisse vorgestellt, die sich bei der Auswertung der Fragebögen ergaben. Kapitel 7 zeigt ein kurzes Fazit, einen Ausblick sowie einige Anregungen zur eventuellen weiteren Entwicklung der einzelnen Steuerungen und Simulationen.

2 Grundlagen

In diesem Kapitel beschreibe ich die physikalischen Gesetze der Gravitation und Trägheit. Im Anschluss daran untersuche ich die unterschiedlichen Steuerungsmodelle von verschiedenen bekannten Weltraumsimulationen und ähnlichen Computerspielen.

2.1 Gravitation

Bei der Gravitation, vom lateinischen *gravitas* – Schwere, handelt es sich um eine der vier Grundkräfte der Physik. Diese sind:

- starke Wechselwirkung
- elektromagnetische Wechselwirkung
- schwache Wechselwirkung
- Gravitation.

An dieser Stelle ist aber nur die Gravitation von Relevanz. Diese bezeichnet die Anziehungskraft zwischen unterschiedlichen Massen. So sorgt die Gravitation etwa dafür, dass Gegenstände auf den Boden fallen. Sie werden von der Erde angezogen. Genau betrachtet zieht der Gegenstand die Erde auch an sich heran, durch den riesigen Massenunterschied ist das aber zu vernachlässigen.

Des Weiteren ist die Gravitation für die Umlaufbahnen der Planeten um die Sonne und für die Umlaufbahnen von Monden und Satelliten um Planeten verantwortlich. Dieser Aspekt ist für diese Ausarbeitung der wichtigste Punkt.

Der deutsche Astronom und Naturphilosoph Johannes Kepler formulierte die Geometrie und Kinematik der Planetenlaufbahnen in drei Gesetzen, welche Keplersche Gesetze genannt werden. Diese drei Gesetze sind:

1. *Die Planeten bewegen sich auf Ellipsen, in deren gemeinsamem Brennpunkt die Sonne steht.*
2. *Die Verbindungsgerade Sonne-Planet überstreicht in gleichen Zeiten gleiche Flächen.*
3. *Die Quadrate der Umlaufzeiten der Planeten verhalten sich wie die Kuben der großen Halbachsen ihrer Bahnen um die Sonne, also $r^3 / T^2 = \text{konstant}$ [Kuc96].*

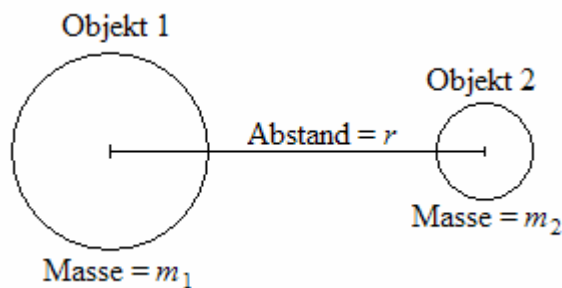
Diese Gesetze können als exakte Lösung für ein System, welches aus zwei kugelsymmetrischen Himmelskörpern besteht, angesehen werden. Sind in dem System noch weitere Massenkörper enthalten, so wirken sich deren Anziehungskräfte auf die Umlaufbahnen aus, so dass keine exakten Ellipsen entstehen. Es müssen allerdings noch folgende beiden Einschränkungen getroffen werden:

- nichtgravitative Kräfte sind zu vernachlässigen
- es wird angenommen, dass sich die Gravitationswirkung unendlich schnell ausbreitet

Isaac Newton stellte die drei Keplerschen Gesetze später in den allgemeinen Zusammenhang mit seinem Gravitationsgesetz, welches er 1686 in seinem Werk *Philosophiae Naturalis Principia Mathematica* wie folgt formulierte:

„Jede Masse, genauer jeder Massenpunkt, zieht jeden anderen Massenpunkt mittels einer Kraft an, die entlang der Verbindungslinie gerichtet ist. Der Betrag dieser Gravitationskraft ist proportional zum Produkt der beiden Massen und umgekehrt proportional zum Quadrat des Abstandes der beiden Massen.“ [New88]

Die Gravitationskraft F eines Objektes mit der Masse m_1 auf ein zweites Objekt mit der Masse m_2 lässt sich wie folgt berechnen:



$$F = G * \frac{(m_1 * m_2)}{r^2}$$

Abb. 2.1 Gravitation zwischen 2 Objekten unterschiedlicher Masse

Der Faktor G in der Formel in Abb. 2.1 stellt die von Isaac Newton als Naturkonstante eingeführte Gravitationskonstante dar. Diese hat im internationalen Einheitensystem nach dem *Committee on Data for Science and Technology* (CODATA) 2006 einen Wert von [WFI09]:

$$G = (6,67428 \pm 0,00067) * 10^{-11} \frac{m^3}{kg * s^2}$$

Abb. 2.2 Gravitationskonstante G

2.2 Massenträgheit

Das Prinzip der Massenträgheit wurde ebenfalls von Isaac Newton definiert. Eigentlich wurde es von Newton nur neu definiert und korrigiert. Schon in der Antike wurde von Aristoteles die These aufgestellt, dass für die Aufrechterhaltung einer Bewegung immer eine entsprechende Kraft aufgebracht werden müsse. Newton hingegen stellte folgendes Axiom auf:

„Ein Körper verharrt in seinem Zustand der Ruhe oder der gleichförmigen geradlinigen Bewegung, solange die Summe aller auf ihn einwirkenden Kräfte Null ist.“ [New88]

Dieses Axiom besagt, dass ein Gegenstand, solange keinerlei Kraft auf ihn wirkt, seine momentane Bewegung immer weiter ausführen wird. Diese Bewegung kann auch darin bestehen, bewegungslos an einer Stelle zu liegen. So würde sich aber ein Gegenstand, der sich mit einer gleichmäßigen Geschwindigkeit in eine Richtung bewegt, auch solange weiter in diese Richtung bewegen, bis irgendwann eine andere Kraft auf ihn wirkt und somit seine Bewegung ändert. Unter idealen Umständen und auf einem reibungsfreien Untergrund könnte man eine Kugel anstoßen und diese würde endlos weiter rollen und niemals zum Stillstand kommen.

Im Weltraum hingegen fällt die Reibung, der Gegenstände auf der Erde aufgrund der Atmosphäre ausgesetzt sind, weg. Ein Gegenstand würde dort immer weiter in der entsprechenden Bewegung verbleiben. Es mag zwar keine Reibung im Weltraum geben, aber dennoch wirken auch dort unterschiedliche Kräfte auf den Gegenstand, so dass seine Bewegung durch diese immer wieder verändert wird. So wirken die Gravitationskräfte unterschiedlicher Sterne und Planeten etwa auf den Gegenstand ein und verändern damit seine Flugbahn.

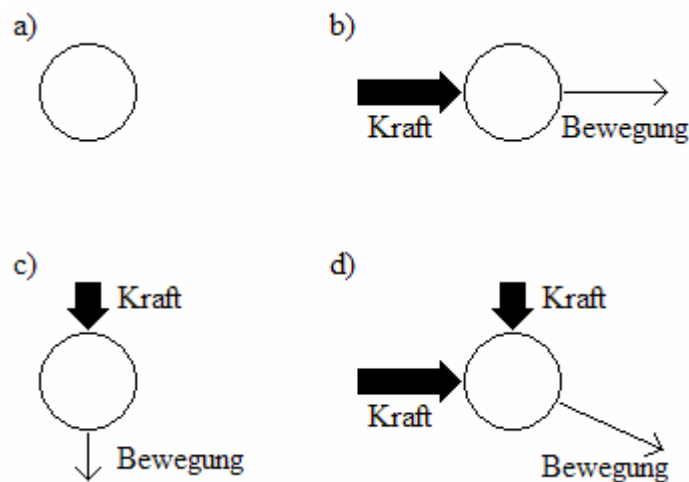


Abb. 2.3 Auswirkungen unterschiedlicher Kräfte auf einen Gegenstand(1)

Die Abbildung 2.3 zeigt in a) einen Gegenstand, auf den keinerlei Kräfte einwirken und der sich in Ruhe befindet.

In Abbildung b) wird auf den ruhenden Gegenstand eine Kraft von links eingewirkt, so dass der Gegenstand sich mit einer der Kraft entsprechenden Geschwindigkeit nach rechts bewegt. Diese Geschwindigkeit lässt sich anhand folgender Formeln sehr einfach berechnen:

$$\text{Kraft} = \text{Masse} \cdot \text{Beschleunigung}$$

$$\text{Geschwindigkeit} = \text{Beschleunigung} \cdot \text{Beschleunigungsdauer}$$

Abbildung c) zeigt wie auf den ruhenden Gegenstand aus a) eine Kraft von oben einwirkt. Daraus resultiert eine Beschleunigung nach unten. Die Kraft ist allerdings deutlich kleiner als die Kraft, die in b) auf den Gegenstand gewirkt wird. Somit ist auch die Beschleunigung deutlich geringer als die des Gegenstandes in b).

In Abbildung d) hingegen wirken sowohl die Kraft aus b) als auch die Kraft aus c) auf den ruhenden Gegenstand. Somit werden nun beide Bewegungen aufgenommen und der Gegenstand bewegt sich nach rechts und etwas nach unten.

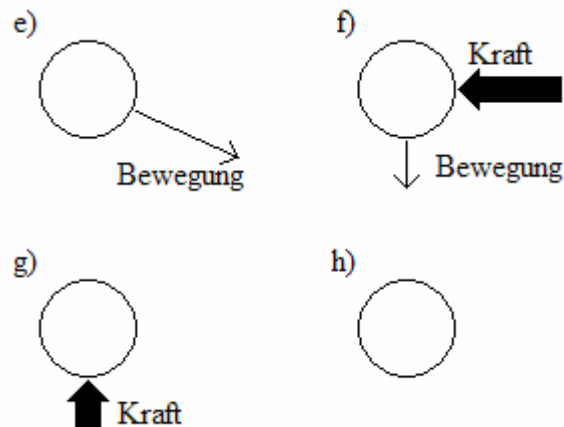


Abb. 2.4 Auswirkungen unterschiedlicher Kräfte auf einen Gegenstand(2)

Wenn nun die Kräfte wieder aufgehoben werden, wie in Abbildung 2.4 e), so wird der Gegenstand allerdings nicht wieder in den Ruhezustand zurückkehren, sondern die entsprechende Bewegung, welcher er unterliegt, weiter ausführen. Um den Gegenstand wieder zurück in seinen Ruhezustand zu überführen, müssen Kräfte entgegen seiner momentanen Bewegung auf diesen eingewirkt werden.

Nun wird in Bild f) auf den sich bewegenden Gegenstand eine Kraft exakt entgegengesetzt der Kraft aus b) ausgeübt. Wenn diese Kraft genauso stark und genauso lange auf den Gegenstand ausgeübt wird wie die Kraft in b), so wird der Gegenstand in diese Richtung gebremst sich nur noch nach unten bewegen.

Bild g) zeigt nun, wie eine Kraft genau entgegengesetzt der Kraft aus c) auf den Gegenstand gewirkt wird. Auch hier gilt wie in f), dass wenn die Kraft genauso stark und genauso lange auf den Gegenstand ausgeübt wird wie in c), so wird die Bewegung auch in dieser Richtung abgebremst.

Somit gelangt der Gegenstand wieder in seine Ruheposition, in der er keinerlei Bewegung mehr ausführt, bis neue Kräfte auf ihn einwirken.

2.3 Unterschiedliche Flugmodelle in Computerspielen

Diese physikalischen Gesetze der Gravitation und Massenträgheit sind teilweise auch in diversen Computerspielen umgesetzt worden. Für diese Arbeit sind vor allem solche Weltraumspiele interessant, in denen unterschiedliche Flugmodelle realisiert wurden. Allerdings wird nicht bei allen Weltraumsimulationen auf diese Form des Realismus Wert gelegt, so dass es auch Spiele gibt, bei denen Gravitation und Massenträgheit keine Rolle spielen.

Im Folgenden werde ich einige dieser Flugmodelle vorstellen:

2.3.1 Lunar Lander

Ein Beispiel für die Anwendung der Gravitationsgesetze in Computerspielen ist das 1979 von Atari entwickelte Spiel *Lunar Lander* (Abbildung 2.5). Dabei steuert der Spieler in einer Seitenansicht eine Mondlandefähre und muss diese an einem bestimmten Zielpunkt landen. Dazu muss der Spieler die Landefähre so steuern und Schub geben, dass sie nicht auf der Mondoberfläche zerschellt. Zum Einen wird dies dadurch erschwert, dass wenn der Spieler Schub nach recht gibt, die Mondfähre nicht nur ein Stück nach rechts fliegt, sondern durch Trägheit weiteren Schwung nach rechts erhält, selbst wenn die Steuerdüse wieder deaktiviert wird. Zum Anderen darf die Fähre auch nicht zu schnell auf dem Boden aufsetzen, so dass der Spieler auch an dieser Stelle durch entsprechenden Schub die Geschwindigkeit der Landefähre soweit reduzieren muss, dass sie sicher auf der Oberfläche aufsetzen kann [AH09].

