Marlene Dorfinger, Muhammedmehdi Kanyildiz

Visit our solar system …

Inhalt

[1. Aufgabenstellung 2](#_Toc413669676)

[2. Zeitaufzeichnung 3](#_Toc413669677)

[3. Designüberlegung 4](#_Toc413669678)

[3.1 UML 4](#_Toc413669679)

[3.3 Splashscreen Prototyp 5](#_Toc413669680)

[3.4 Verwendete Libraries 5](#_Toc413669681)

[4. Funktionalität 5](#_Toc413669682)

[4.1 Pygame installieren 5](#_Toc413669683)

[4.2 PyOpenGL und Pillow installieren 5](#_Toc413669684)

[4.3 Texturen mit Pillow 5](#_Toc413669685)

[4.4 Sonnensysteminfos 6](#_Toc413669686)

[4.5 Prototyp 6](#_Toc413669687)

[4.6 Startbildschirm 6](#_Toc413669688)

[5. Fortschritt 7](#_Toc413669689)

[6. Fehleranalysen 7](#_Toc413669690)

[Quellen 7](#_Toc413669691)

# Aufgabenstellung

Wir wollen nun unser Wissen aus Medientechnik und SEW nützen um eine etwas kreativere Applikation zu erstellen.

Eine wichtige Library zur Erstellung von Games mit 3D-Grafik ist Pygame. Die 3D-Unterstützung wird mittels PyOpenGL erreicht.

Die Kombination ermöglicht eine einfache und schnelle Entwicklung.

Während pygame sich um Fensteraufbau, Kollisionen und Events kümmert, sind grafische Objekte mittel OpenGL möglich.

Die Aufgabenstellung:

Erstellen Sie eine einfache Animation unseres Sonnensystems:  


In einem Team (2) sind folgende Anforderungen zu erfüllen.

* Ein zentraler Stern
* Zumindest 2 Planeten, die sich um die eigene Achse und in elliptischen Bahnen um den Zentralstern drehen
* Ein Planet hat zumindest einen Mond, der sich zusätzlich um seinen Planeten bewegt
* Kreativität ist gefragt: Weitere Planeten, Asteroiden, Galaxien,...
* Zumindest ein Planet wird mit einer Textur belegt (Erde, Mars,... sind im Netz verfügbar)

Events:

* Mittels Maus kann die Kameraposition angepasst werden: Zumindest eine Überkopf-Sicht und parallel der Planentenbahnen
* Da es sich um eine Animation handelt, kann diese auch gestoppt werden. Mittels Tasten kann die Geschwindigkeit gedrosselt und beschleunigt werden.
* Mittels Mausklick kann eine Punktlichtquelle und die Textierung ein- und ausgeschaltet werden.
* Schatten: Auch Monde und Planeten werfen Schatten.

Hinweise:

* Ein Objekt kann einfach mittels glutSolidSphere() erstellt werden.
* Die Planten werden mittels Modelkommandos bewegt: glRotate(), glTranslate()
* Die Kameraposition wird mittels gluLookAt() gesetzt
* Bedenken Sie bei der Perspektive, dass entfernte Objekte kleiner - nahe entsprechende größer darzustellen sind.  
  Wichtig ist dabei auch eine möglichst glaubhafte Darstellung. gluPerspective(), glFrustum()
* Für das Einbetten einer Textur wird die Library Pillow benötigt! Die Community unterstützt Sie bei der Verwendung.

 Tutorials:

