Cindy Lehner und Lukas Zainzinger, 21.12.2015, 5AHITM

Solar-System

Protokoll

Inhalt

[1. Aufwandsschätzung 2](#_Toc438489090)

[2. Zeitaufzeichnung 2](#_Toc438489091)

[3. Tatsächlicher Aufwand 2](#_Toc438489092)

[4. Aufgabenstellung 3](#_Toc438489093)

[5. UML-Diagramm 4](#_Toc438489094)

[6. Benutzer-Interface 4](#_Toc438489095)

[7. Installation 5](#_Toc438489096)

[8. Code 6](#_Toc438489097)

[7.1 Planeten 6](#_Toc438489098)

[7.2 Fixstern 7](#_Toc438489099)

[7.3 Mond 7](#_Toc438489100)

[7.4 Licht 8](#_Toc438489101)

[7.5 Texturen 9](#_Toc438489102)

[7.6 Kamera Perspektive 10](#_Toc438489103)

[7.7 Key-Events 10](#_Toc438489104)

[7.8 Hilfe-Menü 11](#_Toc438489105)

[9. Quellen 12](#_Toc438489106)

**Git-Repository: https://github.com/lzainzinger/sunsystem**

# Aufwandsschätzung

|  |  |
| --- | --- |
| Art | Stunde(n) |
| Recherche | 5 |
| Installation | 2-3 |
| Programmieren | 11 |
| Protokoll | 1 |

Aufwand in Stunden: 20-21

# Zeitaufzeichnung

In Minuten.

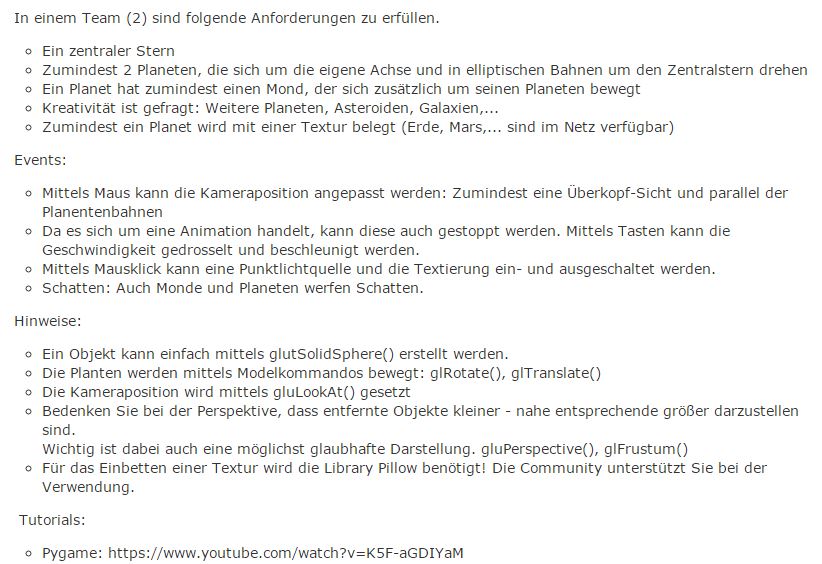
|  |  |  |  |
| --- | --- | --- | --- |
| Lehner | | Zainzinger | |
| 15.11.2015 | 60 | 15.11.2015 | 60 |
| 22.11.2015 | 75 | 22.11.2015 | 75 |
| 26.11.2015 | 90 | 26.11.2015 | 110 |
| 29.11.2015 | 40 | 29.11.2015 | 50 |
| 14.12.2015 | 80 | 13.12.2015 | 130 |
| 20.12.2015 | 150 | 20.12.2015 | 150 |
| 21.12.2015 | 150 | 21.12.2015 | 210 |

