

Hochschule Mittweida

Fakultät Angewandte Computer- und Biowissenschaften



Bachelorarbeit zum Thema:

**Interaktive Partikelsimulationen - Eine Untersuchung der  
Visualisierung astrophysikalischer Prozesse**

Vorgelegt von:

**Hans Standfest**

Studiengang:

**Medieninformatik & interaktives Entertainment**

Seminargruppe:

**Mi20w3-b**

Erstprüfer:

**Prof. Dr.-Ing. Christian Roschke**

Zweitprüfer:

**B.Sc. Joshua Queck**

Einreichung:

**Berlin, 08.01.2024**

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis.....</b>	<b>I</b>
<b>1 Einleitung.....</b>	<b>1</b>
<b>2 Astrophysikalische Grundlagen.....</b>	<b>3</b>
2.1 Grundbegriffe.....	3
2.2 Klassifizierung.....	4
2.3 Stellare Evolutionspfade.....	6
2.4 Verhältnismäßigkeiten.....	7
2.4.1 Absolute Helligkeit.....	7
2.4.2 Spektralklasse.....	8
2.4.3 Lebenszeit.....	8
2.5 Sternentstehung.....	9
2.5.1 Interstellare Wolken.....	9
2.5.2 Protostern.....	10
2.5.3 Hauptreihenstern.....	11
2.5.4 Brauner Zwerg - fehlgeschlagener Stern.....	13
2.6 Sternverfall.....	14
2.6.1 Roter Riese - massearme Sterne.....	14
2.6.2 Roter Überriese - massereiche Sterne.....	15
2.6.3 Neutronenstern.....	16
2.6.4 Schwarzes Loch.....	17
<b>3 Informationsvermittlung durch interaktive Simulationen.....</b>	<b>18</b>
3.1 Visualisierung.....	18
3.2 Interaktion.....	19
3.3 Dynamik.....	20
3.4 Die Rolle des Experimentierens im Physikunterricht.....	20
<b>4 Simulationen als Mittel zur Prozessvisualisierung.....</b>	<b>22</b>
4.1 Partikelsimulationen und deren Anwendungsgebiete.....	22
4.1.1 Anwendungsgebiet Wissenschaft.....	22
4.1.2 Anwendungsgebiet Filmindustrie.....	23
4.1.3 Anwendungsgebiet Spieleindustrie.....	24
4.2 Houdini als Simulationstool - Aufbau der Umgebung.....	25
4.3 POPs - Grundprinzipien von Partikelsimulationen in Houdini.....	28
4.4 HDAs - Der Aufbau eigener Nodes als Werkzeuge.....	32
4.5 Houdini im Vergleich mit anderen Möglichkeiten der Visualisierung.....	33
<b>5 Erstellung einer interaktiven Simulation des stellaren Lebenszyklusses.....</b>	<b>35</b>
5.1 Rahmenbedingungen.....	35
5.2 Angestrebtes Ziel.....	36

5.3 Vordiskussion.....	37
5.4 Erstellung der Simulationen.....	41
5.4.1 Interstellare Wolke.....	41
5.4.2 Protostern.....	43
5.4.3 Teilbereiche der Sphärischen Objekte.....	45
5.4.4 Hauptreihensterne.....	47
5.4.5 Roter Riese, Roter Überriese.....	48
5.4.6 Weißer Zwerg.....	49
5.4.7 Schwarzes Loch.....	49
5.4.8 Neutronenstern.....	50
5.4.9 Helium Blitz.....	51
5.4.10 Supernova.....	52
5.4.11 Zusatzinformationen.....	52
5.5 Logik und Verarbeitung von Usereingaben.....	53
5.5.1 Struktur.....	53
5.5.2 Logik.....	54
5.6 Ergebnis.....	56
<b>6 Fazit.....</b>	<b>59</b>
<b>Literaturverzeichnis.....</b>	<b>62</b>
<b>Anhang.....</b>	<b>66</b>
<b>Eidesstattliche Erklärung.....</b>	<b>76</b>

# Abbildungsverzeichnis

Abbildung 1: "Spektralklassen der Sterne"	4
Abbildung 2: "Hertzprung Russel Diagramm"	5
Abbildung 3: "Stellare Evolutionspfade"	6
Abbildung 4: "Verhältnis von Masse und Spektralklasse"	8
Abbildung 5: "Aufbau eines Protosterns"	10
Abbildung 6: "Aufbau der Sonne"	11
Abbildung 7: "Konvektion und Strahlung in Sternen"	12
Abbildung 8: "Struktur des Kernes eines Roten Überriesen"	15
Abbildung 9: "Aufbau eines Neutronensterns"	16
Abbildung 10: "Aufbau eines schwarzen Lochs"	17
Abbildung 11: "Modell des Text- und Bildverstehens"	18
Abbildung 12: "Beispiel für prozedurale Operation in Houdini"	25
Abbildung 13: "Übersicht über die Houdini Benutzeroberfläche"	26
Abbildung 14: "Aufbau eines Popnets"	28
Abbildung 15: "Beispiel Für VEX-Operation mit if-Abfrage"	30
Abbildung 16: "Arbeitsblatt für digitales Schülerexperiment zur Sternentwicklung"	75
Abbildung 17: "Aufnahmen eines Schwarzen Lochs, Protosterns, Neutronensterns"	37
Abbildung 18: "Schematische Darstellung der erwarteten Probleme mit Lösungsansatz"	40
Abbildung 19: "Rosettennebel, NGC 2237"	41
Abbildung 20: "Grundkörper der Schichten der interstellaren Wolke"	66
Abbildung 21: "Farbanpassung mit Hilfe einer Ramp"	66
Abbildung 22: "Protostern (W33A) mit Jets und Akkretionsscheibe"	43
Abbildung 23: "Spektralklassen der Hauptreihensterne"	47
Abbildung 24: "Kern des Überriesen"	48
Abbildung 25: "Farbverlust des weißen Zwerges"	49
Abbildung 26: "Neutronenstern mit Pulsaren"	50
Abbildung 27: "Benutzeroberfläche des HDA"	74
Abbildung 28: "innere Struktur des HDA"	53
Abbildung 29: "Die fünf möglichen evolutionären Pfade"	54
Abbildung 30: "Übersicht über die Teilbereiche der Simulation"	56
Abbildung 31: "Struktur der Funktionen"	73
Abbildung 32: "Simulation des Hauptreihensternes (Spektralklasse G)"	71
Abbildung 33: "Simulation der interstellaren Wolke"	67
Abbildung 34: "Simulation des schwarzen Lochs mit Akkretionsscheibe"	68
Abbildung 35: "Simulation des Neutronensterns & Querschnitt der Simulation"	68
Abbildung 36: "In Planetaren Nebel zerfallener roter Riese"	69
Abbildung 37: "Außenhülle des roten Riesen und des roten Überriesen"	69
Abbildung 38: "Simulation des Protosterns"	70
Abbildung 39: "In einer Supernova Kollabierender Kern des roten Überriesen"	72



Abbildung 40: “Zerstörung der Außenhülle des roten Überriesen durch die Supernova”..	72
Abbildung 41: “Die fünf Arrays der Pfade mit enthaltenen Stadien als Strings” .....	74

# 1 Einleitung

Kaum ein Film kommt heutzutage ohne den Einsatz von Spezialeffekten in Form von Partikelsimulationen aus. Das Einsatzgebiet dieser ist groß, da sich mit ihnen von realistischen Staub-, Feuer- oder Wassersimulationen bis hin zu abstrakten magischen Effekten alles kreieren lässt.

Sie werden allgemein meist dann genutzt, wenn Prozesse zu komplex sind, um diese mit praktischen realen Mitteln zu visualisieren. Zu jenen zählen auch astrophysikalische Prozesse, da diese auf einer Zeit und Raum Skalierung ablaufen, die nur schwer beobachtbar und vorstellbar ist. Diese Arbeit beschäftigt sich mit Partikelsimulationen, denen das Ziel zugrunde liegt, jene astrophysikalischen Prozesse zu visualisieren.

Dabei wird im Speziellen darauf eingegangen, ob und inwieweit es möglich ist, den Aspekt der Interaktivität in solchen Simulationen einzubringen, da diese allgemein als förderlich für den Erkenntnisgewinn angesehen wird. Das Ziel besteht darin zu prüfen, ob es möglich ist, Partikelsimulationen zu erschaffen, welche der Nutzer weitestgehend selbst steuert, um sich so individuell Wissen über astrophysikalische Prozesse anzueignen.

Dafür wird eine solche interaktive dreidimensionale Partikelsimulation erschaffen, welche das Ziel hat, Schülern den Prozess der stellaren Evolution und im Speziellen die Unterscheidung der Evolutionsstufen verständlich zu machen. Es wird diskutiert, inwiefern Aspekte der Interaktivität oder technische Limitationen hinderlich für die Umsetzung einer solchen Simulation sind und welche Vorgehensweisen in Frage kommen.

Zunächst werden dazu, im zweiten Kapitel, die simulierten astrophysikalischen Prozesse thematisch untergliedert. Jede Evolutionsstufe eines Sternes wird einzeln beschrieben und allgemeine Gesetzmäßigkeiten, welche die Abbildung solcher Prozesse in einer Simulation ermöglichen, werden aufgeführt. Darauf folgend wird in Kapitel 3 auf didaktische Aspekte eingegangen, welche in einer Anwendung allgemein zur Informationsvermittlung beitragen. Dabei wird ebenfalls erörtert, inwiefern die Vermittlung von Wissen im Physikunterricht Mängel aufweist, welchen die Arbeit mit einer interaktiven Partikelsimulation entgegenwirken könnte. Im vierten Kapitel wird nun auf die technische Umsetzung von Partikelsimulationen eingegangen. Verschiedene Möglichkeiten werden evaluiert und die Umsetzung in der 3D- Software Houdini wird genauer beschrieben. Im Methodenteil werden daraufhin in Kapitel 2 thematisierte Prozesse mit den in Kapitel 4 vorgestellten technischen Vorgehensweisen in einer interaktiven Partikelsimulation umgesetzt, welche die didaktischen Aspekte aus Kapitel 3 zu erfüllen versucht. Probleme

und Konflikte bei der Umsetzung werden thematisiert und Lösungen vorgestellt. Anschließend wird erörtert, inwiefern das Endergebnis die didaktischen Ziele einhält und die Forschungsthese wird beantwortet.

## **2 Astrophysikalische Grundlagen**

### **2.1 Grundbegriffe**

Während des Lebenszyklus eines Sternes werden dessen Elemente durch diverse Kräfte beeinflusst. Diese definieren die Lebensphasen, welche er durchlaufen wird, ebenso wie die Klassifizierung, in die er sich in jeder Lebensphase einordnen lässt. Da auf jeden Stern die gleichen physikalischen Parameter wirken, ist es möglich, Prognosen aufzustellen, die den Lebenszyklus skizzieren.

#### **Gravitation**

Das grundlegendste Konzept, welches zur Bewegung von Materie im interstellaren Raum führt, ist die Gravitation. Mit der allgemeinen Relativitätstheorie ist die Krümmung der Raumzeit durch massive Objekte definiert. Daraus geht hervor, dass Objekte mit einer Masse eine Vertiefung in der Raumzeit erzeugen und somit andere Objekte anziehen. (Göbel, 2016, S. 7)

#### **Interstellares Medium**

Sterne bilden sich im sogenannten interstellaren Medium. Darunter wird der Raum verstanden, welcher Sterne umgibt. Er enthält Gas- und Staubpartikel, die aus Zerfallsprozessen anderer Sterne hervorgehen und gleichzeitig als Grundlage für die Geburt neuer stellarer Objekte dienen. (Kippenhahn, 1970, S.157) Durch Strahlung, Magnetfelder und Gravitation sind sie konstant in Bewegung. Dabei handelt es sich größtenteils um Wasserstoff und Helium. Schwerere Elemente sind in deutlich geringeren Mengen vorhanden. (Snell, 2011, S. 837)

#### **Kernfusion**

Erreichen Atome eine entsprechend hohe Temperatur, gehen sie vom Gasförmigen in den Plasmazustand über. Plasma enthält freie Elektronen und Ionen, da sich bei diesen Temperaturen Elektronen vom Atomkern lösen. Ist die richtige Temperatur und das nötige Verhältnis an ungebundenen Teilchen vorhanden, kommt es zur Fusion und es bildet sich ein neues, schwereres Element. Die für die Fusion notwendige Temperatur steigt mit der Schwere der Elemente an. Für die Fusion von Wasserstoff zu Helium wird beispielsweise eine Temperatur von rund 10 Millionen °C benötigt. Durch diesen Prozess werden große Mengen an Energie in Form von Strahlung freigesetzt. (Baker, 2009, S.140ff)

## 2.2 Klassifizierung

### Spektralklasse

Sterne können anhand ihrer Eigenschaften klassifiziert werden. Hierzu zählt unter anderem die Spektralklasse, welche eine Unterteilung nach Aussehen des Lichtspektrums darstellt. Die Spektren von Hauptreihensternen sind, wie in Abbildung 1 zu sehen, unterteilt.

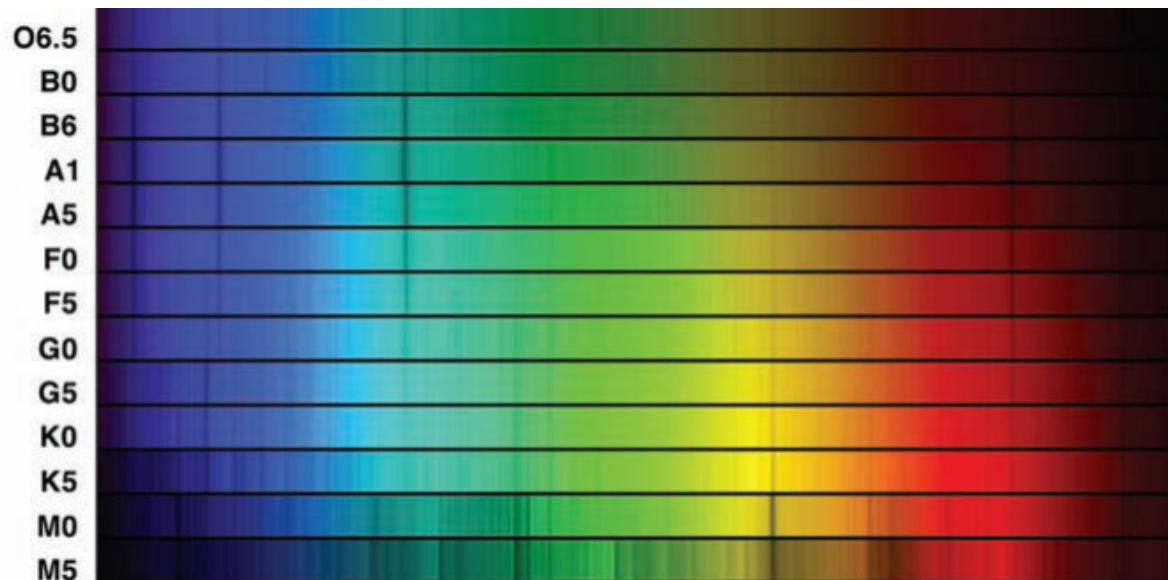


Abbildung 1: "Spektralklassen der Sterne" (Young, 2012, S.21)

Die Spektralfarbe eines Sternes steht dabei im Zusammenhang mit dessen Temperatur, da je nach Art und Häufigkeit der Strahlungsreaktionen, welche im Kern ablaufen, unterschiedliche Lichtspektren erzeugt werden. Allgemein lässt sich sagen, dass sich heißere Sterne weiter im Blau-Spektrum und kühlere eher im Rot-Spektrum befinden. (Young, 2012, S.21)

## Hertzsprung-Russell Diagramm

Ein weiterer Faktor für die Klassifizierung von Sternen ist die absolute Helligkeit. Diese Vereinheitlicht die Größenordnung der Helligkeit von selbstleuchtenden Objekten, da sie den Abstand zum beobachteten Objekt mit einbezieht. (Boblest et al., 2015, S.22f) Sie ist also für alle Sterne universell anwendbar. Lichtspektrum bzw. Temperatur werden ins Verhältnis mit der absoluten Leuchtkraft gesetzt, um Sterne auf einem Diagramm einordnen zu können. (Young, 2012, S.21) Dieses wird als Hertzsprung-Russell Diagramm bezeichnet und ist auf der nachfolgenden Abbildung zu sehen.

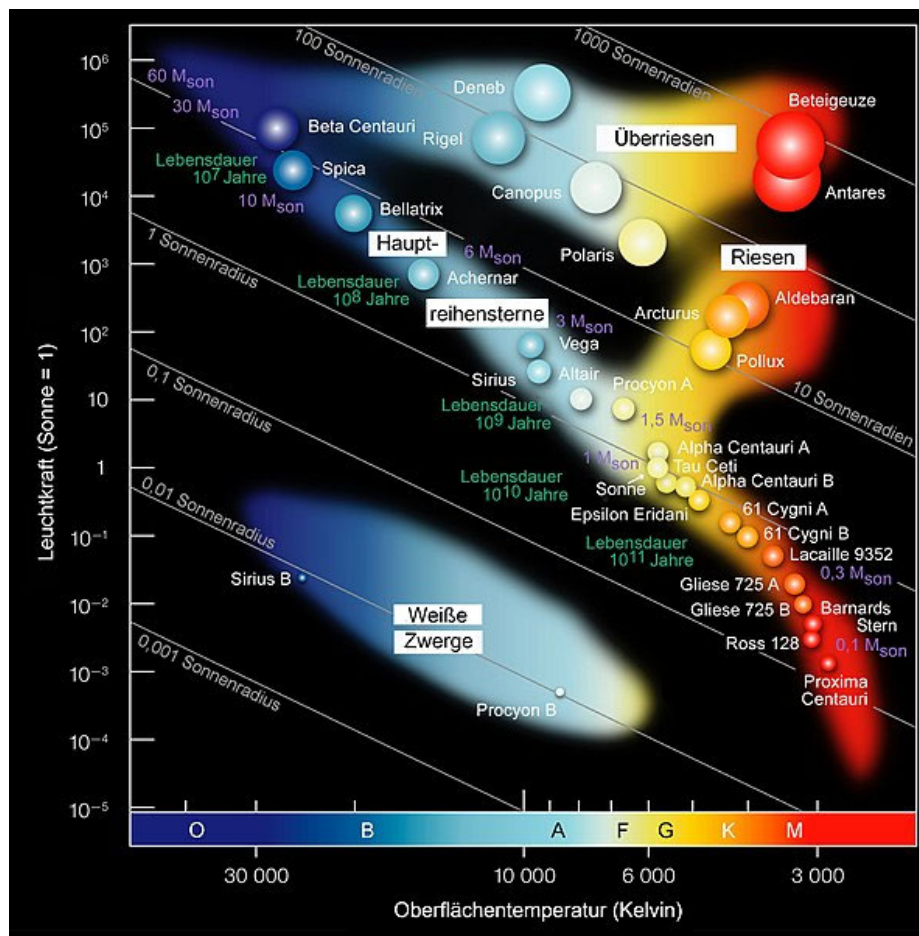


Abbildung 2: "Hertzsprung Russel Diagramm" (in Anlehnung an Young, 2012, S.22)

Im spezifischen Beispiel (Abb. 2) wird auf der X-Achse die Spektralklasse bzw. die Temperatur dargestellt, da diese beiden Parameter miteinander korrelieren. Auf der Y-Achse ist die absolute Helligkeit zu sehen. Nun lässt sich erkennen, dass die eingetragenen Sterne in einigen Regionen des Diagramms konzentrierter auftreten, was es ermöglicht, Schlüsse auf einige Eigenschaften wie Alter, chemische Zusammensetzung oder Masse zu ziehen. (Young, 2012, S.22)

## 2.3 Stellare Evolutionspfade

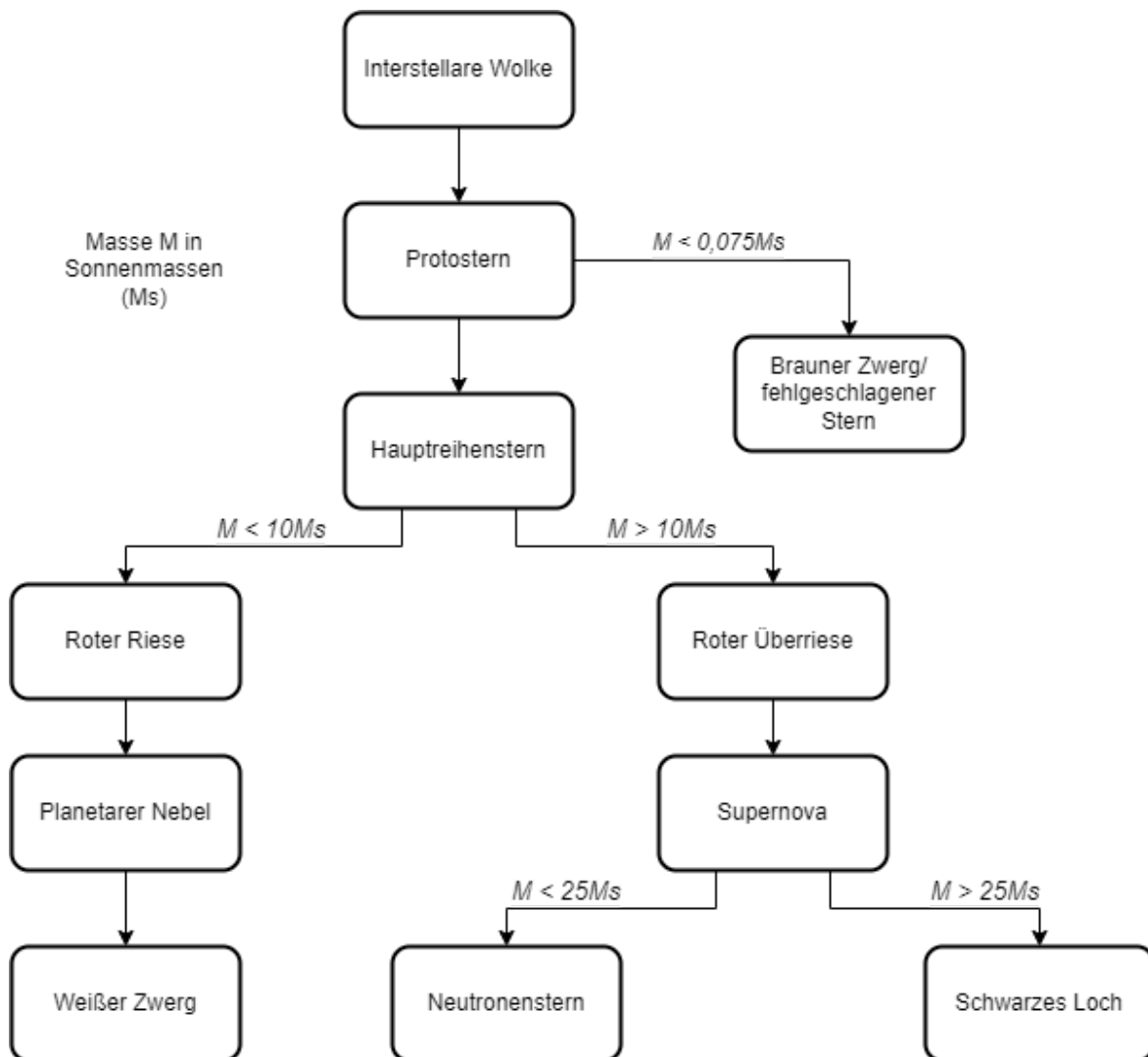


Abbildung 3: “Stellare Evolutionspfade” (In Anlehnung an Camenzind, 2016, S.62ff)

Das zu sehende Diagramm (Abb. 3) stellt die unterschiedlichen Phasen der Sternentwicklung dar. Der durchlaufene Pfad ist dabei von der Masse des jeweiligen Sterns abhängig. (Massey, 2001, S.1)

Alle Sterne werden zunächst in interstellaren Wolken geboren. (Kippenhahn, 1970, S.157) Sind sie nach der Protostern-Phase in der Lage zur Wasserstofffusion, beginnt die Hauptreihenphase. Im Hertzsprung-Russell Diagramm (Abb. 2) lässt sich eine diagonale Linie einzeichnen, auf welcher sich die meisten Sterne für den Großteil ihrer aktiven Lebenszeit befinden. Diese Diagonale wird als Hauptreihe bezeichnet. Nach Beendigung der längsten Lebensphase beginnt der Verfall des Sternes. Hier trennen sich die Pfade von

massiven Sternen mit mehr als circa zehn Sonnenmassen und nicht massiven Sternen unter diesem Wert. Nicht massive Sterne verstreuen nach einer Phase als roter Riese den Großteil ihrer Materie in einen planetaren Nebel und enden als weißer Zwerg. (Camenzind, 2016, S.69) Massive Sterne durchschreiten zunächst eine Phase als roter Überriese und beenden diese mit einer Supernova. Diese kann entweder einen Neutronenstern oder, bei besonders massiven Sternen, ein schwarzes Loch hervorbringen. (Camenzind, 2016, S.62)

## 2.4 Verhältnismäßigkeiten

Durch das Russell-Vogt-Theorem wird definiert, dass sich aus der stellaren Masse und der chemischen Zusammensetzung eines Sternes alle weiteren Eigenschaften ableiten lassen. (Massey, 2001, S.1) Im theoretischen Umfeld dieser Arbeit dient dieser Zusammenhang als Grundlage für die Abschätzung bestimmter Werte, da im Methodikteil Annahmen über Sterneigenschaften anhand der gegebenen Masse getroffen werden.

### 2.4.1 Absolute Helligkeit

Das Verhältnis zwischen Masse (M) und absoluter Helligkeit (L) ist für unterschiedliche Masse Abstufungen wie folgt definiert:

$L \sim M^{1,6}$ , für Sterne mit rund 100 Sonnenmassen.

$L \sim M^{3,1}$ , für Sterne mit rund 10 Sonnenmassen.

$L \sim M^{4,7}$ , für Sterne mit Sonnenmasse.

