Marlene Dorfinger, Kanyildiz

Visit our solar system …

Inhalt

[1. Aufgabenstellung 2](#_Toc412458753)

[2. Zeitaufzeichnung 3](#_Toc412458754)

[3. Designüberlegung 4](#_Toc412458755)

[1. UML 4](#_Toc412458756)

[4. Funktionalität 5](#_Toc412458757)

[1. Pygame installieren 5](#_Toc412458758)

[2. PyOpenGL installieren 5](#_Toc412458759)

[5. Fehleranalysen 5](#_Toc412458760)

[Quellen 5](#_Toc412458761)

# 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

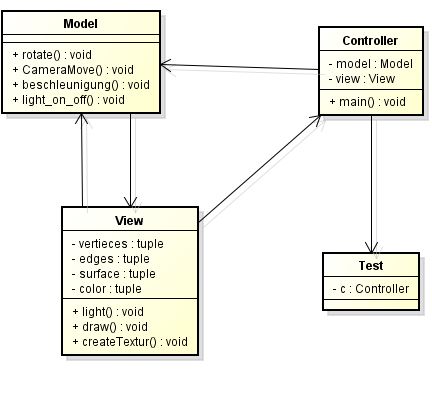
Viel Erfolg!

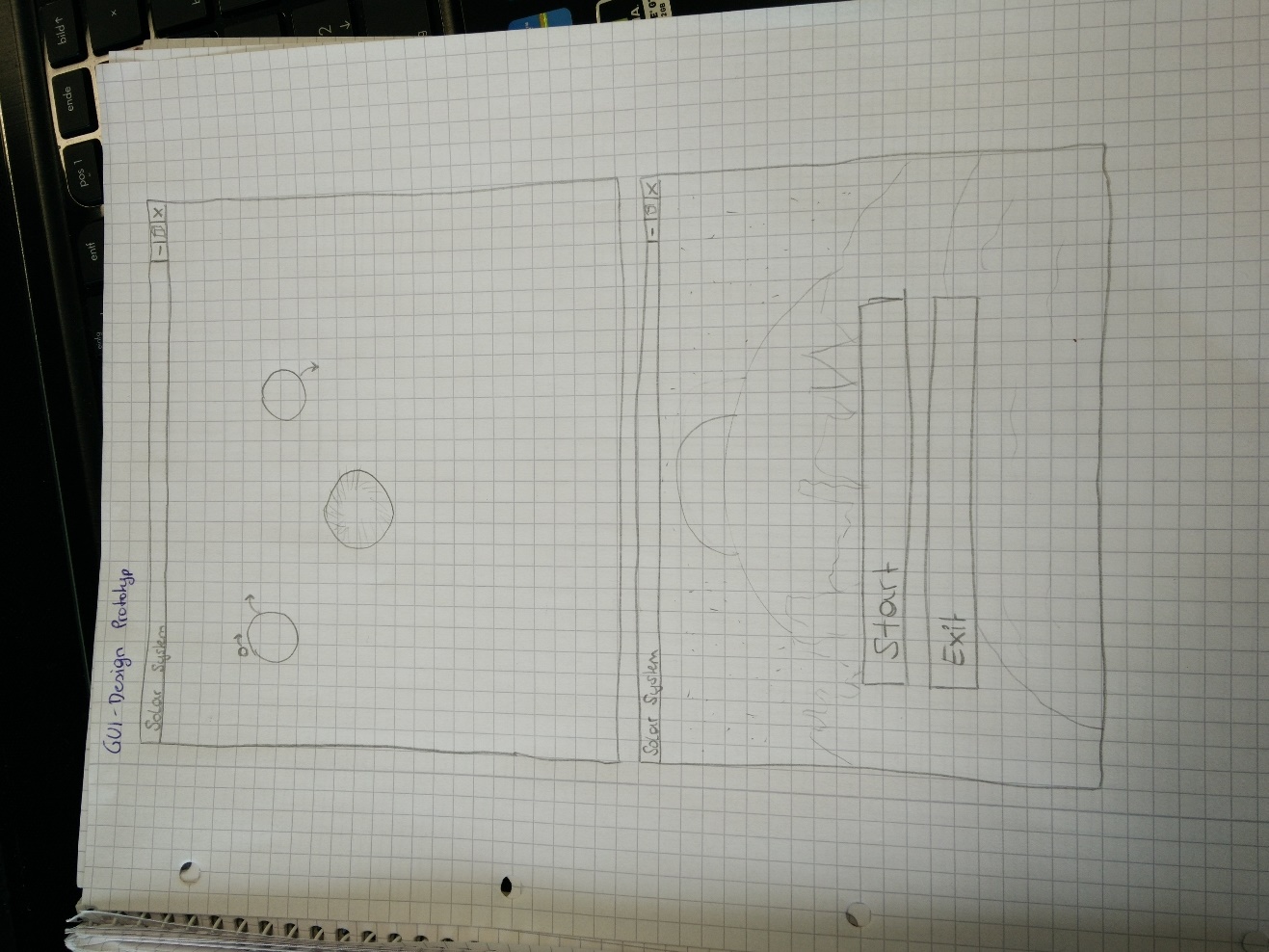
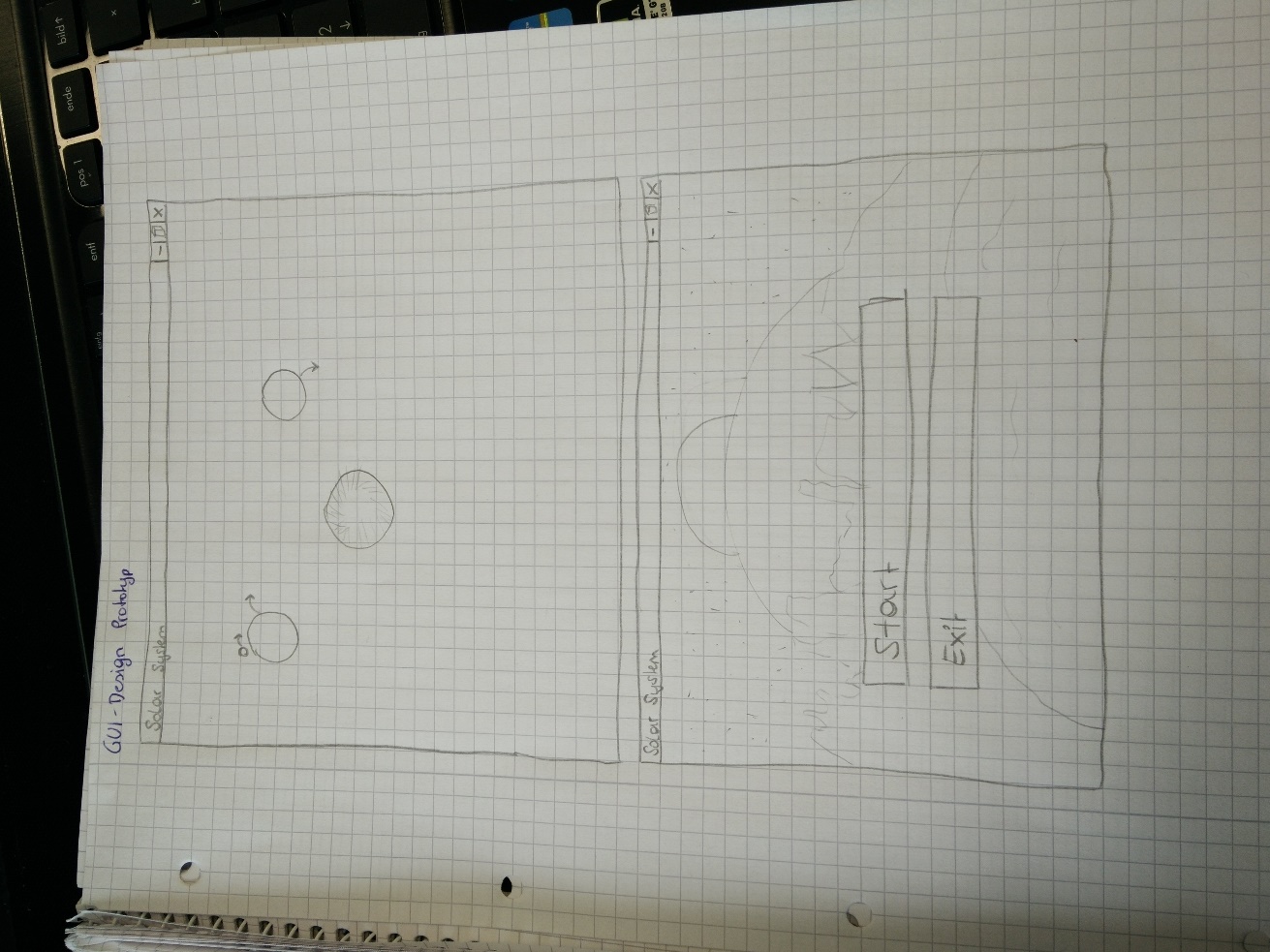
# Zeitaufzeichnung

