

Visit our solar system

Daniel Bracher, Martin Suschny

Inhaltsverzeichnis

[1 Aufgabenstellung 1](#_Toc413666411)

[1.1 Zusätzliche Information 1](#_Toc413666412)

[2 Zeitaufzeichnung 2](#_Toc413666413)

[3 Design-Entwurf 3](#_Toc413666414)

[3.1 UML-Diagramm 3](#_Toc413666415)

[3.2 GUI 3](#_Toc413666416)

[4 Arbeitsvorgang 4](#_Toc413666417)

[4.1 Bracher 4](#_Toc413666418)

[4.2 Suschny 4](#_Toc413666419)

[5 Testdokumentation 5](#_Toc413666420)

[Quellen 6](#_Toc413666421)

TODO: Libraries beschreiben

# 1 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.

## 1.1 Zusätzliche Information

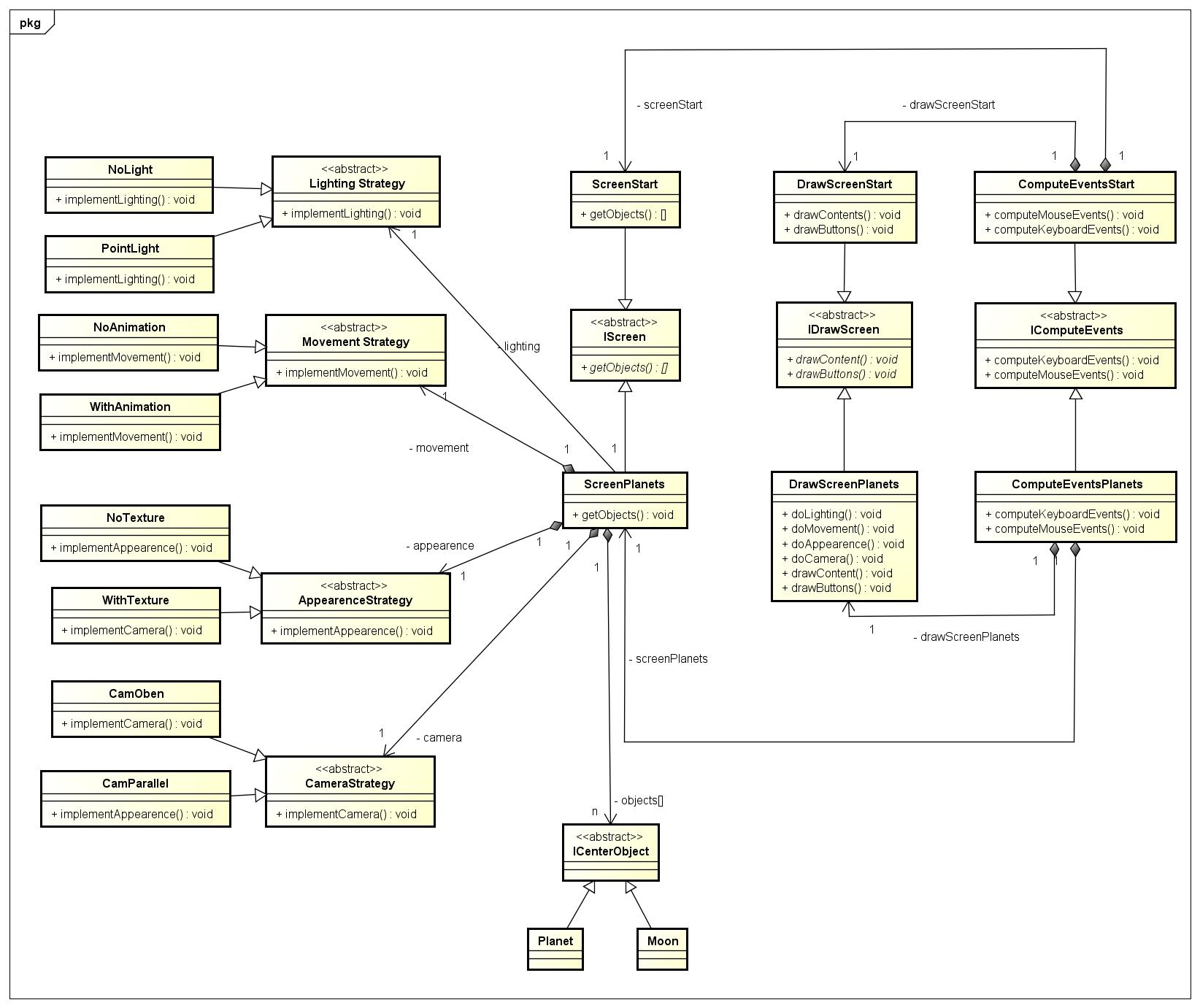
* Ein Objekt kann einfach mittels glutSolidSphere() erstellt werden.
* Die Planten werden mittels Modelkommandos bewegt: glRotate(), glTranslate()
* Die Kameraposition wird mittels gluLookAt() gesetzt
* Bedenken Sie bei der Perspektive, dass entfernte Objekte kleiner - nahe entsprechende größer darzustellen 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.