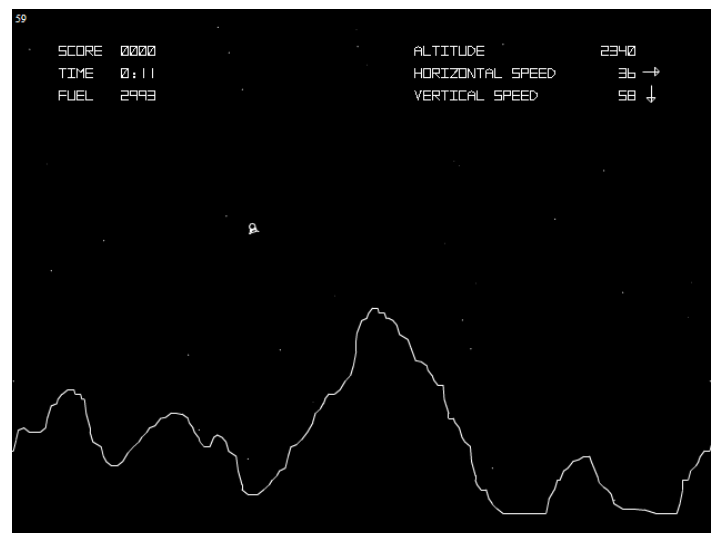


Abb. 2.5 Spielszene aus *Lunar Lander*

Mittlerweile gibt es zahlreiche Varianten des Spiels, einige davon bieten sogar die Möglichkeit, die Gravitationskraft zu variieren. So kann man beispielsweise bei einer Umsetzung als Java Applet zwischen Simulationen der Gravitationskräfte des Mondes, der Erde oder des Jupiters wählen [Fla97].

2.3.2 Asteroids

Ein weiteres Spiel im 2D-Bereich ist das von Lyle Rains und Ed Logg 1979 für Atari entwickelte Spiel *Asteroids* (Abbildung 2.6). In diesem Spiel übernimmt der Spieler die Steuerung eines Raumschiffs in einem Asteroidenfeld. Die Asteroiden fliegen chaotisch über den Bildschirm und drohen das Raumschiff zu treffen und damit zu zerstören. Der Spieler hat nun die Aufgabe das Raumschiff so zu steuern, dass es von den Asteroiden nicht getroffen wird, und dabei die Flugkörper mit Hilfe der Bordkanone zu zerstören. Dabei werden die Asteroiden aber erstmal nur in kleinere Brocken gespalten, welche wiederum umherfliegen und auch zerstört werden müssen. Die Asteroiden üben allerdings keine Anziehungskraft auf das Raumschiff aus. Die Schwierigkeit besteht darin, dass das Raumschiff, sobald es einmal Schub gegeben hat, weiter fliegt, und nur durch entsprechenden Gegenschub wieder abgebremst werden kann. Somit tritt in diesem Spiel eher die Trägheit als die Gravitation auf [AH09].

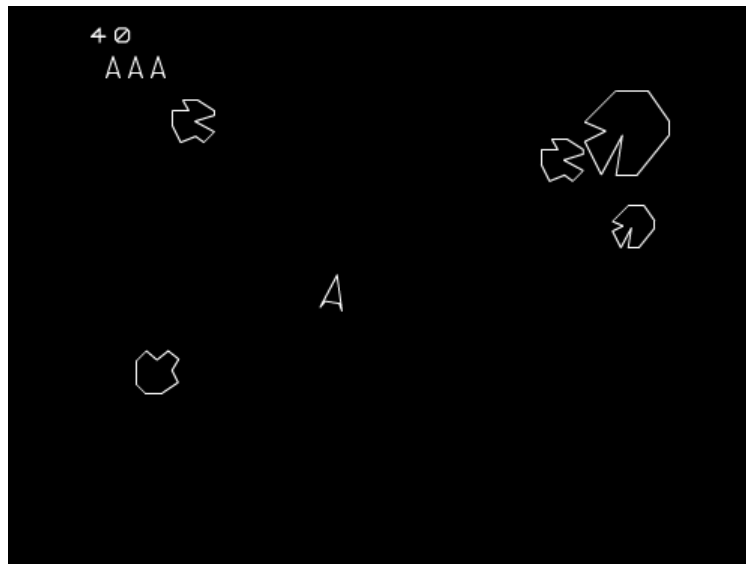


Abb. 2.6 Spielszene aus *Asteroids*

2.3.3 Elite

Bei dem 1985 von David Braben und Ian Bell entwickelten Spiel *Elite* handelt es sich um eine 3D-Weltraumsimulation (Abbildung 2.7). Der Spieler steuert sein Raumschiff im dreidimensionalen Raum von einer Raumstation zu einer anderen, um dort unterschiedliche Aufträge zu erfüllen. Im ersten Teil der Serie verhielt sich die Steuerung von Geschwindigkeit und Kurvenflug noch wie in einem mit Luft gefüllten Raum und nicht wie in einem Vakuum. Ebenso war kein Gravitationsverhalten der einzelnen Planeten enthalten [Sch09].



Abb. 2.7 Spielszene aus *Elite*

Der 1993 erschienene Nachfolger namens *Frontier* hingegen enthielt nicht nur ein weitaus größeres Universum mit deutlich mehr Planeten und Sonnensystemen, sondern auch der Raumflug wurde realistischer umgesetzt. So wurden die physikalischen Gesetze von Gravitation, Massenträgheit und Beschleunigung integriert. Der Spieler musste somit durch entsprechenden Gegenschub sein Raumschiff wieder zum Stillstand bringen. Auch wirkten sich die einzelnen Himmelskörper mit ihrer Anziehungskraft auf die Flugbahn des Raumschiffs aus [MG09].

2.3.4 Wing Commander

Der erste Teil der bisher fünfteiligen Computerspielreihe *Wing Commander* wurde 1990 von Chris Roberts für die Firma *Origin* entwickelt (Abbildung 2.8). Der Spieler schlüpft in die Rolle eines Raumjägerpiloten. Die einzelnen Missionen werden aus der Sicht des Piloten im Cockpit gesteuert. Das Hauptaugenmerk bei diesen Missionen sind in der Regel Raumschlachten oder Eskortflüge durch den Weltraum. Die Darstellung des Weltraums unterliegt allerdings nicht den physikalischen Gesetzen, so dass Anziehungskräfte von Planeten nicht vorhanden sind und auch die Massenträgheit nicht umgesetzt ist.



Abb. 2.8 Spielszene aus *Wing Commander*

Neben den insgesamt fünf Teilen der Reihe gibt es zu den einzelnen Spielen teilweise mehrere Erweiterungen, die zusätzliche Aufträge bieten und die Handlung weiterführen [MG09].

2.3.5 Star Wars: X-Wing

Ähnlich wie bei *Wing Commander* steht auch bei dem im *Star Wars*-Universum angesiedelten Weltraum Simulator *X-Wing* von *LucasArts* aus dem Jahr 1993 eher der Action-Anteil im Vordergrund als die physikalische Umsetzung von Gravitation und Massenträgheit (Abbildung 2.9). Der Spieler hat unterschiedliche Missionen zu absolvieren, die ähnlich der Missionen bei *Wing Commander* großteils aus Raumnäpften bestehen. Eine Besonderheit bieten so genannte „historische Missionen“, bei denen der Spieler teilweise die Raumnäpften aus den *Star Wars*-Filmen nachspielen kann [MG09].



Abb. 2.9 Spielszene aus *Star Wars – X-Wing*

3 Umsetzung

Im folgenden Kapitel werde ich mich mit der Implementierung der Weltraumumgebung und einzelner Steuerungsarten befassen. Dabei werden sowohl ein zweidimensionales als auch ein dreidimensionales Raummodell genutzt.

3.1 *JLI Asteroids*

Das Programm für die 2D-Umsetzung der einzelnen Steuerungsarten (siehe Kapitel 4) habe ich unter Visual C++ entwickelt. Es wurde in Teilen durch das Softwarebeispiel *JLI Asteroids* [Rou03] inspiriert, welches eine Umsetzung des Spieleklassikers *Asteroids* (siehe Kapitel 2.3.2) ist. Abbildung 3.1 zeigt eine Spielszene aus *JLI Asteroids*. Dieses Softwarebeispiel dient als Lehrprogramm zur Programmierung der Visualisierung von Szenen, Kollisions- und Steuerungsabfragen. Hierbei wird für die Umsetzung der Steuerung mit dem Joystick und des Bildschirmaufbaus *DirectX 9.0* verwendet. Davon habe ich lediglich das Prinzip der Steuerungsabfragen und des Bildschirmaufbaus übernommen.

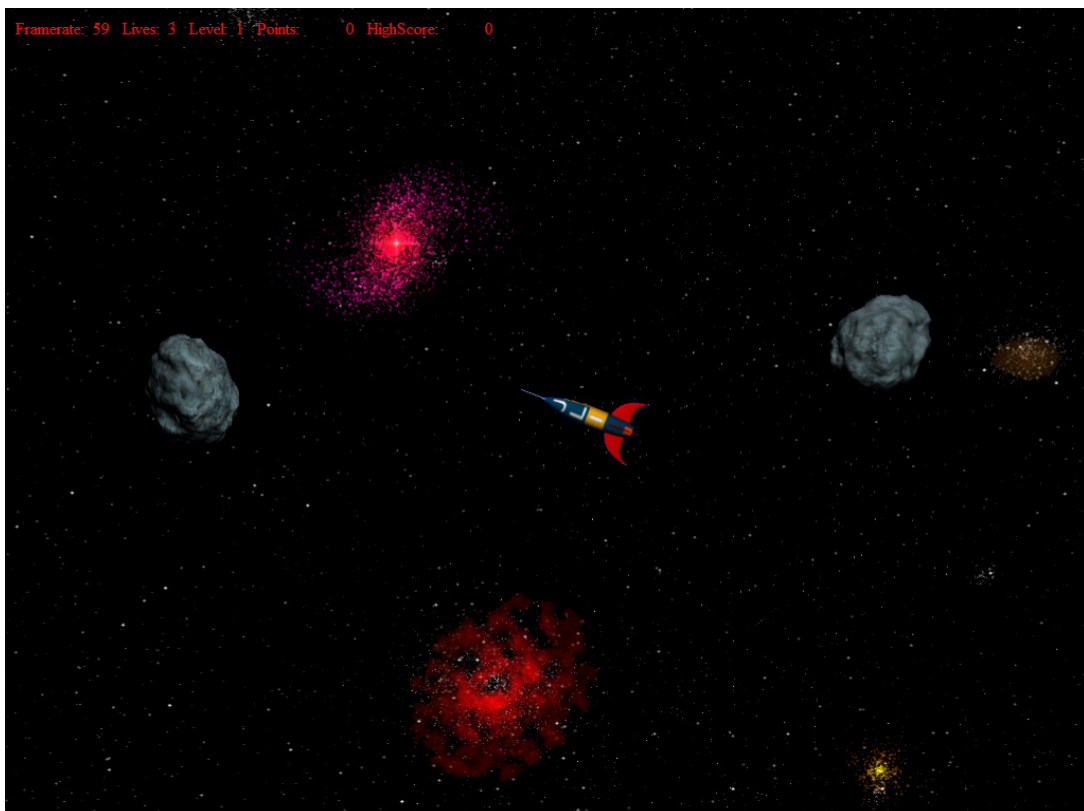


Abb. 3.1 Spielszene aus *JLI Asteroids*

Von mir wurde das Programm dahingehend erweitert, dass die einzelnen Planeten, Zielpunkte und das Raumschiff als *struct* implementiert wurden. Die benötigten Daten für die Objekte werden aus einer Konfigurationsdatei ausgelesen. Ebenso wurden die Gravitation und

auch die Trägheit als zusätzliche Funktionen implementiert. Eine weitere neue Funktion ist das Ein- und Ausschalten einer möglichen Verfolgerspur. Diese zeigt die Flugbahn des Raumschiffs als gepunktete Linie an. Dabei habe ich mich aus Speicherplatzgründen entschieden, nur alle 10 Frames einen Punkt der Linie einzuzeichnen.

3.2 *MinSG*

Als 3D-Umgebung wurde das Framework *MinSG* (Minimalistischer Szenengraph) der Arbeitsgruppe Meyer auf der Heide der Universität Paderborn verwendet [UPB09]. Dieses Framework bietet die Möglichkeit, unterschiedliche Szenen dreidimensional darzustellen. Dabei können verschiedene Objekte eingefügt oder bewegt werden. Abbildung 3.2 zeigt eine Beispielszene, welche mit *MinSG* erzeugt wurde. Im Gegensatz zu der 2D-Umsetzung wird bei *MinSG* kein *DirectX* verwendet, sondern *OpenGL* und *SDL* (*Simple DirectMedia Layer*). Dabei handelt es sich um Multimedia-Bibliotheken, welche eine Schnittstelle für Grafik-, Sound- und Eingabegeräte bieten.