$L \sim M^{2,7}$ , für Sterne mit rund 0,1 Sonnenmassen. (Massey, 2001, S.1)



### 2.4.2 Spektralklasse

Anhand der gegebenen Beziehung zwischen Helligkeit und Masse lässt sich die Spektralklasse eines Hauptreihensternes mit dem HR-Diagramm (Abb. 2) abschätzen, indem für den gegebenen Helligkeitswert die entsprechende Position auf der Hauptreihe gefunden wird. Die Spektralklasse kann nun abgelesen werden. Im Verhältnis zur Masse lässt sich daraus folgende Unterteilung ableiten:

Masse in Sonnenmassen ( $M_{\odot}$ )	Spektralklasse
$\geq 25 M_{\odot}$	O
$\geq 2,5 M_{\odot}$	B
$\geq 1,25 M_{\odot}$	F
$\geq 1 M_{\odot}$	G
$\geq 0,8 M_{\odot}$	K
$< 0,8 M_{\odot}$	M

Abbildung 4: "Verhältnis von Masse und Spektralklasse" (in Anlehnung an Massey 2001)

### 2.4.3 Lebenszeit

Die Lebenszeit eines Sterns ist von einer Reihe an Faktoren abhängig. Die äußeren Bedingungen und Abstände zu anderen Stellaren Objekten spielen eine wichtige Rolle, ebenso wie das Verhältnis an schwereren Elementen in der Wolke, aus welcher der Stern sich formt. Betrachtet man einen Stern in einem theoretischen, isolierten Umfeld, ohne stellare Winde oder vergleichbare externe Einflüsse, ist die Lebenszeit im Wesentlichen von zwei Faktoren abhängig: der Masse und dem Energieverbrauch. Die Frage ist also, wie viel Materie zur Energiegewinnung vorhanden ist und wie schnell diese verbraucht bzw. fusioniert wird. Der Energieverbrauch kann anhand der absoluten Helligkeit bestimmt werden. Diese dient als Indikator für die Menge an ablaufenden Fusionsreaktionen, da die Strahlung, wie in Kapitel 2.1 beschrieben, ein Produkt der Reaktion ist. (Massey, 2001, S.1) Da die Absolute Helligkeit ( $L$ ) wiederum ebenfalls von der Masse ( $M$ ) abhängig ist können aus der Formel zur Abschätzung der Lebenszeit ( $t$  in  $10^{10}$  Jahre) mit der absoluten Helligkeit:  $t \sim M/L$ , stellare Lebenszeiten anhand der Masse wie folgt abgeschätzt werden:  $t \sim M^{-3,7}$ . (Massey, 2001, S.1)

## 2.5 Sternentstehung

### 2.5.1 Interstellare Wolken

Die Materie im interstellaren Medium ist nicht gleichmäßig verteilt, da sie durch Gravitation, Elektromagnetismus und weitere Prozesse, welche der Lebenszyklus von Sternen hervorruft, in bestimmten Regionen konzentrierter auftritt. (*Kippenhahn, 1970, S.157*) Die daraus entstehenden interstellaren Wolken bestehen zum Großteil aus Wasserstoff. Sie müssen nun einen Zustand gravitativer Instabilität erreichen, damit sich aus ihnen Sterne bilden können. Dies geschieht unter anderem dadurch, dass die Gase durch externe Ereignisse in Bewegung gesetzt werden und sich in einigen Regionen weiter konzentrieren. (*Werner, 1981, S. 105*) Wird im Prozess dieser Fluktuation eine kritische Masse überschritten, ist die Wolke gravitativ instabil. Diese Masse wird als "Jeans-Masse" bezeichnet. (*Kippenhahn, 1970, S.160*) Temperatur und Dichte der Gaswolke stellen dabei die beiden ausschlaggebenden Faktoren dar. So wird entweder bei gegebener Temperatur eine sehr hohe Dichte oder bei gegebener Dichte eine sehr niedrige Temperatur benötigt, um die Gegebenheiten für die Sternentstehung zu schaffen. Je niedriger die gegebene Temperatur, desto weniger dicht kann die Wolke sein, um gravitative Instabilität zu erzeugen. (*Werner, 1981, S. 106*)

## 2.5.2 Protostern

Ist der Punkt der Gravitativen Instabilität erreicht, bildet sich ein verdichtetes Zentrum, mit einem gravitativen Sog (der tatsächliche Protostern) (*Kippenhahn, 1970, S.160*). Um den Stern bildet sich eine scheibenförmige Wolke aus Staub und Gas, welche durch die Anziehung in einem Strudel nach innen rotiert. Zentrifugalkraft und Gravitation wirken entgegengesetzt und sorgen dafür, dass die Scheibe zunächst erhalten bleibt und nicht sofort nach innen kollabiert. (*Wambsganß, 2012, S.202*)

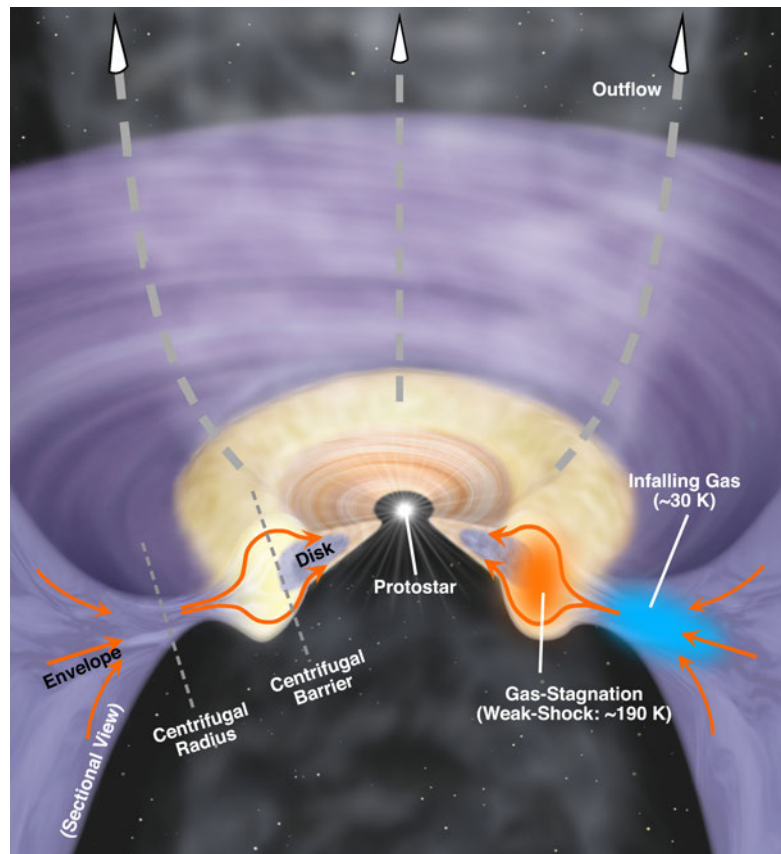


Abbildung 5: "Aufbau eines Protosterns" - (*Güdel et al, 2018, S.5*)

Von der Scheibe aus betrachtet orthogonal, in beide Richtungen bilden sich zwei sogenannte "Jets", aus elektromagnetischer Strahlung, die sich vom Stern weg bewegen. Durch sie werden Staub und Gas aus dem Gravitationsfeld des Sternes hinaus ins Weltall transportiert. Die Dichte und Temperatur im Kern des Sternes wachsen auf diese Weise nun immer weiter, bis er die Protostern-Phase überwunden hat. Die Dauer dieser Phase verkürzt sich mit geringerer Masse. (*Wambsganß, 2012, S.202*)

### 2.5.3 Hauptreihenstern

Ist der Kern des Protosternes nach einigen Millionen Jahren (abhängig von der Masse der ursprünglichen Gaswolke) verdichtet genug, wird eine Temperatur von rund 10 Millionen °C erreicht (*Wambsganß, 2012, S.202*). Bei dieser Temperatur beginnt der Wasserstoff im Kern des Sterns in den Plasmazustand überzugehen, was die Fusion zu Helium ermöglicht. (*siehe 2.1*) Da beim Prozess der Nuklearen Fusion eine signifikante Menge an Energie freigesetzt wird, die vom Kern nach außen hin wirkt, entsteht eine Kraft, welche der Gravitation entgegengesetzt ist. Ist die Fusionsenergie signifikant genug, stoppt der gravitative Kollaps des Sternes und er wird stabil. Dieser Ausgleich von Gravitation und Fusion wird als “Hydrostatisches Gleichgewicht” verstanden. (*Boblest et al., 2015, S. 309*)

Ist dieses Gleichgewicht erreicht, hat der Stern einen sphärischen Aufbau, bestehend aus mehreren Schichten, in denen unterschiedliche Druck- und Temperaturverhältnisse vorherrschen. Der Aufbau eines Sonnenähnlichen Sternes wird im Folgenden von innen nach außen beschrieben.

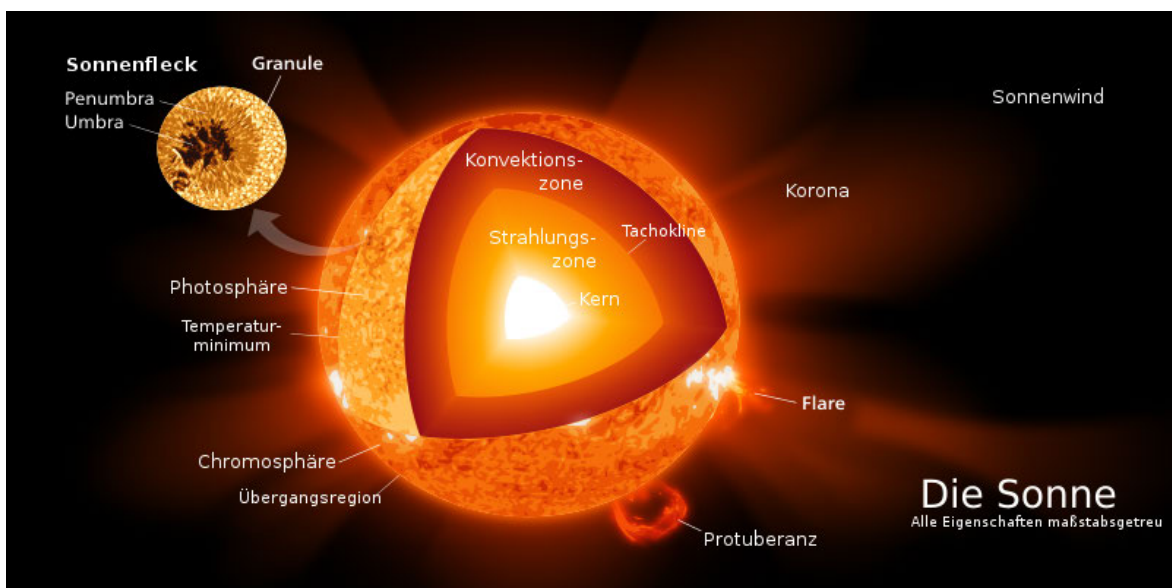


Abbildung 6: “Aufbau der Sonne” - (In Anlehnung an Chandel, 2021, S.1)

#### Kern

Dieser bildet die dichteste Region des Sternes. Die Kernfusion, durch die das hydrostatische Gleichgewicht aufrechterhalten wird, findet hier statt, da im Kern die für den Prozess notwendige Temperatur erreicht wird.

### Strahlungsschicht

Die durch Fusionsreaktionen erzeugte Strahlung wird in dieser Schicht nach außen transportiert. Die Bewegung erfolgt dabei nur in eine Richtung und ist das Resultat eines flachen Temperaturgradienten in der Region. (Hansen et al., 2004, S.194)

### Konvektionszone

In diesem Bereich fällt die Temperatur verhältnismäßig steil ab, wodurch ein Gradient erzeugt wird, welcher Konvektion ermöglicht. Im Gegensatz zur Strahlung beinhaltet der Prozess der Konvektion Temperatur-Ströme in beide Richtungen. (Hansen et al., 2004, S.214)

### Photosphäre und Chromosphäre

Diese beiden Schichten bilden die Abgrenzung des Sternes nach außen. Die Temperatur ist hier am niedrigsten. Wird ein Stern beobachtet, sind diese Regionen sichtbar und ermöglichen die Einordnung in eine Spektralklasse, bzw. eine Messung der Oberflächentemperatur.

### Strahlung und Konvektion

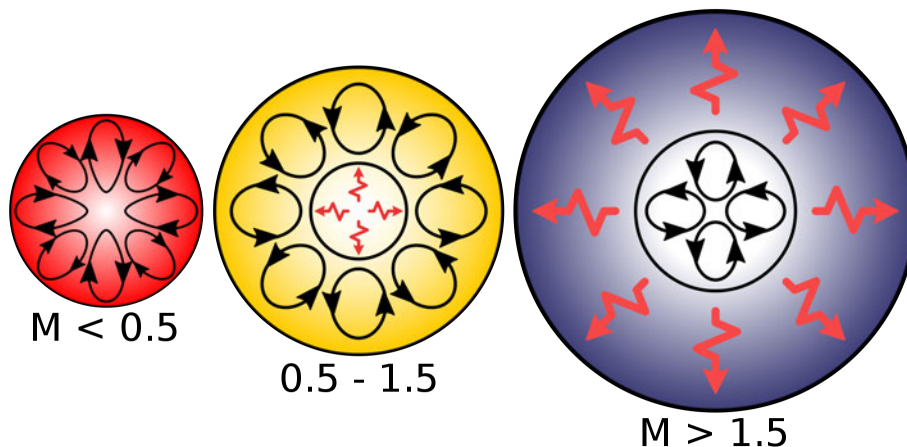


Abbildung 7: "Konvektion und Strahlung in Sternen" - (Hanslmeier, Arnold, 2020, S.401)

Sterne, welche massiver als 1.5 Sonnenmassen sind, setzen bei ihrer Fusion mehr Energie frei als weniger massive Sterne. Ihre Kerntemperatur ist aus diesem Grund höher, was dazu führt, dass im Stern ein steiler Temperaturgradient herrscht. So haben massive Sterne meist eine Konvektionszone direkt um den Kern und eine Strahlungszone außen, da hier die Temperatur wieder gleichmäßiger verteilt ist. Abbildung 7 veranschaulicht diesen Zusammenhang. (Hanslmeier, 2020, S.400)

## 2.5.4 Brauner Zwerg - fehlgeschlagener Stern

Wie im vorhergehenden Kapitel thematisiert, ordnet sich ein Stern ab dem Zeitpunkt in die Hauptreihe ein, ab dem die Wasserstofffusion im Kern beginnt. Die dafür nötigen Bedingungen sind, wie im Kapitel 2.1 erläutert, eine ausreichende Temperatur und genügend Druck, um den vorhandenen Wasserstoff zu Plasma umzuwandeln. Diese Parameter sind von der vorhandenen Gravitation abhängig, welche wiederum durch die Masse des Sterns bestimmt wird. Nun wird folglich eine bestimmte Masse benötigt, um die Bedingungen für die Fusion zu schaffen. Diese liegt bei ca. 0.075 Sonnenmassen. Objekte, welche sich unter diesem Schwellwert befinden, werden braune Zwerge (auch fehlgeschlagene Sterne) genannt.

Sie halten ihren inneren Druck durch Deuteriumfusion aufrecht, ein Prozess, welcher auch bei niedrigeren Temperaturen möglich ist und weniger Energie erzeugt als die Proton-Proton-Fusion, wie sie Hauptreihensterne betreiben. So befinden sie sich auf dem Hertzsprung-Russell-Diagramm (*Abb. 2*) nie auf der Hauptreihe, da ihnen die Temperatur bzw. Strahlung fehlt, um das Lichtspektrum von Hauptreihensternen zu erreichen. (*Allard & Homeier, 2007*)

Der Fusionsprozess findet im Kern des Zwerges statt. Ihn umgibt eine Konvektionszone, welche sich bis an die Außenschicht erstreckt. Eine Strahlungszone besitzen braune Zwerge nicht, da der Temperaturgradient im Inneren dafür nicht steil genug ist. Nach der ersten Lebensphase eines braunen Zwerges kommt die Deuteriumfusion ebenfalls zum Erliegen. Durch die geringe Masse kann der Druck der entarteten Materie, welche zu Beginn des Lebenszyklus entstanden ist, den Zwerg gegen die Gravitation stabilisieren (ähnlich wie bei weißen Zwergen). So überdauert der fehlgeschlagene Stern Milliarden von Jahren, in denen er sich schließlich langsam auf die Temperatur des interstellaren Mediums abkühlt. (*Burrows & Liebert 1993, S. 308f*)

## 2.6 Sternverfall

Die Hauptreihenphase eines Sternes endet dann, wenn die vorhandene Menge an Wasserstoff im Kern nicht mehr genügt, um durch Fusion dem Gravitationsdruck entgegenzuwirken. Zunächst komprimiert sich der Kern unter Erhöhung der Temperatur, welche nun ausreicht, um den in der Schale vorhandenen Wasserstoff zu fusionieren. Dadurch wirkt mehr Energie auf die äußeren Schichten, wodurch diese nach außen gedrückt werden und sich ausbreiten. Der Stern verändert sein Lichtspektrum, da die Oberfläche nun deutlich größer und weiter entfernt vom Kern ist. Dies führt dazu, dass er die Hauptreihe auf dem Hertzsprung-Russell Diagramm (*Abb. 2*) verlässt. Er wird zum Riesenstern und der Beginn seines Verfalls ist eingeleitet. Dieser Prozess kann, abhängig von der Masse des Sterns, auf unterschiedlichen Wegen ablaufen. (*Camenzind, 2016, S.67*)

### 2.6.1 Roter Riese - massearme Sterne

Sterne mit einer Masse von weniger als neun Sonnenmassen gehen nach der Hauptreihenphase in das Stadium des roten Riesen über. (*Camenzind, 2016, S.65*) Nachdem ihr Kern weit genug komprimiert wurde und dadurch eine Temperatur von 100 Millionen Grad Kelvin erreicht ist, beginnen sie das vorhandene Helium zu Kohlenstoff und Sauerstoff zu fusionieren. So werden zwei Schalen, eine Wasserstoff Fusionierende und eine Helium Fusionierende, erzeugt. (*Camenzind, 2016, S.68*) Mit Beginn dieser Reaktion kommt es zu einer weiteren Expansion der Hülle und der Stern erreicht ein Vielfaches seines Radius aus der Hauptreihenphase. Bei dieser Expansion geht bereits ein Teil der Materie verloren, da die Gravitation ab einem bestimmten Abstand zum Kern nicht mehr stark genug wirkt, um die äußeren Hüllen zusammenzuhalten. Massearme Sterne (Sterne mit weniger als neun Sonnenmassen) können nicht die notwendige Temperatur erzeugen, um den Kohlenstoff im Kern zu fusionieren. Folglich kommt diese Reaktion zum Erliegen, wenn der Großteil des Heliums zu Kohlenstoff umgewandelt wurde. Der größte Teil des Sternes ist bis dahin bereits nach außen in das interstellare Medium abgegeben worden. Zurück bleibt nun der verdichtete, entartete Kohlenstoff aus dem Kern, welcher einen weißen Zwerg formt. Dieser ist anfänglich sehr heiß und kühlt nun, ähnlich wie ein brauner Zwerg, über Jahrmilliarden langsam auf die Temperatur des interstellaren Mediums ab. Die übrige Materie, die durch den Verfall des Roten Riesen abgegeben wurde, ist weit verstreut um den weißen Zwerg herum und bildet einen planetarischen Nebel. (*Camenzind, 2016, S.69ff*)

## 2.6.2 Roter Überriese - massereiche Sterne

Sterne mit mehr als neun Sonnenmassen durchlaufen zunächst eine ähnliche Entwicklung wie die massearmen Roten Riesen. Jedoch können sie genügend Gravitationsdruck und damit auch die nötige Temperatur erzeugen, um auch Kohlenstoff zu fusionieren. Die Temperaturen erhöhen sich weiter und es werden zunächst Neon, Sauerstoff, Silikon und schließlich Eisen erzeugt.

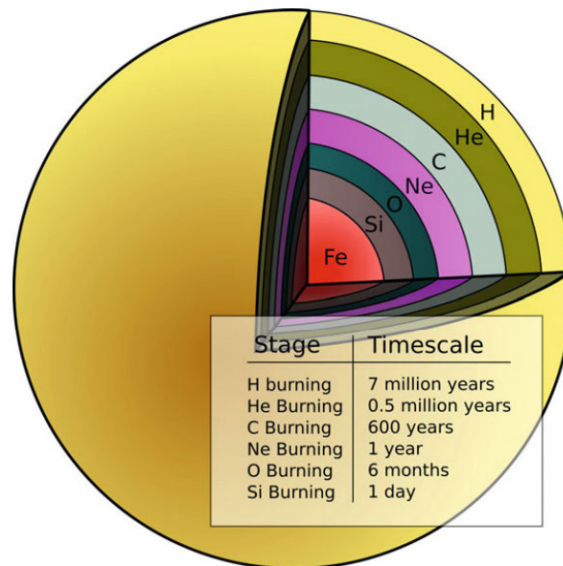


Abbildung 8: "Struktur des Kernes eines Roten Überriesen" (Camenzind, 2016, S.77)

Diese Schichten sind der Schwere der Elemente nach wie in Abbildung 8 geordnet. (Camenzind, 2016, S.77) Das erzeugte Eisen sammelt sich im Zentrum des Kernes und nimmt immer weiter an Masse zu. Dieser entartete Eisenkern wird nicht weiter fusioniert und seine Temperatur steigt bis auf fünf bis zehn Milliarden Kelvin. An diesem Punkt werden die Atome des Kernes instabil und Elektronen und Protonen verschmelzen miteinander zu Neutronen. Dieser Prozess komprimiert den Kern in Bruchteilen einer Sekunde so stark, dass er mit 25 prozentiger Lichtgeschwindigkeit in sich zusammenfällt. Nun hat der Kern nahezu die Dichte der Teilchen eines Atomkerns, ohne an Masse verloren zu haben, wodurch eine extreme Gravitationskraft erzeugt wird. Diese zieht die äußeren Schichten des Kernes mit hoher Geschwindigkeit auf ihn hinunter. Wenn der Kern erreicht wird, prallt die Materie von ihm ab und erzeugt eine Druckwelle, welche nach außen hin wirkt und nach einigen Stunden die äußeren Schichten erreicht. So wird fast der gesamte Stern mit hoher Geschwindigkeit in das Weltall abgegeben. Dieser Prozess wird als Supernova bezeichnet. (Camenzind, 2016, S.78f)



### 2.6.3 Neutronenstern

Von der Supernova eines Sternes mit weniger als 40 Sonnenmassen bleibt der dicht komprimierte Kern zurück. Dieser wird als Neutronenstern bezeichnet, da er aus den bei der Supernova entstandenen Neutronen besteht. Er besitzt einen Radius von ca. 30 Kilometern bei einer Dichte, vergleichbar mit den Teilchen eines Atomkerns. Er besitzt eine Temperatur von ca. einer Millionen °C und rotiert anfänglich sehr schnell.

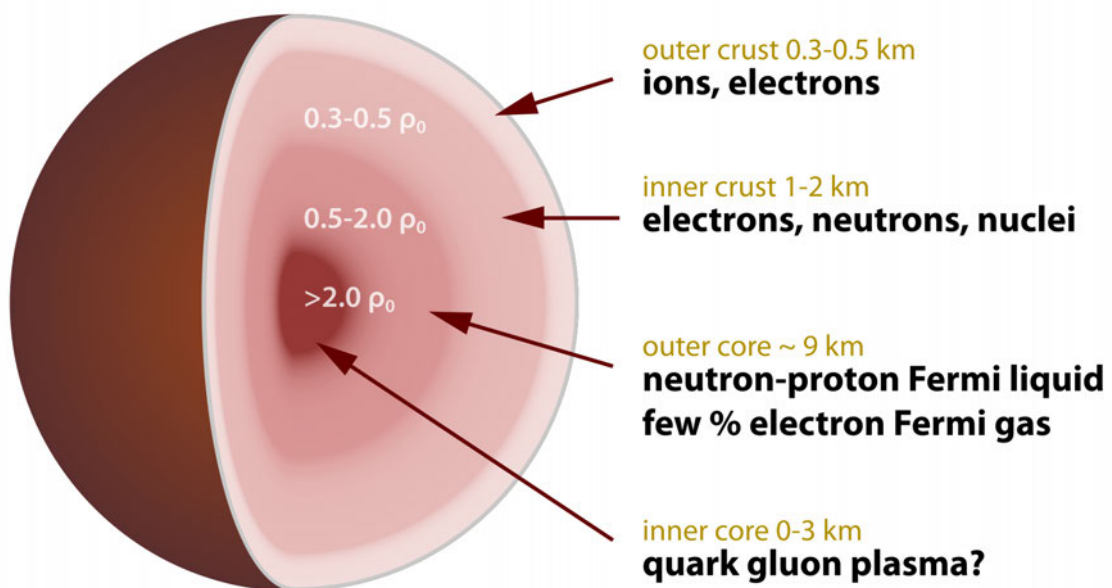


Abbildung 9: "Aufbau eines Neutronensterns" - (Albakry, 2015, S.1500)

Der Neutronenstern hat eine feste Außenhülle und hat wie in Abb. dargestellt drei Schichten im Inneren. In diesen Bereichen herrschen (abgesehen vom Inneren eines schwarzen Lochs) die extremsten Druck Bedingungen des Universums. Die genaue Zusammensetzung ist bis heute nicht vollständig wissenschaftlich erklärt. (Albakry, 2015, S. 1500)

## 2.6.4 Schwarzes Loch

Ein schwarzes Loch entsteht dann, wenn der kollabierende Stern eine Masse von über 40 Sonnenmassen hat. Die Materie des Kernes stürzt in sich zusammen und wenn der entstandene Neutronenstern das sogenannte "Tolman-Oppenheimer-Volkoff-Limit" überschreitet, reicht der Druck der entarteten Materie des Kernes nicht mehr aus um der Gravitation entgegenzuwirken. Es kommt zum Kollaps des Sterns, welcher nun zu einem schwarzen Loch wird. (*O'Connor & Ott, 2011, S.1*)

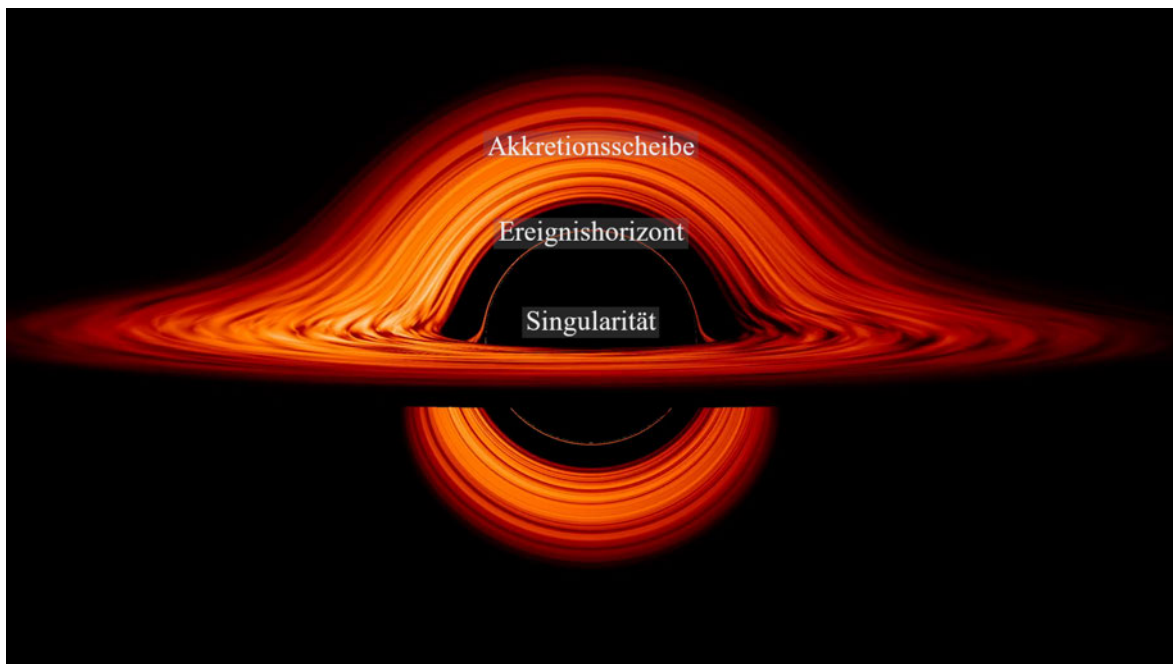


Abbildung 10: "Aufbau eines schwarzen Lochs" - (in Anlehnung an Schnittman, 2019)

Das schwarze Loch hat eine Akkretionsscheibe aus Materie, welche sich im Orbit um den Körper bewegt. Der Ereignishorizont ist ein theoretisches Konzept, welches sich auf den letzten Punkt bezieht, an dem die Fluchtgeschwindigkeit für Teilchen noch nicht die Lichtgeschwindigkeit (300000 km/s) überschreitet. Die sogenannte Singularität befindet sich hinter dem Ereignishorizont. Sie ist ein theoretischer Punkt mit unendlich kleiner Größe bei unendlich großer Dichte, welcher durch die Raumzeitkrümmung folglich unendliche Gravitation erzeugt. Das Innere von schwarzen Löchern ist aufgrund des Ereignishorizonts nicht beobachtbar oder messbar. Aus diesem Grund sind die Bedingungen innerhalb des schwarzen Lochs wissenschaftlich nicht genau definiert. (*Nowikow, 1981, S. 40ff*)

## 3 Informationsvermittlung durch interaktive Simulationen

### 3.1 Visualisierung

Modelle, welche der Visualisierung komplexer Zusammenhänge dienen, sind aus vielerlei Hinsicht sinnvoll. Sie geben der lernenden Person einen visuellen Reiz, welcher, wie in Abbildung 11 durch das Modell von Wagner (2013) visualisiert, verarbeitet wird.