# 

# 2 Zeitaufzeichnung

# 3 Design-Entwurf

## 3.1 UML-Diagramm



## 3.2 GUI

## 3.3 Verwendete Libraries

# 4 Arbeitsvorgang

## 4.1 Bracher

1.3.2015 -> UML-Diagramm entworfen

3.3.2015 -> UML Diagramm aktualisiert (ICenterObject hinzugefügt)

8.3.2015 -> UML Diagramm aktualisiert (einzelne Relationen hinzugefügt)

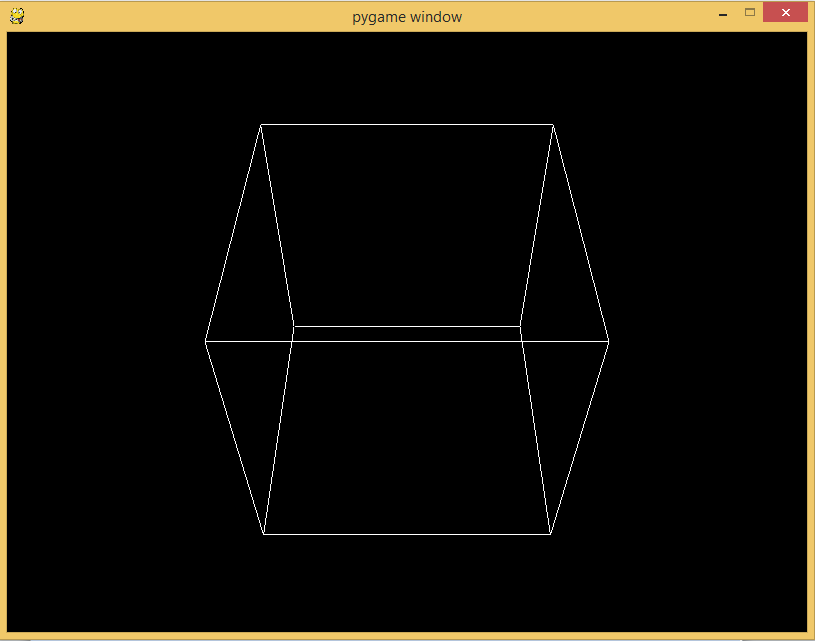
Klassen und Interfaces nach UML-Diagramm erstellt

Mittels PyQt GUI fuer die Planeten erstellt

PyQt-Code in Programm eingefügt

Problem: Controller der Planeten kann nur von einem Interface erben, nicht von zwei (Qt- und Controller-Interface)  
Loesungsansatz1: Problem koennte an gleicher Benennung der Funktionen liegen

9.3.2015-> Implementierung des im Tutorial angefuehrten Codes (GUI + sich drehender Wuerfel) [1]



10.3.2015-> Dynamisches Zeichnen aller in der Model-Klasse erstellten Objekte

11.3.2015 -> BIG RELEASE  
  
Mehrstufiges Abfragen der Events implementiert

def computeKeyboardEvents(self, event):

"""

Verarbeitet die Keyboard-Events

:return: Nichts

"""

#if event.key == pygame.K\_LEFT:

def computeMouseEvents(self, event):

"""

Verarbeitet die Mouse-Events

:return: Nichts

"""

if event.type == pygame.QUIT: #Falls das Fenster geschlossen werden soll

pygame.quit()

quit()

elif event.type == pygame.MOUSEBUTTONDOWN: #Falls etwas mit der Maus gedrückt wurde

if event.button == 1: #Falls linke Maustaste gedrückt wurde

self.screenContent.changeMovement() #Movement-Strategie aendern

elif event.button == 3: #Falls rechts Maustaste gedrückt wurde

self.screenContent.changeLighting()

elif event.button == 4: #Falls Maus-Rad nach vorne gescrollt wird

glTranslatef(0.0,0.0,1)

elif event.button == 5: #Falls Maus-Rad nach hinten gescrollt wird

glTranslatef(0.0,0.0,-1)

Aendern der Movement-Strategie und Lighting-Strategie während der Laufzeit implementiert

def changeMovement(self):

#Falls Movement derzeit NoAnimation-Strategie implementiert

if isinstance(self. movement, NoAnimation):

#Auf WithAnimation-Strategie aendern

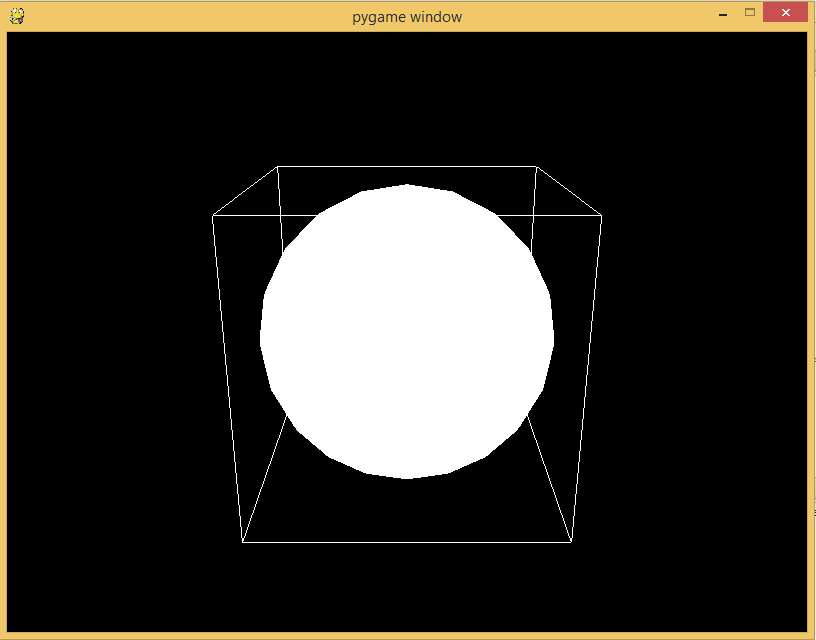
self.movement = WithAnimation()

#Falls WithAnimation-Strategie implementiert

elif isinstance(self.movement, WithAnimation):

#Auf NoAnimation-Strategie aendern

self.movement = NoAnimation()  
Sonne als Kugel implementiert



## 4.2 Suschny

# 5 Testdokumentation

# Quellen

1. Online-Tutorial; <https://www.youtube.com/watch?v=R4n4NyDG2hI>; zuletzt besucht: 9.3.2015