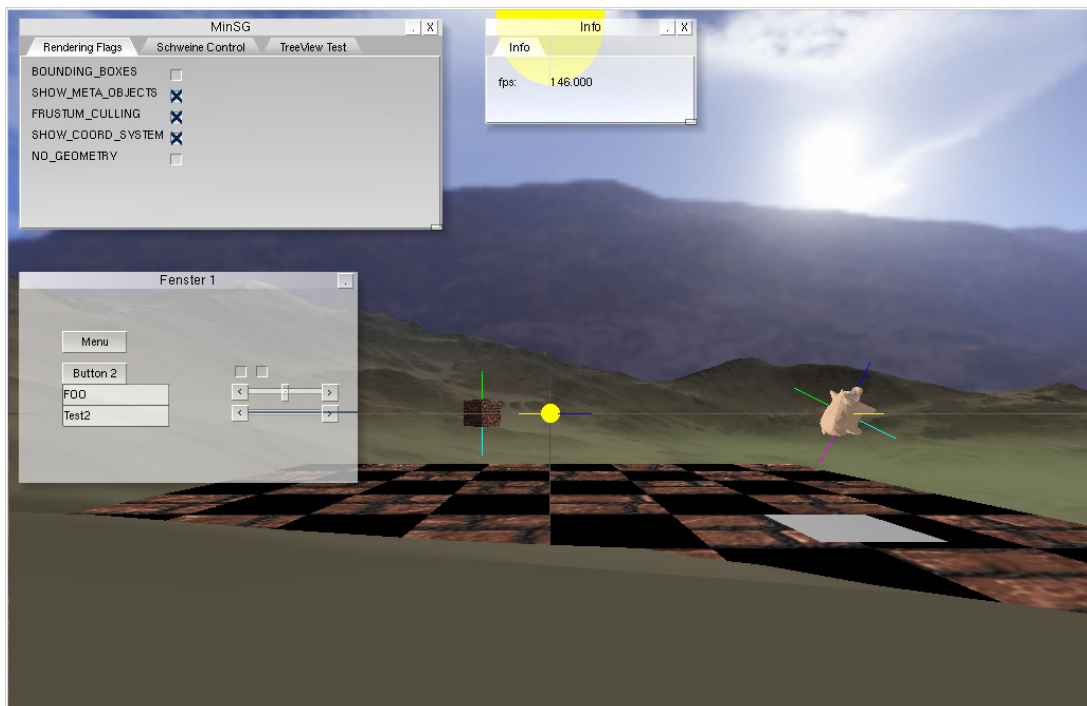


Abb. 3.2 Beispielszene aus *MinSG*

Ursprünglich wird die Abfrage der Eingabegeräte über einen Eventhandler und entsprechende Wrapper-Klassen realisiert. Ich habe aber bei der Umsetzung der einzelnen Steuerungsmodelle auf den Eventhandler verzichtet. Die Joystick- und Tastaturevents werden direkt abgefragt, um so auch feinere Steuerungsformen realisieren zu können. Neben der Implementierung der einzelnen Steuerungsmodelle wurde das Framework noch, analog zu der

2D-Umsetzung, um die Strukturen für die Planeten, Zielpunkte und das Raumschiff erweitert. Auch die Gravitation und Trägheit sind zusätzlich implementierte Funktionen.

Als Raumschiff wurde ein Modell eines fiktiven *Cobra MK II Space Fighters* aus dem Internet verwendet [TS09].

3.3 Config-Datei

Um die Szene, sowohl in der 2D- als auch der 3D-Umsetzung, einfach anpassen zu können, werden die einzelnen Werte der Planeten, Zielpunkte und des Raumschiffs, sowie der Gravitationskonstanten, aus einer Konfigurationsdatei ausgelesen. Dazu wird die Datei *config.txt* verwendet.

Die Konfigurationsdatei wird dazu so ausgelesen, dass in ihr nach entsprechenden Schlüsselwörtern gesucht wird und dann die nachfolgenden Zeichen der Zeile als die zu verwendenden Werte für das gefundene Objekt verwendet werden.

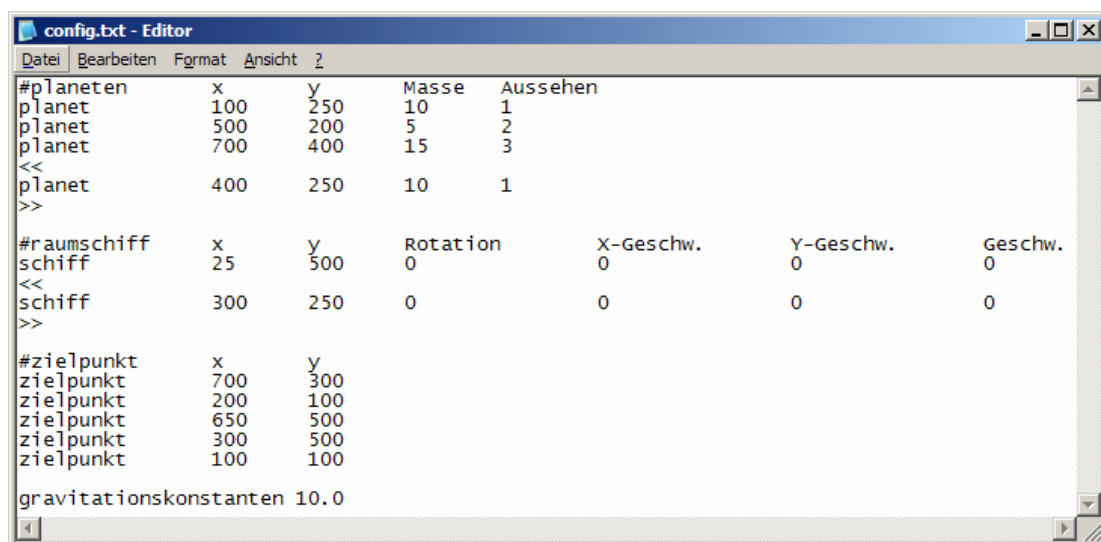


Abb. 3.3 2D-Konfigurationsdatei config.txt

Für die unterschiedlichen Planeten werden aus der Konfigurationsdatei in Abbildung 3.3 die Werte für die X- und Y-Position ausgelesen, ebenso die Masse und das Aussehen. Für das Aussehen sind unterschiedliche Grafiken hinterlegt, welche durchnummeriert sind. Dabei wird dann, je nachdem welcher Wert für das Aussehen in der *config.txt* hinterlegt wurde, die entsprechende Grafik geladen.

Auch für das Raumschiff sind Startwerte für die X- und Y-Position enthalten. Der Wert *Rotation* gibt den Ausrichtungswinkel des Raumschiffs zu Beginn des Programms an. Für die Geschwindigkeit des Raumschiffs werden unterschiedliche Geschwindigkeitswerte ausgelesen. Dabei wird zwischen Geschwindigkeit in X- und in Y-Richtung unterschieden. Zusätzlich gibt es noch einen generellen Geschwindigkeitswert. Je nachdem welches Steuerungsmo-

dell ausgewählt wird, werden entweder die X- und Y-Geschwindigkeiten oder aber die generelle Geschwindigkeit genutzt (vergleiche Kapitel 4).

Des Weiteren werden die Werte für mögliche Zielpunkte abgefragt, die angefliegen werden können, Für diese werden nur X- und Y-Position angegeben. Sie benötigen keinerlei Masse, da die Zielpunkte nur als Markierungen gelten und nicht als zusätzliche Gravitationspunkte.

Der letzte auszulesende Wert ist die Gravitationskonstante. In der Realität ist sie zwar konstant, allerdings kann in dieser Simulation mit einer Anpassung der Gravitationskonstanten die Stärke der Gravitation verändert werden, um unterschiedliche Szenarien zu erzeugen.

In der Konfigurationsdatei für die 3D-Umgebung sind für alle Objekte zusätzlich Z-Koordinaten enthalten. Außerdem wird für das Raumschiff eine Geschwindigkeit in Z-Richtung angegeben. Des Weiteren wird die Rotation in der 3D-Variante in zwei unterschiedliche Werte aufgeteilt, eine Rotation um die X- und eine um die Z-Achse.

3.4 Berechnung von Gravitation und Trägheit

Für die Berechnung der Gravitation wird für jeden Planeten einzeln die auf das Raumschiff wirkende Anziehungskraft berechnet. Dazu wird eine Schleife solange durchlaufen, bis für jeden Planeten die Anziehungskraft berechnet und zur gesamten Anziehungskraft in Richtung der X- und Y-Achse addiert wurde. Dabei geben die Werte *ship.x* und *ship.y* die X- und Y-Position des Raumschiffs an. Die Werte *gravX* und *gravY* geben die Summe der Gravitationskräfte an.

```
//Berechnung der Gravitation

for( int j=planetenCount; j>0; j--)
{
    planeten[j-1].grav = gravConst * planeten[j-1].masse /
        (pow(sqrt(pow((planeten[j-1].x - ship.x), 2) +
            pow((planeten[j-1].y - ship.y), 2)), 3));
    gravX += ((planeten[j-1].x - ship.x) * planeten[j-1].grav);
    gravY += ((planeten[j-1].y - ship.y) * planeten[j-1].grav);
}
ship.x += gravX * timediff / 100;
ship.y += gravY * timediff / 100;
```

Abb. 3.4 Berechnung der Gravitation für die 2D-Steuerung

Nachdem die Schleife in Abbildung 3.4 durchlaufen wurde, wird der daraus resultierende Gravitationsvektor auf die Position des Raumschiffs addiert. Dabei wird allerdings noch ein zeitlicher Faktor berücksichtigt, so dass die Bewegung von der Frame-Rate unabhängig ist. Neben dem zeitlichen Faktor wird auch noch ein konstanter Wert zur Skalierung eingerechnet. Dieser Wert unterscheidet sich teilweise in den einzelnen Steuerungsmodellen und wurde

nach eingehenden Tests so gewählt, dass sich das Raumschiff in angemessener Geschwindigkeit bewegt.

Dabei ist zu beachten, dass die Werte *gravX* und *gravY* nicht auf Null zurückgesetzt werden, nachdem sie auf die Werte *ship.x* und *ship.y* addiert wurden. Die bisherige Bewegung des Raumschiffs, welche aus der Gravitation resultiert, muss beibehalten werden, da im All keinerlei Reibung existiert und somit das Raumschiff dem Gesetz der Trägheit folgt und nicht abgebremst wird.

Neben der Gravitation und der daraus resultierenden Trägheit wird für einige Steuerungsmodelle, davon unabhängig, zusätzlich die Trägheit bezüglich der eigenen Bewegung des Raumschiffs berechnet. Allerdings hängt die Implementierung der Trägheit von den unterschiedlichen Steuerungsmodellen ab.

```
moveVecX -= ship.velX * timediff / 50;
moveVecY -= ship.velY * timediff / 50;

ship.x -= moveVecX * timediff / 50;
ship.y -= moveVecY * timediff / 50;
```

Abb. 3.5 Variante a) zur Berechnung der Trägheit für die Raumschiffbewegung für die 2D-Steuerung

Abbildung 3.5 zeigt die Berechnung der Geschwindigkeit des Raumschiffs, sofern die Blickrichtung nicht für die Steuerung relevant ist. In diesem Fall wird die Geschwindigkeit des Raumschiffs in die beiden Werte *ship.velX* und *ship.velY* unterteilt. Dabei gibt jeder Wert jeweils die Geschwindigkeit in Richtung der X- und Y-Achse an. Daraus resultiert dann in skaliertem Form der Bewegungsvektor des Raumschiffs.

```
moveVecX += sin(ship.rotation) * ship.vel * timediff / 100;
moveVecY += cos(ship.rotation) * ship.vel * timediff / 100;

ship.x -= moveVecX * timediff / 100;
ship.y -= moveVecY * timediff / 100;
```

Abb. 3.6 Variante b) zur Berechnung der Trägheit für die Raumschiffbewegung für die 2D-Steuerung

In der Berechnung in Abbildung 3.6 ist hingegen die Blickrichtung des Raumschiffs ausschlaggebend für die Bewegungsrichtung bzw. die Schubrichtung des Raumschiffs. In diesem Fall wird mit Hilfe der Sinus- und Cosinus-Funktion der Bewegungsvektor berechnet. In beiden Fällen wird der Bewegungsvektor nach seiner Berechnung auf die Position des Raumschiffs addiert.

Bei der Implementierung der Gravitation und Trägheit im dreidimensionalen Raum verläuft die Berechnung analog zum zweidimensionalen Raum. Es wird lediglich die Z-Koordinate mit in die Berechnungen einbezogen.

3.5 Berechnung der Energie-Steuerung

Eine besondere Form der Steuerung existiert bei der Energie-Steuerung. Diesem Steuerungsmodell liegen einige mathematische Formeln zugrunde, welche mir durch Herrn Dr. Ziegler zur Verfügung gestellt wurden (siehe Anhang a)). Dabei wird das Raumschiff auf unterschiedlichen, aus der potentiellen Energie des Raumschiffs resultierenden Umlaufbahnen gesteuert. Der Initialwert für die potentielle Energie des Raumschiffs wird anhand seiner Position ermittelt.

```
deltaPhiX = 0;
deltaPhiY = 0;

for( int b = planetenCount; b > 0; b-- )
{
    deltaPhiX += (planeten[b-1].masse * (ship.x - planeten[b-1].x)) /
                (pow(sqrt(pow((ship.x - planeten[b-1].x), 2) +
                pow((ship.y - planeten[b-1].y), 2)), 3));

    deltaPhiY += (planeten[b-1].masse * (ship.y - planeten[b-1].y)) /
                (pow(sqrt(pow((ship.x - planeten[b-1].x), 2) +
                pow((ship.y - planeten[b-1].y), 2)), 3));
}

ship.x += ship.velX * deltaPhiY * timediff * 1000.0 ;
ship.y -= ship.velX * deltaPhiX * timediff * 1000.0 ;
```

Abb. 3.7 Berechnung der Raumschiffbewegung bei Links-Rechts-Steuerung

In Abbildung 3.7 wird die Berechnung der Raumschiffbewegung für die Links-Rechts-Steuerung des Joysticks dargestellt. Zuerst werden die einzelnen Gravitationskräfte der Planeten, welche auf das Raumschiff einwirken, berechnet und addiert. Mit Hilfe der Werte *deltaPhiX* und *deltaPhiY* und der X-Achsenausrichtung des Joysticks wird dann die neue Position des Raumschiffs auf der Umlaufbahn um die Planeten berechnet. Dabei hat das Raumschiff an jedem Punkt dieser Umlaufbahn dieselbe Energie.