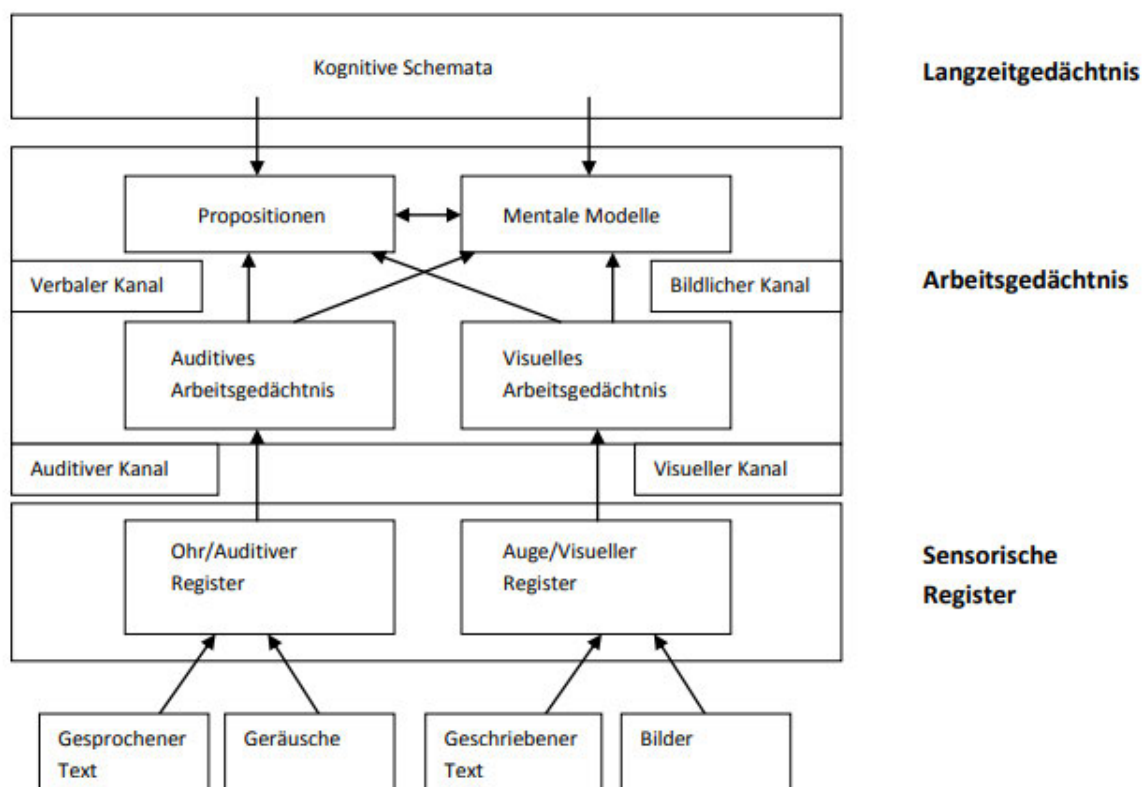


Abbildung 11: "Modell des Text- und Bildverstehens" (Wagner, 2013, S. 23)

Es ist zu sehen, dass beim Lernen ein mentales Modell des Sachverhaltes konstruiert wird, durch das Informationen im Arbeitsgedächtnis gespeichert oder Sachverhalte verstanden werden. Hervorzuheben ist, dass der Informationszugang über zwei sensorische Kanäle: das Auditive- und das Visuelle Register erfolgt. Daraus lässt sich ableiten, dass es sinnvoll ist, neben der verbalen Vermittlung von Informationen die visuelle Ebene ebenfalls zu bedienen, da dies die Konstruktion eines mentalen Modells fördert. So würden dreidimensionale Modelle zur Visualisierung von Prozessen das Verständnis dieser

vereinfachen, da sie eine weitere sensorische Ebene bedienen, welche allein durch verbale Informationsvermittlung nicht abgedeckt werden kann. (*Wagner, 2013, S. 22ff*)

### **3.2 Interaktion**

Interaktive Simulationen bringen neben dem Aspekt der visuellen Darstellung auch die Interaktion des Lernenden mit dem Programm mit sich. Der Nutzer erarbeitet sich seine Informationen selbst. Dies kann auf verschiedene Arten geschehen. Eine Aufteilung in verschiedene Ebenen der Interaktion nahm beispielsweise Schulmeister (2002, 2005) vor. (*Wedekind, 2007, S.3*) Eine Interaktive Partikelsimulation würde hiernach in die Klassifizierung "Den Inhalt der Komponente beeinflussen" eingeordnet werden. Der Lernende erhält dabei die Möglichkeit zur freien Manipulation verschiedener Parameter, wodurch neue Darstellungsweisen erzeugt werden. Diese Form der Lehre funktioniert nur, wenn der Nutzer selbst aktiv wird, da das Programm ohne Benutzereingaben keine Informationen darstellt. So wird die Aufmerksamkeit des Lernenden automatisch auf das zu bearbeitende Themengebiet gelenkt. Weiterhin werden durch die Manipulation der Parameter des Programmes kontinuierlich angepasste Darstellungsformen erzeugt, was dafür sorgt, dass die Aufmerksamkeit beim Lernen länger erhalten bleibt, da es von Abwechslung und Erfolgserlebnissen begleitet wird.

Wie Wedekind (2007, S.9) betont, tragen interaktive Visualisierungen zur Erschließung neuer mentaler Zugänge zu bestimmten Themenbereichen bei. Dies ist vor allem dann förderlich, wenn es sich bei den visualisierten Prozessen um zeitliche Abläufe oder abstrakte Zusammenhänge handelt, was besonders auf einige Bereiche der Naturwissenschaften zutrifft. Auf diese Weise können Simulationen als Experimente dienen, welche in Bereichen eingesetzt werden können, in denen das Experimentieren sonst kaum möglich ist. (*Zendler, 2018, S.13ff*)

Interaktive Partikelsimulationen haben neben der thematisierten Manipulation von Parametern noch eine weitere Ebene der Interaktion. Da die Simulationen im dreidimensionalen Raum ablaufen, ist es möglich, erstellte Modelle aus mehreren Perspektiven zu betrachten. Dadurch wird ein entscheidender Vorteil gegenüber der bloßen Visualisierung durch Bilder gegeben, da dreidimensionale Darstellungen, welche sich aus allen Perspektiven betrachten lassen, näher an der Wahrnehmung realer Objekte aus dem Alltag liegen. So wirken diese weniger abstrakt und ihr Verständnis wird weiter vereinfacht.

### 3.3 Dynamik

Ein weiterer essenzieller Vorteil, den Partikelsimulationen bieten, ist die Dynamik. Sie sind animiert und dadurch ist es möglich, Konzepte der Bewegung zu visualisieren, was mit Lehrbüchern, Skizzen und Erläuterungen unmöglich ist. Schüler haben beispielsweise häufig unrealistische oder unphysikalische Vorstellungen physikalischer Prozesse, da diese die mentale Abstraktionsfähigkeit stark herausfordern. Wie durch Wilhelm und Heuer (1995) experimentell bewiesen, hat ein Großteil der Schüler auch nach erhaltenem Unterricht immer noch kein oder ein fehlerhaftes mentales Modell im Bezug auf Prozessabläufe. Das von Wilhelm und Heuer (1995) vorgeschlagene Konzept zur Lösung dieses Defizits umfasst einen experimentellen Aufbau, welcher den Schülern dynamisch den jeweiligen Prozess visualisiert. Eine Partikelsimulation könnte eben diese dynamische Visualisierung rein digital abdecken. Im Bereich der Astrophysik könnten auf diese Weise zahlreiche dynamische Phänomene visualisiert werden. Beispielhaft zu nennen sind Teilchenbewegungen, Temperatur- und Lichtveränderung, Sternen und Planetenbewegungen bis hin zu Supernovae.

### 3.4 Die Rolle des Experimentierens im Physikunterricht

Im Unterricht wird zu großen Teilen das Konzept des klassischen Frontalunterrichts angewandt. Durch die Lehrkraft werden Inhalte erklärt und die Schüler sind aufgefordert, für sich relevante Informationen festzuhalten. Seltener werden Konzepte durch den Lehrer experimentell präsentiert und noch seltener erarbeiten sich Schüler die Informationen durch Experimentieren selbst. Das sind Erkenntnisse aus einer IPN-Videostudie der Zeitschrift für Pädagogik. Das Lernen der Schüler wird "auf die Rezeptive Verarbeitung von Inhalten" eingeschränkt. (Seidel et al., 2006, S.805)

Weiterhin zeigt die Studie, dass es selten gelingt den Schülern zu vermitteln, auf welchen Inhalt sich die Unterrichtseinheit konzentriert bzw. welches Lernziel angestrebt wird. Der Lehrer-Schüler-Dialog wird stark vom Lehrer dominiert und die gestellten Fragen sind meist nur mit Stichworten zu beantworten. Die Studie analysierte ebenfalls die Trennung von Lern- und Leistungsphasen. Dabei wurde festgestellt, dass die Trennung selten klar vorgenommen wird und Schüler so bei der Erarbeitung von Informationen das Gefühl bekommen, bewertet zu werden. Das führt dazu, dass "eigene Ideen und Überlegungen sehr viel vorsichtiger geäußert werden". (Seidel et al., 2006, 806)

Das Experimentieren wird im Physikunterricht zu selten eingesetzt, um Informationen zu vermitteln, und wenn experimentiert wird, erfolgt das oft auf wenig effektiven Wegen. Schülerexperimente sind eng mit vordefinierten Ergebnissen durchgeplant. Den Schülern wird insgesamt zu wenig Freiraum zum offenen Experimentieren und zum selbstständigen Erarbeiten von Informationen geboten. (*Braun & Backhaus, 2006, S.1f*)

Ein offenes Experimentieren wird allgemein als sinnvoll angesehen, da sich Schüler so ohne Bewertungs Druck physikalische Konzepte und mentale Modelle erarbeiten können. Ihr persönliches Interesse an den Themengebieten wird erhöht und die intrinsische Motivation zum Lernen gestärkt. (*Braun & Backhaus, 2006, S.1ff*)

Dieses offene Experimentieren benötigt jedoch das passende Umfeld, welches in der Umsetzbarkeit stark vom jeweiligen Thema abhängig ist. Im Bereich der Astronomie wird das Experimentieren daher schnell eine Herausforderung, da sich kosmische Zusammenhänge schwierig mit einfachen Mitteln experimentell abbilden lassen. Eine Simulation würde eben diese Umstände digital nachbilden und Schülern die Möglichkeit zum selbstständigen, offenen Experimentieren geben. Durch eine visuelle Nachbildung von realen Phänomenen werden diese beobachtbar. Dies würde einerseits Interesse fördern und andererseits ein besseres Lernumfeld mit weniger Leistungsdruck aufbauen. Jeder Schüler könnte in seinem individuellen Tempo an einem individuellen Teilaspekt des Themenbereichs arbeiten und die Simulation dient dabei als Werkzeug. Da sie digital abläuft, lässt sie sich ohne großen Ressourcenaufwand abrufen und es müssen keine Einschränkungen getroffen werden, da die Schüler bei dieser Form des Experiments keine Fehler machen können und Sicherheitsrisiken ausgeschlossen sind. So bietet eine Simulation als Lernwerkzeug den Schülern die Möglichkeit zum selbstständigen und freien Erarbeiten von Informationen. Dies wäre eine sinnvolle Kompensation für Defizite, welche durch die derzeitige Unterrichtsgestaltung auftreten.

## 4 Simulationen als Mittel zur Prozessvisualisierung

### 4.1 Partikelsimulationen und deren Anwendungsgebiete

#### Partikel

Die Definition des Begriffes unterscheidet sich je nach Gebiet. In der Physik versteht man unter Partikeln in der Regel ein kleines Teilchen wie beispielsweise ein Elektron oder ein ganzes Atom. (*Oxford Dictionary, 2023, "particle, n."*)

#### Simulation

Als Simulation wird die künstliche Nachbildung eines Prozesses verstanden, der in der Realität existieren könnte. (*Oxford Dictionary, 2023, "simulation, n."*)

Als Partikelsimulation lässt sich folglich die Nachbildung eines Prozesses, bestehend aus einzelnen Teilchen, verstehen. Die Partikel bilden ein Modell, welches sich mit der Zeit anhand von gegebenen Eigenschaften dynamisch verändert. Partikel interagieren dafür mit ihrer Umgebung und untereinander und definieren so den Ablauf der Simulation. Im Rahmen dieser Arbeit werden digitale Simulationen thematisiert. Der Computer übernimmt hier die Berechnung der Partikelbewegungen anhand gegebener Parameter. Partikelsimulationen finden in vielerlei Gebieten Einsatz. Sie kommen immer dann infrage, wenn die Simulation eines bewegten Mediums angestrebt wird. Reale Materie besteht ebenfalls aus Partikeln, die miteinander interagieren. Bildet man diese nun digital nach und weist ihnen bestimmte Eigenschaften zu, ist es möglich komplexe Prozesse realitätsgetreu zu simulieren.

#### 4.1.1 Anwendungsgebiet Wissenschaft

In der Physik können Partikelsimulationen beispielsweise eingesetzt werden, um wissenschaftliche Prognosen für bestimmte Prozesse aufzustellen. Dabei werden Simulationen für Prozesse auf allen Größenordnungen von molekularer Ebene bis hin zu astronomischen Phänomenen eingesetzt. Für die Berechnung des Verhaltens von Materie im Plasmazustand gibt es beispielsweise einige Algorithmen, welche konstant weiterentwickelt und in der Wissenschaft eingesetzt werden. (*Verboncoeur, 2005, S.1*) Da Plasma extreme Temperaturbedingungen benötigt, welche sich im Labor kaum nachbilden lassen, bieten sich digitale Repräsentationen in diesem Fall an. In größerem Maßstab gibt es physikbasierte Partikelsimulationen beispielsweise für die Nachbildung der

Sternformation und dem Verhalten von Gasen im Weltraum (*Navratil et al., 2007, S. 1717*), bis hin zur Nachbildung der Raumzeit-Hierarchie in kosmischen Partikel Nebeln. (*Fraedrich & Westermann, 2012, S.3*) Für die Entwicklung von wissenschaftlichen Simulationen werden verschiedene Programme verwendet. Beispielhaft zu nennen sind: "IFrIT - Ionization Front Interactive Tool", "MayaVi Data Visualizer" oder "ParaView". (*Kapferer & Riser, 2008, S.5f*)

Bei dieser Art von Simulationen steht das Ziel der Erkenntnisgewinnung im Vordergrund. Man erhofft sich, Prozesse so physikalisch korrekt wie möglich nachzubilden, um Schlüsse auf die reale Welt ziehen zu können. Zur Abbildung realer Bedingungen sind sehr komplexe Berechnungen vonnöten, welche große Datenmengen produzieren. Dafür sind entsprechende Hardware sowie ein hohes fachliches und informatisches Verständnis zur Auswertung und Erstellung der Simulationen vonnöten. (*Kapferer & Riser, 2008, S.8*)

#### **4.1.2 Anwendungsgebiet Filmindustrie**

In der Filmindustrie werden Simulationen eingesetzt, um verschiedene Prozesse abzubilden, welche zur Erzählung einer Geschichte beitragen. Da beispielsweise große Explosionen, abstrakte Magieeffekte oder Wassermassen nicht ohne weiteres real aufzunehmen sind, werden diese meist digital nachgebildet und im Nachhinein in die Aufnahmen integriert. Das Einsatzfeld von Partikelsimulationen ist auch hier sehr weitreichend. Angefangen bei Wasser und Ozeanen (*Koshizuka, 2012, S.155ff*) über Rauch- und Feuersimulationen (*Minor, 2007, S.3ff*) bis hin zu Zerstörung und Staub. (*Liu et al., 2007*)

Im Gegensatz zur Wissenschaft ist es in diesem Anwendungsfeld nicht beabsichtigt, spezifische Erkenntnisse aus den Simulationen zu gewinnen. Ihre Bedeutung liegt vielmehr in der künstlerischen Gestaltung und soll dem Zuschauer ein bestimmtes Gefühl vermitteln. Dabei können die Effekte subtil wirken, um nicht den realen Look des Filmes zu brechen oder absichtlich surreal erscheinen, um die Szene optisch aufzuwerten. Häufig besteht das Ziel darin, einen realen Prozess wie beispielsweise einen Hausbrand für den Film nachzubilden. Da die Zuschauer ungefähr wissen, wie ein Brand aussehen muss, ist es in diesem Fall das Ziel, die Flammen und die Zerstörung des Gebäudes so realitätsnah wie möglich abzubilden. Dafür werden nun, ähnlich wie im wissenschaftlichen Anwendungsfeld, physikalische Parameter genutzt, um ein realistisches Ergebnis zu erhalten. Im Bereich der Film- und VFX (Visual Effects) Industrie, wird hierfür vor allem



die 3D-Software “Houdini” eingesetzt. Der Fokus dieses Programms liegt dabei auf dem größtmöglichen Grad an Kontrolle, da oft ein sehr spezifischer Look für die Simulation angestrebt wird, um letztendlich den kreativen Wünschen der Ersteller zu entsprechen. Daher ist es nicht nötig, jeden Partikel mit allen Parametern detailliert zu simulieren, sondern lediglich den für den Zuschauer sichtbaren Bereich. Weiterhin können Prozesse, welche nicht im Fokus der Kamera stehen oder in sehr kleinem Maßstab ablaufen, stark vereinfacht simuliert werden, um so Ressourcen zu sparen. Allgemein lässt sich sagen, dass die benötigte Hardware und Simulationszeit hier stark von der Komplexität der Simulation abhängt, welche wiederum durch die kreativen Ansprüche bestimmt wird.

#### **4.1.3 Anwendungsgebiet Spieleindustrie**

Auch in der Spieleindustrie sind Partikelsimulationen ein häufig genutztes Mittel zur visuellen Aufwertung des Spieles, weswegen die Anwendungsgebiete denen der Filmindustrie stark ähneln. Es wird auch hier ein möglichst ansprechender Look angestrebt, um dem Spieler ein bestimmtes Gefühl zu vermitteln. Der wichtigste Unterschied besteht in der Umsetzung dieses Zieles.

Da Spiele interaktive Echtzeitanwendungen sind, ist es vonnöten, die Simulationen in Echtzeit durchzuführen. Da sie außerdem darauf ausgelegt sind, auf vergleichsweise leistungsschwacher Hardware zu funktionieren, ergibt sich ein anderer Arbeitsablauf für die Umsetzung von Simulationen. Die Bewegung jedes Partikels muss 60 mal in der Sekunde aktualisiert werden (die Aktualisierungsrate der meisten Spiele), um dem Spieler das Gefühl von Dynamik zu vermitteln. Um dies zu ermöglichen, müssen die Berechnungen so effizient wie möglich ablaufen und alle rechenaufwändigen Prozesse müssen so weit wie möglich vereinfacht werden. Physikalisch korrekte, bzw. realistische Simulationen stellen daher eine große Herausforderung für die Spieleindustrie dar. Ein stilisiertes Aussehen lässt sich trotzdem recht gut erreichen, beispielsweise mit der “Unity” Entwicklungsumgebung. Das Erreichen eines realistischen Looks ist weiterhin ein Feld, welches es auszubauen gilt. Vorreiter hierbei ist die “Unreal Engine”, eine Entwicklungsumgebung, mit der man sehr effizient auch aufwendige Simulationen in Echtzeit berechnen kann. (Oliver, 2012)

## 4.2 Houdini als Simulationstool - Aufbau der Umgebung

Houdini ist eine von SideFX entwickelte 3D Software, die ein breites Feld an Funktionen für die Erstellung dreidimensionaler Computergrafik bietet. Beispielhaft zu nennen sind Modelling, Animation, Partikelsysteme, Licht, Kamera, Rigging und Rendering.

Neben den aufgeführten Features hat die Software noch einen weiteren Vorteil, der sie vor allem für große Produktionen interessant macht: die Prozeduralität. Houdini ist Node basiert, was bedeutet, dass alle Daten und alle Modifikationen in sogenannten Nodes gespeichert werden. Diese Nodes sind visuelle Repräsentationen der Datenblöcke, die im Code des Programmes ausgeführt werden. Bei der Arbeit mit Houdini erschafft der Nutzer ein Netzwerk aus verschiedenen Nodes, die dann nacheinander angewandt werden. Jede Operation und jeder Parameter kann dabei zu jedem Zeitpunkt verändert werden und das Netzwerk wird neu berechnet.

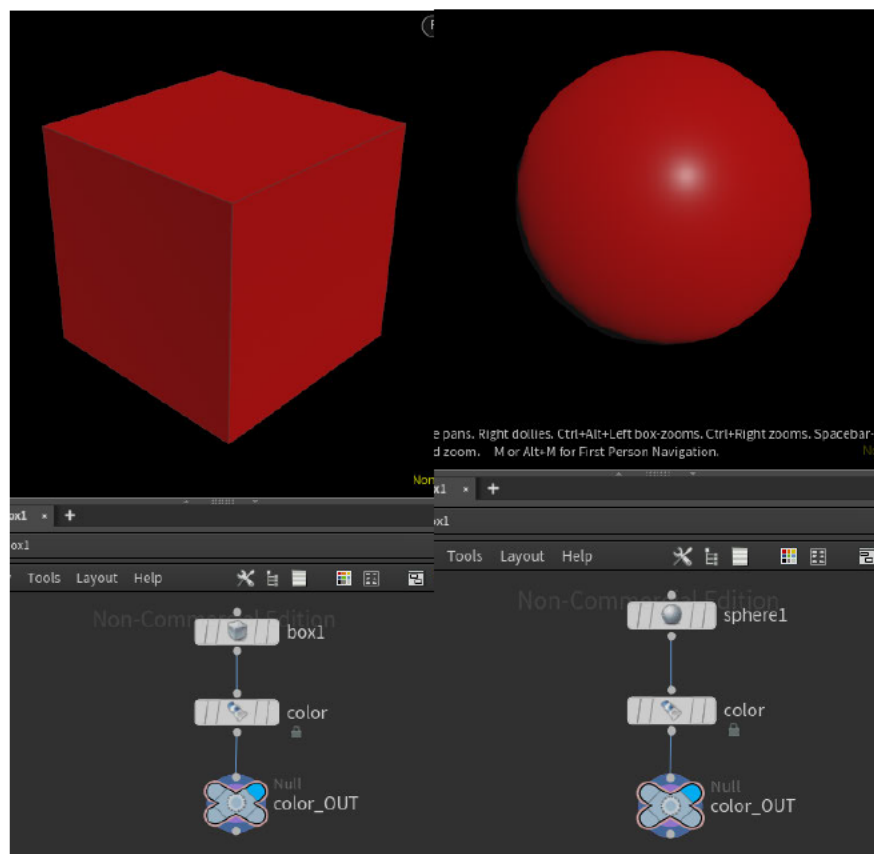


Abbildung 12: “Beispiel für prozedurale Operation in Houdini”

Auf Abb. 12 wird ein Beispiel für die beschriebene Funktionsweise veranschaulicht. Auf der linken Seite ist ein Würfel zu sehen, welcher durch eine “color”-Node rot eingefärbt wird. Durch die prozedurale Funktionsweise Houdinis, ist es möglich, wie auf der rechten

Seite zu sehen, den Würfel durch eine Kugel zu ersetzen, ohne dass das gesamte Netzwerk neu aufgesetzt werden muss.

Durch diesen grundlegenden Mechanismus erhält Houdini seine Prozeduralität. Das Programm ermöglicht es, erstellte Arbeit flexibel zu modifizieren und erleichtert so beispielsweise die Umsetzung von Feedback und Kundenwünschen. Weiterhin ist es möglich, die Umgebung durch verschiedene Programmiersprachen (VEX, HScript, Python) selbst zu modifizieren. Aus diesem Grund lässt sich Houdini gut in verschiedenste Pipelines (Arbeitsketten) einbinden. [1]

Die Houdini Oberfläche ist grundlegend in folgende Abschnitte aufgeteilt:

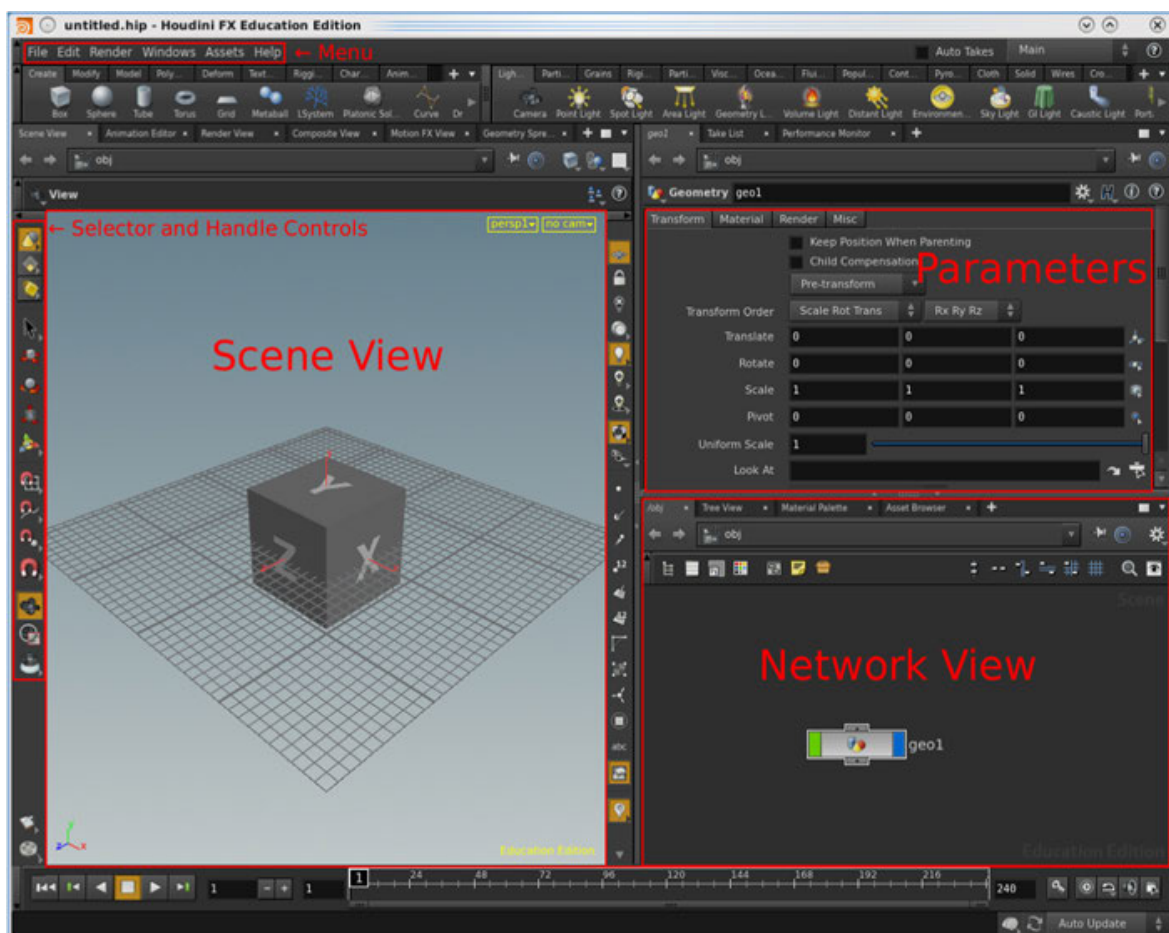


Abbildung 13: “Übersicht über die Houdini Benutzeroberfläche”

### Scene View

Hier ist in der Regel eine Vorschau des dreidimensionalen Objektes zu sehen, das durch die Nodes erstellt wird. Einige Operationen können direkt im 3D Raum durchgeführt werden.

## **Parameters**

In diesem Bereich werden alle Parameter der ausgewählten Node angezeigt, die vom Benutzer modifiziert werden können.

## **Network View**

Im Network View erfolgt der größte Teil der Arbeit bei der Benutzung von Houdini. Hier wird das Node Netzwerk erstellt und modifiziert, welches die 3D-Szene kreiert.

## **Playbar**

Eine Leiste, auf der die ausgewählte Frame Range angezeigt wird, mit der Animationen abgespielt werden können.

