

# A07-Solarsystem

---

Isabella Dall'Oglio, Bernhard Schwertberger

## Inhaltsverzeichnis

1. Aufgabenstellung .....	3
2. Aufwandschätzung .....	4
3. GUI – Design .....	5
Startscreen: .....	5
Nach dem Start: .....	6
Unser Sonnensystem.....	6
4. Design – Evaluation .....	6
5. Probleme.....	7
6. Quellen: .....	7

## 1. 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 Planetenbahnen
- 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 Texturierung ein- und ausgeschaltet werden.
- Schatten: Auch Monde und Planeten werfen Schatten.

Hinweise:

- Ein Objekt kann einfach mittels `glutSolidSphere()` erstellt werden.
- Die Planeten 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.

Viel Erfolg!

## 2. Aufwandschätzung

User Story/Anforderung	Priorität	Verant- wortlicher	Zeit in Minuten	Geschätz t	ist	Sta tus
	Subtask					
Zentraler Stern						
	Modellierung		D	5	5	F
	Texturierung		D	60	180	F
	Positionierung in der Applikation		D	120	60	F
Zumindest 2 Planeten						
	Modellierung		S	60	60	F
	Texturierung		S	60	60	F
	Drehung um Zentralstern(Ellipse)		S	180	120	F
	Berechnung der Geschwindigkeiten		D	60	120	F
	Positionierung in der Applikation		S	120	60	F
	Drehung um eigene Achse		S	30	30	F
Mond						
	Modellierung		D	30	5	F
	Texturierung		D	60	30	F
	Drehung um eigene Achse		S	30	30	F
	Drehung um zugehörigen Planeten		S	120	60	F
	Positionierung in der Applikation		D	120	60	F
Kamerasteuerung						
	Positionsänderung mittels Maus		S	120	160	F
Animationssteuerung						
	Start/Stop		D	30	30	F
	Geschwindigkeitsänderung mittels Pfeiltasten		D	60	30	F
Licht						
	Punktlichtquelle erstellen/positionieren		S	60	60	F
	Steuerung mittels Mausklick(An/Aus)		S	10	20	F
Recherche						
	OpenGL:		D, S			
	Modellierung			120	90	F

	Texturierung		60	180	F
	Buffering		60		
	Rotation		60	30	F
	Translation		60	30	F
	pyGame:	Dall'Oglio, Schwertberger			
	Steuerung		60	60	F
	Darstellung		60	30	F
			30 h 15m	25 h 45 min	