# Tatsächlicher Aufwand

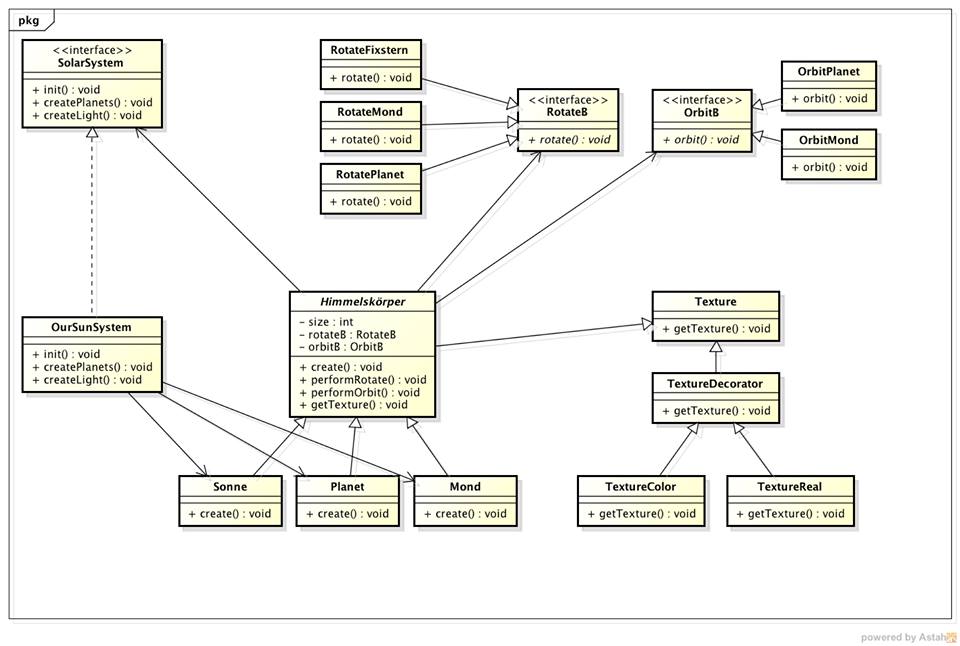
In Stunden

|  |  |
| --- | --- |
| Soll-Zustand | Ist-Zustand |
| 20-21 | 24 |

# Aufgabenstellung



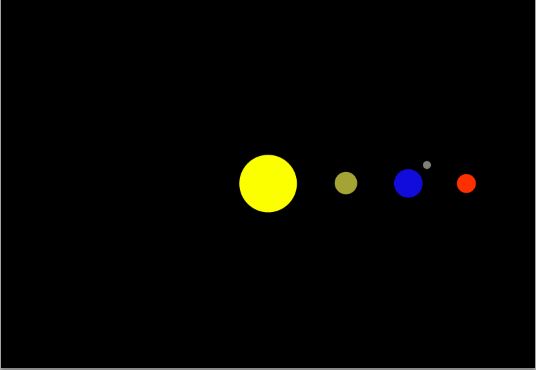
# UML-Diagramm



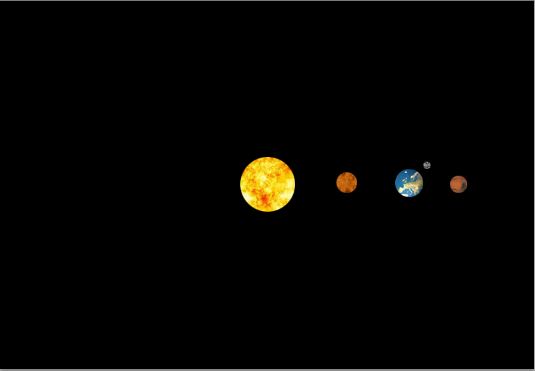
# Benutzer-Interface

Wir werden unser Solarsystem mit der Sonne, drei Planeten (Venus, Erde und Mars) und einem Mond, der um die Erde kreist, ausstatten. Für Erstbenutzer, beziehungsweise um die verschiedenen Eigenschaften, die unser Programm bietet einzusehen, gibt es ein Hilfe-Menü.

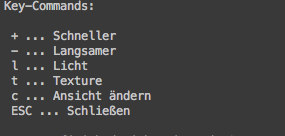
Sonnen-System:



Mit Textur:



Hilfe:



# Installation

Wir haben unser Projekt mit PyGame umgesetzt. Damit das Ganze funktioniert müssen vorerst ein paar Libraries installiert werden.

Python

PyGame

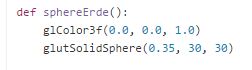
Pillow

PyOpenGL

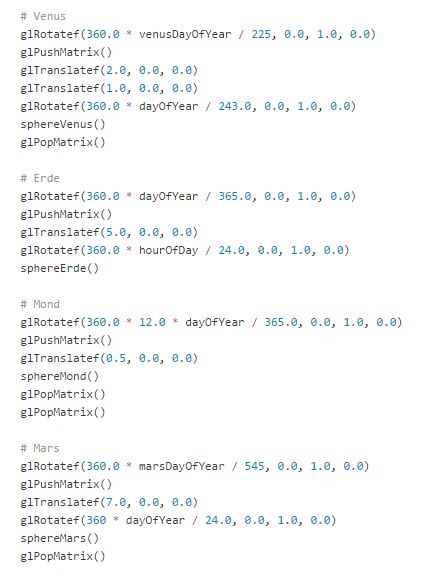
# Code

# Planeten

Wir haben drei Planeten: die Venus, die Erde und den Mars. Planeten werden mit folgenden Zeilen erstellt.