* Pygame: https://www.youtube.com/watch?v=K5F-aGDIYaM

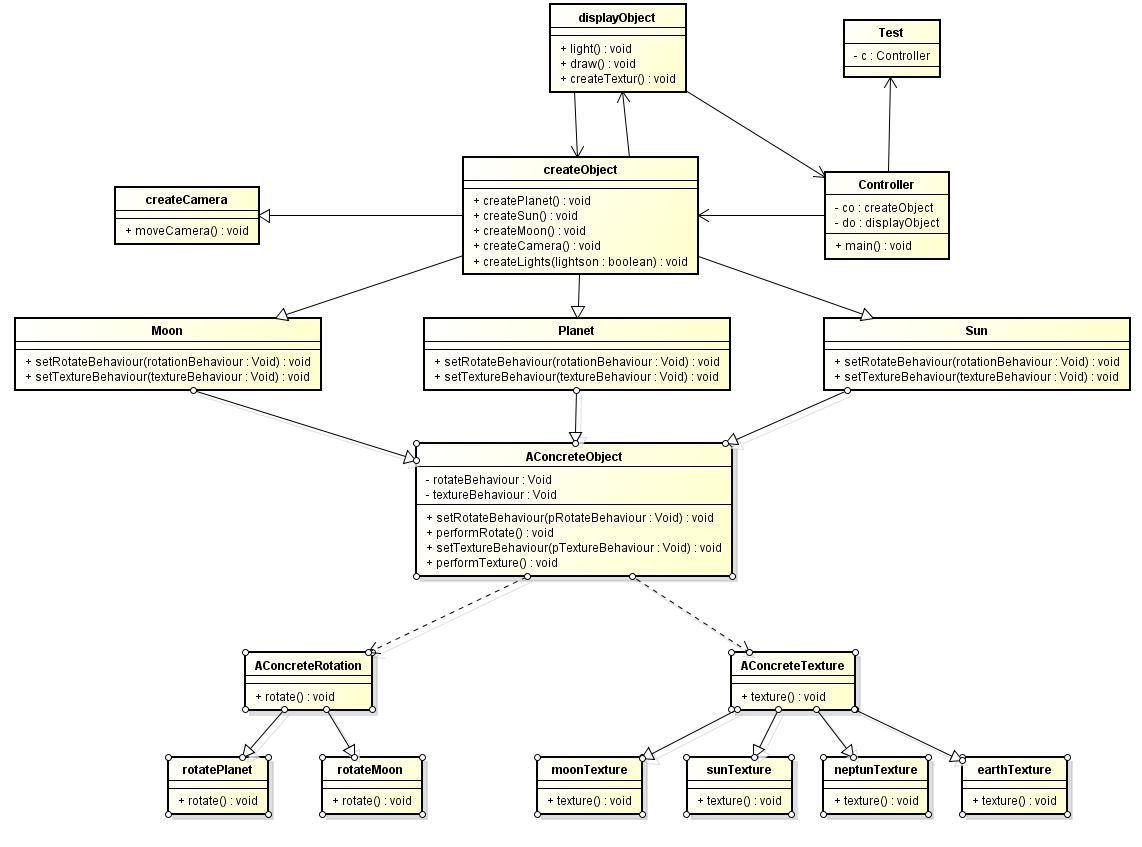
Viel Erfolg!