Es gibt verschiedene sogenannte "Kontexte", in denen man Netzwerke in Houdini erstellen kann. Sie repräsentieren die interne Ordnerstruktur, in die Node-Netzwerke eingeteilt werden. Sie erfüllen unterschiedliche Funktionen, welche für die Erstellung von Szenen relevant sind. Die verfügbaren Nodes unterscheiden sich je nach Kontext und man kann zwischen ihnen am oberen Rand des Network View wechseln. Die für den Rahmen dieser Arbeit relevanten Kontexte sind:

### **Obj (Object)**

Auf diesem Level werden alle erstellten Objekte angezeigt. Es können einige Modifikationen durchgeführt werden, welche sich auf das gesamte Objekt beziehen.

### **SOPs (Surface Operators)**

Wurde im Obj-Tab eine Node erstellt, dient diese als eine Art Ordner für die enthaltenen SOPs. Hierbei handelt es sich um Nodes zur Erstellung und Manipulation von Objekten.

### **DOPs (Dynamic Operators)**

Netzwerke zur Erstellung von Simulationen. Als Grundlage dient in den meisten Fällen eine SOP-Node, welche importiert und daraufhin durch die Simulation modifiziert wird.

## 4.3 POPs - Grundprinzipien von Partikelsimulationen in Houdini

Simulationen werden in Houdini wie im vorangegangenen Teil geschildert durch DOP-Netzwerke umgesetzt. Erstellt man ein solches Netzwerk, befindet sich darin zunächst nur eine Output Node. Sie bildet das Ende des Simulations-Netzwerkes und gibt das Ergebnis im SOP Kontext aus. Die übrigen für die Simulation benötigten Nodes können nun selbst erstellt werden. Für die Erstellung einer Partikelsimulation eignet sich ein Popnet. Dabei handelt es sich um ein Dopnet, welches bereits alle Nodes, die für eine funktionierende Partikelsimulation benötigt werden, enthält.

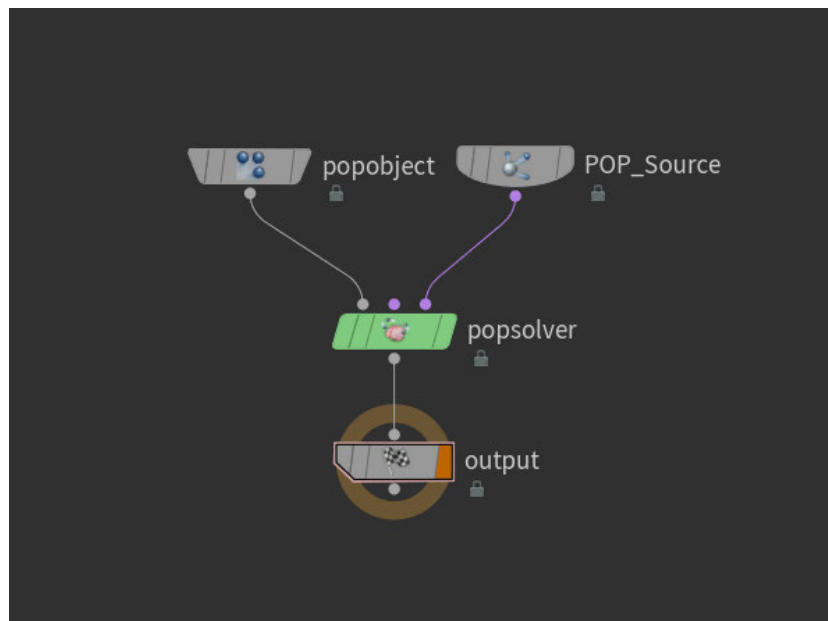


Abbildung 14: "Aufbau eines Popnets"

### POP Object

Diese Node konvertiert ein Partikelsystem in ein dynamisch animiertes Objekt, welches mit anderen DOP Objekten interagieren und vom Popsolver gelesen werden kann.

### Pop Source

Die Pop Source Node generiert Partikel auf Grundlage einer aus dem SOP-Kontext eingelesenen Node (z.B. ein Mesh). Sämtliche Parameter (Anzahl, Lebensdauer etc.) sind dabei konfigurierbar.

### Pop Solver

Hier werden die Bewegungen der vom Pop Object geschaffenen Partikel anhand gegebener Eigenschaften berechnet. Das Resultat wird im Output ausgegeben.

Mit diesen Nodes können Partikel erschaffen werden, die Punkte im 3D-Koordinatensystem repräsentieren. Jeder Punkt hat dabei einige Eigenschaften, welche im sogenannten “Geometry-Spreadsheet” angezeigt werden. Diese können unter anderem vom Datentyp Integer (Ganzzahlen), Float (Fließkommazahlen), Vector, oder String (Text) sein. Einige für Partikelsimulationen relevante Parameter sind:

**Index (id)**

Ein integer Wert, welcher jedem Punkt individuell zugeordnet wird.

**Position (P)**

Dieser Vektor speichert die drei Koordinaten des jeweiligen Punktes als Float Werte ( $P[x]$ ,  $P[y]$ ,  $P[z]$ ).

**Color-Data (Cd)**

Der Farbwert des Partikels als Vektor im RGB-Format. Die einzelnen Farbwerte werden jeweils als float gespeichert ( $Cd[r]$ ,  $Cd[g]$ ,  $Cd[b]$ ).

**Velocity (v)**

Die Bewegungsrichtung des Partikels wird in diesem Vektor gespeichert. Die Länge des Vektors bestimmt die Geschwindigkeit. Die drei Werte sind im float Format ( $v[x]$ ,  $v[y]$ ,  $v[z]$ ).

**age**

Das Alter des Partikels in Sekunden als float Wert.

Damit der Solver eine Bewegung für Partikel berechnen kann, müssen einige dieser Eigenschaften modifiziert werden. Am wichtigsten dafür ist der Velocity Wert, da der Solver diesen jeden Frame zum Positionsvektor des Partikels addiert und so die Bewegung erzeugt wird. Dafür eignen sich beispielsweise sogenannte “Fields”. Darunter versteht man dreidimensionale Gitter, welche an jedem Punkt einen individuellen Wert besitzen, durch den ein bestimmtes Attribut abgebildet wird. Ein Strudel kann beispielsweise durch ein Field erschaffen werden, welches aus kreisförmig ausgerichteten Vektoren besteht. Wenn dadurch nun der Velocity Vector manipuliert wird, bewegen sich Partikel im Kreis zum Zentrum hin.

Die Manipulation der Partikeleigenschaften und somit deren Animation kann auf verschiedenen Wegen erfolgen. Houdini bietet im DOP-Kontext eine Reihe an Nodes, welche einem ermöglichen, über die Manipulation von Parametern zu kontrollieren, wie die Bewegung der Partikel ablaufen soll. Einige Beispiele hierfür sind:

### **POP Attract**

Zieht die Partikel an eine bestimmte Koordinate oder zu festgelegter Geometrie.

### **POP Drag**

Definiert einen Luftwiderstand und/oder eine Windrichtung, welche die Bewegung der Partikel verlangsamen oder modifizieren.

### **POP Axis Force**

Lenkt die Partikel um eine festgelegte Achse herum und gibt ihnen damit eine kreisende Bewegung.

### **POP Curve Force**

Lenkt Partikel entlang eines Pfades, der durch eine im SOP-Kontext erstellte Kurve definiert wird.

Neben diesen Nodes, die festgelegte Funktionen erfüllen, gibt es die Möglichkeit, Eigenschaften durch Code in "Wrangle"-Nodes zu manipulieren. Hier wird in der Regel die Programmiersprache VEX verwendet. Es können unter anderem neue Eigenschaften erschaffen werden, die Syntax dafür lautet: "*Datentyp@Name = Wert;*". Weiterhin können verschiedene Vergleichsoperationen wie if-statements durchgeführt werden.

A screenshot of a VEX expression editor window. The title bar says "VEXpression". The editor contains three lines of code: line 1: `if (@age > 5){`, line 2: `removepoint(0, @ptnum);`, line 3: `}`. The code is color-coded: `if` is purple, `@age` is orange, `>` is blue, `removepoint` is green, `(0,` is blue, `@ptnum)` is orange, and `;` is blue.

*Abbildung 15: "Beispiel Für VEX-Operation mit if-Abfrage"*

Im Beispiel aus Abb. 13 wird anhand des age-Attribute geprüft, ob ein Partikel ein Alter von fünf Sekunden überschritten hat. Trifft die Bedingung zu, wird dieser mit der removepoint Methode entfernt.

Auf diese Weise ist es möglich, die Simulation sehr spezifisch zu modifizieren und an die jeweiligen Wünsche anzupassen. Der Rechenaufwand, der vonnöten ist, um die Simulation zu kalkulieren, steigt mit der Partikelanzahl und den durchgeführten Operationen stark an. Ab einer bestimmten Komplexität wird die Simulations-Vorschau im Viewport nur sehr stockend angezeigt. Um trotzdem ein flüssiges Bild zu erhalten, werden in der Regel Cache-Nodes verwendet, welche die Simulation zwischenspeichern. Diese Node berechnet die Simulation vollständig und speichert die erhaltenen Positionen und Eigenschaften der Partikel für jeden Frame ab. Dieser Cache muss nun lediglich ausgelesen werden, ohne Berechnungen durchzuführen, wodurch die Darstellung im Viewport wieder deutlich flüssiger wird. Der Nachteil daran ist jedoch, dass jedes Mal, wenn eine Änderung vor dem Cache getroffen wird, dieser neu berechnet werden muss. [1]



## 4.4 HDAs - Der Aufbau eigener Nodes als Werkzeuge

Als HDA versteht man ein "Houdini digital asset", welches es Usern ermöglicht, eigene Funktionalitäten in einer selbst erstellten Node zusammenzufassen. Dabei kann nahezu jede Node zu einem HDA zusammengefasst und daraufhin individuell modifiziert werden. digitale Assets haben zunächst den Vorteil, dass sie sich beliebig oft replizieren lassen. Dadurch lassen sie sich zum Beispiel gut zur Generierung von prozeduralem Content einsetzen. Das zugehörige SOP Netzwerk wird als HDA zusammengefasst und nun kann es sehr effizient dupliziert werden und durch die Modifikation von wenigen Parametern können Variationen erschaffen werden. Digitale Assets lassen sich außerdem auf andere Systeme übertragen. Dazu müssen sie lediglich über die Funktion "Import digital Asset" referenziert werden und sind sofort als Node in der Datei verfügbar. Aufgrund dieses unkomplizierten Installationsprozesses ist diese Funktionalität vor allem für die Zusammenarbeit in Firmen interessant. Features und Node-Setups können in einer HDA gespeichert und übertragen werden und der User kann sie wie jede andere Node verwenden, ohne dabei die innere Funktionalität verstehen zu müssen. Ein großer Teil der standardmäßig in Houdini vorhandenen Nodes sind ebenfalls HDAs, welche von Houdini Nutzern oder Entwicklern erstellt wurden.

Nachdem ein Node-Setup als digitales Asset zusammengefasst wurde, ist es möglich, eine eigene Benutzeroberfläche für die entstandene Node anzulegen. Darauf lassen sich alle Parameter verlinken ("promoten"), die sich im unterliegenden Setup befinden. Mit Hilfe dieser Parameter können die Ergebnisse der Node modifiziert werden. So könnte beispielsweise, bei einer HDA, zur Generierung von Gebäuden der "Uniform-Scale" Parameter der Ursprungs-Geometrie promoted werden. Dieser Wert kann nun in der HDA-Node selbst angepasst werden und das erhaltene Gebäude ändert seine Größe entsprechend, ohne dass der Nutzer dafür den Wert im Node Setup erst suchen muss. Parameter können dabei nicht nur direkt verlinkt werden. Es ist möglich, ihnen Logik in Form eines Python-Skripts zu hinterlegen. Außerdem können Usereingaben durch verschiedene UI-Elemente getätigt werden (Button, Slider, Drop-down-menu), welche ebenfalls eingelesen und durch Python Logik verarbeitet werden können. HDAs können also als Werkzeug zur Erstellung eigener Funktionalitäten in Houdini verwendet werden, welche sich effizient an andere Nutzer übertragen lassen. [1]

## 4.5 Houdini im Vergleich mit anderen Möglichkeiten der Visualisierung

Für die Erstellung der im Rahmen dieser Arbeit entstandenen Simulation, wurde die Houdini Software genutzt. Diese Entscheidung ist durch mehrere Faktoren begründet. Wie im Kapitel 4.1 dargestellt, hätte die Umsetzung im Wesentlichen mit drei Ansätzen erfolgen können: mit wissenschaftlicher, physikalisch korrekter Simulationssoftware, mit einer Entwicklungsumgebung für Spiele oder mit 3D VFX Software.

Wissenschaftliche Software hat gegenüber dem gewählten Ansatz den Vorteil, dass sie ein sehr exaktes und realistisches Resultat erschafft. Alle denkbaren Faktoren werden mit größtmöglicher Genauigkeit in die Berechnung einbezogen. Bestünde das Ziel der Anwendung in der Überprüfung wissenschaftlicher Zusammenhänge, wäre diese Art von Software das am besten geeignete Mittel zur Umsetzung. Da das Hauptziel jedoch in der Visualisierung und Informationsvermittlung besteht, wird keine wissenschaftlich exakte Simulation benötigt. Die ausgegebenen Modelle der entsprechenden Software sind außerdem meist darauf ausgelegt, bestimmte Zusammenhänge isoliert zu betrachten, beispielsweise die Temperaturverteilung in einer jungen Galaxie. Der Fokus liegt dabei auf den ausgelesenen mathematischen Werten und nicht auf der dreidimensionalen Darstellung von visuell verständlichen Modellen. Betrachtet man Zielgruppe und Anwendungsfeld lässt sich ebenfalls feststellen, dass physikalische Simulationssoftware für den Anwendungsfall ungeeignet ist. Einerseits ist die Bedienung der Anwendungen ohne astrophysikalische Vorkenntnisse und Programmierkenntnisse nahezu unmöglich. Andererseits genügt die in Schulen vorhandene Hardware nicht, um solche komplexen Simulationen durchführen zu können.

Die Umsetzung der Simulation mit Hilfe einer Entwicklungsumgebung für Spiele erscheint zunächst naheliegend, da die Interaktion einen wesentlichen Bestandteil des didaktischen Konzeptes darstellt. (*Siehe Kapitel 3.2*) Bei Spielen handelt es sich um interaktive Anwendungen, weswegen diese Entwicklungsumgebungen großen Fokus auf die Interaktion und Performance legen. Es soll während des Spielens so wenige Unterbrechungen wie möglich geben, um dem Spieler ein immersives Erlebnis zu bieten. Wie in Kapitel 4.1.3 ausgeführt, müssen deshalb alle Aspekte der Anwendung dem Zweck der Performance untergeordnet werden. Da es sich bei Simulationen aber um komplexe Prozesse handelt, müssen diese von der Game-Engine vereinfacht werden, um während der Laufzeit berechenbar zu bleiben. Die Simulationen beruhen daher meistens kaum auf

realen physikalischen Konzepten und die Möglichkeiten der Umsetzung sind im Vergleich zu denen von 3D Software stark begrenzt. Es wäre also theoretisch möglich gewesen, die Anwendung mit Hilfe einer Spiele-Entwicklungsumgebung umzusetzen, jedoch hätten dafür Abstriche in der Komplexität und somit teilweise auch der Verständlichkeit der Modelle gemacht werden müssen. Da auf diesen jedoch der größte Fokus liegt, wurde dafür die Einbuße im Bereich der Interaktivität in Kauf genommen.

Der größte Nachteil in der Verwendung von Houdini sind die unumgänglichen Ladezeiten. Das Programm ist nicht zur Entwicklung von Echtzeitsimulationen geschaffen. Es ist zwar möglich, einen Cache zu generieren, um die entsprechende Simulation flüssig zu betrachten, jedoch ist die Berechnung des Caches ebenfalls mit einer Ladezeit verbunden. Dieser Aspekt könnte sich negativ auf die Interaktivität der Simulation auswirken, da der Nutzer weniger gewillt sein könnte, mit den Werten zu experimentieren, wenn jede Veränderung eine Wartezeit mit sich bringt. Dieser Aspekt wird im Methodik-Teil weiter betrachtet. Die Nutzung von Houdini als Simulationstool bringt jedoch auch einige entscheidende Vorteile mit sich. Durch die Vielzahl an Möglichkeiten zur Erstellung von Partikelsimulationen, können alle Prozesse visuell ansprechend dargestellt werden. Die User Eingaben können verarbeitet und auf die Simulation angewandt werden, um für jeden veränderten Parameter ein individuelles Ergebnis auszugeben. Mit Hilfe von HDAs und Python-Code lässt sich ein übersichtliches User Interface zur Zusammenfassung aller relevanten Parameter erstellen. Simulationen können in ihrer Bewegung betrachtet werden und mit Hilfe der Playbar kann jeder Frame pausiert und einzeln angesehen werden. Die freie Betrachtung des Modells im 3D Koordinatensystem ist ebenfalls möglich.

Zusammenfassend lässt sich feststellen, dass wissenschaftliche Simulationstools aufgrund von zu hoher Komplexität nicht für die Umsetzung von interaktiven Simulationen infrage kämen. Spiel-Entwicklungsumgebungen ermöglichen zwar die Umsetzung interaktiver Anwendungen, jedoch sind die Kontrollierbarkeit sowie die vorgegebenen Möglichkeiten zur Erstellung realistischer Simulationen aufgrund der nötigen Performance stark vereinfacht. Houdini ist für die optisch ansprechende Visualisierung von Prozessen ausgelegt und bietet eine Reihe an Funktionen, welche dieses Ziel unterstützen. Daher ist Houdini das im Methodikteil genutzte Programm.

## **5 Erstellung einer interaktiven Simulation des stellaren Lebenszyklusses**

### **5.1 Rahmenbedingungen**

Zur Überprüfung der Forschungsfrage soll im Folgenden eine dreidimensionale Partikelsimulation erstellt werden. Diese richtet sich an Schüler in der Sekundarstufe II im Rahmen des Physikunterrichts. Das Feld der Astronomie ist in den meisten Lehrplänen als fakultatives Modul vorgesehen, wie beispielsweise in Bayern in der zwölften Klasse. (*ISB, 2004*)

Die Simulation dient dabei als digitales Experiment, wie in 3.4 beschrieben und kann beispielsweise im Rahmen einer Doppelstunde zum Themenkomplex “Sterne” eingesetzt werden. Jeder Schüler kann sie selbst bedienen, wofür pro Schüler ein Computer mit einer installierten Version von Houdini (19.5 aufwärts) benötigt wird. Die entsprechende HDA sollte sich in der Houdini Ordnerstruktur befinden. Da nicht alle visualisierten Prozesse bis ins Detail erklärt werden, ist es von Vorteil, wenn die Schüler bereits ein gewisses astrophysikalisches Vorwissen besitzen. Es sollte ein grobes Verständnis von Sternen und stellarer Evolution sowie von den Konzepten der Gravitation und Kernfusion bestehen. Die Lehrkraft sollte grundlegende Fähigkeiten im Umgang mit der Houdini Software besitzen, damit bei auftretenden Fragen und Problemen geholfen werden kann.

Für die Durchführung des Experiments wird den Schülern zunächst durch die Lehrkraft demonstriert, wie die Simulation zu bedienen ist. Dann kann sich für einige Minuten ausprobiert werden, um sich mit dem Umgang mit den Parametern und der Navigation im 3D Viewport vertraut zu machen. Dann bearbeiten die Schüler individuell einige Aufgaben, die sie mit Hilfe der Simulation lösen können. Der Fokus liegt hierbei auf der allgemeinen Erfassung aller stellaren Evolutionsstufen und dem Einfluss der Masse auf die Entwicklungsgeschwindigkeit eines Sternes. Zu bearbeitende Aufgaben sind beispielhaft auf Abbildung 16 in Form eines Arbeitsblattes aufgeführt. Da die Simulation deutlich mehr Informationen enthält als durch die gewählten Aufgaben abgefragt werden, besteht die Möglichkeit für die Lehrkraft ein individuelles Arbeitsblatt zu entwickeln, um den Umfang zu erhöhen oder zu senken.

## 5.2 Angestrebtes Ziel

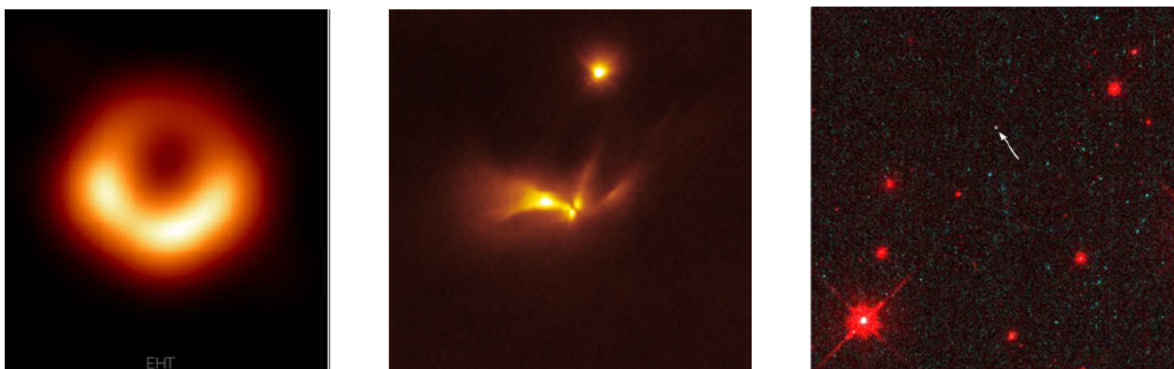
Die Simulation soll in Houdini als HDA komprimiert sein, damit der User lediglich eine Node hinzufügen muss und keine weiteren Schritte zum Setup der Datei notwendig sind. Die Viewport-Konfigurationen sind entsprechend vorgenommen, um die Partikel optimal betrachten zu können. Dafür ist das "high-quality-lighting with shadows" aktiviert, der Hintergrund ist auf "dark" gestellt, das 3D-Grid ist deaktiviert und die Point-Guides haben die entsprechende Größe. Diese Einstellungen können zuvor in Houdini als Standard festgelegt werden, weswegen sie nur einmal nach der Installation durchgeführt werden müssen.

Ist das Stern-Simulator HDA in die Datei hinzugefügt, erscheint im "Parameters"-View das UI der Node. Hier soll es Eingabemöglichkeiten für Masse und vergangene Zeit geben, welche vom User eingestellt werden können. Über einen Knopf soll der User die Simulation starten können. Es soll nun durch das HDA berechnet werden, welche stellare Evolutionsstufe der Stern mit der gegebenen Masse nach der gegebenen Zeit erreicht hätte. Die entsprechende Simulation soll dem User angezeigt werden. Er soll diese nun mit der Playbar abspielen und sie sich in Bewegung anschauen können. Seine Eingaben sollen dabei nicht nur die Art der angezeigten Simulationen verändern, sondern ebenso bestimmte Parameter Simulations-intern skalieren. Das bedeutet beispielsweise, dass die Geschwindigkeit der Partikel Bewegungen einer interstellaren Wolke mit der Zeit ansteigen soll. Über zwei weitere Knöpfe soll es möglich sein, eine Evolutionsstufe nach vorn oder nach hinten zu springen, um alle Stufen einer Masse-Einstellung problemlos beobachten zu können. Ist eine Simulation angesteuert, sollen dem User Informationen zum stellaren Objekt in Textform angezeigt werden. Über einen weiteren Knopf soll bei den meisten Simulationen (außer Supernova, Planetarer Nebel und Schwarzes Loch) ein Querschnitt aktivierbar sein, um das Verständnis für den inneren Aufbau zu fördern. Im Network-View-Fenster soll gleichzeitig zu jedem stellaren Objekt ein individueller Erklärtext angezeigt werden, welcher sowohl allgemeine Informationen zum geschaffenen Stern, als auch spezifische Eigenschaften wie Lebenszeit und Spektralklasse enthalten soll. Diese Parameter sollen durch die Eingaben des Users beeinflusst werden, um das Verständnis für die Zusammenhänge der wirkenden Kräfte und deren Relation mit Masse und Zeit zu fördern.

## 5.3 Vordiskussion

Bei der Umsetzung der Simulation gibt es eine Reihe an Faktoren, welche es abzuwägen gilt. Im Wesentlichen soll die Anwendung zum Verständnis der astrophysikalischen Prozesse aus Kapitel 2 beitragen, indem die in Kapitel 3 erörterten Faktoren möglichst gut adressiert werden. Die dargestellten Modelle sollen also möglichst visuell ansprechend, gleichzeitig aber auch dynamisch sein. Dabei muss das größtmögliche Level an Interaktion erhalten bleiben. Der User sollte nicht mit Informationen überflutet werden, gleichzeitig müssen relevante Fakten offensichtlich erkennbar sein.

Da der Zweck der Visualisierung zur Verständnis Vereinfachung im Fokus der Arbeit steht, werden die Simulationen keinem realistischen Darstellungsstil folgen. Dies ist zum einen dadurch begründet, dass wie in Abbildung 17 zu sehen, von vielen der stellaren Objekte bisher keine hochauflösenden Bilder existieren, welche als Vorlage verwendet werden könnten.



*Abbildung 17: “Aufnahmen eines Schwarzen Lochs (links), Protosterns (mitte), Neutronensterns (rechts)” (Muzerolle et al. ,2013), (Walter & NASA, 1997), (EHT Collaboration, 2022)*

Zum Anderen ist es für das Verständnis des Aufbaus eines Sternes deutlich sinnvoller, diesen nicht real abzubilden. Würde man einen Hauptreihenstern wie die Sonne beispielsweise mit realen Parametern von naher Distanz darstellen, müsste der Stern so hell erscheinen, dass er praktisch nicht mehr erkennbar ist. Für die Darstellung wurde der Stil einer Partikelwolke gewählt, da diese deutlich performanter im Viewport darzustellen sind als beispielsweise Rauch-Simulationen. Weiterhin haben sie den Vorteil, dass man durch sie hindurch sehen kann, was den Aufbau verständlicher präsentiert und beispielsweise den Ladungszustand und die Temperatur der Teilchen besser abbildet, da

sich jeder Partikel einzeln bewegen kann. Damit die Simulationen jederzeit ein verständliches Modell präsentieren, mussten weiterhin einige Details hervorgehoben bzw. abgewandelt werden. Der Kern eines Überriesen ist beispielsweise so klein im Verhältnis zur äußeren Hülle, dass man kaum beide Komponenten gleichzeitig sehen könnte. Da im Kern jedoch für das Verständnis relevante Prozesse ablaufen, wird er für die Simulation vergrößert dargestellt.

Eine weitere Abwägung muss im Bezug auf den Detailgrad getroffen werden. Je weiter die Partikelanzahl erhöht wird, desto eindeutiger wird das Modell der Simulation zu erkennen sein. Durch eine Trail Node kann beispielsweise ein Partikel repliziert werden, wobei der zweite Partikel, versetzt, den gleichen Pfad durchläuft wie der erste. Dadurch können Ströme besser dargestellt werden, z.B. bei den Strudeln eines Protosternes, jedoch erhöht diese Node die Partikelanzahl drastisch. Um Wartezeiten so gering wie möglich zu halten, aber trotzdem weiterhin eine eindeutige Darstellung aller Simulationen erzeugen zu können, wurde hierfür festgelegt, dass die initiale Ladezeit einer Simulation zwischen fünf und zehn Sekunden betragen sollte. Dieser Wert wird hauptsächlich durch die Regulierung der Partikelanzahl erreicht.

Für die flüssige Betrachtung der Animationen mit mindestens 24 Bildern pro Sekunde, ist es meist notwendig, einen Cache zu erstellen. Dieser benötigt jedoch einige Sekunden Zeit zur Berechnung, die der User abwarten muss. Jedes Mal, wenn ein Parameter verändert wird, muss daraufhin der Cache neu berechnet werden, was erneut mit Wartezeiten einhergeht. Die Arbeit mit den Simulationen soll so interaktiv wie möglich gestaltet sein, was durch diesen Aspekt erschwert wird, da der User weniger dazu motiviert wird Eingaben auszuprobieren, wenn er weiß dass Änderungen mit Ladezeiten einhergehen. Um die Interaktivität zu wahren, aber gleichzeitig zu ermöglichen, die Animationen flüssig zu betrachten, wird eine Option implementiert, welche das Cachen aktiviert oder deaktiviert. Mit deaktiviertem Caching werden Simulationen nahezu ohne Ladezeit direkt nach Änderung der Parameter angezeigt. Sie können dann als statisches Modell betrachtet werden. In der Timeline können weiterhin auch andere Frames angesehen werden. Um die Simulation flüssig animiert zu betrachten, kann die Cache-Funktion aktiviert werden und nach 15-30 Sekunden Wartezeit ist dieser fertig geschrieben.