F → Fertig

I → Implementierung

T → Test

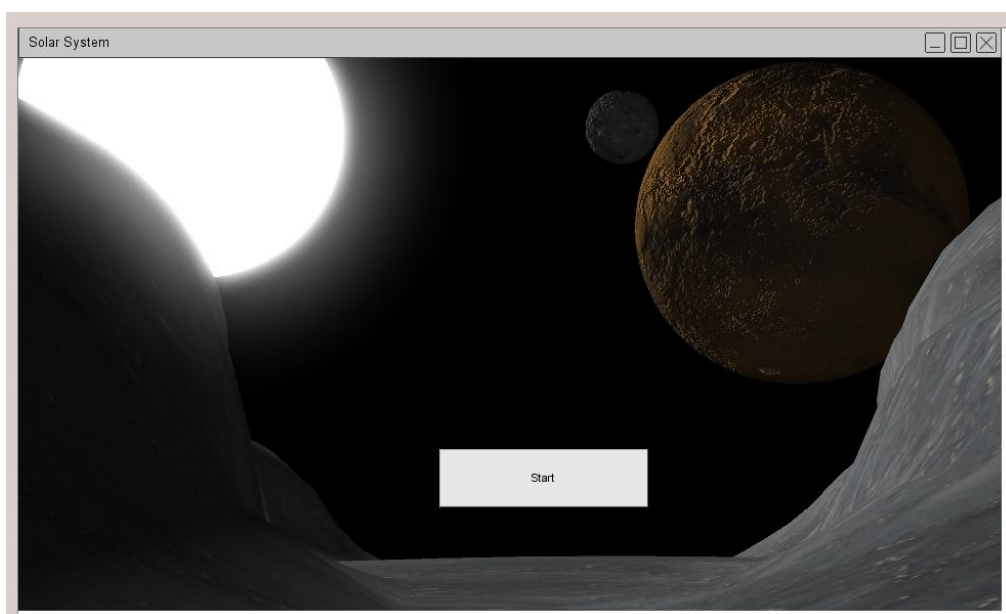
D → Dokumentation

D → Dall'Oglio

S → Schwertberger

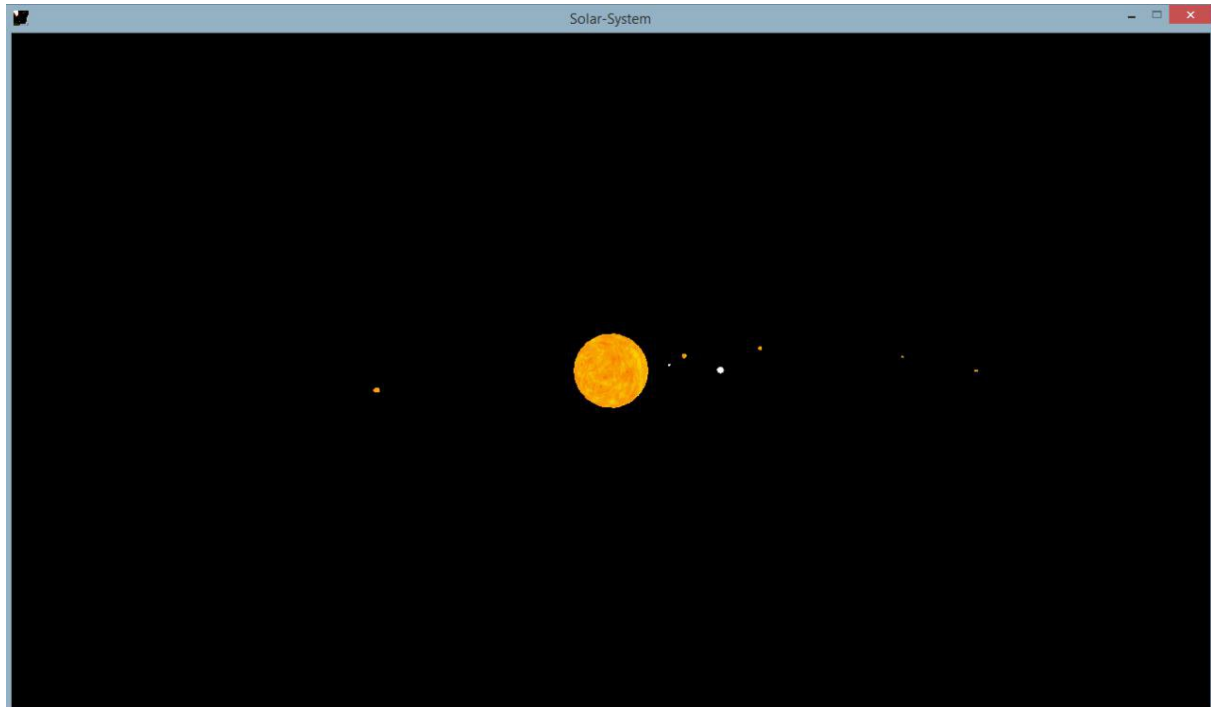
### 3. GUI – Design

Startscreen:



## Nach dem Start:

Unser Sonnensystem



## 4. Design – Evaluation

Libraries

### **Pygame**

Pygame als Container für die in PyOpenGL erstellten Grafiken.

[1]

### **PyOpenGL**

PyOpenGL für die Erstellung der Grafiken (Sonne, Planeten, ...)

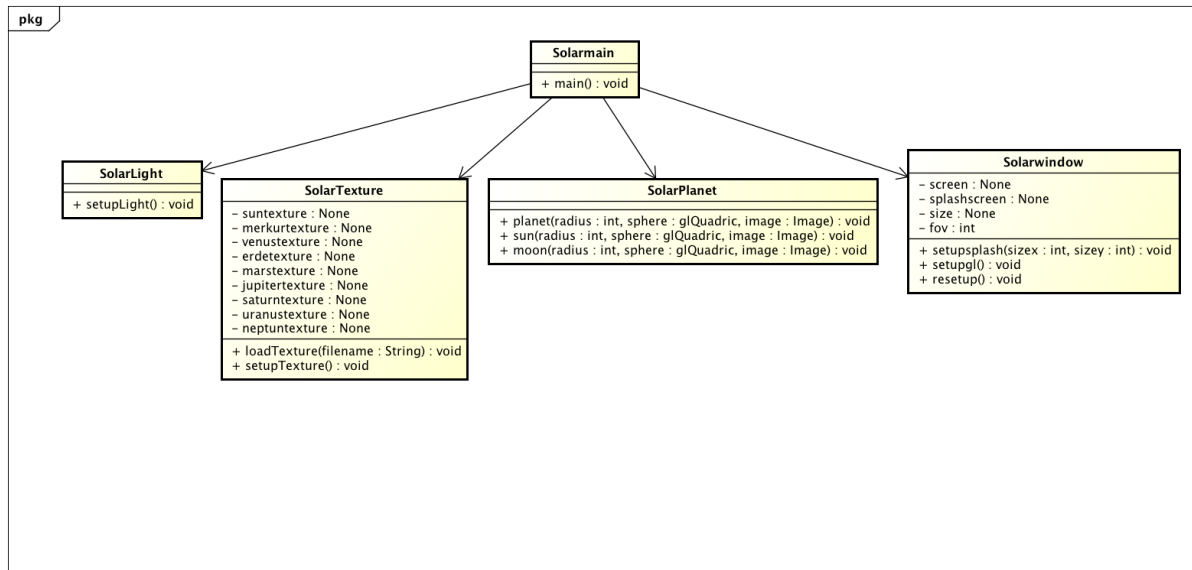
[2]

### **PyGlet**

PyGlet für die Texturierung

[3]

## UML



## 5. Probleme

Sphinx funktioniert nicht.

## 6. Quellen:

- [1] <http://pygame.org> , zuletzt besucht: 29.3.15
- [2] <http://pyopengl.sourceforge.net/> , zuletzt besucht: 29.3.15
- [3] <http://www.pyglet.org/doc/api/index.html> , zuletzt besucht: 29.3.15