|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Erwartete Zeit | Tatsächliche Zeit | Zuständigkeit |
| OpenGL lernen | 2:00 h |  | Dorfinger, Kanyildiz |
| Librarys suchen | 1:00 h |  | Dorfinger, Kanyildiz |
| Informieren über Sonnensystem | 1:00 h |  | Dorfinger |
| Planeten implementieren (mind. 2) | 1:00 h |  | Kanyildiz |
| Zentralstern implementieren | 1:00 h |  | Kanyildiz |
| Mond(e) implementieren | 1:00 h |  | Kanyildiz |
| Lichtquellen erstellen | 1:00 h |  | Dorfinger |
| Texturen erstellen oder aus Internet suchen | 2:00 h |  | Dorfinger |
| Texturen auf Objekte legen | 1:00 h |  | Dorfinger, Kanyildiz |
| Zentralstern dreht sich um sich selbst | 1:00 h |  | Kanyildiz |
| Planeten drehen sich um eigene Achse und um Zentrealstern | 1:00 h |  | Dorfinger |
| 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 |  | Kanyildiz |
| Einschalten/Ausschalten von Textur und Lichtquelle | 1:00 h |  | Kanyildiz |
| Weitere Planeten, Asteoriden, Galaxien, … | 1:00 h |  | Dorfinger, Kanyildiz |
| Schatten zu Lichtern | 1:00 h |  | Dorfinger, Kanyildiz |
| Perspektive ändern wen näher/weiter weg | 1:00 h |  | Kanyildiz |
| 3D-Splashscreen | 2:00 h |  | Dorfinger |
| Gesamt | 20:00h |  |  |

# Designüberlegung

## 1. UML

  
2. GUI-Design Prototyp

# Funktionalität

## 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

## 2. PyOpenGL installieren

Mit pip install pyopengl installiert man sich die Library PyOpenGL.

# 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