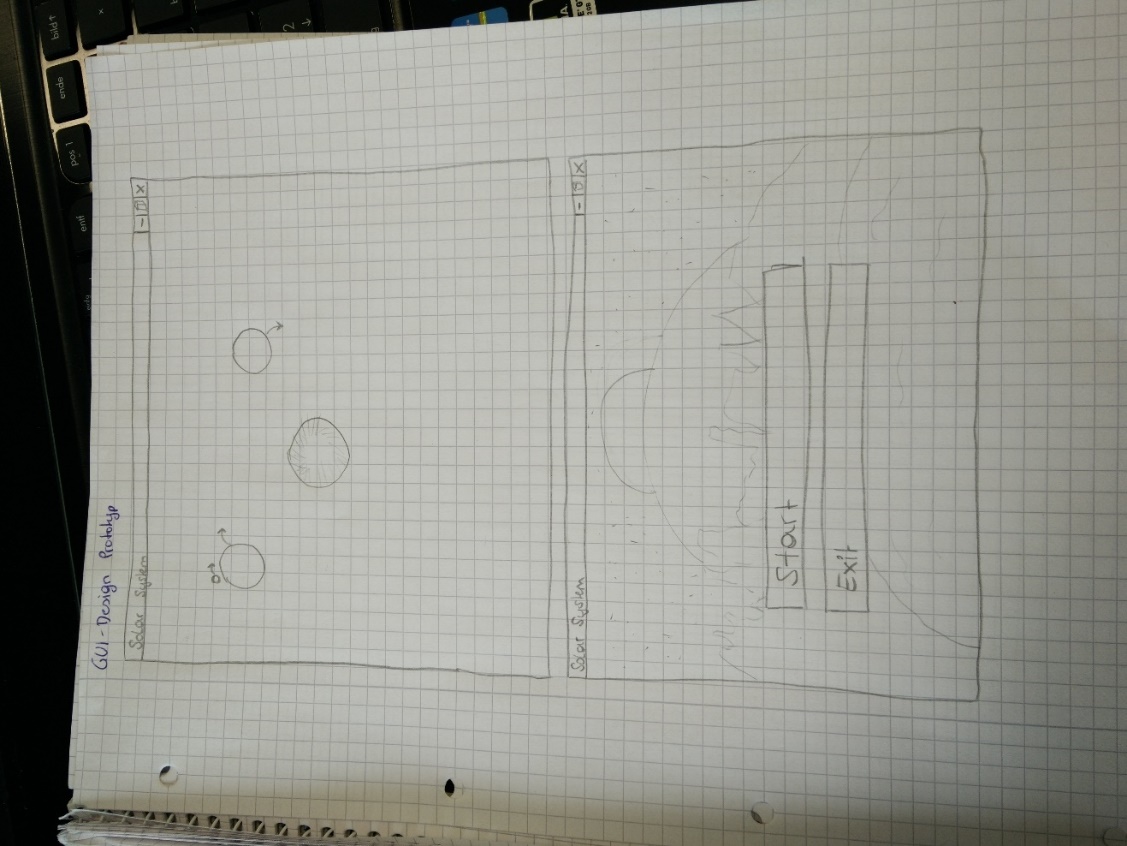
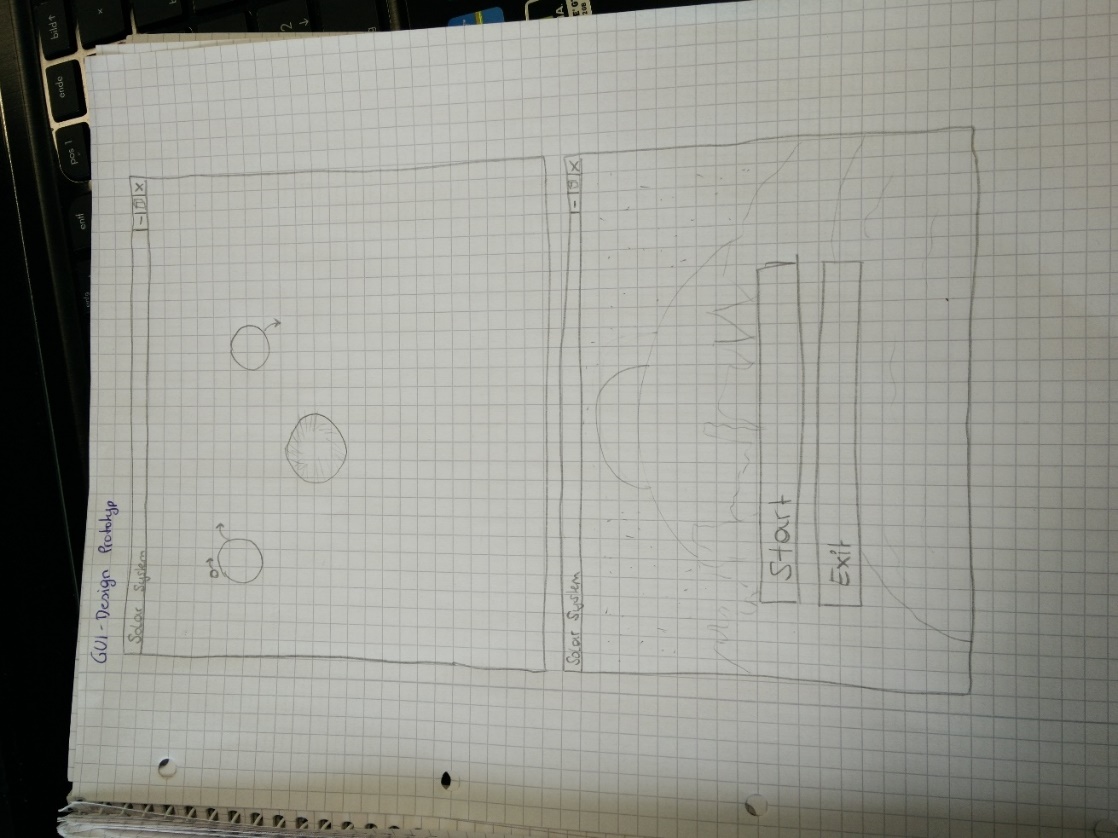
# Zeitaufzeichnung