Ein weiterer kritischer Punkt besteht in der Darstellung der Informationen außerhalb der Partikel. Um die zu sehenden Objekte in einen Kontext einordnen zu können, ist es notwendig, Zusatzinformationen einzublenden. Der Nutzer sollte jedoch nicht mit zu vielen Informationen auf einmal konfrontiert werden. So wird eine Sticky-Note verwendet,

um nur die relevantesten Informationen direkt im Network-View-Fenster darzustellen. Die Datei mit dem zum stellaren Objekt zugehörigen Text wird bei der Aktivierung der Simulation automatisch ausgewählt.

Komplikationen können auch im Rahmen der zu bearbeitenden Parameter auftreten. Diese müssen in eine für den Nutzer vorstellbare Größenordnung gebracht werden und gleichzeitig intuitiv zu regulieren sein. Die Masse wird deswegen in Sonnenmassen angegeben, um einen Bezug zu dem Nutzer bekannten Sternen herzustellen. Problematisch ist dabei, dass die Sternenmasse sowohl ziemlich weit unter der Sonnenmasse, als auch sehr weit darüber liegen kann (0.01 - 120 Sonnenmassen). Um dieses Problem zu adressieren, wird die Maßeinstellung über einen logarithmischen Float Slider vorgenommen. Durch die Manipulation mit einem Slider wird einerseits die Interaktivität erhöht, da Werte nicht über die Tastatur eingegeben werden müssen, andererseits stellt dieser bis zur Mitte nur die Werte von null bis eins dar und alle anderen Werte auf der zweiten Hälfte. Dadurch wird der User motiviert, trotz der weiten Spanne an Werten auch kleinere Eingaben auszuprobieren.

Die Regulierung des Zeitwertes wird aus denselben Gründen ebenfalls über einen Slider gelöst. Jedoch ist diese Problematik noch etwas komplexer. Die Zeitliche Entwicklung und die Dauer der Phasen ändert sich rapide mit Zu- und Abnahme der Masse. Ein Stern von 100 Sonnenmassen verbringt keine 1000 Jahre in der Protostern-Phase, während diese für einen Stern mit 0.01 Sonnenmassen Millionen von Jahren anhalten wird. Derart signifikante Unterschiede lassen sich nicht mit einer einheitlichen Zeiteinheit abbilden. Zusätzlich wird daher zum Slider noch ein Drop-down-Menü eingeführt, welches verschiedene Einheiten anbietet (Jahre, Jahrhundert, Jahrtausend, Jahrmillionen), damit der User sowohl langsam als auch schnell entwickelnde Sterne betrachten kann. Ein weiteres Problem besteht in diesem Zusammenhang mit der Dauer der unterschiedlichen Phasen. Diese variiert ebenfalls stark mit der Hauptreihenphase, welche bis zu 3000 Milliarden Jahre andauern kann und beispielsweise einer Supernova, welche binnen weniger Stunden abläuft. Für den Nutzer ist es fast unmöglich, die korrekte Zeiteinstellung zu finden, zu der die Supernova stattfindet. Deswegen werden zwei Knöpfe implementiert, die es ermöglichen, eine Phase nach vorn sowie nach hinten zu springen. So ist es nicht nötig, den korrekten Zeitwert jeder Phase durch das Verstellen des Sliders herauszufinden. Der gesamte stellare Verlauf kann einfach durch die beiden Knöpfe abgerufen und jede Phase simuliert werden.



Ein weiterer Faktor, den es zu betrachten gilt, ist die durch den Umfang der Simulation bedingte wissenschaftliche Ungenauigkeit. Die dargestellten Prozesse des stellaren Lebenszyklus sind zu komplex, um bei der Nachbildung alle Faktoren einzubeziehen. In der Realität wirken Stern- Entstehung und Verfall von unterschiedlichen Sternen wechselseitig aufeinander ein. Die durch eine Supernova freigesetzten interstellaren Winde sorgen für den Kollaps interstellarer Wolken, welche nun neue Sterne bilden. Die Zusammensetzung der Staub- und Gasnebel unterscheidet sich an jeder Stelle des Universums, wodurch die thematisierten Prozesse überall eine unterschiedliche Zeitskalierung aufweisen. Damit das didaktische Ziel, also die Visualisierung des Gesamtprozesses möglich ist, sind Simplifizierungen von Nöten. Es muss dabei in Kauf genommen werden, dass nicht alle erhaltenen Ergebnisse in Gänze wissenschaftlich korrekt sind. Es wird im theoretischen Umfeld der Simulation davon ausgegangen, dass die gezeigte Materie eine ähnliche chemische Zusammensetzung wie die Sonne besitzt. Der Lebenszyklus der Sonne kann nun betrachtet werden, da dieser größtenteils wissenschaftlich erforscht und bewiesen ist. Die erhaltenen Werte können in Formeln wie in Kapitel 2 beschrieben zusammengefasst und vereinfacht werden, bis lediglich eine Abhängigkeit vom Parameter der Masse besteht, welcher vom Benutzer manipuliert wird.

<b>Konflikt/Problem</b>	<b>Lösungsansatz</b>
Konflikt zwischen Realismus und Verständlichkeit	schematische Darstellung als Partikelwolke
Konflikt zwischen Performance und Eindeutigkeit	initiale Ladezeit von 5-10 Sekunden als Kompromiss
Konflikt zwischen Dynamik und Interaktivität	Caching deaktivierbar und wenn aktiv nicht über 30 Sekunden
Einordnung der Simulationen in einen Kontext	Zusatzinformationen in Textform in einer Sticky Note
Kontrolle der Massewerte	Logarithmischer Slider
Kontrolle der Zeitwerte	Dropdown Menü für Zeiteinheit
Auffindbarkeit kurzer Phasen	Knöpfe zum springen zwischen Stadien
Konflikt zwischen Simplizität und wissenschaftlicher Genauigkeit	Annahme Sonnenähnliches Umfeld, Vereinfachung der Parameter auf Masse

*Abbildung 18: "Schematische Darstellung der erwarteten Probleme mit Lösungsansatz"*

## 5.4 Erstellung der Simulationen

### 5.4.1 Interstellare Wolke



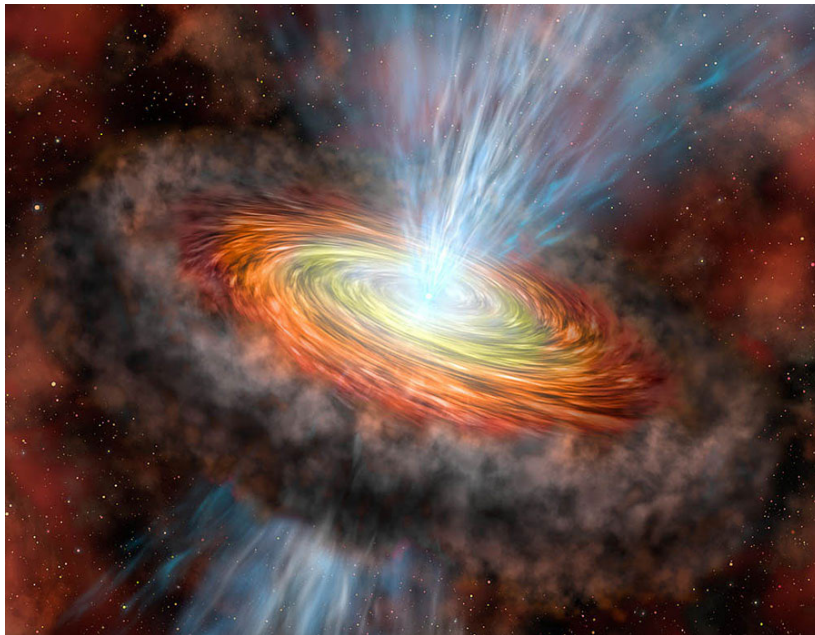
*Abbildung 19: “Rosettennebel, NGC 2237” (Jones 2019)*

Das Modell der interstellaren Wolke orientiert sich in Farbgebung und Form in Grundzügen am auf Abb. zu sehenden Rosettennebel, NGC 2237. Im Wesentlichen galt es als Ziel, eine Partikelwolke mit chaotischen, aber nicht zu schnellen Partikel Bewegungen abzubilden. Die Geschwindigkeit der fluktuierenden Partikelströme sollte sich mit der Zeit erhöhen, um den sich anbahnenden gravitativen Kollaps zu verdeutlichen.

Als Grundkörper der Wolke wird eine Kugel verwendet, die durch eine Attribute-Noise Node zufällig deformiert wird. In den entstandenen Körper werden nun mit Hilfe einer points-from-volume Node Punkte verteilt und ihre Position innerhalb des Körpers wird randomisiert, um eine chaotische Verteilung zu erreichen. Abbildung 20 zeigt die entstandene Partikelwolke. Um diese nun in Bewegung zu versetzen, wurde ein Popnet verwendet. Hier wird mit Hilfe einer Pop-Wind Node ein Windstrom simuliert, welcher dafür sorgt, dass sich die dargestellten Partikel in Strömen zufällig bewegen. Die Größe der Ströme wird hierbei durch den Swirl-Scale Parameter bestimmt. Dieser ermöglicht es, eine Art Zusammenhalt der Partikel zu erzeugen, wodurch der Effekt der Gravitation visualisiert werden kann. Um die Bewegung noch weiter zu randomisieren, wurde an dieser Stelle eine weitere Attribute-Noise Node hinzugefügt. In dieser Node ist der Offset Parameter an die Sternmasse gekoppelt. Dadurch wird die Position der verwendeten Noise für Sterne mit unterschiedlicher Masse im 3D Koordinatensystem verschoben. Daraus

folgt, dass jede neu generierte Wolke eine andere Form hat, wodurch die Simulation interessant bleibt. Die angestrebte Erhöhung der Partikelgeschwindigkeit wurde mit Hilfe des Pulse-Duration Parameters umgesetzt. Dieser wird durch den eingegebenen Zeitwert geteilt, was dazu führt, dass er mit erhöhter Zeit geringere Werte annimmt. Auf diese Weise wird mit höherer Zeit eine kürzere Pulsierungsdauer der Noise erzeugt, was den angestrebten Effekt des gravitativen Kollaps hervorhebt. Anschließend werden die Partikel mit Hilfe von drei Nodes eingefärbt. Durch eine Wrangle Node wird mit Hilfe von VEX Code ein Attribut erschaffen, welches die Länge des Positionsvektors, also den Abstand zum Koordinatenursprung der Partikel speichert. Dieses Attribut wird zunächst randomisiert, um eine natürliche Farbverteilung zu erreichen. Mit einer Color Node werden die Punkte nun mit einer Ramp anhand des gespeicherten Attributes wie auf Abbildung 21 zu sehen eingefärbt. Die inneren Teile sind farblos und hell und die äußeren Parts dunkler und Rot-Violett wie beim Rosettennebel. (*Abb. 19, Abb. 33*)

### 5.4.2 Protostern



*Abbildung 22: "Protostern (W33A) mit Jets und Akkretionsscheibe" (Cook, 2010)*

Die Simulation des Protosternes basiert auf Cook's (2010) Darstellung des Sternes W33A (Abb. 22). Sie besteht aus vier Teilbereichen, die den Aufbau des Protosterns repräsentieren.

#### **Äußere Staubwolke**

Als Grundkörper wurde hier ein Torus verwendet. In diesen wurden mit der bereits beschriebenen Vorgehensweise zufällig Partikel verteilt. Über eine Wrangle Node wurde diesen dann eine randomisierte Velocity zugewiesen. Die Rotation der Wolke wird durch ein Popnet erzeugt. Die Punkte werden hier durch eine Pop-Axis-Force Node innerhalb eines gegebenen Bereichs um die y-Achse rotiert. Über die Parameter Orbit-Speed und Suction-Speed wird die Geschwindigkeit der Bewegung reguliert. Die Node berechnet die Umlaufbahn der Partikel und multipliziert den Velocity-Vector mit dem Orbit Speed, damit die Bewegung um das Zentrum erzeugt wird. Ist die Geschwindigkeit der Partikel jedoch zu hoch, können sie den Orbit verlassen und bewegen sich nach außen vom Zentrum weg. Um dem entgegenzuwirken, kann der "Suction Speed" erhöht werden, da dieser einen Sog zum Zentrum hin erzeugt. Da die Gravitation des Protosternes im Bereich der äußeren Wolke verhältnismäßig gering ist, wurden beide Parameter so konfiguriert, dass die Partikel zwar rotieren, sich jedoch nicht zum Zentrum bewegen. Ein Teil der Partikel wird

nach außen abgegeben. Die Farbe der Wolke wird anhand des Age Attributes bestimmt, welches auf verschiedene Graustufen einer Rampe projiziert wird.

### **Akkretionsscheibe**

Für die Akkretionsscheibe wurde eine kleinere skalierte Version des Grundkörpers der äußeren Wolke verwendet. Die Velocity der Punkte wurde über VEX Code zum Koordinatenursprung hin ausgerichtet. Die Partikel wurden daraufhin ebenfalls durch eine Pop-Axis-Force Node in Rotation versetzt. Um den für Protosterne charakteristischen, gravitativen Strudel zu erzeugen, wurde hier Suction-Speed im Verhältnis zu Orbit-Speed weiter erhöht. Über eine Popkill Node, welche den Kern als Grundkörper verwendet, werden alle Partikel, welche das Zentrum erreicht haben, gelöscht. So wird die Partikelanzahl konstant gehalten, da in den Außenbereichen stetig neue erschaffen werden. Um den Eindruck von gravitativen Strömen weiter zu verstärken, wurden über eine Trail Node Klone der Partikel erschaffen, welche versetzt deren Bewegung folgen und so einen Strom erzeugen. Die Farbgebung der Scheibe erfolgt anhand der Distanz zum Zentrum. Orientiert an (Abb. 22) ist das Zentrum hierbei hell und gelb gefärbt und die äußeren Bereiche rot und dunkler, um den Temperaturabfall zu repräsentieren.

### **Kern**

Der Kern verwendet einen sphärischen Grundkörper. Die verteilten Punkte werden hier nicht über ein Popnet, sondern lediglich über eine Attribute-Noise Node animiert, da diese weniger Rechenleistung verwendet und der Kern im Gesamtbild der Simulation kaum sichtbar ist. Die Farbgebung erfolgt über die Distanz zum Koordinatenursprung.

### **Jets**

Die Jets verwenden den Kern als Grundkörper. In einem Popnet werden sie über eine Pop-Axis-Force Node in Bewegung gesetzt. Als Form dient hierbei, im Gegensatz zur Staubwolke und Akkretionsscheibe ein Torus. Der Suction-Speed erhält einen negativen Wert, was dazu führt, dass Partikel nach oben und unten fontänenartig abgegeben werden, da sie sich mit hoher Geschwindigkeit vom Zentrum weg bewegen. (Abb. 38)

### 5.4.3 Teilbereiche der Sphärischen Objekte

Im Folgenden wird die Erstellung einzelner Elemente des Sternaufbaus dokumentiert. Die entsprechenden Sterne wurden aus diesen Teilbereichen gemäß des in Kapitel 2 geschilderten Aufbaus zusammengesetzt.

Der Aufbau des braunen Zwerges und des Hauptreihensternes ist, wie in Kapitel 2 geschildert, grundlegend ähnlich. Der wichtigste Unterschied besteht in der Verteilung der Strahlungs- und Konvektionszonen. Braune Zwerge haben, wie im zugehörigen Kapitel beschrieben, keine Strahlungszone, sondern sind vom Kern bis zur Hülle konvektiv. Weiterhin unterscheiden sich Hauptreihensterne wie in Abbildung 7 dargestellt, mit unterschiedlicher Masse ebenfalls in ihrem Aufbau. Die jeweiligen Zonen wurden daher getrennt erstellt und daraufhin zu den zugehörigen Modellen zusammengesetzt. Die Umsetzung der individuellen Besonderheiten der stellaren Objekte wird in den zugehörigen Kapiteln dokumentiert.

#### **Kern**

Eine Kugel diente als Grundkörper des Kernes. In diese wurden Partikel verteilt, wobei die Dichteverhältnisse des Sternes auch in der Partikelanzahl repräsentiert wurden. Der Kern hatte folglich eine höhere Dichte an Partikeln als die äußeren Schichten. Weiterhin sollten die Temperaturverhältnisse einerseits durch Färbung und andererseits durch Bewegungsgeschwindigkeit der Partikel visualisiert werden. So sind die Partikel des Kernes bei den dargestellten Simulationen meist hellblau bis weiß. Weiterhin bewegen sie sich immer schneller als Partikel in den weiter außen liegenden Schichten. Um den Kern möglichst chaotisch wirken zu lassen bzw. den Prozess der Fusion zu visualisieren, wurde das Velocity Attribut der zugehörigen Partikel mit einer Attribute-Randomize Node randomisiert. In einem Popnet wurde der Birth-Rate Parameter auf einen verhältnismäßig hohen Wert gesetzt, was dazu führt, dass konstant viele neue Partikel erscheinen. Eine Popkill Node entfernt alle Partikel, die den Kern verlassen, um ihn visuell klar von den anderen Bereichen zu trennen.

### **Strahlungszone**

Die Strahlungszone geht ebenfalls von einer Kugel als Grundkörper aus. Die Punkte werden hier jedoch nicht im gesamten Volumen, sondern lediglich auf der Oberfläche verteilt. Dies ist notwendig, da sich Partikel in der Strahlungszone nur von innen nach außen bewegen sollen. Als Velocity Vector wird der Positionsvektor, also der Abstand zum Koordinatenursprung, verwendet. Dadurch bewegen sich alle Punkte geradlinig vom Kern weg. Für eine natürliche Bewegung wird der Velocity Vektor daraufhin mit Hilfe der Rand() Funktion mit einer für jeden Partikel unterschiedlichen Zahl multipliziert. Ein Popnet sorgt dafür, dass die Punkte der gegebenen Bewegung folgen und löscht über eine Popkill Node alle Punkte, welche das Ende der Strahlungszone erreichen. Die Strahlungszone erhält eine gelbliche Färbung, um die hohe Temperatur zu visualisieren und sie vom Kern zu trennen.

### **Konvektionszone**

In der Konvektionszone gibt es im Gegensatz zur Strahlungszone Bewegung in beide Richtungen. So wurde eine Kugel wie im vorangegangenen Abschnitt verwendet, um Punkte nach außen zu bewegen und eine weitere Kugel, um Punkte nach innen zu bewegen. Die nach innen gerichteten Ströme haben dafür eine negative Velocity und bewegen sich folglich zum Kern hin. Mit Hilfe von Pop-Kill Nodes in einem Popnet wird die Konvektionszone auf der Innen- und Außenseite abgegrenzt. Um die Konvektionsströme besser als solche hervorzuheben, wird eine Trail Node verwendet, die Partikel Bewegungen verdeutlicht. Die Partikel wurden farblich etwas dunkler als die Strahlungszone eingefärbt, um diese Schichten klar zu differenzieren.

### **Außenhülle (Chromosphäre, Photosphäre)**

Die Hüllen sollen den Stern nach außen abgrenzen und ihm seine Farbe verleihen. Zwei Kugeln wurden hierfür als Grundkörper verwendet. Auf ihnen wurden zufällig Partikel verteilt und mit Hilfe einer Attribute-Noise Node animiert. Um die niedrigen Temperaturen in den Hüllen zu verdeutlichen und die vorherrschende Stabilität des Sternes zu visualisieren, sind die Animationen eher subtil. Farblich wurden beide Schichten primär in der Helligkeit voneinander abgesetzt, um den Stern optisch interessant zu halten.

#### 5.4.4 Hauptreihensterne

Die Hauptreihensterne wurden entsprechend der in Kapitel 2 beschriebenen Struktur zusammengesetzt. Die Verteilung der Strahlungs- und Konvektionszonen wurde wie in Abbildung 7 für unterschiedliche Massen differenziert. Um das Modell des Hauptreihensternes noch einprägsamer zu gestalten, wurden hier speziell noch Protuberanzen und Spektralklassen implementiert. Die Spektralklassen differenzieren Hauptreihensterne mit unterschiedlichen Massen voneinander. In der Simulation wurden diese über Color Nodes umgesetzt, welche mit einer Switch Node verbunden sind. Alle Spektralklassen haben die zugehörige Farbe entsprechend der Harvard Klassifizierung (Abb. 1) erhalten. Die Auswahl der korrekten Spektralklasse erfolgt über die Python Logik des Programms. Die Protuberanzen sollen den Stern klar als Hauptreihenstern erkenntlich machen und die hohen vorherrschenden Temperaturen unterstreichen. Sie verwenden die Außenhülle als Grundlage. Für die Animation wird eine Attribute-Noise Node verwendet, welche über die Add Operation lediglich nach außen wirkt. Die dahinter befindliche Trail Node sorgt dafür, dass die sichtbaren Strukturen als Ströme erkenntlich sind. Über eine Delete Node werden Partikel gelöscht, sobald sie wieder die Sonnenoberfläche erreichen. So werden chaotische Ströme und Bögen erzeugt, die an zufälligen Stellen des Sternes aus der Hülle nach außen vordringen und darauffolgend wieder in den Stern zurückfallen. (Abb.32)

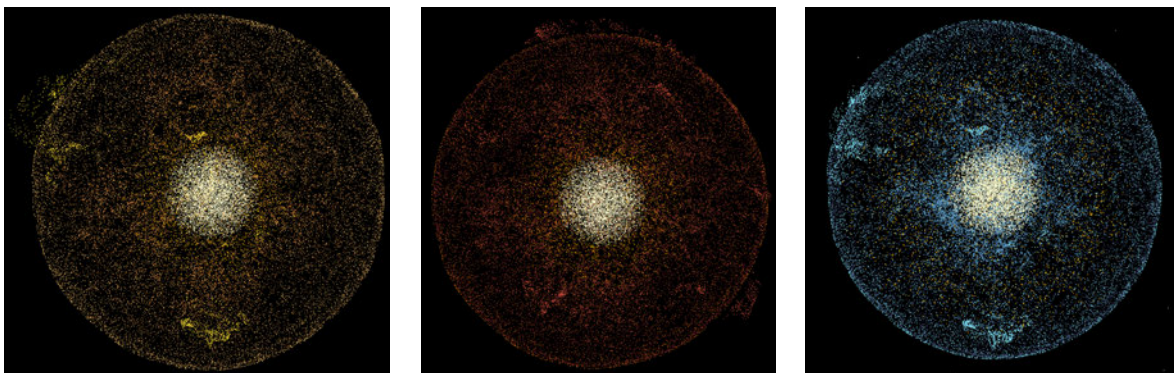
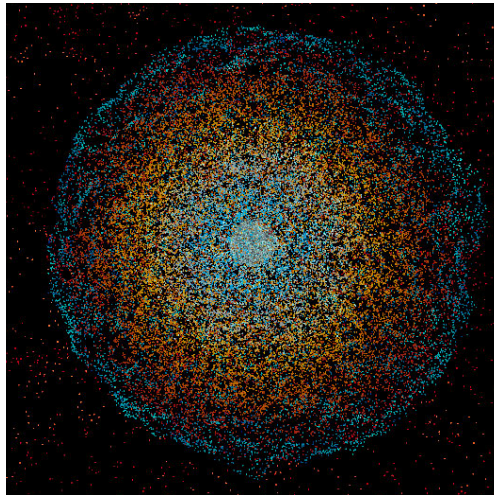


Abbildung 23: "Spektralklassen der Hauptreihensterne"



### 5.4.5 Roter Riese, Roter Überriese

In der Simulation der Riesensterne sollten einige Aspekte besonders hervorgehoben werden. Einerseits sollten die verschiedenen Elemente des Kernes erkennbar dargestellt werden und andererseits sollte die sich ausbreitende Außenhülle, welche Materie abgibt, sichtbar sein. Die unterschiedlichen Schichten des Kernes wurden daher, wie im Kapitel 2.6.2 beschrieben, mit Hilfe eine Farbgradienten differenziert.



*Abbildung 24: "Kern des Überriesen"*

Dieser wird über ein Attribut aufgebaut, welches die Länge des Positionsvektors, also den Abstand zum Koordinatenursprung der Punkte speichert. So ist es möglich, allen Punkten bis zu einem bestimmten Abstand vom Zentrum eine Farbe zuzuweisen, welche die Zugehörigkeit zu einem Element repräsentiert. (Abb. 8) Die Außenhülle der Riesensterne breitet sich aus und verliert dabei einen Teil ihrer Masse nach außen. Um diesen Zusammenhang darzustellen, wurde der Scale Parameter über eine Transform Node animiert. Zur Darstellung des Masseverlusts wurden die im vorangegangenen Kapitel dokumentierten Protuberanzen verwendet und auf die entsprechende Skalierung gebracht. Da die Hülle in diesem evolutionären Stadium bereits sehr weit vom Kern entfernt ist, stellt die gleichzeitige Darstellung beider Teilbereiche eine Herausforderung dar. Um dem Nutzer dennoch zu ermöglichen, Kern und Hülle zu sehen, wurden die Kameras der Riesensterne animiert. Zunächst sind diese einige Sekunden auf den Kern gerichtet, so dass die Schichten hier eindeutig sichtbar sind. Darauffolgend bewegt sich die Kamera nach außen und betrachtet den gesamten Stern. Der Kern ist nun nur noch sehr klein im Zentrum erkennbar, aber die Außenhülle und ihre Ausbreitung sind in vollem Umfang zu sehen. (Abb. 36, 37)

### 5.4.6 Weißer Zwerg

Da im weißen Zwerg keine Fusionsreaktionen mehr ablaufen, wurde dieser nahezu statisch dargestellt. Um die stetige Abkühlung auf die Temperatur des interstellaren Mediums zu repräsentieren, wurde der Farbgradient des Sternes mithilfe des Zeitwerts der Simulation so animiert, dass alle Farben mit vergehender Zeit immer dunkler werden. So wird die Simulation des weißen Zwerges dunkler, umso höher der vom Nutzer eingegebene Zeitwert ist.