Um die potentielle Energie des Raumschiffs zu verändern, wird die Y-Achsenausrichtung des Joysticks abgefragt. Die Berechnung dafür wird in Abbildung 3.8 dargestellt.

```

Phi = 0;

for( int b = planetenCount; b > 0; b-- )
{
    Phi -= planeten[b-1].masse / (sqrt(pow((ship.x - planeten[b-1].x), 2) +
        pow((ship.y - planeten[b-1].y), 2)));
}

ship.energy += (ship.velY / ship.masse) * (timediff / 5000.0);

for( float c = ceil(log(abs(100 * (ship.energy - Phi) / ship.energy)));
    c > 0; c-- )
{
    Phi = 0;
    deltaPhiX = 0;
    deltaPhiY = 0;

    for( int b = planetenCount; b > 0; b-- )
    {
        [neue Berechnung von Phi wie oben]
        [neue Berechnung von deltaPhiX wie oben]
        [neue Berechnung von deltaPhiY wie oben]
    }
    tempVarU = (ship.energy - Phi) / (pow(deltaPhiX, 2) + pow(deltaPhiY, 2));
    ship.x += tempVarU * deltaPhiX;
    ship.y += tempVarU * deltaPhiY;
}

```

Abb. 3.8 Berechnung der Änderung der Energie und der Energie-Umlaufbahn

Die zweite for-Schleife in Abbildung 3.8 wird nur dann ausgeführt, wenn die Energie sich verändert hat, da ansonsten die Abbruchbedingung sofort erfüllt wird. Bei Ausführung dieser for-Schleife werden die Werte *Phi*, *deltaPhiX* und *deltaPhiY* neu berechnet. Mit Hilfe dieser Werte und der neuen potentiellen Energie des Raumschiffs wird die neue Position mit einer veränderten Entfernung zu den Planeten berechnet.

4 Steuerungsarten

Für diese Arbeit habe ich verschiedene Steuerungsmodelle implementiert. In einigen davon wurde das physikalische Gesetz der Trägheit berücksichtigt. Des Weiteren unterscheiden sich die Umsetzungen auch darin, dass einige der Modelle im zweidimensionalen und einige im dreidimensionalen Raum implementiert wurden. Allen Modellen ist gemein, dass die Planeten unterschiedliche Gravitationskräfte auf das Raumschiff ausüben. In diesem Kapitel werde ich auf die einzelnen Steuerungen und ihre näher eingehen.

4.1 Steuerungsmodelle in 2D

4.1.1 „absolute Steuerung“

Bei diesem Steuerungsmodell wird das Raumschiff direkt nach rechts, links, oben und unten gesteuert. Das bedeutet, dass das Raumschiff auf einen Tastendruck der rechten Cursor-Taste mit einer festen Geschwindigkeit nach rechts fliegt (Abbildung 4.1). Analog bedeutet ein Drücken der Cursor-Taste nach oben, dass das Raumschiff sich nach oben bewegt (Abbildung 4.2). Das Raumschiff behält aber diese Geschwindigkeit bei auch, wenn die Taste wieder losgelassen wird. Das Raumschiff gehorcht also der Massenträgheit.

Das Raumschiff kann aber nicht nur per Tastatur gesteuert werden, sondern auch mittels des Joysticks. Dabei reagiert das Raumschiff genauso auf die entsprechenden Richtungen wie bei der Tastatursteuerung.

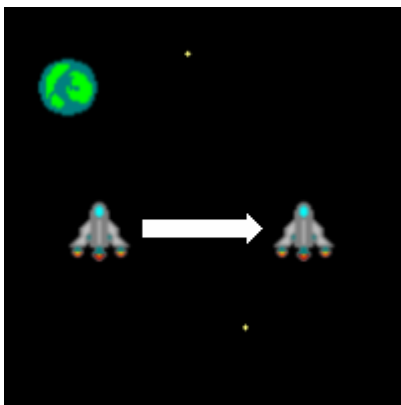


Abb. 4.1 Steuerung nach rechts

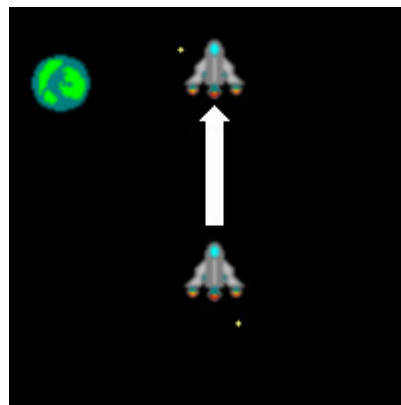


Abb. 4.2 Steuerung nach oben

Diese Steuerung wird mithilfe von zwei unterschiedlichen Geschwindigkeiten realisiert. Im Programm heißen diese *ship.velX* und *ship.velY*. Sie bezeichnen jeweils die Geschwindigkeiten, mit der sich das Raumschiff in Richtung der X- bzw. der Y-Achse fliegt. Dabei bedeutet ein negativer Wert für *ship.velX*, dass das Raumschiff nach links fliegt, ein positiver Wert lässt sich das Raumschiff nach rechts bewegen. Ähnlich steuert ein negativer Wert für *ship.velY* das Raumschiff auf dem Bildschirm nach oben, ein positiver Wert nach unten.

4.1.2 „direkte Steuerung“

Diese Steuerungsart ist der absoluten Steuerung sehr ähnlich. Ebenso wie bei der absoluten Steuerung fliegt das Raumschiff bei entsprechendem Tastendruck in die angegebene Richtung. Es gibt aber dennoch einige Unterschiede zu dem absoluten Steuerungsmodell.

Die Tastatursteuerung ist in diesem Fall fast zu vernachlässigen. Sie unterscheidet sich nur dahingehend, dass ein Loslassen der Taste auch ein Setzen der Geschwindigkeit auf null bedeutet. Das Raumschiff unterliegt also nicht der Massenträgheit. Die Planeten hingegen üben immer noch ihre Anziehungskräfte auf das Raumschiff aus.

Das Besondere an diesem Steuerungsmodell ist hingegen die Joysticksteuerung. Denn bei dieser Variante ist der Ausschlag des Joysticks wichtig für die Geschwindigkeit, mit der das Raumschiff sich bewegen soll. So bedeutet etwa eine schwache Auslenkung des Joysticks nach rechts, dass auch das Raumschiff sich nur recht langsam nach rechts bewegt. Wird der Joystick allerdings sehr stark nach links ausgeschlagen, so bewegt sich auch das Raumschiff mit einer deutlich höheren Geschwindigkeit nach links.

Zu diesem Steuerungsmodell wurden insgesamt zwei unterschiedliche Varianten umgesetzt.

- a) Bei der ersten Variante wird der Ausschlag des Joysticks linear auf die Geschwindigkeit abgebildet (Abbildung 4.3). Die Geschwindigkeit verdoppelt sich also wenn auch der Joystick doppelt so weit ausgeschlagen wird.
- b) In der zweiten Variante wird der Ausschlag des Joysticks in dritter Potenz als Geschwindigkeit übernommen (Abbildung 4.4). Ein sehr kleiner Ausschlag des Joysticks bedeutet somit eine sehr kleine Geschwindigkeit, wobei ein weiterer Ausschlag deutlich höhere Geschwindigkeiten als die lineare Variante bietet.

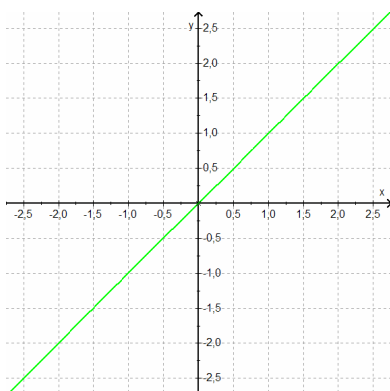


Abb. 4.3 $f(x) = x$

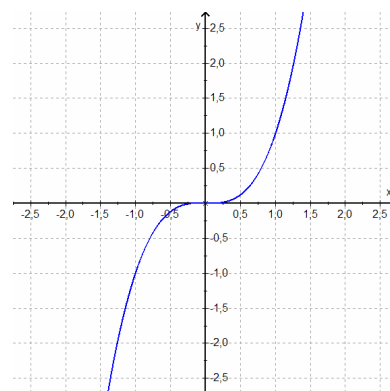


Abb. 4.4 $f(x) = x^3$

Für diese beiden Varianten der Steuerung wurde auch die Abstandsbedingung der Zielpunkte etwas angepasst. Da mit diesen Modellen eine sehr genaue Steuerung des Raumschiffs möglich ist, ist auch der benötigte Abstand zum entsprechenden Zielpunkt deutlich kleiner als bei den anderen Steuerungsmodellen.

4.1.3 „Fahrradsteuerung“

Die bisher beschriebenen Steuerungsmodelle haben alle direkt die Richtungen des Joysticks bzw. der Cursor-Tasten an das Raumschiff weitergegeben. Die folgende Steuerung hingegen unterscheidet sich davon erheblich.

So bedeutet bei diesem Modell etwa eine Linksbewegung des Joysticks nicht, dass sich das Raumschiff nach links bewegt, sondern das Raumschiff führt eine Rotation nach links aus. Analog dazu wird durch einen Ausschlag des Joysticks nach rechts eine Rechtsrotation durchgeführt. Ebenso führen die Links- und Rechtstasten der Tastatur zu entsprechenden Rotationsbewegungen.

Wird der Joystick nach vorne bewegt, so gibt das Raumschiff Schub in die momentane Blickrichtung. Analog bedeutet eine Bewegung des Joysticks nach hinten, dass Rückwärtsschub gegeben wird. Abbildung 2.5 zeigt eine Flugbahn des Raumschiffs um einen Planeten. Dabei ist die Gravitation des Planeten deaktiviert, um die Flugbahn deutlicher darzustellen.

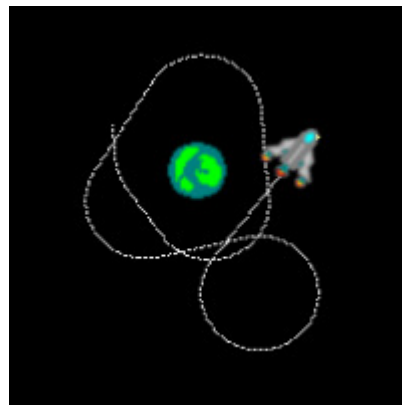


Abb. 4.5 Bewegung des Raumschiffs mit eingeschalteter Verfolgerspur

Das Raumschiff bleibt zwar beim Loslassen des Joysticks nicht stehen, so dass zumindest der Schub beibehalten wird, allerdings wird ansonsten keine Trägheit des Raumschiffs berücksichtigt. Das Raumschiff bewegt sich immer in genau die Richtung, in die es auch deutet. Daher auch der Name „Fahrradsteuerung“, denn wie beim Fahrradfahren wirkt sich eine Änderung der Bewegungsrichtung direkt auf die Fahrtrichtung aus.

4.1.4 „realistische Steuerung“

Dieses Modell bildet, zumindest im zweidimensionalen Raum, die realistischste Steuerung ab. Ähnlich der Fahrradsteuerung bewirkt eine Links- bzw. Rechtsbewegung des Joysticks eine Rotation des Raumschiffs. Auch der Schub des Raumschiffs wird durch eine Vorwärtsbewegung des Joysticks bewirkt.

Allerdings fliegt das Raumschiff nun nicht mehr direkt dorthin wohin es gesteuert wird, sondern es bleibt immer noch der Impuls in die vorherige Richtung bestehen. Die „neue“ Flugrichtung wird zwar dahingehend angepasst, dass der Schub in die entsprechende Richtung auch zu einer neuen Bewegungsrichtung führt, allerdings lässt die alte Schubrichtung das Raumschiff auch weiter dahin treiben.

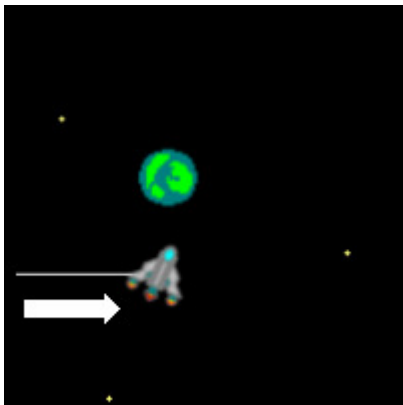


Abb. 4.6 Das Raumschiff bewegt sich trotz Rotation weiter nach rechts

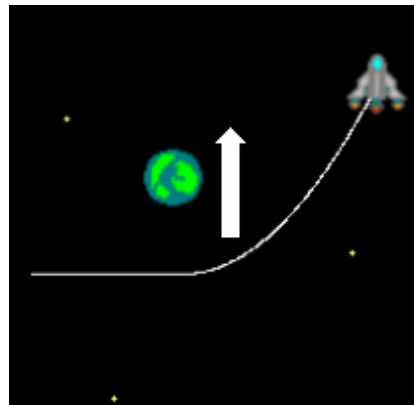


Abb. 4.7 Auch bei Schub in Blickrichtung treibt das Raumschiff zusätzlich weiter in die alte Bewegungsrichtung

4.1.5 Energie-Steuerung

Die Energie-Steuerung unterscheidet sich grundlegend von den bisher vorgestellten Steuerungsmodellen. Das Raumschiff kann nicht mehr direkt frei auf dem Bildschirm bewegt werden, sondern es fliegt auf entsprechenden Energiebahnen um die Planeten herum. Dabei wird dasselbe Prinzip verwendet wie auch für Satelliten, die in einer konstanten Umlaufbahn um einen Planeten fliegen.