|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Erwartete Zeit | Tatsächliche Zeit | Zuständigkeit |
| OpenGL lernen | 2:00 h | 3:00h | Dorfinger, Kanyildiz |
| Libraries suchen | 1:00 h | 0:30 h | Dorfinger, Kanyildiz |
| Informieren über Sonnensystem | 1:00 h | 0:30h | Dorfinger,  Kanyildiz |
| Planeten implementieren (mind. 2) | 1:00 h | 1:00h | Kanyildiz |
| Zentralstern implementieren | 1:00 h | 0:30h | Kanyildiz |
| Mond(e) implementieren | 1:00 h |  | Kanyildiz |
| Lichtquellen erstellen | 1:00 h | 1:30h | Dorfinger |
| Texturen erstellen oder aus Internet suchen | 2:00 h | 1:00 h | Dorfinger |
| Texturen laden | 0:30 h | 0:30 h | Dorfinger |
| Texturen auf Objekte legen | 1:00 h |  | Dorfinger |
| Zentralstern dreht sich um sich selbst | 1:00 h | 0:50h | Kanyildiz |
| Planeten drehen sich um eigene Achse und um Zentrealstern | 1:00 h | 2:00h | Kanyildiz |
| Monde drehen sich um sich selbst, um Planeten und um den Zentralstern | 1:00 h |  | Dorfinger |
| Implementieren der Tastensteuerung (Animation und Geschwindigkeit) | 1:00 h | 0:30h | Kanyildiz |
| Einschalten/Ausschalten von Textur und Lichtquelle | 1:00 h | 0:30h | Dorfinger,  Kanyildiz |
| Schatten zu Lichtern | 1:00 h | 1:00h | Kanyildiz |
| Perspektive ändern wen näher/weiter weg | 1:00 h | 0:30h | Kanyildiz |
| 3D-Splashscreen | 2:00 h | 1:30 h | Dorfinger |
| Gesamt | 20:30h | 14:60 h |  |

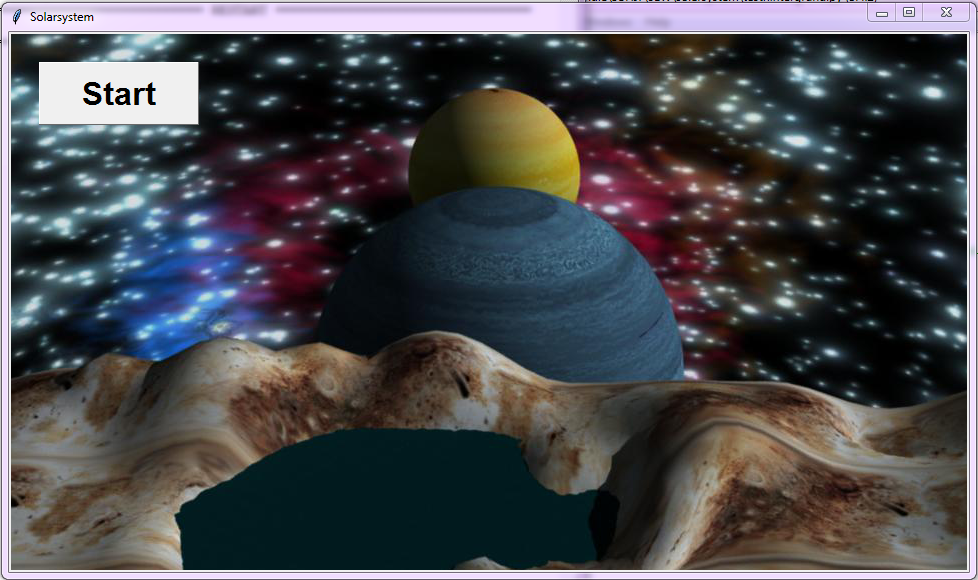
# Designüberlegung

## 3.1 UML

  
  
3.2 GUI-Design Prototyp

## 3.3 Splashscreen Prototyp



## 3.4 Verwendete Libraries

* PyOpenGL  
  Verwendet für: Objekte, Lighting, Texturen
* PyGame  
  Verwendet für: Benutzersteuerung
* Tkinter  
  Verwendet für: Starbildschirm
* Pillow  
  Verwendet für: Texturen

# Funktionalität

## 4.1 Pygame installieren

Man geht auf die Seite <http://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame> und lädt sich die Version Pygame für Python 3.4 runter. Diese installiert man mit:

pip install --no-index --find-links=LocalPathToWheelFile PackageName

## 4.2 PyOpenGL und Pillow installieren

Mit pip install pyopengl installiert man sich die Library PyOpenGL.  
Mit pip install pillow installiert man sich die Library Pillow

## 4.3 Texturen mit Pillow

Mit der Library Pillow kann man Bilder öffnen und anzeigen lassen.   
Man muss die Library einbinden mit from PIL import Image.  
Mit im = Image.open("test.jpg") lädt man die Bilder.  
Mit im.show() kann man sich die Bilder anzeigen lassen.  
Mit print(im.format,im.size,im.mode) bekommt man Infos der Bilder.