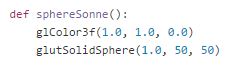
Mit glColor3f wird die Farbe festgesetzt. glutWireSphere bestimmt den Radius und gibt dann an, wieviele Wires gezeichnet werde. Da es solid ist sieht man diese aber natürlich nicht.



In der Rotate Funktion berechnen wir uns die Umlaufbahn um die Sonne und um die eigne Achse. Der Mond dreht sich um die Erde und ist deshalb innerhalb der Matrix der Erde.

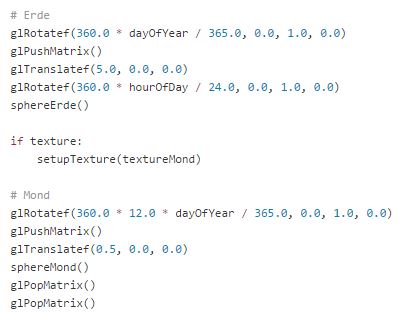
# Fixstern

Unser Fixstern, als das Zentrum um das sich die Planeten drehen, ist die Sonne.



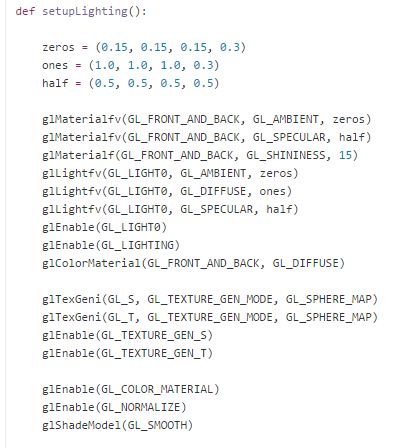
# Mond

Mond sind die einzigen Sphären, die sich nicht um die Sonne, sondern einen Planeten drehen. Daher befindet er sich in der Matrix der Erde.



# Licht

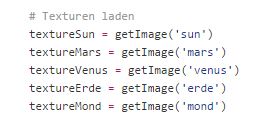
Mit der Taste „L“ wird das Licht ein- und ausgeschalten.



Anschließend wird nur noch programmiert, dass bei dem Tastaturbefehl, die Funktion ausgeführt wird.

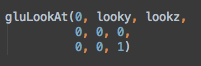
# Texturen

Die Funktion getImage() holt sich die Bilder für die Planeten vom Dateipfad und lädt sie. Mit setUpTexture werden die Texturen auf die Planeten gelegt und wenn das richtige KeyEvent gedrückt wird angezeigt.

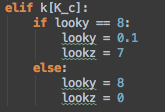


# Kamera Perspektive

Die Kameraperspektive wird durch den Befehl „glLookAt()“ geändert. In unserem Projekt gibt es eine Draufsicht und eine Ansicht von der Seite.

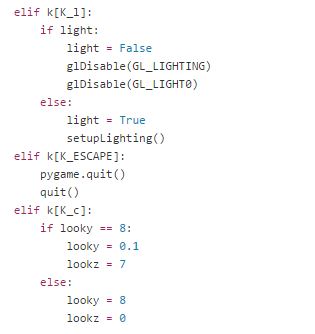


Die Werte werden mithilfe einer IF-Anweisung verändert:



# Key-Events

Um die verschiedenen Events auszuführen, haben wir uns jeweils einen Buchstaben auf der Tastatur ausgesucht.

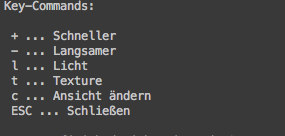


Mit „L“ wird das Licht ein- und ausgeschalten. Mit „P“ wird das Bild gestoppt und auf Pause gesetzt. Mit „+“ bzw. „-„ werden die Planeten schneller bzw. langsamer gemacht. Mit „T“ wird die Textur ein- oder ausgeschalten und mit

# Hilfe-Menü

Nachdem wir Probleme bei unserem erst gedachten Design hatten, haben wir uns dazu entschieden am Anfang beim Starten des Programmes alle Key-Events in der Konsole auszugeben.





# Quellen

PyGame Tutorials:

<https://www.youtube.com/watch?v=K5F-aGDIYaM>

Tutorial für das Erstellen von Buttons:

<https://pythonprogramming.net/pygame-buttons-part-1-button-rectangle/>

Die PyGame Dokumentation generell:

<https://www.pygame.org/docs/>

Die API von OpenGL im Allgmeinen:

<https://www.opengl.org/documentation/>

<http://www.stackoverflow.com/>

<https://pythonprogramming.net/>