Anfangs erhält das Raumschiff eine Startposition. Mit dieser Startposition, bzw. dem Abstand zu dem Planeten, wird dann die potentielle Energie des Raumschiffs berechnet. Diese potentielle Energie bleibt solange konstant, bis der Joystick nach vorne oder nach hinten bewegt wird, oder die nach-oben, bzw. nach-unten Tasten gedrückt werden. Dadurch wird dann die Energie des Raumschiffs erhöht und gesenkt, wodurch sich der Abstand zu dem Planeten ändert. Eine Links- oder Rechtsbewegung des Joysticks bewirkt, dass sich das Raumschiff auf der Energiebahn, auf der es sich gerade befindet, hin und her bewegt.

Bei diesem Steuerungsmodell ist es nicht nur möglich, sich um einen einzelnen Planeten auf unterschiedlichen Energiekurven zu bewegen (siehe Abbildung 4.8). Es können auch mehrere Planeten in der Szene vorhanden sein, so dass sich für die unterschiedlichen Energiewerte des Raumschiffs nicht nur Kreisbahnen ergeben. Es können dabei auch unterschiedlich geformte Bahnen entstehen (siehe Abbildung 4.9).

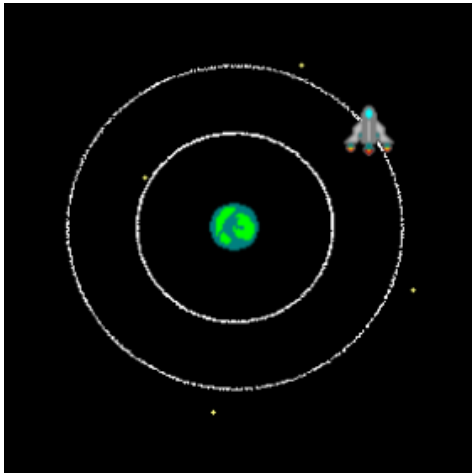


Abb. 4.8 Unterschiedliche Umlaufbahnen der Energie-Steuerung mit einem Planeten

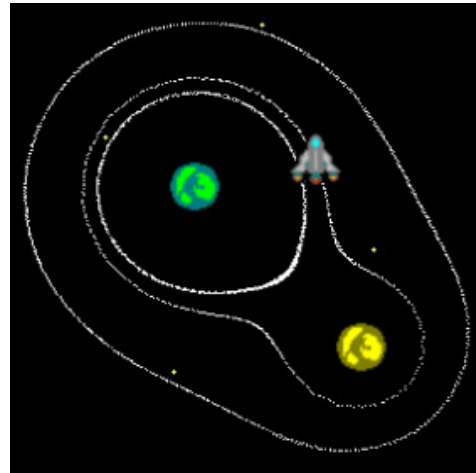


Abb. 4.9 Unterschiedliche Umlaufbahnen der Energie-Steuerung mit zwei Planeten

4.2 Steuerungsmodelle in 3D

Bei den dreidimensionalen Steuerungen steht die Joystick-Steuerung im Vordergrund. Die Steuerung per Tastatur ist diesen Steuerungsmodellen zwar auch möglich und implementiert, allerdings unterscheiden sich die einzelnen Tastatursteuerungsmodelle kaum von einander. Zusätzlich ist es hilfreich, für die dreidimensionalen Steuerungsmodelle einen Joystick mit Schubkontrolle zu verwenden (etwa den Joystick ST30 von Saitek). Abbildung 4.10 zeigt eine Beispielszene der 3D-Steuerung.

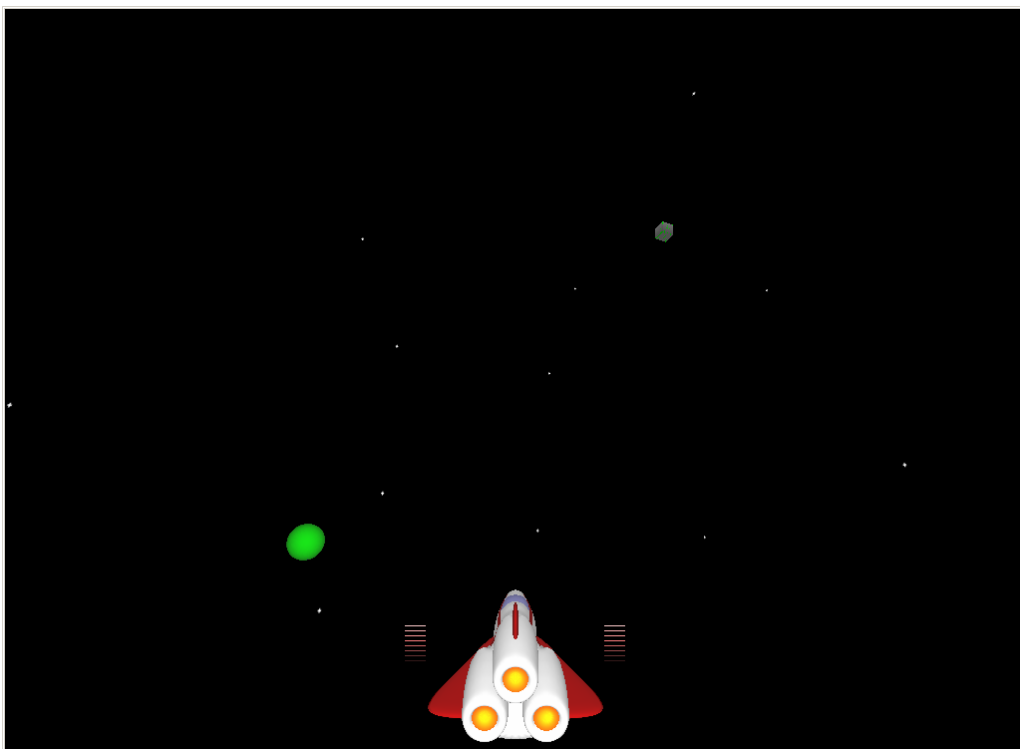


Abb. 4.10 Beispielszene der 3D-Steuerung

4.2.1 Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit

Bei diesem dreidimensionalen Steuerungsmodell wird mit Hilfe der Pfeiltasten bzw. durch Bewegen des Joysticks die Blickrichtung des Raumschiffs verändert. Dabei bedeutet eine Ausrichtung des Joysticks nach rechts bzw. links eine Rotation in die entsprechende Richtung. Ebenso wird durch Drücken der Pfeiltasten nach rechts bzw. links das Raumschiff in die entsprechende Richtung gedreht. Ein Heranziehen des Joysticks oder ein Drücken der Pfeiltaste „nach unten“ lässt das Raumschiff sich nach oben neigen. Im Gegensatz dazu bewirkt ein Wegdrücken des Joysticks oder das Drücken der Pfeiltaste „nach oben“ eine Neigung des Raumschiffs nach unten. Diese Steuerung ist der realen Flugzeugsteuerung nachempfunden.

Zur Beschleunigung und zum Abbremsen des Raumschiffs kann entweder die Schubkontrolle des Joysticks oder die Tastatur verwendet werden. Hierbei dient die Taste W zum Beschleunigen und S zum Abbremsen.

Bei diesem ersten Steuerungsmodell im dreidimensionalen Raum bewirkt eine Ausrichtung des Joysticks oder ein Betätigen der entsprechenden Tasten eine Bewegung oder Rotation um einen konstanten Wert. Dabei ist irrelevant wie weit der Joystick ausgerichtet wird, die Geschwindigkeit der Rotation oder die Geschwindigkeit des Raumschiffs bleibt immer gleich. Erst wenn der Joystick wieder in die neutrale Position gebracht oder die entsprechende Taste losgelassen wird, werden Beschleunigung und Rotation gestoppt.

4.2.2 Steuerung ohne Geschwindigkeitsabstufungen, aber mit Trägheit

Dieses Steuerungsmodell unterscheidet bezüglich der Steuerung des Raumschiffs nicht von der Steuerung des Modells unter 4.2.1. Allerdings unterliegt das Raumschiff in diesem Flugmodell nicht nur der Gravitation der einzelnen Planeten, sondern auch der Trägheit. So bewirkt ein Loslassen der Schub-Taste etwa nicht, dass das Raumschiff stehen bleibt. Erst wenn entsprechender Gegenschub gegeben wird, kommt das Raumschiff wieder zum Stillstand. Die Rotation hingegen unterliegt nicht der Trägheit.

4.2.3 Steuerung mit Geschwindigkeitsabstufungen, aber ohne Trägheit

Bei diesem Steuerungsmodell wird zusätzlich zur Richtungsabfrage ermittelt, wie stark die Auslenkung des Joysticks und der Schubkontrolle ist. Je weiter der Joystick bewegt wird, desto schneller wird das Raumschiff gedreht. Analog dazu wirkt sich die Auslenkung der Schubkontrolle auf die Geschwindigkeit aus. Eine leichte Auslenkung bewirkt eine geringe Geschwindigkeit, eine starke Auslenkung hingegen eine hohe Geschwindigkeit.

4.2.4 Steuerung mit Geschwindigkeitsabstufungen und mit Trägheit

Auch dieses Steuerungsmodell unterscheidet sich in Bezug auf die Steuerung des Raumschiffs nicht von der Steuerung 4.2.3. Im Gegensatz dazu unterliegt das Raumschiff aber auch hier wieder der Trägheit. Ein Zurücksetzen der Beschleunigung bewirkt kein Abbremsen des Raumschiffs. Dazu muss auch in diesem Modell Gegenschub gegeben werden.

5 Entwicklung des Fragebogens

Dieses Kapitel beschreibt die einzelnen Fragestellungen und die Entwicklung des Fragebogens, der zur Analyse der untersuchten Kriterien wie Schwierigkeit, Realismus und Lernfähigkeit der einzelnen Steuerungsmodelle verwendet wurde.

5.1 Unterschiedliche Testszenarien

Um die einzelnen Steuerungsarten miteinander zu vergleichen habe ich einen Fragenbogen entwickelt, welcher von unterschiedlichen Testkandidaten ausgefüllt wurde. Dabei unterteilt sich der Fragebogen in zwei grundlegende Abschnitte. Im ersten Abschnitt beantwortet der Testkandidat subjektiv seine Erfahrungen in Bezug auf den Testdurchlauf mit den Steuerungsmodellen. Der zweite Abschnitt hingegen ist rein objektiver Natur und wird nicht von dem Testkandidaten ausgefüllt, sondern durch entsprechende zeitliche Messungen zum Erreichen der einzelnen Zielpunkte in der Testszene bestimmt.

Für den subjektiven Test durften die Kandidaten zuerst mit den einzelnen Steuerungen experimentieren. Dabei konnten sie Zielpunkte anfliegen und erkennen, wie leicht oder schwer diese zu erreichen sind. Die Probanden sollten parallel zu den Testdurchläufen einen Fragebogen ausfüllen und die einzelnen Steuerungen bewerten (siehe Anhang b)). Dabei testeten die Kandidaten jede der implementierten Steuerungen. Ein solcher Testdurchlauf dauerte in etwa 15 bis 20 Minuten, wobei manche Testteilnehmer die einzelnen Steuerungen auch durchaus länger ausprobiert haben.

Der objektive Test stellt eine Erweiterung des subjektiven Tests dar. Zusätzlich zu den Fragen des Fragebogens sollten die Testkandidaten ein vorgefertigtes Szenario durchspielen. Dabei dienten die einzelnen Zielpunkte als zu erreichende Koordinaten. Die zum Erreichen der Zielpunkte benötigten Zeiten wurden zur Auswertung notiert. Da für jede Steuerung mehrere Testdurchläufe zu absolvieren waren, betrug die Dauer, zusätzlich zu dem subjektiven Abschnitt, durchschnittlich weitere 30 bis 45 Minuten.

5.2 Subjektiver Abschnitt des Fragebogens

5.2.1 Vorkenntnisse

Bei der Implementierung einiger Steuerungen habe ich mich an Flugmodellen aus mir bekannten Computerspielen wie etwa *Star Wars – X-Wing* oder *Wing Commander* orientiert. Daher stellte ich mir die Frage, ob Vorkenntnisse solcher Flugsimulationen einen positiven Einfluss auf das Beherrschen entsprechender Steuerungsmodelle haben könnten.

Aufgrund dessen entschloss ich mich, einen kurzen Katalog an Computerspielen auszuwählen und die Kandidaten zu fragen, ob sie diese Spiele schon mal gespielt haben. Dabei wollte

ich mich aber nicht nur 3D-Flugsimulationen beschränken, sondern auch auf Spiele in 2D-Umgebung, wie eben *Lunar Lander* oder *Asteroids* eingehen.

Elite bzw. den Nachfolger *Frontier* habe ich ausgesucht, da es sich hierbei um ein Vorreiterspiel für Weltraumsimulationen im dreidimensionalen Raum handelt. Die Fortsetzung *Frontier* enthielt als eines der ersten Computerspiele eine Umsetzung der physikalischen Prinzipien von Gravitation und Trägheit.

Die Liste an 3D-Simulationen hätte beliebig lang erweitert werden können, allerdings entschied ich mich dafür, nur die fünf folgenden bekannten Serien und Vorreiterspiele aufzulisten (davon zwei 2D-Spiele und drei 3D-Spiele). Eine kurze Erläuterung zu den einzelnen Spielen ist in Kapitel 2.3 zu finden:

- 2D-Spiele:
 - *Lunar Lander*
 - *Asteroids*
- 3D-Spiele
 - *Elite*
 - *Wing Commander*
 - *Star Wars – X-Wing*

5.2.2 Einarbeitungszeit

Als nächstes wollte ich wissen, ob die Kandidaten die einzelnen Steuerungen zügig beherrschen konnten. Dieser Punkt steht in direktem Zusammenhang mit dem objektiven Abschnitt des Fragebogens. Daher habe ich diese Frage bewusst etwas knapp gehalten. Ziel war lediglich, eine Einschätzung der Testkandidaten zu erhalten, ob die Steuerungen für sie intuitiv und sofort leicht verständlich sind.