Um ein Bild zu öffnen und es anzeigen zu lassen verwendet man folgenden Code:

from PIL import Image

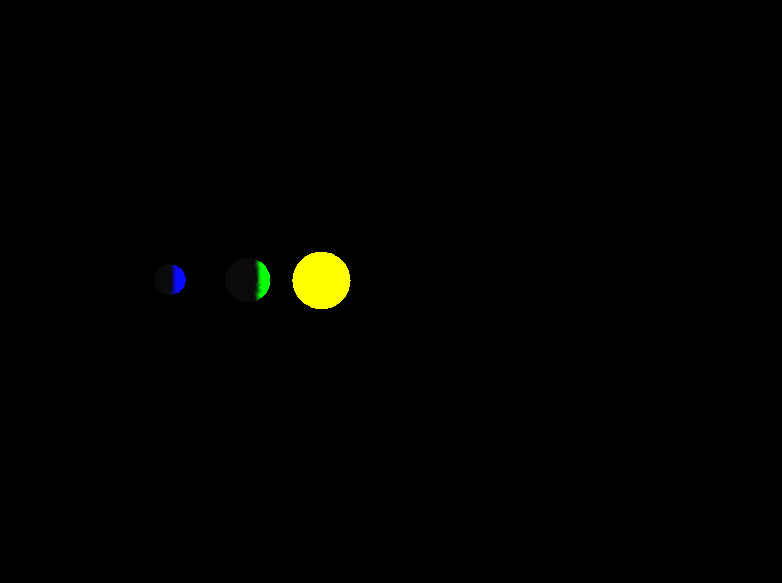
try:  
 im = Image.open("test.jpg") #image laden  
 print(im.format,im.size,im.mode) #infos ueber das bild  
 im.show() #image zeigen

except:  
 print ("Unable to load image")

## 4.4 Sonnensysteminfos

|  |  |  |
| --- | --- | --- |
| Name | Größe (Durchmesser) | Geschwindigkeit um Sonne (km/s) |
| Sonne | 1.390.000 km | - |
| Merkur | 4.900 km | 48 |
| Venus | 12.100 km | 35 |
| Erde | 12.800 km | 29.8 |
| Mars | 6.800 km | 24 |
| Jupiter | 143.000 km | 14 |
| Saturn | 120.500 km | 9.6 |
| Uranus | 51.100 km | 6.8 |
| Neptun | 49.500 km | 5.4 |

## 4.5 Prototyp



## 4.6 Startbildschirm

Der Startbildschirm wurde mit Tkinter implementiert.   
Mit image = tk.PhotoImage(file="splash.png") kann man ein Bild hochladen, das man mit backg.create\_image(0, 0, anchor='nw', image=image) anzeigen lässt.   
Außerdem braucht man einen Button mit dem man das eigentliche Programm startet, diesen macht man mit startb = tk.Button(None, text="Start", bd=1, height=1, width=8, font=('Arial', 24, 'bold')) und lässt ihn mit backg.create\_window(30,30, window=startb, anchor='nw') anzeigen.  
Um in Tktinder ein neues Fenster zu machen, braucht man folgende Befehle:

win = tk.Tk() #erstellt ein Fenster  
win.title("Solarsystem") #gibt dem Fenster einen Titel  
win.geometry('960x540') # Größe des Fensters  
win.mainloop() # Startet das Fenster

# Fortschritt

**Geschafft:**

* Die Sonne dreht sich um seine eigenen Achse
* Es drehen sich mehrere Planeten um die Sonne
* Die Belichtung funktioniert(Die Sonne ist vollkommen beleuchtet während nur die eine Hälfte der Planeten beleuchtet ist)
* Kamerasteuerung funktioniert zu 80%

**Ausstehend:**

* Die Texturierung
* Unterschiedliche Geschwindigkeit beim Rotieren
* Monde
* Steuerung optimieren

# Fehleranalysen

# Quellen

1. Playlist zu Pygame (Python Game Development) von thenewboston, <https://www.youtube.com/watch?v=K5F-aGDIYaM&list=PL6gx4Cwl9DGAjkwJocj7vlc_mFU-4wXJq>   
   gesehen 23.02.2015
2. Unofficial Window Binaries für Python Extension Packages, <http://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame>   
   gesehen 23.02.2015
3. How do I install Python libraries?, <http://stackoverflow.com/questions/21222114/how-do-i-install-python-libraries>   
   gesehen: 23.02.2015
4. Reference, <http://pillow.readthedocs.org/reference/index.html>   
   gesehen 01.03.2015
5. Umlaufgeschwindigkeiten von Planeten, <http://www.astrologie.de/forum/astrologie-allgemein-f1/umlaufgeschwindigkeit-der-planeten-t1933.html>   
   gesehen 02.03.2015
6. Größenvergleich der Planeten unseres Sonnensystems, <http://www.astronomie.de/astronomie-fuer-kinder/interessantes-fuer-lehrer-eltern/in-der-schule/groessenvergleich-der-planeten/>   
   gesehen 02.03.2015