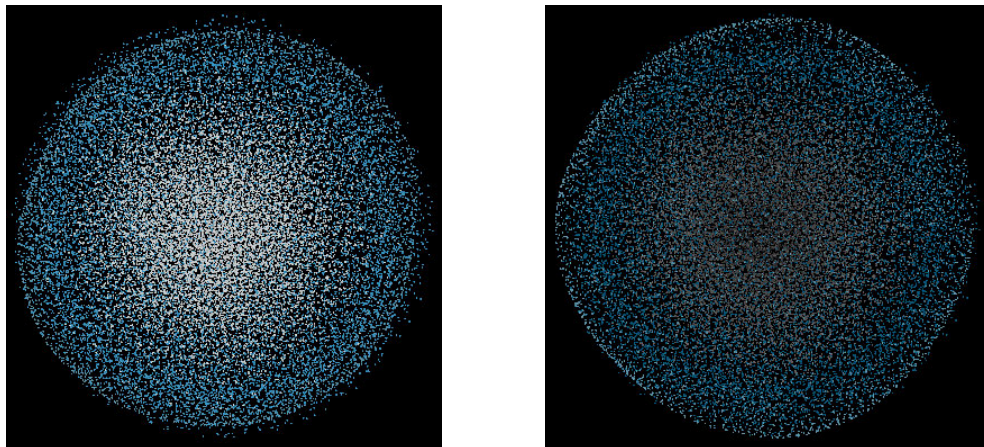


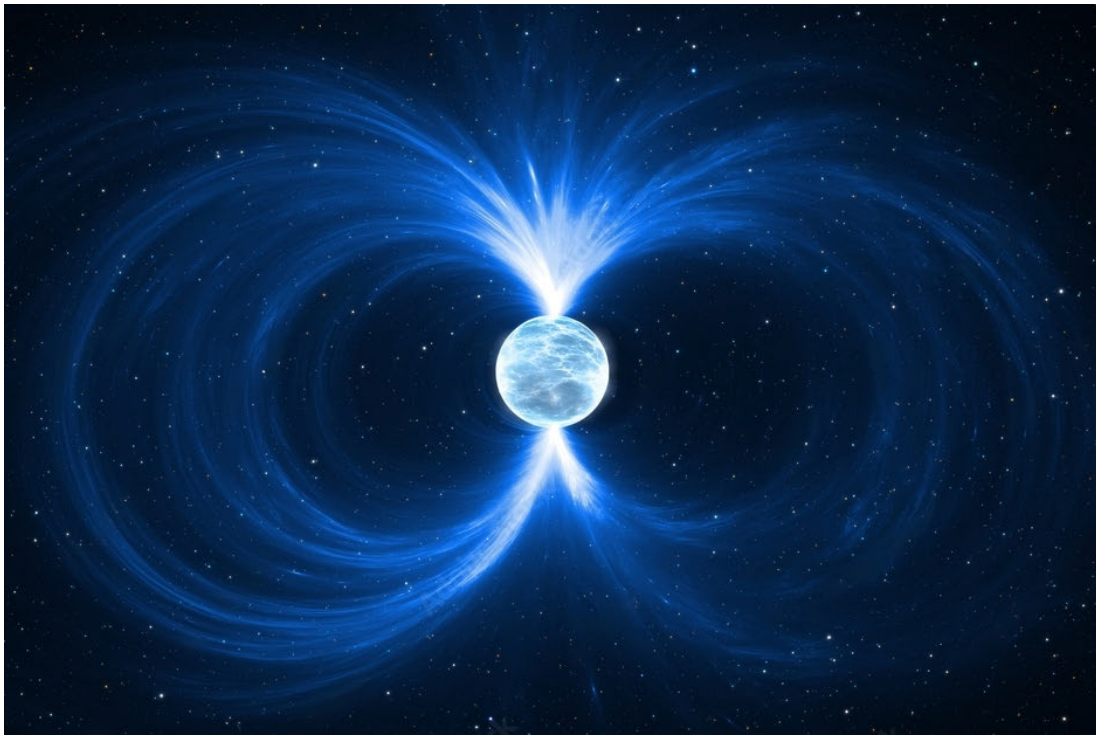
Abbildung 25: “Farbverlust des weißen Zwerges”

### 5.4.7 Schwarzes Loch

Das schwarze Loch wurde gemäß dem Aufbau aus Kapitel 2.6.4 visualisiert. Problematisch war hierbei die Verzerrung des Lichts im Umkreis des schwarzen Lochs, da diese sich mit der Betrachtungsperspektive verändert. Damit es jedoch möglich ist, das schwarze Loch von allen Seiten zu betrachten, wurde die Form der Akkretionsscheibe vereinfacht und in Strudelform visualisiert. Der Ereignishorizont wurde durch eine Kugel repräsentiert, welche mit Hilfe einer Color Node schwarz gefärbt ist. Die Darstellung der Akkretionsscheibe erfolgt über die Vorgehensweise für die Visualisierung der Akkretionsscheibe des Protosterns. Um das Ansteigen der Temperatur zum Zentrum hin zu repräsentieren, wurde auch hier ein Farbgradient anhand des Abstandes zum Zentrum generiert. Punkte, welche sich näher am Ereignishorizont befinden, sind dabei deutlich heller gefärbt als weiter außen befindliche Punkte. Überschreitet ein Punkt den Ereignishorizont, wird dieser durch eine Popkill Node gelöscht, so bleibt die Grenze des Horizonts klar erkennbar. (Abb. 34)

### 5.4.8 Neutronenstern

Neben den in Kapitel 2.6.3 beschriebenen Schichten, welche durch Color Nodes differenziert wurden, sollte die Simulation die für Neutronensterne typischen Pulsare zeigen. Diese entstehen durch das starke Magnetfeld des Sterns und verlaufen Kreisbahnen zwischen den beiden magnetischen Polen.



*Abbildung 26: "Neutronenstern mit Pulsaren" (Jurik, 2021)*

Abbildung 26 zeigt die dafür verwendete Referenz. Deutlich zu erkennen ist dabei der Helligkeitsverlauf der Pulsare mit unterschiedlicher Distanz zum Stern selbst. Als Grundlage für die Form des Magnetfeldes wurde ein Torus verwendet, welcher mit seinem Zentrum durch die Kugel des Kernes hindurch verläuft. Die vertikalen Linien des Körpers sollten als Bahnen für die Pulsare dienen. Um diese möglichst natürlich anzuordnen, wurden ihre Positionen durch eine Attribute-Noise Node randomisiert. Die Bewegung wird nun in einem Popnet mithilfe einer Popaxisforce Node umgesetzt. Der Shape Parameter der Node wurde auf Torus umgestellt und die Radius- und Height Parameter wurden entsprechend der Größe des Grundkörpers angepasst. Die Färbung der Partikel erfolgte anhand des Abstandes vom Kern, wobei entferntere Partikel dunkler eingefärbt wurden als Nähere. So wird ebenfalls die Leuchtkraft des Sterns visualisiert. (Abb. 35)

Ein weiterer Aspekt, der in der Simulation ersichtlich sein sollte, ist die Rotation des Sterns. Um den Unterschied zwischen Rotationsachse und der Achse des Magnetfeldes zu verdeutlichen, wurde der Stern leicht geneigt. So ist zu erkennen, dass Neutronensterne sich nicht zwangsläufig entlang der magnetischen Pole rotieren. Weiterhin ist die Rotationsgeschwindigkeit unmittelbar nach der Entstehung am höchsten und verlangsamt sich mit der Zeit immer weiter. Dieser Zusammenhang wurde durch die Animation über den Zeit Parameter der HDA Node umgesetzt. Die Rotation erfolgt dabei auf der Y-Achse anhand der \$F-Variable, welche den derzeitigen Frame speichert. Der Zeitwert wird mit 0.001 multipliziert und dient dann als Divisor für die vorherige Rotation anhand des Frames. So rotiert der Neutronestern bei höheren Zeitwerten langsamer.

### **5.4.9 Helium Blitz**

Die Endphase des Roten Riesen sollte anhand mehrerer Aspekte dargestellt werden. Einerseits sollte die Kontraktion des Kernes zu erkennen sein, wenn die Heliumfusion beginnt. Andererseits sollte erkennbar sein, dass die dabei freigesetzte Energie die äußeren Hüllen in das Weltall abgibt und einen planetaren Nebel entstehen lässt. Da sich beide Prozesse stark in ihrer Skalierung unterscheiden, wurde ähnlich zur Simulation der Riesensterne eine Kamerafahrt verwendet, die zunächst einige Sekunden den Kern zeigt und sich daraufhin nach außen bewegt, sodass der Zerfall der Außenhüllen zu erkennen ist. Als Grundlage für Kern und Außenhülle dienten die Partikel des roten Riesen, welche durch Objekt-Merge Nodes importiert wurden. Der Kern wurde über eine Attribute Noise Node mit niedrigem Pulse-Length Parameter animiert und sehr hell eingefärbt, um den energiereichen Helium Blitz zu verdeutlichen.

Die Animation der Hülle erfolgt über den Scale Parameter einer Transform Node. Mit einer Trail Node wurde die Velocity anhand dieser Animation berechnet. Um diese nun natürlich zu verlangsamen, damit sich Partikel nicht unendlich weit nach außen bewegen, wurde ein Popnet mit einer Pop-Wrangle Node verwendet. Hier wird die Länge des Velocity-Vektors, also die Geschwindigkeit der Partikel, mit einer If-Abfrage überprüft. Unterschreitet diese einen Wert von 0.1, wird sie auf Null gesetzt, was den Partikel zum Anhalten bringt. Eine Attribute-Noise Node randomisiert die Bewegung der Partikel und sorgt dafür, dass am Ende die Struktur eines planetaren Nebels entsteht.

#### **5.4.10 Supernova**

Für die Supernova wurde ähnlich wie beim Helium Blitz eine Kamerafahrt verwendet, um zunächst den Zusammenfall des Kernes und darauffolgend die Zerstörung der Außenhülle durch die entstandene Druckwelle zu zeigen. Die Animation des Kernes erfolgt innerhalb eines Popnets, durch eine Pop-Attract Node. Zunächst wird ein Burst-Frame definiert, der den Zeitpunkt des Rückstoßes der Partikel nach dem Kollaps repräsentiert. Der Force-Scale Parameter wird innerhalb einer If Abfrage animiert. Diese vergleicht die Nummer des derzeitigen Frames mit dem Burst-Frame und setzt die Force-Scale auf einen negativen Wert solange der Burstframe nicht erreicht wurde. So bewegen sich die Partikel des Kernes nach innen zum Koordinatenursprung. Ist der Burst-Frame erreicht, wird der Parameter invertiert und sorgt dafür, dass die Partikel sich nun explosionsartig nach außen zur Hülle bewegen. Diese wird nun scheibenförmig durchbrochen und nach außen abgegeben. Für die Animation der Hülle wurde der Velocity-Wert zunächst dem Positionsvektor gleichgesetzt und danach durch den Betrag der Y-Koordinate der Partikel geteilt. So bewegen sich Punkte mit größeren Y-Werten schneller nach außen, was die scheibenförmige Zerstörung der Außenhülle erzeugt, wie sie bei Supernovae bekannt ist. (Abb. 39, 40)

#### **5.4.11 Zusatzinformationen**

Zusätzlich zur gezeigten dreidimensionalen Partikelsimulation sollten dem Nutzer einige Informationen in Textform angezeigt werden. Einerseits einige relevante Informationen über die Entstehung und den Aufbau des Sternes, wie sie in Kapitel 2 beschrieben wurden und andererseits einige spezifische auf den Benutzereingaben basierende Parameter wie aktive Lebenszeit und Spektralklasse. Die beschriebenen Informationen werden nach Betätigung des “Start” Knopfes im Network Editor in Form einer Text-Box eingeblendet. Anhand der Namen der Simulationen wird eine entsprechend benannte Textdatei mit den allgemeinen Informationen zum stellaren Objekt geladen und der Inhalt wird in die Textbox übernommen. Über weitere Methoden werden die spezifischen Informationen zunächst berechnet, gerundet und danach der Text-Box hinzugefügt.



## 5.5 Logik und Verarbeitung von Usereingaben

### 5.5.1 Struktur

Die HDA Node des Sternen Simulators wird auf SOP-Ebene erstellt. Die Benutzung erfolgt ebenfalls auf dieser Ebene, da hier alle Parameter zugänglich sind, im Viewport die korrekte Simulation angezeigt wird und Informationstexte eingeblendet werden. So kann die Simulation ohne Vorkenntnisse über die Funktion der Ebenen der Houdini Umgebung erfolgen. Abbildung 27 zeigt die Benutzeroberfläche der HDA Node, welche im Parameterfenster angezeigt wird, nachdem die Node erstellt wurde. Um diese Art der Nutzung zu ermöglichen, ist es vonnöten, dass sich alle zugehörigen Daten innerhalb des HDA befinden.

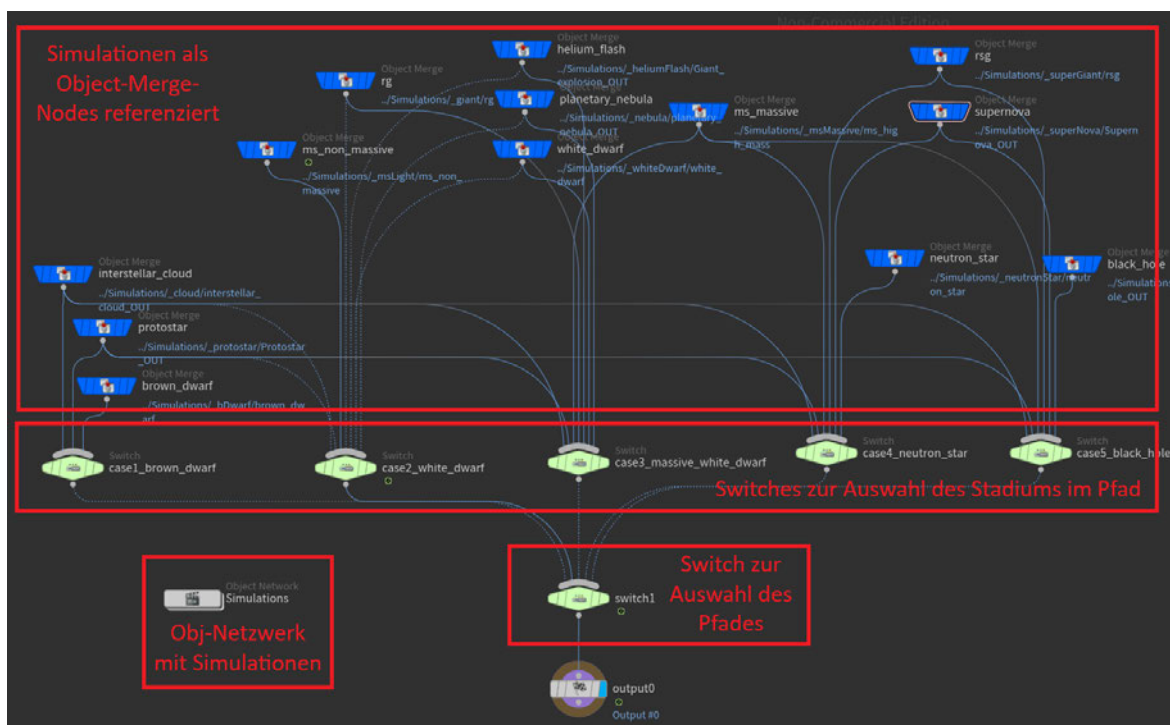


Abbildung 28: "innere Struktur des HDA"

Abbildung 28 zeigt die innere Struktur der Node. Enthalten ist ein Obj-Netzwerk, in dem sich die Simulationen und die zugehörigen Kameras befinden. Diese werden durch die blauen Objekt-Merge Nodes referenziert und wenn eine der Nodes mit der Output Node verbunden wird, ist die jeweilige Simulation im Viewport zu sehen. Um zu steuern welche Simulation zu welchem Zeitpunkt angezeigt wird, sind die importierten Simulationen durch ein Netzwerk aus grünen Switch Nodes unterteilt. Diese kontrollieren über einen Integer Wert, welcher Input angesteuert wird und eignen sich somit für die Manipulation durch Python Code. Der untere Switch steuert dabei, welcher der fünf auf Abbildung 29

dargestellten evolutionären Pfade angezeigt wird. Die oberen Switches repräsentieren jeweils einen der fünf Pfade und kontrollieren, welches Stadium entlang des Pfades zu sehen ist.

### 5.5.2 Logik

Das HDA besitzt ein hinterlegtes Python-Skript, welches die in Kapitel 5.2 beschriebenen Funktionalitäten umsetzt. Zentral ist hierbei die Auswahl der richtigen Simulation anhand gegebener Übergabeparameter. Anhand des Diagramms auf Abbildung 3 lassen sich grundlegend fünf mögliche evolutionäre Wege einteilen, die abhängig von der Masse durchlaufen werden können.

Fall	Masse	Stadien
1	$M < 0,075M_{\odot}$	Wolke, Protostern, Brauner Zwerg
2	$0,075M_{\odot} \leq M < 1,5M_{\odot}$	Wolke, Protostern, HR-Stern(klein), Roter Riese, Nebula, Weißer Zwerg
3	$1,5M_{\odot} \leq M < 10M_{\odot}$	Wolke, Protostern, HR-Stern(groß), Roter Riese, Nebula, Weißer Zwerg
4	$10M_{\odot} \leq M < 25M_{\odot}$	Wolke, Protostern, HR-Stern(groß), Roter Überriese, Supernova, Neutronenstern
5	$25M_{\odot} \leq M < 120M_{\odot}$	Wolke, Protostern, HR-Stern(groß), Roter Überriese, Supernova, Schwarzes Loch

*Abbildung 29: “Die fünf möglichen evolutionären Pfade”*

Zunächst werden notwendige Variablen initialisiert. Dazu zählen unter anderem die fünf auf Abbildung 29 zu sehenden Pfade als String Arrays. Die enthaltenen Strings repräsentieren die Namen der Simulationen in der korrekten Reihenfolge (Abb. 30). Die globale “case” Variable speichert den Pfad und die “index” Variable die korrekte Position der derzeitigen Phase innerhalb des Arrays als Integer Wert. Anhand dieser Variablen können alle Methoden auf die richtige Simulation verweisen. Abbildung 31 zeigt die Struktur der im Skript enthaltenen Funktionen, mit den zugehörigen Übergabeparametern. Anhand der Pfeile lassen sich Funktionsaufrufe erkennen. Drückt der Nutzer den “Start” Knopf der Node, wird die Start() Funktion aufgerufen. Diese Funktion steuert die Logik des HDA und ruft alle weiteren notwendigen Funktionen auf. Zunächst werden Masse und Zeit aus den vom Nutzer eingegebenen Werten ausgelesen und in Variablen gespeichert.

Der Zeitwert wird dabei mit dem zugehörigen Faktor, welcher über das Dropdown-Menü für die Zeiteinheit festgelegt wurde, multipliziert. So kann dieser Wert im Code einheitlich verarbeitet werden, unabhängig von der durch den Nutzer ausgewählten Einheit. Nachfolgend werden die Switch Nodes zur Regulierung der angezeigten Simulation als “Parm” (ein Datentyp welcher in Houdini UI Elemente repräsentiert) eingelesen (*Abb. 28*). Nun wird anhand von If-Abfragen, welche die Masse mit den Grenzwerten der fünf Pfade (Tabelle) vergleichen geprüft, welcher Pfad für die Eingabe des Nutzers zutrifft. Das String Array des zutreffenden Pfades wird daraufhin der globalen Case Variable zugewiesen und der untere Switch (*Abb. 28*) wird auf die korrekte Einstellung aktualisiert. Nun berechnet die CalcState() Funktion anhand der eingegebenen Zeit das derzeitige Stadium innerhalb des Pfades, in welchem sich der Stern befindet. Dafür wird die CalcTime() Funktion aufgerufen, die anhand der Lebenszeitformel aus Kapitel 2.4.3 die Länge der einzelnen Phasen des stellaren Lebenszyklus berechnet. Ist das korrekte Stadium ermittelt, wird die zugehörige Position des String Werts im Case Array als Integer in die globale Index Variable gespeichert. Darauffolgend wird die SetStateDur() Funktion aufgerufen, welche den prozentualen Fortschritt innerhalb der derzeitigen Phase berechnet und den erhaltenen Wert in der “Perc” Variable speichert, welche durch die Simulationen referenziert wird. Dieser Parameter wird innerhalb der Simulationen verwendet, um einen zeitlichen Fortschritt anzudeuten, beispielsweise durch die schnellere Bewegung der interstellaren Wolke oder den Helligkeitsverlust des Weißen Zwerges mit vergehender Zeit. Befindet sich der Stern im Hauptreihenstadium, wird nun ebenfalls die Spektralklasse anhand der CalcSpec() Funktion berechnet und die zugehörige Farbe innerhalb der Simulation wird aktualisiert. Schließlich setzt die SetState() Funktion den oberen Switch (*Abb. 28*) auf das ermittelte Stadium und führt alle notwendigen Funktionen zum korrekten Laden und Anzeigen der Simulation aus. Dazu zählen das Setzen der Kamera, damit der User die angezeigte Simulation in Gänze sehen kann, das Caching falls es nicht durch die “Cache überspringen” Checkbox deaktiviert wurde und das Laden des Textes in die Infobox, damit alle Zusatzinformationen in schriftlicher Form angezeigt werden.



## 5.6 Ergebnis

Das Ergebnis des Projektes ist die Stern Simulator Node. Diese kann als HDA auf SOP-Ebene der Houdini Datei hinzugefügt werden. Ist das HDA erstellt, kann der Nutzer das Experimentieren beginnen und sich anhand der dreidimensionalen Partikelsimulationen selbstständig Wissen erarbeiten.

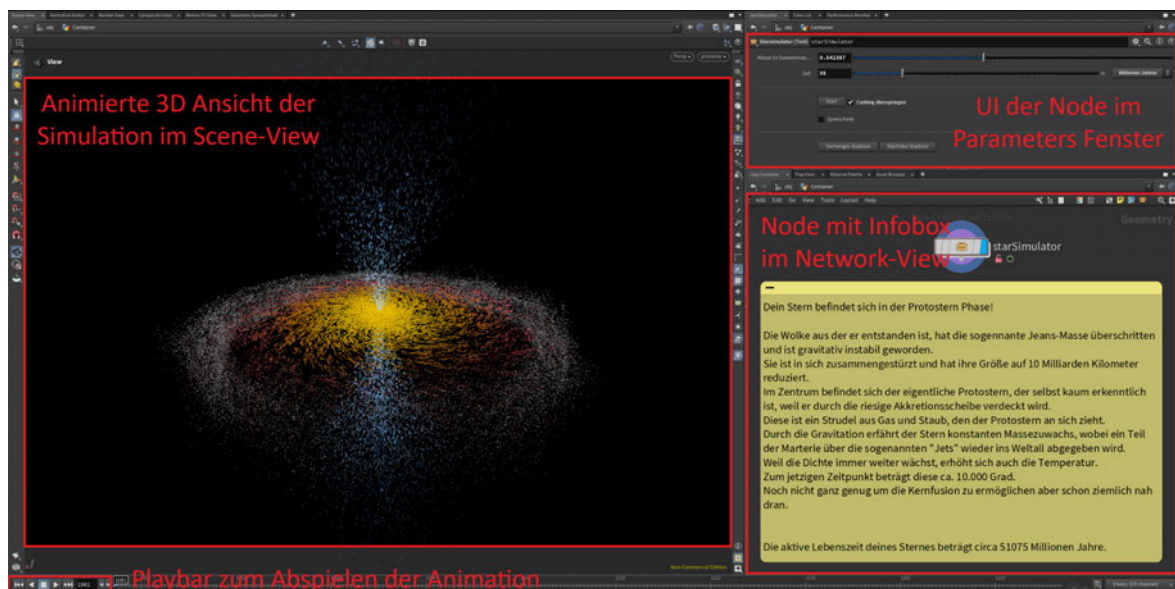


Abbildung 30: "Übersicht über die Teilbereiche der Simulation"

Auf Abbildung zu sehen ist eine Übersicht der Teilbereiche, mit denen der Nutzer interagieren kann. Zunächst können im Parameters Fenster (Abbildung Benutzeroberfläche des HDA) Masse und Zeit eingegeben werden. Der Wert der Masse wird hierbei in Sonnenmassen angegeben und die Einheit des Zeitwertes kann über das Dropdown Menü rechts neben dem Slider reguliert werden. So ist es möglich, sowohl kleinere als auch große Zeitabschnitte präzise auszuwählen. Die Checkbox "Caching überspringen" sorgt, wenn sie angewählt ist, für ein schnelleres Laden der Simulationen (je nach Hardware drei bis zehn Sekunden). So kann schnell experimentiert werden, wenn der Fokus nicht auf der Betrachtung der Animation, sondern auf dem erhaltenen Modell liegt. Um die Simulation flüssig animiert anzuschauen, kann die Checkbox deaktiviert werden. Dann wird vor dem Anzeigen ein Cache Vorgang ausgeführt, welcher zehn bis 15 Sekunden dauert und ein flüssiges Abspielen der Animation ermöglicht. Nach dem Eingeben der Werte kann der Nutzer den "Start" Knopf drücken, um die Berechnung zu starten. Nun wird im Scene-View-Fenster die errechnete Simulation angezeigt, welche durch die automatisch ausgewählte Kamera in Gänze zu sehen ist. Mithilfe der Playbar kann die zehn sekündige

(240 Frames lange) Animation abgespielt werden. Ist der Nutzer mit der Navigation im 3D-Viewport vertraut, kann der Simulierte Stern außerdem aus allen Perspektiven betrachtet werden. Um eine bessere Ansicht des inneren Aufbaus des stellaren Objektes zu gewinnen, kann mit der “Querschnitt” Checkbox eine Querschnittsansicht des Modells angezeigt werden. So können beispielsweise Schalen des Kernes eines Roten Überriesen besser unterschieden werden. Im Network-View-Fenster wird neben der HDA Node eine Infobox eingeblendet. Diese enthält einen kurzen Erklärtext mit allgemeinen Informationen zum angezeigten stellaren Objekt wie Entstehung, Größe, chemische Zusammensetzung und Temperatur. Außerdem wird in der Box die aktive Lebenszeit des Sternes (die Zeit bis zur Beendigung der Riesenstern Phase) und bei Hauptreihensternen die Kennung der Spektralklasse angegeben. Um auch sehr kurze Phasen wie die Supernova ohne manuelle Eingabe des Zeitwertes zu finden, kann über die beiden Knöpfe “vorheriges Stadium” und “nächstes Stadium” zwischen den unterschiedlichen Stadien eines Massewerts gesprungen werden. So können insgesamt zwölf Simulationen durch den Nutzer experimentell ermittelt und betrachtet werden und mit Hilfe der Infoboxen kann zusätzliches Wissen über die stellaren Evolutionsstufen erlangt werden.

Die entwickelte Lernumgebung innerhalb der Houdini Software erfüllt die in Kapitel 3 erörterten Kriterien zur Förderung der Informationsvermittlung. Durch die dreidimensionalen Simulationen werden beim Lernenden die visuellen Register angesteuert. Dadurch wird ein mentales Modell des Prozesses erzeugt, welches es erleichtert, diesen einzuprägen. Der Erklärtext der stellaren Evolutionsstufen wird nicht nur gelesen, sondern mit diesem mentalen Modell verknüpft. Da sich die im Text beschriebenen Informationen in der Simulation widerspiegeln, sind sie leichter zu verstehen. Weiterhin sind die dargestellten Prozesse animiert, was, wie in Kapitel 3.3 erläutert, das Verständnis besonders für bewegte Prozesse fördert. Die Animationen bieten dadurch einen entscheidenden Vorteil gegenüber dem Lernen durch statische Bilder wie in Lehrbüchern. Beispielsweise die Rotation der Akkretionsscheibe eines Protosterns, eine Supernova oder abstraktere Zusammenhänge wie Druck und Temperatur können besser durch Animationen visualisiert werden. Die Ebene der Interaktion wird ebenfalls abgedeckt, da die Eingaben des Nutzers das angezeigte Ergebnis bestimmen. Im Zusammenhang mit dem in Kapitel 5.1 erarbeiteten Umfeld für ein Schülerexperiment bildet die Simulation eine Möglichkeit für ein praktisches Experiment im Zusammenhang mit dem sonst schwer vorstellbaren Bereich der Astrophysik.

Die Schüler erarbeiten sich dabei ihre Informationen selbst was, wie in Kapitel 3.2 erläutert, essentiell zum Verständnis für Zusammenhänge beiträgt. Weiterhin ist das dargestellte Experiment eine Kompensation für die in Kapitel 3.4 aufgezeigten Defizite des derzeitigen Physikunterrichts an Schulen. Die Rolle der Schüler ist dabei oft zu passiv, was dazu führt, dass Zusammenhänge nicht greifbar und damit auch nicht vorstellbar sind. Weiterhin wird beim Experimentieren oft zu wenig Freiheit zur Findung des Lösungsweges gelassen. Der Ablauf wird genau vorgeschrieben und der Schüler erarbeitet sich Informationen nicht, sondern führt lediglich vorgegebene Schritte aus, was der Grundidee des Experimentierens widerspricht. Im Kontext des dargestellten Experiments könnten Schüler in einer dedizierten Doppelstunde frei mit dem HDA experimentieren. Jeder könnte sich dabei auf sein individuelles Lerntempo angepasst, selbständig Informationen erarbeiten. Die aufgeführten Aufgaben geben keine Anleitung, wie man zum Ergebnis kommt, sondern fordern den Schüler auf, selbst einen Weg zur Beantwortung der Fragen zu finden. Auf diese Weise würden die Schüler dazu angeregt werden, sich tiefgründig mit der Thematik zu befassen und die selbständige Erarbeitung des Lösungsweges würde die Entwicklung von intrinsischem Interesse fördern.