5.2.3 Schwierigkeitsgrade der einzelnen Steuerungsmodelle

Ein wichtiger Punkt für den Fragebogen sollte sein, wie schwierig die Testkandidaten die einzelnen Steuerungsmodelle empfanden. Dabei konnten die Testpersonen die Schwierigkeitsgrade auf einer Skala von 1 bis 10 bewerten. Die 1 steht hierbei für „sehr einfach“ und die 10 für „nahezu unsteuerbar“. In einer frühen Version des Fragebogens sollte 10 noch für „sehr schwer“ stehen. Allerdings wurde die Begrifflichkeit dahingehend angepasst, dass dem Testkandidaten hiermit ein deutlich extremes Kriterium zur Verfügung steht.

5.2.4 Besondere Schwierigkeiten beim Steuern

Ein weiteres Ziel des Fragebogens war es zu erfahren, womit genau die verschiedenen Testkandidaten bei den unterschiedlichen Steuerungsmodellen Schwierigkeiten hatten. Nach eigener Einschätzung erwies sich die Trägheit als ein sehr großes Problem bei der Steuerung in den dreidimensionalen Umgebungen. Das Raumschiff konnte, sobald es einmal in Bewegung war, kaum kontrolliert abgebremst und zielgenau gesteuert werden. Da die Steuerung aber möglichst realistisch sein sollte, wollte ich das Prinzip der Trägheit nicht vollständig ausklammern. Ich hatte mich daher entschieden einzelne Steuerungsmodelle ohne Trägheit einzusetzen. So verwendete ich gerade die dreidimensionalen Steuerungsmodelle jeweils doppelt, einmal mit und einmal ohne Trägheit. Ziel war es, anhand der Bewertungen der verschiedenen Kandidaten zu überprüfen, ob diese das Prinzip der Trägheit tatsächlich als Schwierigkeit beim Steuern ansehen.

Um weitere Schwierigkeiten bei der Steuerung zu erkennen oder zumindest einzugrenzen, entschied ich mich danach zu fragen, womit die Testpersonen die größten Schwierigkeiten hatten.

5.2.5 Realismus der einzelnen Steuerungsmodelle

Schon recht früh bei der Umsetzung der einzelnen Steuerungsmodelle versuchte ich, sie möglichst realistisch zu gestalten. Etwa bei der Umsetzung der „absoluten Steuerung“ im zweidimensionalen Raum war die Trägheit ein Punkt, welcher auf jeden Fall berücksichtigt werden sollte. Dieses Prinzip sollte aber auch bei der Umsetzung der Steuerung mit einer Rotation des Raumschiffs enthalten sein. Beim ersten Versuch der Implementierung dieses Steuerungsmodells mit Rotation und Trägheit im zweidimensionalen Raum schaffte ich es allerdings, aufgrund eines Denkfehlers, nur die Geschwindigkeit beizubehalten. Das Raumschiff flog immer nur in Blickrichtung. Allerdings habe ich dieses dabei entstandene Steuerungsmodell weiterverwendet und daraus die „Fahrradsteuerung“ entwickelt.

Bei den ersten eigenen Testläufen mit dieser Steuerung hatte ich dennoch den Eindruck, dass es sich dabei um ein sehr realistisches Flugverhalten handelt. Bei näherer Betrachtung erwies sich diese Annahme als fehlerhaft. Daher fragte ich mich, welche Steuerungsmodelle realistisch wirken und wie Testkandidaten auf die Frage nach dem Realismus antworten würden. Um den Kandidaten keine Hinweise auf den Realismus der einzelnen Steuerungen zu geben, wurden neutrale Bezeichnungen in der Steuerungsauswahl und auf dem Fragebogen gewählt. Dabei kam es zu folgenden Zuordnungen:

Bezeichnung der Steuerungen gemäß Kap. 4	Bezeichnung für die Testdurchläufe
absolute Steuerung	2D-Steuerung (A)
Direkte Steuerung – Variante a)	2D-Steuerung (B)
Direkte Steuerung – Variante b)	2D-Steuerung (C)
Fahrradsteuerung	2D-Steuerung (D)
Realistische Steuerung	2D-Steuerung (E)
Energie-Steuerung	2D-Steuerung (F)
Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit	3D-Steuerung (A)
Steuerung ohne Geschwindigkeitsabstufungen, aber mit Trägheit	3D-Steuerung (B)
Steuerung mit Geschwindigkeitsabstufungen, aber ohne Trägheit	3D-Steuerung (C)
Steuerung mit Geschwindigkeitsabstufungen, und mit Trägheit	3D-Steuerung (D)

5.3 Objektiver Abschnitt des Fragebogens

Da sich gerade die 3D-Steuerungsmodelle in Tests als teilweise sehr kompliziert und schwierig zu steuern herausstellten, entschied ich mich dafür, nicht alle implementierten Steuerungsmodelle auch für den objektiven Test zu verwenden. Ausschlaggebend hierfür war der hohe Zeitaufwand pro Test der notwendig gewesen wäre, um aussagekräftige Werte zu ermitteln. Es wäre zwar möglich gewesen, weniger Durchläufe mit jeder einzelnen Steuerung durchzuführen, aber dabei wäre dann ein entsprechendes Lernverhalten der einzelnen Testteilnehmer nicht erkennbar gewesen. Aber genau um dieses Lernverhalten geht es bei diesem objektiven Testabschnitt.

Die für die objektiven Tests ausgesuchten Steuerungen sind:

- 2D-Steuerungsmodelle
 - absolute Steuerung
 - direkte Steuerung – Variante b)
 - Fahrradsteuerung
 - realistische Steuerung
- 3D- Steuerungsmodelle
 - Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit
 - Steuerung mit Geschwindigkeitsabstufungen und mit Trägheit

Jedem Kandidaten wurde für jedes der ausgesuchten Steuerungsmodelle eine Einarbeitungsrunde gestattet. Dabei sollte der Kandidat das Raumschiff zu den unterschiedlichen Zielpunkten in der Testszene fliegen und von dort aus weiter zum jeweils nächsten Zielpunkt. Sowohl in der 2D- wie auch der 3D-Umgebung sind insgesamt fünf Zielpunkte eingefügt. Diese erscheinen in einer festen Reihenfolge und gelten als erreicht, sobald das Raumschiff entsprechend dicht an den Zielpunkt heran geflogen ist. Dabei sind diese Zielpunkte so angelegt, dass das Raumschiff durchaus um einen Planeten herumgeflogen werden sollte, um den entsprechenden Zielpunkt zu erreichen. Einige Zielpunkte hatten einen geringeren Abstand zu den Planeten, so dass das Raumschiff besonders dicht an den Gravitationspunkt heran fliegen musste.

Bereits bei der Einarbeitungsrunde wurden die Zeiten gemessen, die der Kandidat benötigte, um zu den einzelnen Zielpunkten zu gelangen. Diese Einarbeitungszeiten sind aber für die Auswertung nur dahingehend relevant, um zu erkennen, ob Testkandidaten, die sich sehr viel Zeit bei der Einarbeitungsphase genommen haben, ein konstanteres Ergebnis bei den nachfolgenden Testdurchläufen erzielen konnten.

Die Testkandidaten sollten dann mit den ausgesuchten Steuerungsmodellen jeweils fünf weitere Testdurchläufe absolvieren. Auch bei diesen Testdurchläufen wurden die Zeiten zum Erreichen der einzelnen Zielpunkte gemessen.

6 Auswertung der Fragebögen

Dieses Kapitel beschreibt die unterschiedlichen Ergebnisse der Testumfragen anhand des entwickelten Fragebogens, sowie bei der Auswertung aufgetretene Probleme. Für die Tests wurden insgesamt 60 Personen, überwiegend Studenten, befragt. Die Befragungen und die Testdurchläufe wurden großteils in Poolräumen der Universität durchgeführt. Die meisten Kandidaten wurden dafür einzeln befragt und konnten alleine die einzelnen Steuerungen testen, so dass kein Proband von anderen Testkandidaten beeinflusst wurde. Der subjektive Testanteil wurde von allen 60 Kandidaten durchgeführt. Von diesen 60 Kandidaten haben 15 Personen zusätzlich den objektiven Test absolviert.

6.1 Probleme bei den ersten Tests

Bei den ersten Testdurchläufen mit unterschiedlichen Testkandidaten ergaben sich ein paar vorher nicht bedachte Probleme. Ich selber hatte das verwendete Testszenario mehrfach durchgespielt und dabei den Joystick immer nur recht behutsam ausgeschlagen oder häufig auf die Tastatursteuerung zurückgegriffen. Ebenso war mir das entsprechende Verhalten der einzelnen Steuerungen sehr gut bewusst und ich habe meine Steuerversuche dahingehend angepasst, dass ich nicht sonderlich aggressiv gesteuert und den Joystick nicht sofort voll ausgeschlagen habe. Dadurch kam ich bei den Testflügen oft deutlich dichter an den Gravitationspunkten vorbei, so dass die Gravitation einen starken Einfluss auf die Flugbahn nehmen konnte und das Raumschiff spürbar abgelenkt hat.

Einige der Testkandidaten hingegen steuerten das Raumschiff deutlich aggressiver, wodurch die Gravitationskraft der einzelnen im Testszenario verteilten Planeten nicht im selben Umfang wie bei meinen eigenen Testdurchläufen wahrgenommen wurde. Daher musste ich die Anordnung und auch die unterschiedlichen Massen der einzelnen Planeten dahingehend anpassen, dass auch bei solch aggressivem Steuerungsverhalten die Gravitation einen größeren Einfluss auf die Flugbahnen hatte.

Des Weiteren ließ ich bei den ersten Testdurchläufen mit den Probanden die unterschiedlichen Zielpunkte für den subjektiven Anteil der Testdurchläufe weg. Das hatte allerdings zur Folge dass viele der Testkandidaten nicht genau wussten was sie tun sollten. Daher habe ich auch für die subjektiven Tests die einzelnen Zielpunkte eingefügt, um den Kandidaten somit ein entsprechendes Ziel zu geben.

6.2 Auswertung der Ergebnisse des subjektiven Testanteils

6.2.1 Erwartete Ergebnisse

Für die Auswertung des subjektiven Anteils des Tests hatte ich erwartet, dass besonders die Kandidaten, welche schon vorher Erfahrungen mit unterschiedlichen Weltraumflugsimulationen hatten, die Steuerungen als besonders intuitiv empfanden. Auch erwartete ich, dass diese Kandidaten keine lange Einarbeitungszeit für die einzelnen Steuerungsmodelle benötigten. Zusätzlich nahm ich an, dass besonders die unterschiedlichen Steuerungsmodelle, in welchen das Prinzip der Trägheit umgesetzt wurde, als besonders schwer zu steuern empfunden würden und somit die Trägheit auch von vielen Testkandidaten als mit eines der größten Probleme beim Steuern angesehen würde.

Im Folgenden stelle ich eine Sortierung der einzelnen Steuerungsmodelle nach ihrer Schwierigkeit vor, wie ich sie als Ergebnis der Auswertung erwartet hatte. Dabei war ich mir nicht sicher, an welcher Stelle die Energie-Steuerung einzuordnen wäre. Diese empfand ich selber nur in der Anfangsphase als sehr kompliziert. Nachdem man allerdings verstanden hat wie das Raumschiff zu steuern ist, ist diese recht einfach zu beherrschen. Da ich aber durch die Implementierung über die Kenntnisse der genauen Umsetzung verfügte und auch mehr Zeit hatte, den Umgang mit der Steuerung zu erlernen, schätzte ich diese Steuerung für die Testkandidaten als etwas schwieriger ein. Generell hatte ich folgende Sortierung der einzelnen Steuerungsmodelle bezüglich ihres Schwierigkeitsgrads erwartet:

(nach steigendem Schwierigkeitsgrad von „leicht“ bis „schwer“ sortiert)

- Direkte Steuerung – Variante a)
- Direkte Steuerung – Variante b)
- Fahrradsteuerung
- Steuerung mit Geschwindigkeitsabstufungen, aber ohne Trägheit
- Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit
- Absolute Steuerung
- Energie-Steuerung
- Realistische Steuerung
- Steuerung mit Geschwindigkeitsabstufungen und mit Trägheit
- Steuerung ohne Geschwindigkeitsabstufungen, aber mit Trägheit

Da ich mich selber bezüglich des realistischen Flugverhaltens geirrt hatte und während der Implementierung der realistischen Steuerung zuerst fälschlicherweise die Fahrradsteuerung als realistisches Steuerungsmodell angesehen hatte, erwartete ich auch von einigen Testkandidaten, diesem Irrtum zu unterliegen. Generell vermutete ich, dass sehr viele Testkandidaten die realistische Steuerung bei den zweidimensionalen Steuerungsmodellen, und die beiden

unterschiedlichen dreidimensionalen Steuerungsmodelle mit dem Prinzip der Trägheit als tatsächlich realistisch ansehen würden. Allerdings erwartete ich, gerade aufgrund meines eigenen Irrtums, dass einige Probanden auch die Fahrradsteuerung angeben würden. Ebenfalls erwartete ich Nennungen für die beiden dreidimensionalen Steuerungsmodelle ohne Trägheit, da gerade diese beiden Modelle den bekanntesten Weltraumflugsimulationen nachempfunden sind.

6.2.2 Tatsächliche Ergebnisse

Verhältnis zwischen Vorkenntnissen und tatsächlichen Ergebnissen

Von den 60 befragten Personen gaben 16 Kandidaten an, dass sie *Asteroids* gespielt hätten. *Lunar Lander* hingegen hatten nur 5 Probanden jemals gespielt. Keiner dieser Testkandidaten hatte Schwierigkeiten bei der Einarbeitung in die zweidimensionalen Steuerungsmodelle. Die einzige Ausnahme dabei stellt die Energie-Steuerung dar. Allerdings bietet keines der erwähnten 2D-Spiele eine ähnliche Steuerung. Daher ist diese Steuerung bei einem möglichen Zusammenhang zwischen Vorkenntnissen durch Computerspiele auszuklammern.

Im Bereich der 3D-Simulationen ergab sich folgendes Problem. 22 Kandidaten gaben an, dass sie mindestens eine der genannten Simulationen bereits gespielt hatten. Allerdings gaben auch einige Kandidaten an, dass sie die Spiele vom Namen und Spielprinzip kennen würden, aber sie niemals gespielt hätten. Zusätzlich nannten einige aber noch weitere 3D-Weltraumsimulationen, die ich nicht in der Liste aufgeführt hatte. Dazu zählten aktuellere Titel wie *Freelancer* oder *X: Beyond the Frontier*. Daher bat ich die entsprechenden Testkandidaten, die genannten Spiele zusätzlich auf dem Fragebogen anzugeben. Daraus ergab sich, dass insgesamt 41 Personen mindestens eine 3D-Weltraumsimulation gespielt hatten. Von diesen 41 Kandidaten gab keiner an, dass er Probleme bei der Einarbeitung zu den unterschiedlichen dreidimensionalen Steuerungsmodellen hatte.

Insgesamt gaben nur 7 Probanden an, dass sie Probleme bei der Einarbeitung in die verschiedenen Steuerungsmodelle hatten. Von diesen hatte keiner jemals eines der erwähnten Computerspiele gespielt. Generell lässt sich somit sagen, dass entsprechende Vorkenntnisse einzelner Weltraumsimulationen zwar durchaus hilfreich für das Verstehen und die einfache Anwendung der unterschiedlichen implementierten Steuerungen sind. Allerdings sind diese Vorkenntnisse nicht notwendig, da auch viele Kandidaten ohne diese Vorkenntnisse keinerlei Probleme bei der Einarbeitung hatten.

Verhältnis der einzelnen Schwierigkeitsempfindungen

Nach Angaben der Kandidaten lassen sich die unterschiedlichen Steuerungsmodelle wie in Abbildung 6.1 nach ihrem Schwierigkeitsgrad sortieren:

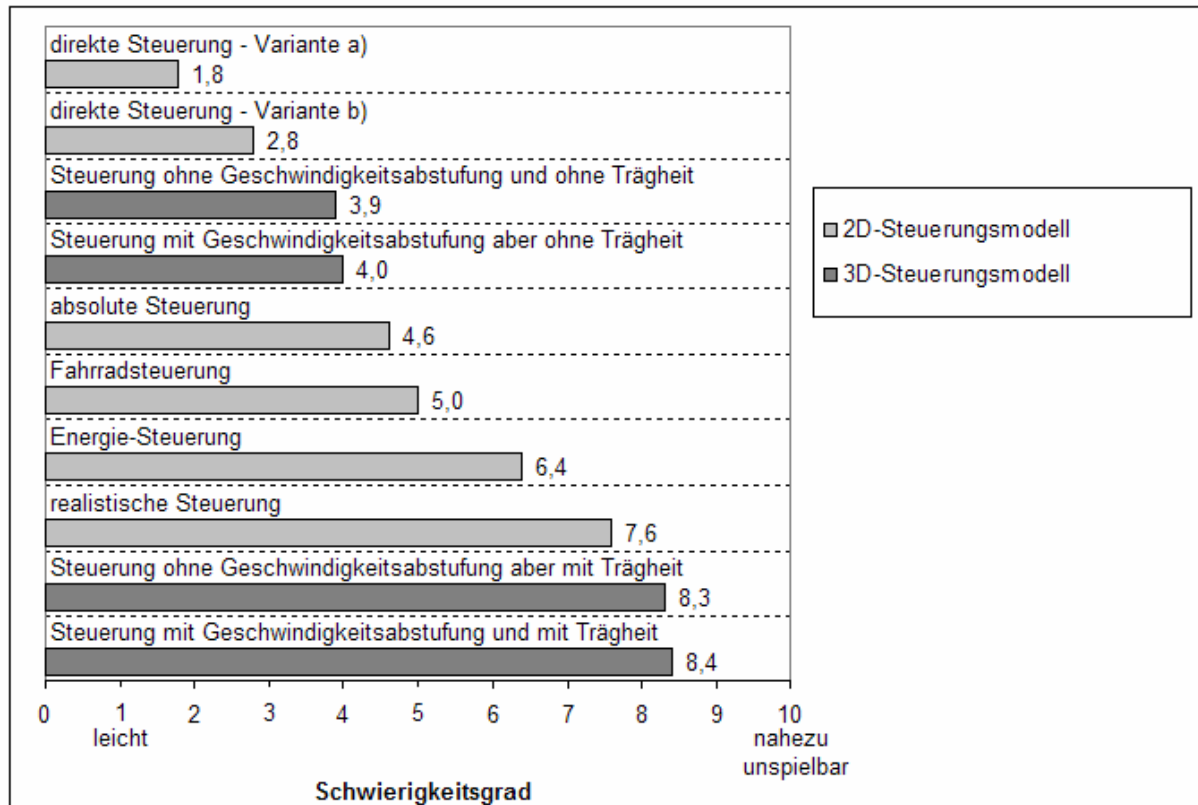


Abb.6.1 durch Auswertung erhaltene Sortierung nach Schwierigkeitsgrad

Dabei ist zu erkennen, dass besonders die Fahrradsteuerung den Kandidaten deutlich mehr Schwierigkeiten bereitete, als ich vorher angenommen hatte. Ansonsten deckt sich das Empfinden der einzelnen Schwierigkeitsgrade in etwa mit meiner Annahme. In den dreidimensionalen Umsetzungen hatte ich angenommen, dass die Varianten mit unterschiedlichen Geschwindigkeitsabstufungen als leichter wahrgenommen würden als die Varianten ohne diese Abstufungen. Hier wurden die Schwierigkeitsgrade jedoch fast identisch wahrgenommen. Die Steuerungsmodelle ohne Geschwindigkeitsabstufung wurden nur als minimal leichter empfunden.

Probleme bei der Steuerung

Für 56 Testkandidaten stellten Trägheit und das Verhalten des Raumschiffs unter Wirken der Trägheit das größte Problem dar. Davon präzisierten 17 Personen das Problem noch dahingehend, dass sie das Prinzip von Schub und Gegenschub als äußerst schwierig erachteten und

sehr große Probleme hatten, das Raumschiff zum Stillstand zu bringen. Besonders in Bezug auf die dreidimensionalen Steuerungsmodelle mit Trägheitsverhalten nannten die Testkandidaten die Prinzipien von Trägheit und Gravitation als besonders großes Problem für die Steuerung.

Zusätzlich war für 47 Kandidaten auch eine fehlende Orientierung in den dreidimensionalen Umgebungen ein großes Problem. Manche gaben die Anregung, eventuell mehr unterschiedliche Planeten oder aber eine andere Möglichkeit der Orientierung einzufügen. Außerdem gaben 6 Testkandidaten an, dass sie sich bei den 3D-Steuerungsmodellen eine Art Zielkreuz wünschten, das anzeigt, wo genau sie hinfliegen. Dieses sollte dazu dienen, das Zielobjekt schon aus größerer Entfernung anzuvisieren und damit ein einfacheres Erreichen der Zielobjekte zu gewährleisten.

Realismus der einzelnen Steuerungsmodelle

Einige der Testkandidaten gaben bei der Frage nach dem Realismus der unterschiedlichen Steuerungsmodelle auch Mehrfachnennungen an. Dabei wurden für die dreidimensionalen Steuerungsmodelle die beiden Modelle genannt, in denen auch die Trägheit implementiert wurde (siehe Kapitel 4.2.2 und 4.2.4). Beide Modelle wurden von insgesamt 35 Kandidaten genannt. Von den Steuerungsmodellen ohne Trägheit wurde das Modell mit Geschwindigkeitsabstufungen fünfmal genannt.

Bei den zweidimensionalen Steuerungsmodellen entschieden sich 48 Kandidaten für die „realistische Steuerung“ (siehe Kapitel 4.1.4). Meine Vermutung, dass sich einige Kandidaten auch für die Fahrradsteuerung als das realistischste Steuerungsverhalten entschieden, wurde allerdings widerlegt. Diese Steuerung wurde von keinem der befragten Testkandidaten angegeben.

Allerdings wurde auch von drei Kandidaten die „Energie-Steuerung“ (siehe Kapitel 4.1.5) als realistisches Steuerungsmodell genannt. Mit diesen Nennungen hatte ich vorher nicht gerechnet und fragte daher die betreffenden Kandidaten zusätzlich, warum sie sich dafür entschieden haben. Dabei stellte sich heraus, dass sie dabei nicht unbedingt die Steuerung an sich als realistisch ansahen, allerdings die Flugbahnen des Raumschiffs für sie am plausibelsten erschienen. Daher nahmen sie an, dass dieses Steuerungsmodell, neben den anderen genannten Modellen, mit am realistischsten sei.

6.3 Auswertung der Testergebnisse der objektiven Tests

6.3.1 Erwartete Ergebnisse

An dieser Stelle war es interessant zu beobachten, ob es einen Zusammenhang zwischen guten Ergebnissen bei den einzelnen Zeitflügen der Kandidaten und Vorkenntnissen durch schon früher gespielte Weltraumsimulationen gibt. Ich hatte erwartet, dass gerade die Ergebnisse von Kandidaten, die eben solche Simulationen schon gespielt hatten, besser ausfallen würden als die von Kandidaten ohne solches Vorwissen. Dabei erwartete ich speziell bei Spielern von *Lunar Lander* oder *Asteroids* bessere Ergebnisse für die zweidimensionalen Steuerungsmodelle, wo hingegen ich bei Spielern mit Kenntnissen von etwa *Elite*, *Wing Commander* oder *Star Wars: X-Wing* bessere Ergebnisse bei den dreidimensionalen Steuerungsmodellen erwartete.

Für die Ergebnisse des objektiven Anteils des Tests war mir schnell klar, dass ich nicht die unterschiedlichen Testzeiten der verschiedenen Steuerungsmodelle miteinander vergleichen, sondern nur jedes Steuerungsmodell für sich betrachten durfte. Zwar wäre es möglich gewesen, die einzelnen zweidimensionalen Modelle untereinander zu vergleichen und auch die dreidimensionalen Modelle für sich genommen untereinander. Aufgrund der unterschiedlichen Dimensionen und Entfernungen innerhalb des 2D- bzw. 3D-Szenarios wären Vergleiche zwischen den Ergebnissen eines 2D-Testdurchlaufes und eines 3D-Testdurchlaufes aber nicht aussagekräftig.

Allerdings erwartete ich, dass sich die einzelnen Zeiten für jeden Testdurchlauf pro Steuerungsmodell mit der Zeit verbessern würden. Somit könnte ein entsprechendes Lernverhalten beobachtet werden.

6.3.2 Tatsächliche Ergebnisse

Bessere Ergebnisse bei Probanden mit Spielerfahrungen?

Von den 15 Testkandidaten, welche auch den subjektiven Test gemacht haben, hatten 9 Kandidaten schon vorher Erfahrungen mit verschiedenen 3D-Weltraumsimulationen gesammelt. Von diesen 9 Kandidaten kamen 4 Probanden tatsächlich mit der dreidimensionalen Steuerungsvariante ohne Trägheit sehr gut zurecht. Bei ihnen fiel die Eingewöhnungsrunde recht kurz aus und auch in den weiteren Testrunden waren die Ergebnisse nahezu konstant. In Abbildung 6.2 sind beispielhaft die Ergebnisse eines dieser Kandidaten zu sehen. Dabei handelt es sich um die Ergebnisse für die dreidimensionale Steuerung ohne Geschwindigkeitsabstufungen und ohne Trägheit. Das dreidimensionale Steuerungsmodell mit Trägheit hingegen bereitete auch ihnen teilweise deutliche Probleme. Das führte dazu, dass die Zeit zum Erreichen des ersten Zielpunktes noch sehr ähnlich der Zeit zum Erreichen des ersten Zielpunktes in dem Steuerungsmodell ohne Trägheit war. Die weiteren Zielzeiten variierten jedoch teilweise sehr stark.