## 6 Fazit

Das Forschungsziel dieser Arbeit war die Untersuchung interaktiver dreidimensionaler Partikelsimulationen mit dem Ziel der Informationsvermittlung über astrophysikalische Prozesse. Dafür wurden zunächst die astrophysikalischen Prozesse selbst untersucht. Für den Rahmen der Arbeit wurde die stellare Evolution als Teilbereich der Astrophysik zum untersuchten Gegenstand. Kapitel 2 legt dafür zunächst alle theoretischen Grundlagen und Parameter fest, anhand derer eine interaktive Simulation möglich ist. Es wurde die Erkenntnis gewonnen, dass für die Bestimmung des evolutionären Stadiums eines Sternes vordergründig seine Masse und sein Alter eine Rolle spielen und dass sich andere Eigenschaften aus diesen Parametern ableiten lassen. Die verschiedenen evolutionären Stadien wurden daraufhin einzeln aufgeschlüsselt, wobei der Fokus vor allem auf dem Aufbau lag, da dieser für die Entwicklung von Modellen zur Visualisierung von zentraler Bedeutung ist.

Kapitel 3 beschäftigt sich zunächst mit den grundlegenden Parametern der Informationsvermittlung. Es wird darauf eingegangen, welche Aspekte zum Verständnis von Zusammenhängen beitragen. Dabei wurde die Erkenntnis gewonnen, dass sowohl Visualisierung als auch Dynamik entscheidend zur Entwicklung mentaler Modelle beitragen und somit das Verständnis für Prozesse vereinfachen. Durch Interaktion wird ebenfalls das Lernen gefördert, da wenn es vonnöten ist, selbst aktiv zu werden und sich Informationen zu erarbeiten, ein persönlicher Bezug zu den erarbeiteten Informationen hergestellt wird. Darauffolgend wurde in diesem Kapitel auf die derzeitige Gestaltung und die Defizite des Physikunterrichts eingegangen. Es wurde diskutiert, wie ein freies Experimentieren dargelegte Defizite ausgleichen könnte. Schüler könnten im individuellen Tempo frei ohne Bewertungsdruck experimentieren und wären zur selbständigen Erarbeitung von Informationen angehalten, was die Entwicklung von Interesse und das Lernen fördern würde.

Kapitel 4 gibt einen Überblick über die Möglichkeiten zur Erstellung dreidimensionaler Partikelsimulationen. Es wurde die Erkenntnis gewonnen, dass sich Houdini als Simulationsumgebung für diesen Zweck eignen würde. Wissenschaftliche Simulationsumgebungen sind nicht nutzerfreundlich gestaltet was eine Bedienung ohne großes physikalisches Vorwissen stark erschwert und sie benötigen verhältnismäßig hohe Rechenleistungen, was die private Nutzung ohne Zugang zu entsprechender Hardware nahezu unmöglich macht. Entwicklungsumgebungen für Spiele würden den Aspekt der

Interaktion hingegen vereinfachen, da die Anwendungen vollkommen individuell und interaktiv gestaltet werden können. Sie haben jedoch den entscheidenden Nachteil, dass die Simulation unbedingt performant sein muss, also 60 mal in der Sekunde aktualisiert wird. Sehr komplexe Berechnungen, welche zur Visualisierung astrophysikalischer Prozesse von Nöten sind, sind daher mit Spiel Entwicklungsumgebungen schwer umzusetzen. Houdini bietet hingegen eine gute Grundlage zur Erstellung von Simulationen, da es dafür seit Jahren in der Filmindustrie eingesetzt wird. Die prozedurale Funktionsweise ermöglicht den Aufbau von Setups, welche sich gegenseitig beeinflussen und dynamisch aktualisiert werden können. POPs ist ein Teilbereich von Houdini, welcher speziell zur Erstellung von Partikelsimulationen ausgelegt ist und verschiedene Möglichkeiten zur Umsetzung dieser bietet. Dafür gibt es einerseits vorgefertigte Nodes, welche durch die Manipulation von Parametern gesteuert werden können und andererseits die Möglichkeit, durch Skripting in VEX Mathematik basierte individuelle Modifikationen vorzunehmen. Houdini bietet folglich viel gestalterische Freiheit bei der Erstellung dreidimensionaler Partikelsimulationen. Weiterhin besteht die Möglichkeit eigene Nodes, sogenannte HDAs zu erschaffen welche Parameter manipulieren und interne Logik in Form von Python Skripten ausführen können. So können in der Umgebung vollkommen individuelle Funktionalitäten implementiert werden, wodurch das Erstellen von interaktiven Simulationen möglich ist.

Im Methodik Teil dieser Arbeit wird eine interaktive dreidimensionale Partikelsimulation erstellt. Dafür wurden die in Kapitel 2 gewonnenen theoretischen Erkenntnisse der stellaren Evolution mit Hilfe der in Kapitel 4 beschriebenen technischen Mittel in der Simulationsumgebung Houdini umgesetzt. Es wurde dabei auf die Erfüllung der Kriterien zur Informationsvermittlung aus Kapitel 3 Wert gelegt. Als Rahmenbedingung für den Einsatz der Simulation wurde ein Schülerexperiment festgelegt. Die Schüler sollen dabei in einer Doppelstunde zum Thema Sterne mit dem praktischen Projekt frei experimentieren können und dabei einige Aufgaben (*Abb. 16*) erfüllen. Die Erkenntnisgewinnung erfolgt hierbei anhand von gestellten Aufgaben, welche selbständig mit der Simulationsumgebung gelöst werden müssen. Grundlegend soll die Anwendung eine HDA Node in Houdini sein, welche die Eingabe von stellarer Masse und Zeit ermöglicht und anhand dieser Parameter das Evolutionsstadium eines Sternes ermittelt. Ein Modell des Sternes im jeweiligen Stadium soll in Form einer animierten dreidimensionalen Partikelsimulation angezeigt werden und so den Zusammenhang visualisieren. Es sollten außerdem einige essenzielle Informationen zum gezeigten Stern in

Textform angezeigt werden. Bei der Erstellung der Experimentierumgebung gab es einige Probleme, welche es abzuwägen galt. Einerseits besteht ein Konflikt zwischen der Interaktion und dem visuellen Verständnis bzw. dem Detailgrad der Darstellungen. Für den maximalen Grad an Interaktion müssten alle Simulationen so performant wie möglich ablaufen, um diese Performanz zu ermöglichen, müssen jedoch Abstriche beim Detailgrad in Kauf genommen werden. Weiterhin müssen die Simulationen so klar wie möglich den Aufbau der stellaren Objekte darstellen, weswegen teilweise Abstraktionen nötig sind. Dadurch dass die Benutzung nicht überfordernd sein darf, müssen Parameter möglichst vereinfacht werden, was eine gewisse wissenschaftliche Ungenauigkeit mit sich bringt, da in der Realität immer deutlich mehr Faktoren von Relevanz sind als im experimentellen Rahmen abgebildet werden können.

Interaktive Partikelsimulationen sind grundlegend umsetzbar, haben aber die beschriebenen technischen Limitationen. Es wäre einerseits nur schwer möglich, eine vollkommen physikbasierte interaktive Simulation umzusetzen, da die Komplexität der Berechnungen hier dafür sorgen würde, dass die Interaktivität aufgrund hoher Ladezeiten verloren gehen würde. Andererseits sind Partikelsimulation als Mittel zur Verständnis Vereinfachung astrophysikalischer Prozesse, wie in Kapitel 5 bewiesen, sehr gut denkbar. Eine Möglichkeit zur Umsetzung dieser wurde hier vorgestellt. Es wurden zunächst einzelne Simulationen der stellaren Evolutionsstufen mit SOPs, POPs und VEX erstellt und eine Benutzeroberfläche in Form eines HDA umgesetzt. Die Auswahl der korrekten Simulation anhand der Benutzereingaben und die Umsetzung weiterer Funktionen zum Experimentieren erfolgte durch Python Logik.

Die umgesetzte Simulation erfüllt die in Kapitel 3 beschriebenen Kriterien der Informationsvermittlung, da sie interaktiv und dynamisch astrophysikalische Prozesse visualisiert. Im beschriebenen Umfeld würde die Simulation dafür sorgen, dass die Schüler ein genaueres Bild der stellaren Objekten entwickeln. Auf diese Weise werden astrophysikalische Prozesse greifbar und einprägsam vermittelt.

# Literaturverzeichnis

Albakry, S. (2015). *Determination the Heating Rate and the Characteristic Age for Pulsars Stars*. Iraqi Journal of Science

Allard, F. & Homeier D. (2007). *Brown Dwarfs*.  
Scholarpedia

Baker, J. (2009). *Kernfusion*. In: 50 Schlüsselideen Physik.  
Spektrum Akademischer Verlag.

Boblest, S. , Müller, T. & Wunner, G. (2015). *Spezielle und allgemeine Relativitätstheorie*  
Springer Berlin, Heidelberg

Braun, T. & Backhaus, U. (2006) *Entwicklung eines Laboratoriums für offene Experimente*. Didaktik der Physik. Frühjahrstagung Kassel 2006

Burrows, A. & Liebert, J. (1993). *The Science of Brown Dwarfs*.  
American Physical Society

Camenzind, M. (2016). *Vom Protostern zum Schwarzen Loch*. In: Gravitation und Physik kompakter Objekte. Springer Spektrum, Berlin, Heidelberg

Chandel, T. (2021). *Performance Evaluation of solar photovoltaic module using photoluminescence image analysis*. (Doktorarbeit) Integral University Lucknow

Cook, L. (2010). *W33A Accretion Disk*.  
International Gemini Observatory

EHT Collaboration (2022). *First image of the black hole at the centre of the Milky Way*.  
Event Horizon Telescope Collaboration

Fraedrich, R. & Westermann, R. (2012). *Motion visualization in large particle simulations*.  
Society of Photo-Optical Instrumentation Engineers

Göbel, H. (2016). *Gravitation und Relativität: Eine Einführung in die Allgemeine Relativitätstheorie*. (2. Aufl.) Berlin, Boston: De Gruyter

Güdel, M., Elmegreen, B. & Toth, L. (2018) *Cosmic Pathways to Life: From Interstellar Molecules to the First Traces of Life*. Proceedings of the International Astronomical Union.

Hansen, C. J. , Kawaler, S. D. & Trimble, V. (2004). *Stellar Interiors*.  
(2. Aufl.). Springer New York

Hanslmeier, A. (2020). *Einführung in Astronomie und Astrophysik*.  
(4 Aufl.) Springer Spektrum Berlin, Heidelberg

ISB, Staatsinstitut für Schulqualität und Bildungsforschung München (2004). *Lehrplan Physik (Pflicht/Wahlpflicht) - Jahrgangsstufe 11, 12*. Bayerisches Staatsministerium für Unterricht und Kultus

Jones, T. (2019) *Rosette Nebula - NGC 2237*.  
Astro Backyard

Jurik, P. (2021) *Magnetar - neutron star in deep space*.  
Adobe Stock

Kapferer W. & Riser T. (2008). *New Journal of Physics*. Visualization needs and techniques for astrophysical simulations. IOP Publishing Ltd

Kippenhahn, R. (1970). *Die Entstehung von Sternen*. In: Verhandlungen der Gesellschaft Deutscher Naturforscher und Ärzte. Springer, Berlin, Heidelberg

Koshizuka S. (2011). *Journal of Nuclear Science and Technology*. Current Achievements and Future Perspectives on Particle Simulation Technologies for Fluid Dynamics and Heat Transfer. Taylor & Francis

Liu, S., Wang, Z., Gong, Z., Huang, L. & Peng Q. (2007) *Physically based animation of sandstorm*. In: Comp. Anim. Virtual Worlds. John Wiley & Sons Ltd

Massey, P. & Michael R. M.. (2001) *Stellar masses*.  
Encyclopedia of Astronomy & Astrophysics. CRC Press.

Minor D. (2007). *Physical Simulation of Fire and Smoke*. (Masterarbeit)  
Bournemouth University



Muzerolle, J., NASA & ESA (2013). *Hubble image of LRL 54361*.

ESA/Hubble Space Telescope

Navratil, P., Johnson, J. & Bromm V. (2007). *Visualization of Cosmological Particle-Based Datasets*. IEEE Transactions on Visualization and Computer Graphics (13. Aufl.). Institute of Electrical and Electronics Engineers

Nowikow, I. D. (1981). *Was ereignet sich innerhalb eines Schwarzen Loches?*. In: Schwarze Löcher im All. Teubner Verlag. Vieweg.

O'Connor, E. & Ott, C. D. (2011). *Black Hole Formation in Failing Core-Collapse Supernovae*. The American Astronomical Society

Oliver, P. (2012). *Unreal Engine 4 Elemental*. In: ACM SIGGRAPH 2012 Computer Animation Festival Association for Computing Machinery, New York

Oxford English Dictionary, (2023)

Oxford University Press

Schnittman, J. (2019). *Black Hole Accretion Disk Visualization*.

NASA Scientific Visualization Studio

Seidel, T., Prenzel, M., Rimele, R., Dalehefte, I. M., Herweg, C., Kobarg, M. & Schwindt, K. (2006) *Blicke auf den Physikunterricht. Ergebnisse der IPN Videostudie*. In: Zeitschrift für Pädagogik. (52 Aufl.)

Snell, R. L. (2011). *Interstellar Medium*. In: Garguad, M. et al. Encyclopedia of Astrobiology. Springer, Berlin, Heidelberg.

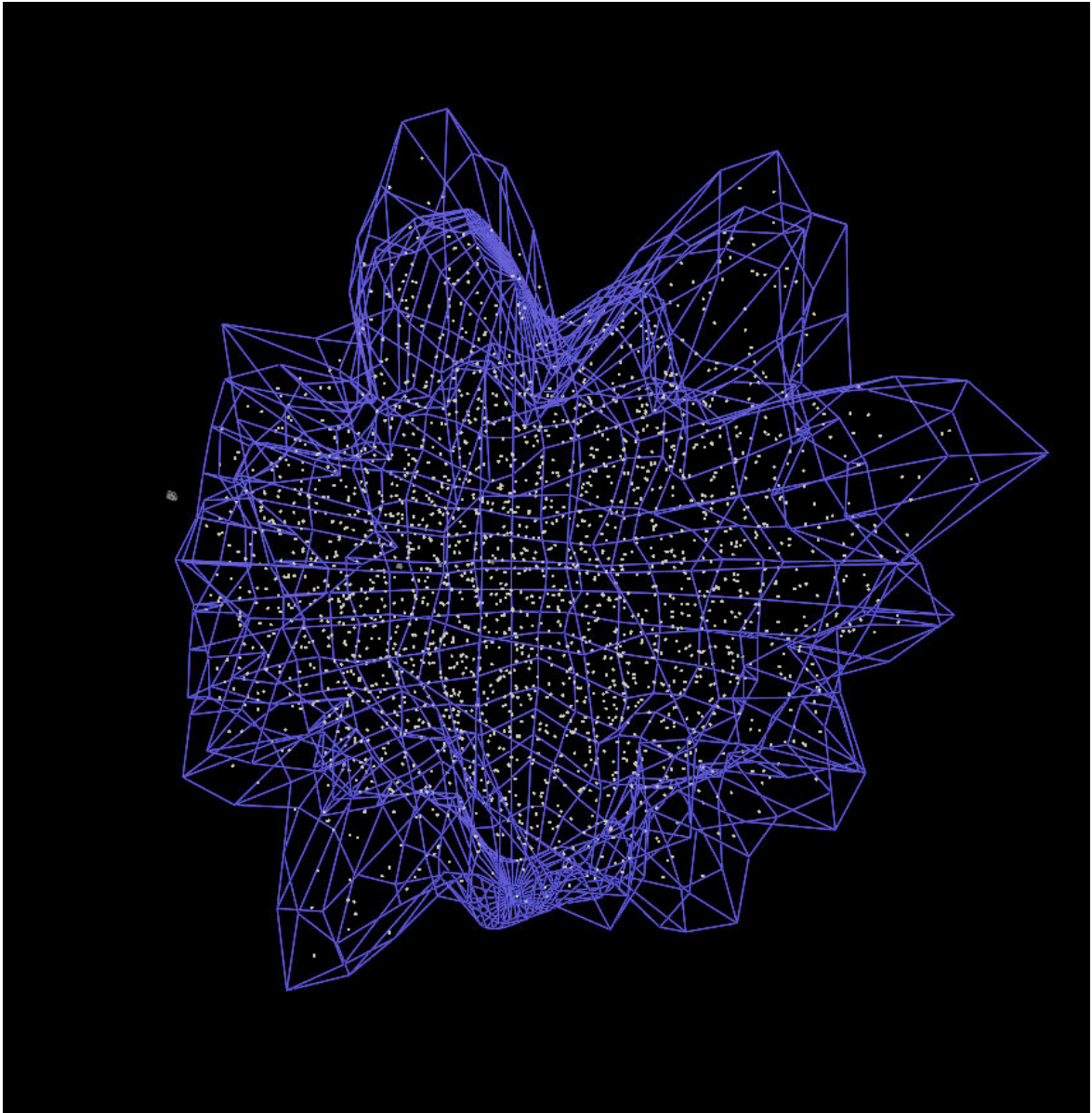
Tscharnuter, W. M. (1981). *Frühphasen der Sternentstehung*. Mitteilungen der Astronomischen Gesellschaft (51. Aufl.), Max-Planck Institut, Garching

Verboncoeur, J. P., (2005). *Plasma Physics and Controlled Fusion*. Particle simulation of plasmas: review and advances. IOP Publishing Ltd

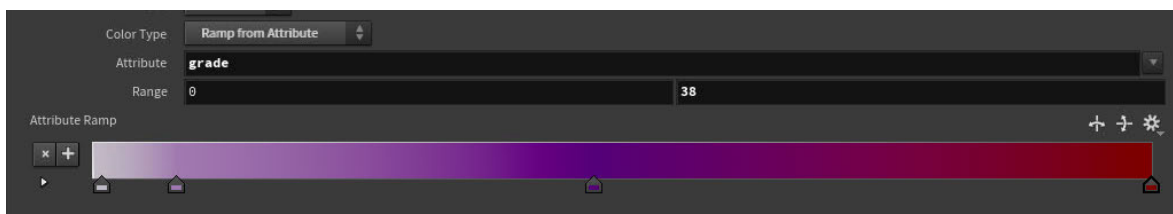
Wagner, I. (2013): *Lernen mit Animationen: Effekte dynamischer und statischer Visualisierungen auf die Bildung perzeptueller und kognitiver Repräsentationen beim Erwerb von Wissen über dynamische Sachverhalte* (Dissertation). Universität Koblenz-Landau

- Walter, F. & NASA (1997). *Isolated Neutron Star RX J185635-3754*  
State University of New York at Stony Brook, NASA
- Wambsganß, J. (2012). *Die Geburt der Sonne*. In: Universum für alle.  
Springer, Berlin, Heidelberg.
- Wedekind, J. (2007). *Lernen mit interaktiven Visualisierungen*. In: Überwindung von  
Schranken durch E-Learning. Studienverlag, Innsbruck-Wien-Bozen
- Wilhelm, T. & Heuer, D. (1995). *Lernen von Konzepten zur Dynamik - dynamische  
Physikrepräsentation am Computer zur Visualisierung*. Leuchtturm-Verlag, Alsbach
- Young, D. (2012). *Pulsating Variable Stars and The Hertzsprung-Russell Diagram*.  
National Earth Science Teachers Association
- Zendler, A. (2018). *MINT-Fächer: Mathematik, Informatik, Naturwissenschaften  
(Biologie, Chemie, Physik), Technik*. In: Unterrichtsmethoden für MINT-Fächer. Springer  
Vieweg, Wiesbaden.
- [1] Informationen zur Houdini Software, enthaltene Nodes und Arbeitsweisen sind in der  
Houdini Dokumentation zu überprüfen: SideFX. *Houdini Documentation*.  
<https://www.sidefx.com/docs/>