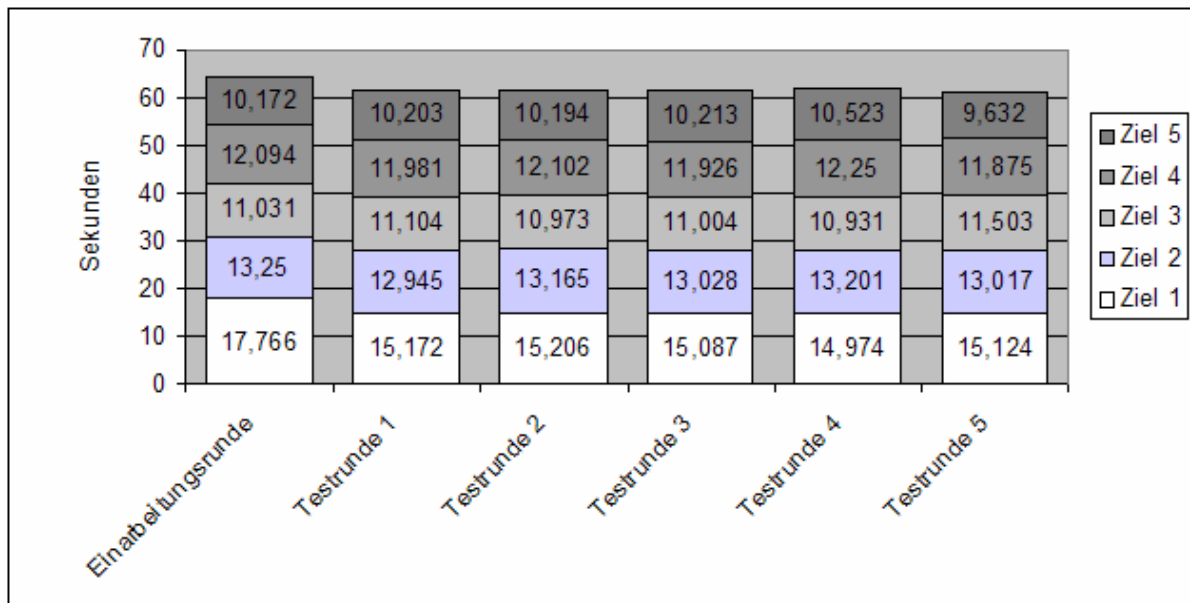


Abb. 6.2 Ergebnisse eines Kandidaten für die dreidimensionale Steuerung ohne Geschwindigkeitsabstufung und ohne Trägheit

Ein ähnliches Verhalten war bei den zweidimensionalen Steuerungsmodellen nicht erkennbar. Nur 3 Probanden gaben an, *Lunar Lander* oder *Asteroids* jemals gespielt zu haben. Von den verbleibenden 12 Personen kannten zumindest 8 Kandidaten die Spiele vom Namen her, hatten sie aber niemals gespielt. Selbst bei den 3 Kandidaten, die Erfahrungen mit den genannten Spielen hatten, waren die Ergebnisse leider nicht aussagekräftig genug. Einer der dieser Kandidaten erzielte sehr gute und auch konstante Ergebnisse. Von den anderen beiden konnte sich einer zumindest kontinuierlich verbessern. Bei dem dritten hingegen schwankten die Ergebnisse zu stark. Es war daher nicht möglich, einen tatsächlichen Zusammenhang zwischen den vorhandenen Vorkenntnissen und den Zeiten zum Erreichen der einzelnen Zielpunkte herzustellen.

Lernverhalten erkennbar?

Wie schon erwähnt erreichten einige der Testkandidaten nahezu konstante Ergebnisse bei ihren Testdurchläufen, so dass bei diesen kein entsprechendes Lernverhalten zu erkennen war. Bei vier der Kandidaten hingegen schwankten die Ergebnisse bei den zweidimensionalen Simulationen so stark, dass auch bei diesen kein Lernverhalten für die einzelnen Steuerungen zu beobachtet werden konnte. Diese Schwankungen traten besonders bei der realistischen 2D-Steuerung auf. Bei drei Kandidaten waren diese Schwankungen auch in den Ergebnissen für die dreidimensionalen Steuerungsmodelle vorhanden.

Einige Kandidaten zeigten tatsächlich anhand ihrer Ergebnisse kontinuierliche Verbesserungen, so dass ein Lernverhalten bezüglich der einzelnen Steuerungsmodelle zu beobachten war. So verbesserten sich bei 8 Kandidaten die einzelnen Zeiten zum Erreichen der Zielpunkte mit den 2D-Steuerungen kontinuierlich, je mehr Testdurchläufe sie absolviert hatten. Abbildung 6.3 zeigt den Testverlauf eines dieser Kandidaten. Von diesen 8 Kandidaten hatten 6

Personen keinerlei Erfahrungen mit Weltraumsimulationen in zweidimensionaler Umgebung. Bei den dreidimensionalen Simulationen war lediglich bei 4 Kandidaten eine kontinuierliche Verbesserung ihrer Zeiten zu erkennen. Zwei dieser Kandidaten hatten auch schon vorher Erfahrungen mit verschiedenen 3D-Simulationen gesammelt.

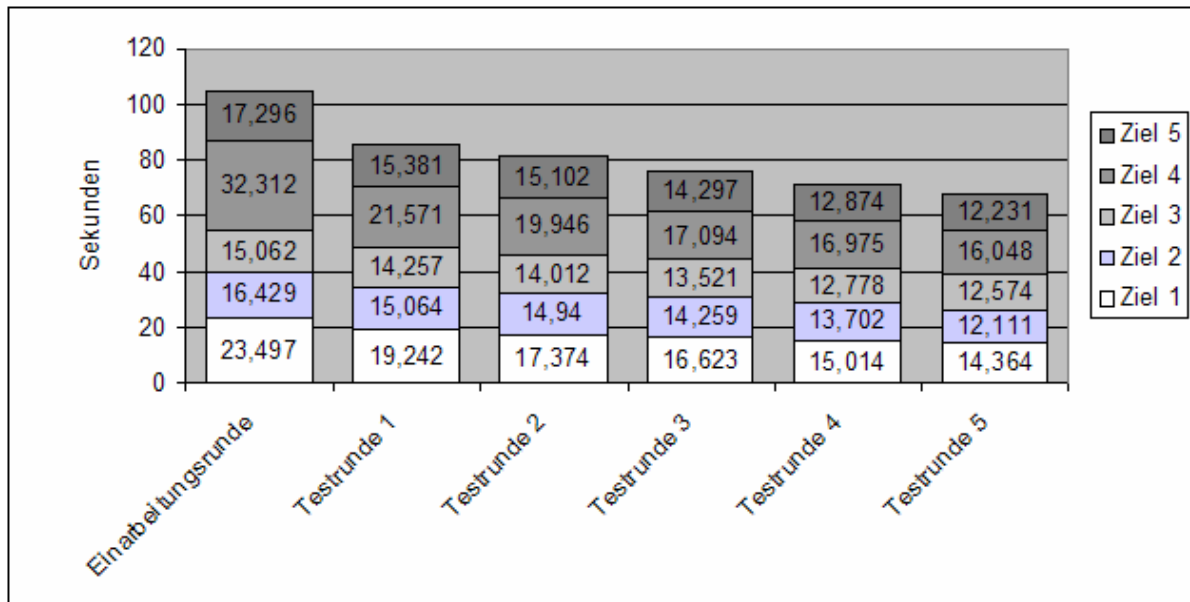


Abb. 6.3 Ergebnisse eines Kandidaten für die „absolute Steuerung“ im zweidimensionalen Raum

Zusammenhang Lernverhalten – Einarbeitungsphase

Es zeigte sich generell bei den Kandidaten, dass sie bei Varianten ohne Trägheit recht schnell zurechtkamen und die Einarbeitungsphasen recht kurz waren. Im Gegensatz dazu ließen sich alle 15 Kandidaten bei den Steuerungsmodellen mit Trägheit deutlich mehr Zeit, um sich an die Steuerungen zu gewöhnen.

Gerade bei diesen langen Eingewöhnungsrunden bestand die Vermutung, dass dadurch die nachfolgenden Testrunden konstante Ergebnisse oder zumindest eine kontinuierliche Verbesserung erkennbar machten. Allerdings war ein solcher Zusammenhang bei nur 3 Kandidaten zu erkennen. Von den Kandidaten konnten 12 Personen gerade bei den Steuerungsmodellen ohne Trägheit konstante Ergebnisse bzw. eine kontinuierliche Verbesserung erzielen. Diese Probanden begannen aber auch schon mit einer recht kurzen Eingewöhnungsrunde. Somit kann kein Zusammenhang zwischen einer verlängerten Eingewöhnungsphase und konstanten Ergebnissen oder kontinuierlichen Verbesserungen bei den unterschiedlichen Steuerungsmodellen geschlussfolgert werden.

7 Fazit und Ausblick

Bei dem Umgang mit den für diese Arbeit umgesetzten Steuerungen hat sich gezeigt, dass alle Probanden, welche schon vorher Erfahrungen mit Weltraumsimulationen hatten, keine Einstiegsschwierigkeiten hatten. Allerdings kamen auch sofort sehr viele Testkandidaten ohne solche Vorkenntnisse gut mit den einzelnen Steuerungen zurecht.

Bei der Auswertung des Fragebogens ergab sich die These, dass je realistischer ein Steuerungsmodell tatsächlich ist, desto schwieriger ist die Handhabung. In den wenigsten kommerziellen Computerspielen, die sich mit dem Thema Raumfahrt oder Weltraumsimulation beschäftigen, werden die physikalischen Gesetze der Gravitation und der Trägheit behandelt. Diese würden den unterschiedlichen Spielen zwar zu einem höheren Grad an Realismus verhelfen, allerdings würde dadurch der Schwierigkeitsgrad der entsprechenden Spiele deutlich ansteigen. Spiele wie *Lunar Lander* oder *Asteroids* beruhen auf diesen Prinzipien. Bei den meisten 3D-Weltraumsimulationen steht aber nicht der Realismus im Vordergrund, sondern der Spielspass. Daher wird auf den Realismus zugunsten einer kurzen Lernphase verzichtet.

Der Schwierigkeitsgrad der einzelnen Steuerungen lässt sich aber nicht nur durch Einbindung von physikalischen Gesetzen anheben, sondern auch schon die Umstellung von einem zweidimensionalen auf ein dreidimensionales Szenario erhöht den Schwierigkeitsgrad der Steuerung. Ein wichtiger Grund dafür ist etwa eine fehlende Orientierung.

Bei einer Weiterentwicklung der einzelnen Steuerungsmodelle und Simulationen wäre interessant zu untersuchen, ob tatsächlich eine bessere Orientierung in dreidimensionalen Umgebungen den Schwierigkeitsgrad senken würde. Eine Überlegung in dieser Hinsicht ist, dass für die dreidimensionalen Steuerungsmodelle eine Richtungsanzeiger oder eine Art Radarschirm implementiert werden könnte. Damit könnten die Kandidaten jederzeit auf zusätzliche Informationen bezüglich der Zielpunkte und ihrer Position zugreifen. Eine weitere Verbesserung wäre, dass mittels der Maus die Kamera um das Raumschiff rotiert werden könnte, dabei die Ausrichtung des Raumschiffs aber nicht verändert wird. Auch dies könnte zur Verbesserung der Orientierung in der dreidimensionalen Umgebung beitragen.

8 Literaturverzeichnis

- [AH09] Alexis Bousiges – arcade history
<http://www.arcade-history.com/>
(abgerufen am 26.08.2009)
- [Fla97] Paul Flavin – Java Application Lunar Lander
http://www.frontiernet.net/~imaging/lunar_lander_game.html
(abgerufen am 24.08.2009)
- [Kuc96] Horst Kuchling – Taschenbuch der Physik,
Fachbuchverlag Leipzig im Carl Hanser Verlag München Wien, 1996
- [MG09] MobyGames
<http://www.mobygames.com/>
(abgerufen am 26.08.2009)
- [New88] Isaac Newton, Ed Dellian (Hrsg.) – Mathematische Grundlage der Naturphilosophie, Felix Mainer Verlag GmbH, Hamburg 1988
- [Rou03] Christian Rousselle – jetzt lerne ich : DirectX 9 und Visual C++,
Markt + Technik Verlag 2003
- [Sch09] Thorsten Schreck – C64-Wiki
[http://www.c64-wiki.de/index.php/Elite_\(Spiel\)](http://www.c64-wiki.de/index.php/Elite_(Spiel))
(abgerufen am 26.08.2009)
- [TS09] TurboSquid Inc. – Cobra MK II Modell
<http://www.turbosquid.com/3d-models/cobra-mk-ii-fighter-space-3ds/256898>
(abgerufen am 24.08.2009)
- [UPB09] Universität Paderborn – AG Meyer auf der Heide
<http://www.whni.uni-paderborn.de/alg/>
(abgerufen am 24.08.2009)

[WFI09] Wikimedia Foundation Inc. - wikimediafoundation.org
<http://de.wikipedia.org/wiki/Gravitationskonstante>
(abgerufen am 26.08.2009)

9 Anhang

9.1 Steuerung der potentiellen Energie separiert von der Navigation auf der Energiefläche

(siehe nächste Seite)

9.2 Fragebogen:

Haben Sie schon mal folgende Spiele gespielt?
(zutreffendes bitte ankreuzen)

Spiele im 2D		
	Ja	Nein
Lunar Lander		
Astroids		
Spiele im 3D		
	Ja	Nein
Wing Commander		
Star Wars: X-Wing		
Elite		

Kamen Sie sofort gut mit den umgesetzten Steuerungsmodellen zurecht?

Bewerten Sie die Steuerungen bezüglich der Schwierigkeit auf einer Skala von 1 bis 10 („1 für sehr einfach“ bis „10 nahezu unsteuerbar“)

	1	2	3	4	5	6	7	8	9	10
2D-Steuerung (A)										
2D-Steuerung (B)										
2D-Steuerung (C)										
2D-Steuerung (D)										
2D-Steuerung (E)										
2D-Steuerung (F)										
3D-Steuerung (A)										
3D-Steuerung (B)										
3D-Steuerung (C)										
3D-Steuerung (D)										

Kamen Sie gut mit den unterschiedlichen Steuerungsarten zurecht?

- Wenn nein, womit genau hatten Sie das größte Problem?

Welche Steuerung empfinden Sie für die physikalisch realistischste?

9.3 Datenträger:

Inhalt des Datenträgers:

2D-Implementierung

 Quellcode

 ausführbares Programm

3D-Implementierung

 Quellcode

 ausführbares Programm

Dokumente

 diese Ausarbeitung (Bachelorarbeit Lucas Jillek.pdf)

 Formeln für die Energie-Steuerung (energy.pdf)

 Fragebogen (Fragebogen.pdf)