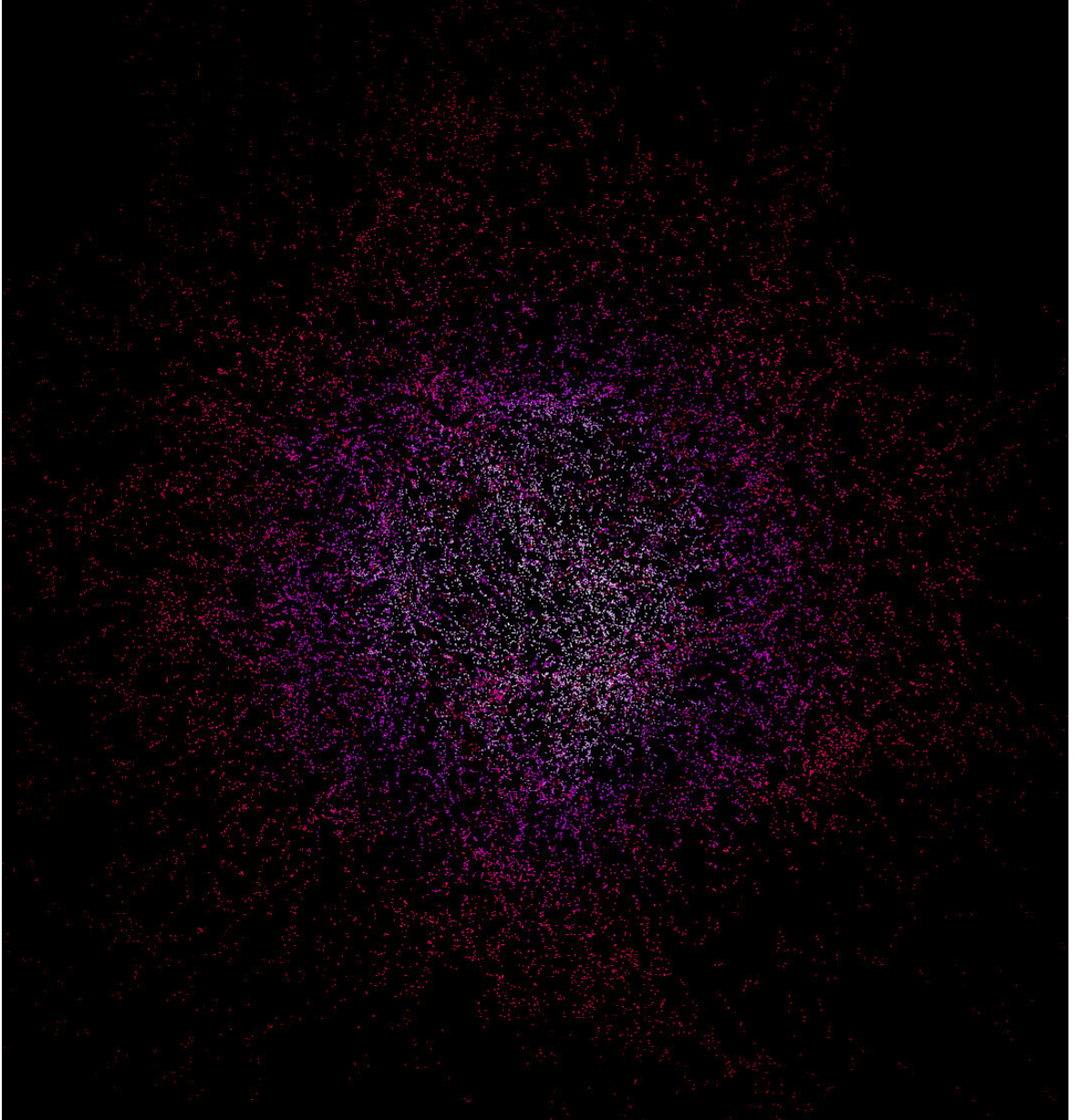
## Anhang



*Abbildung 20: “Grundkörper der Schichten der interstellaren Wolke”*

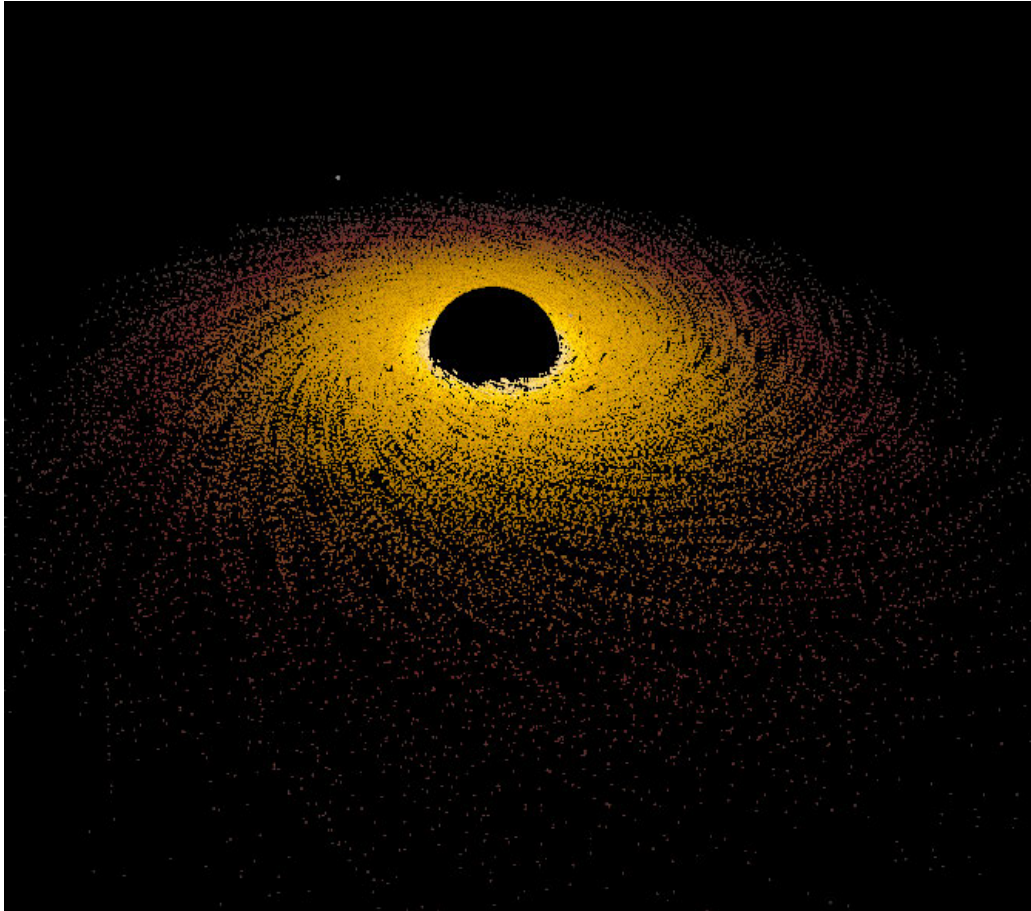


*Abbildung 21: “Farbanpassung mit Hilfe einer Ramp”*

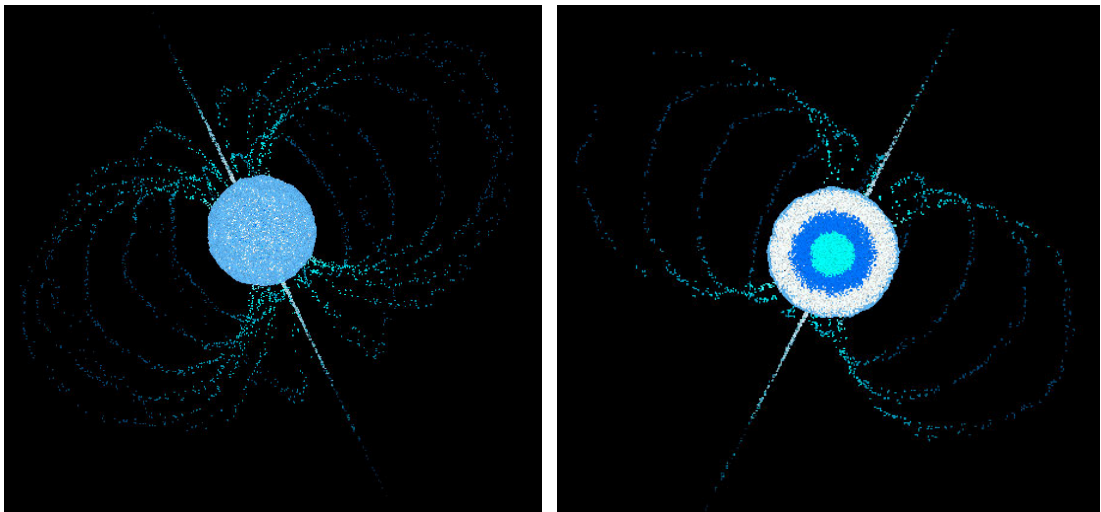


*Abbildung 33: “Simulation der interstellaren Wolke”*

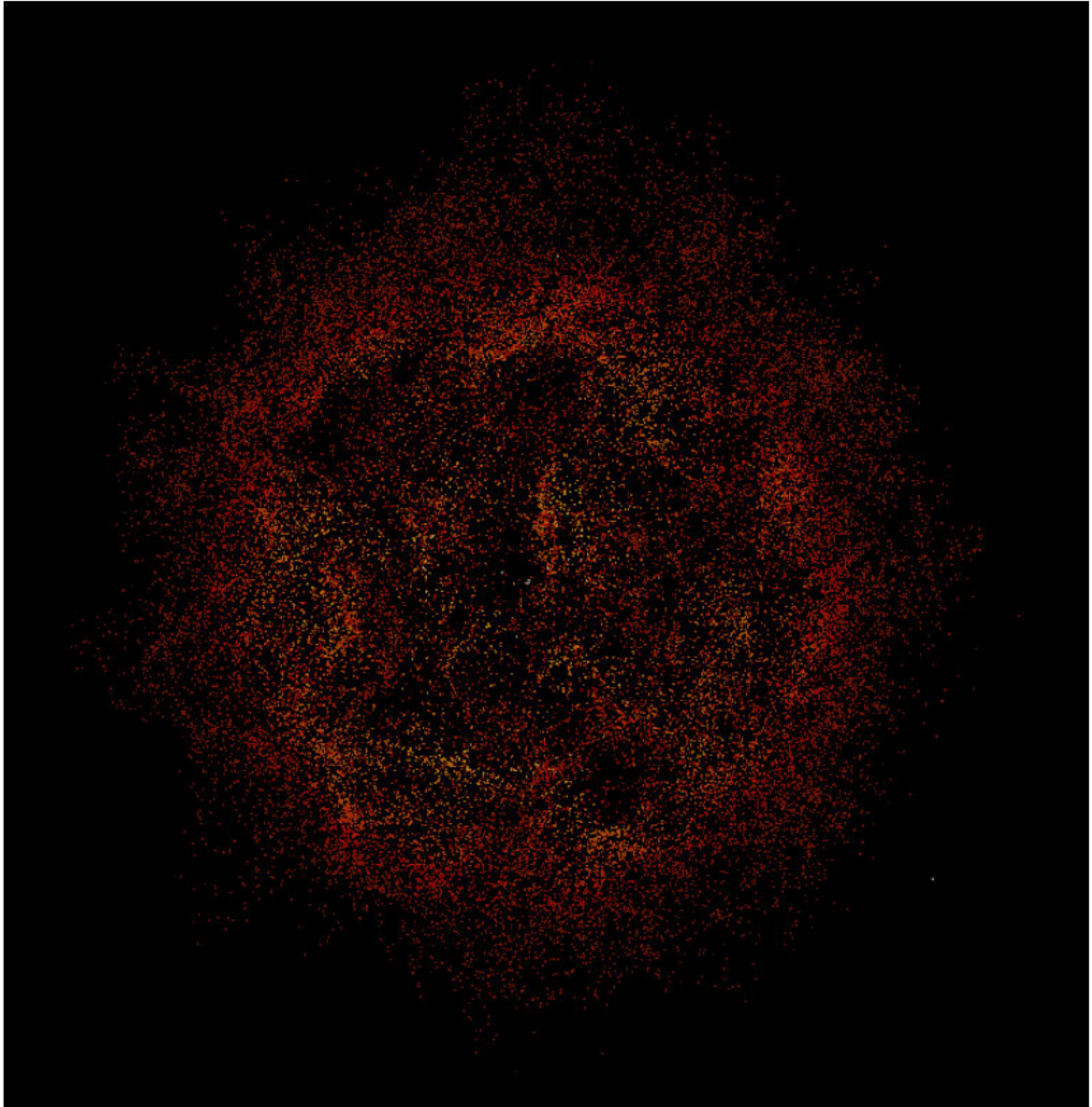




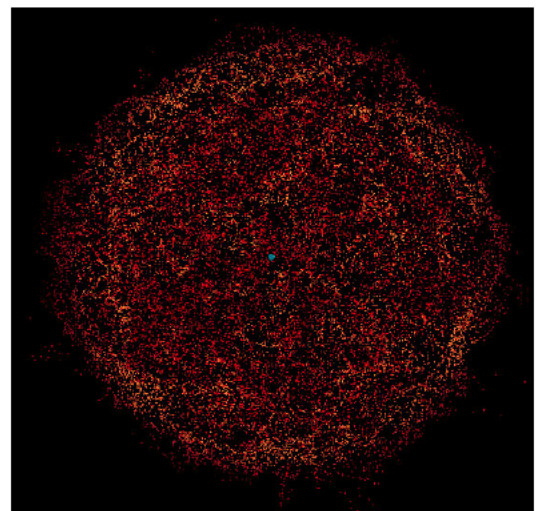
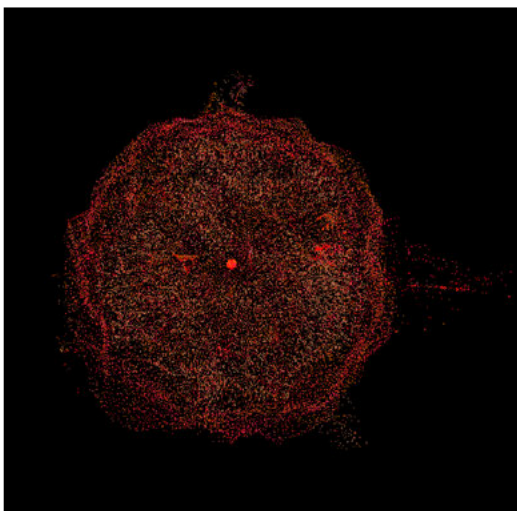
*Abbildung 34: "Simulation des schwarzen Lochs mit Akkretionsscheibe"*



*Abbildung 35: "Simulation des Neutronensterns & Querschnitt der Simulation"*



*Abbildung 36: "In Planetaren Nebel zerfallener roter Riese"*



*Abbildung 37: "Außenhülle des roten Riesen(links) und des roten Überriesen (rechts)"*



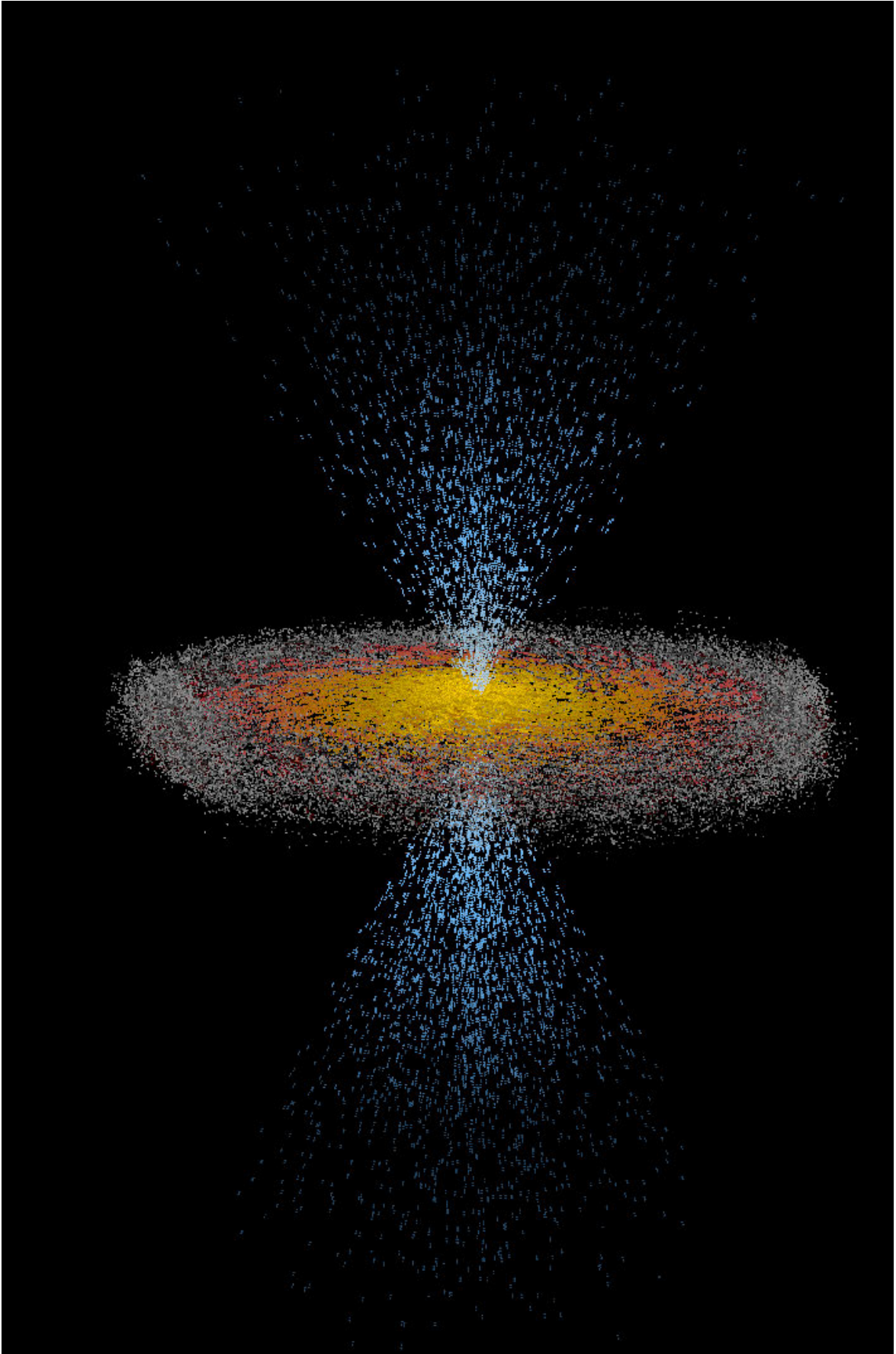
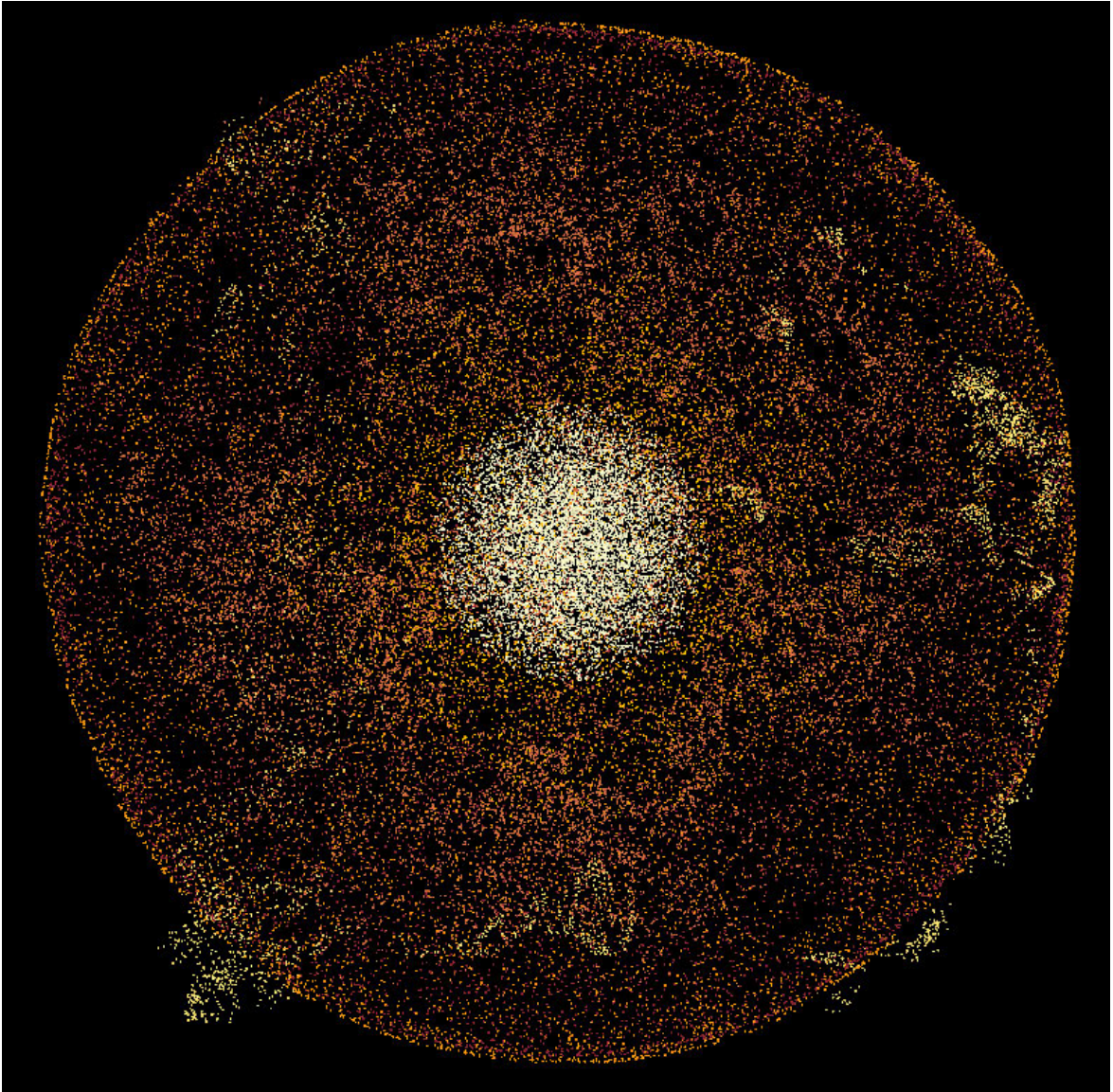
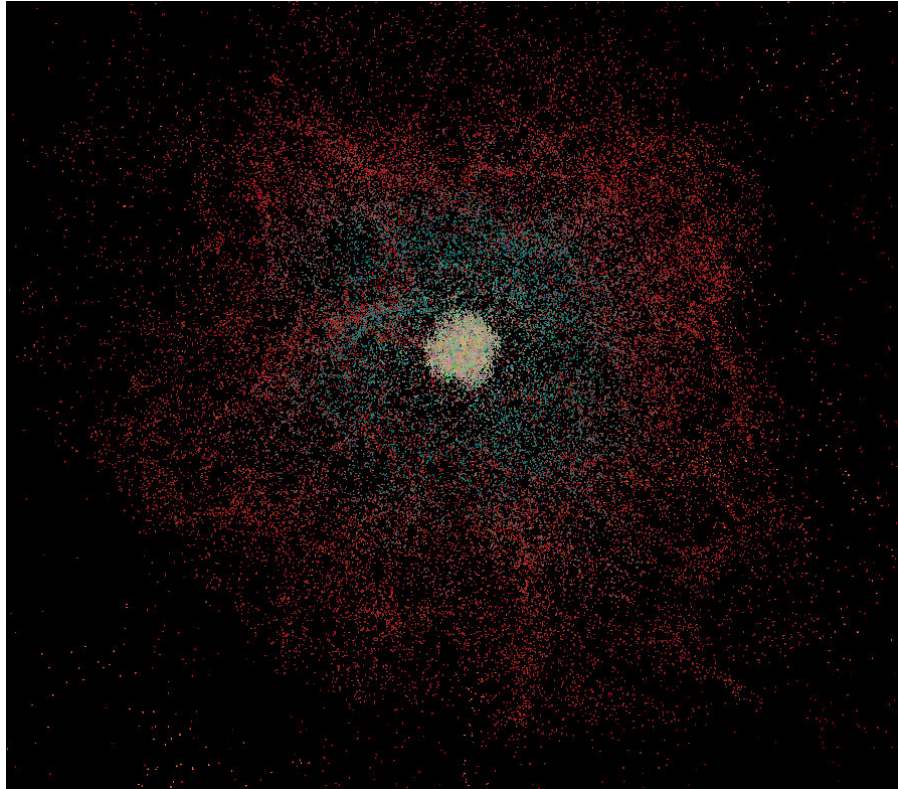


Abbildung 38: "Simulation des Protosterns"

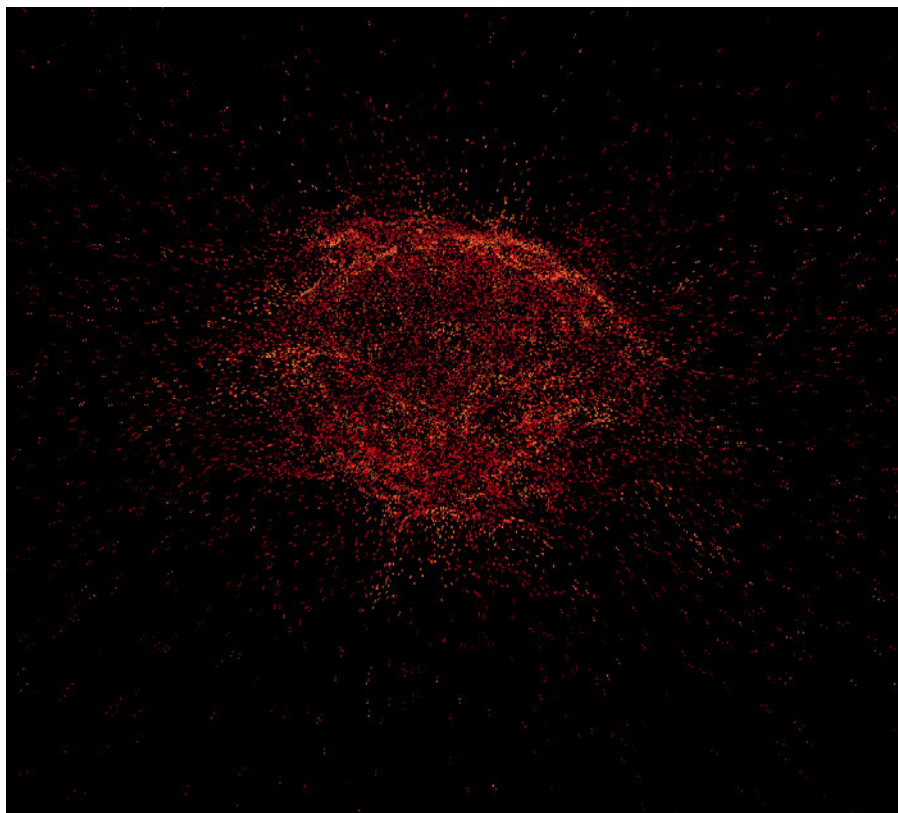


*Abbildung 32: "Simulation des Hauptreihensternes (Spektralklasse G)"*





*Abbildung 39: “In einer Supernova Kollabierender Kern des roten Überriesen”*



*Abbildung 40: “Zerstörung der Außenhülle des roten Überriesen durch die Supernova”*

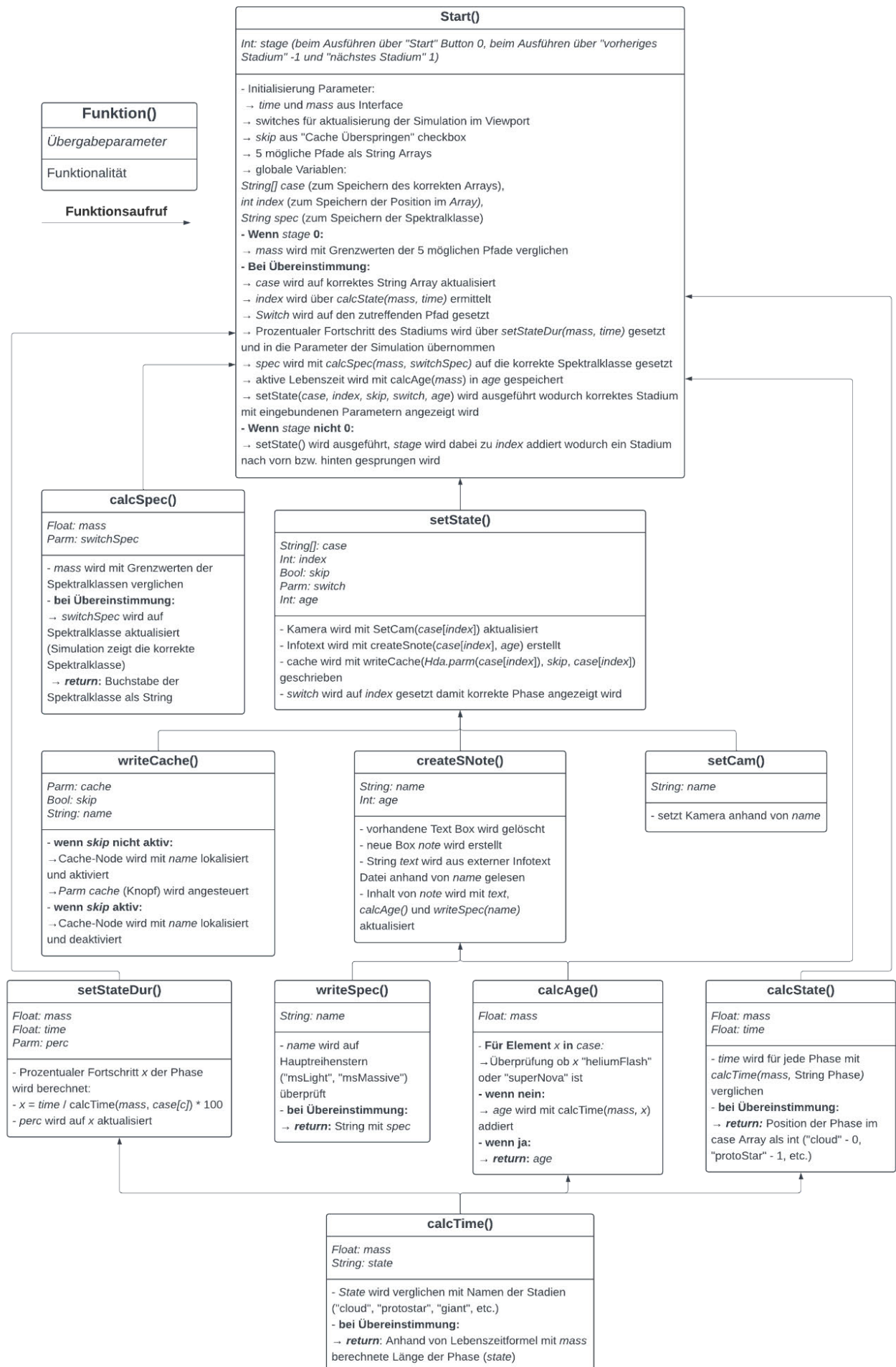


Abbildung 31: "Struktur der Funktionen"

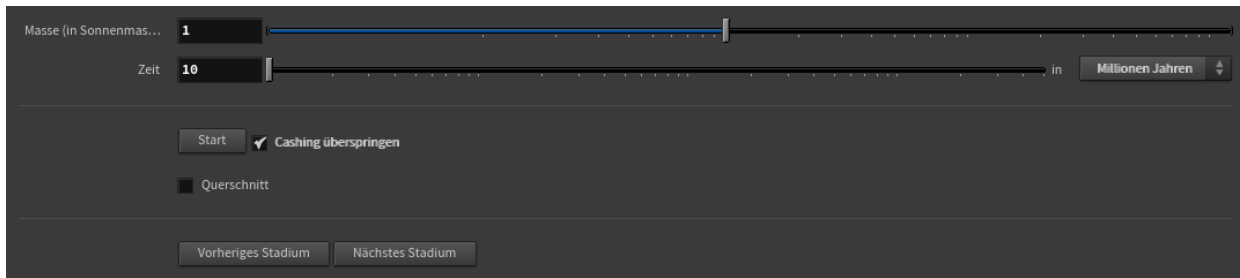


Abbildung 27: "Benutzeroberfläche des HDA"

```
# String state case arrays
brownDwarf = ["cloud", "protostar", "bDwarf"]
msLight = ["cloud", "protostar", "msLight", "giant", "heliumFlash", "whiteDwarf", "nebula"]
msHeavy = ["cloud", "protostar", "msMassive", "giant", "heliumFlash", "whiteDwarf", "nebula"]
neutronStar = ["cloud", "protostar", "msMassive", "superGiant", "superNova", "neutronStar"]
blackHole = ["cloud", "protostar", "msMassive", "superGiant", "superNova", "blackHole"]
```

Abbildung 41: "Die fünf Arrays der Pfade mit enthaltenen Stadien als Strings"

## Leben und Tod eines Sternes

Die Sonne: Ein riesiger Ball aus Gas, unvorstellbar heiß, tausende Kilometer von uns entfernt und doch der Grund, warum wir überhaupt existieren können. Ein einziger Stern, der das Leben auf der Erde möglich macht. Dabei ist sie nur einer von rund 100 Milliarden Sternen in der Milchstraße. Jeder einzelne unterscheidet sich in seinem Aussehen und doch folgen alle den selben universellen physikalischen Gesetzen. Wie entsteht ein Stern? Wie lange wird unsere Sonne uns noch als Lebensgrundlage dienen können und was passiert mit ihr, nachdem sie ihr Leben beendet hat? Diese Fragen wirst du in der nächsten Doppelstunde mit Hilfe eines digitalen Experiments beantworten. Deine Lehrkraft wird dir die technische Funktionsweise des Stern Simulators in Houdini erklären und dann wirst du selbständig folgende Aufgaben bearbeiten:

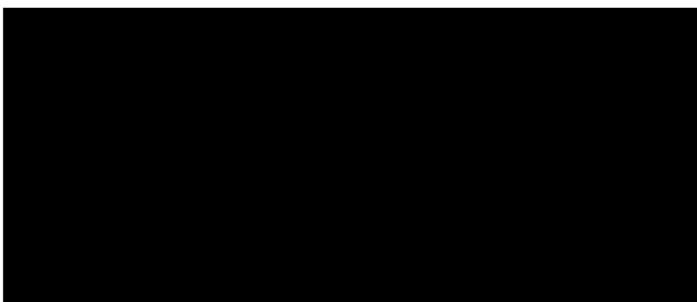
1. Erkläre den Begriff "Hauptreihenstern"! Welches Kriterium muss ein Stern erfüllen, um dieser Kategorie zugeordnet zu werden?
2. Beschreibe was passiert, wenn ein Stern nicht genügend Masse besitzt, um einen Hauptreihenstern zu bilden!
3. Entwickle ein Baumdiagramm, auf dem alle Evolutionspfade abgebildet sind, welche ein Stern durchlaufen kann und markiere darauf den gesamten Lebenspfad unserer Sonne!
4. Nenne, wie sich Hauptreihensterne in ihrem inneren Aufbau unterscheiden können!
5. Nenne alle Spektralklassen von Hauptreihensternen und deren zugehörige Farbe!
6. Ermittle den Massenunterschied zwischen Roten Riesen und Roten Überriesen und ordne die Sonne einer Kategorie zu!
7. Ermittle wie viel initiale Masse ein Stern benötigt, um ein schwarzes Loch zu bilden!
8. Nenne den Namen des spektakulären Prozesses, aus dem Schwarze Löcher und Neutronensterne entstehen!
9. Nenne das kleinste und das größte Stellare Objekt!
10. Finde heraus, mit welchem Lebensmittel das Innere eines Neutronensterns bezeichnet wird!

*Abbildung 16: "Arbeitsblatt für digitales Schülerexperiment zur Sternentwicklung"*

## **Eidesstattliche Erklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe angefertigt habe. Textpassagen, die wörtlich oder dem Sinn nach auf Publikationen oder Vorträgen anderer Autoren beruhen, sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Berlin, 07.01.2024

A large black rectangular box used to redact the signature of the author.

Unterschrift