22.11.2015

5BHITT

Stefan Erceg & Martin Kritzl

Solarsystem

SEW – Python3D-Anwendung

Inhalt

[1 Angabe 2](#_Toc435982167)

[2 Requirementanalyse 3](#_Toc435982168)

[2.1 funktionale Anforderungen 3](#_Toc435982169)

[2.2 nicht funktionale Anforderungen 3](#_Toc435982170)

[3 Evaluierung der grafischen Tools in Python 4](#_Toc435982171)

[4 Designüberlegung 5](#_Toc435982172)

[4.1 grafische Übersicht 5](#_Toc435982173)

[4.2 Abbildung des UML-Diagramms 6](#_Toc435982174)

[4.3 Überlegungen zur Struktur 7](#_Toc435982175)

[4.3.1 Klasse „Luminary“ 7](#_Toc435982176)

[4.3.2 Klasse „RuntimeHandler“ 7](#_Toc435982177)

[4.3.3 Klasse „Camera“ 7](#_Toc435982178)

[4.3.4 Klasse „EventHandler“ 7](#_Toc435982179)

[4.3.5 Klasse „Solarsystem“ 7](#_Toc435982180)

[5 Arbeitsdurchführung 8](#_Toc435982181)

[6 Lessons learned 9](#_Toc435982182)

# Angabe

# Requirementanalyse

## funktionale Anforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Person | Schätzung  (min) | Tatsächlich  (min) | Erledigt |
| **Himmelskörper zur GUI hinzufügen** | **Erceg & Kritzl** | **60** | **80** | **** |
| **Himmelskörper löschen** | **Kritzl** | **20** | **10** | **** |
| **Textierung ein-/ausschalten** | **Kritzl** | **25** | **15** | **** |
| **Kameraposition anpassen** | **Erceg & Kritzl** | **50** | **65** | **** |
| **Geschwindigkeit drosseln** | **Kritzl** | **40** | **30** | **** |
| **Geschwindigkeit beschleunigen** | **Kritzl** | **20** | **15** | **** |
| **Punktlichtquelle ein-/ausschalten** | **Erceg** | **30** |  |  |
| **Schatten zu Himmelskörpern hinzufügen** | **Erceg** | **40** |  |  |

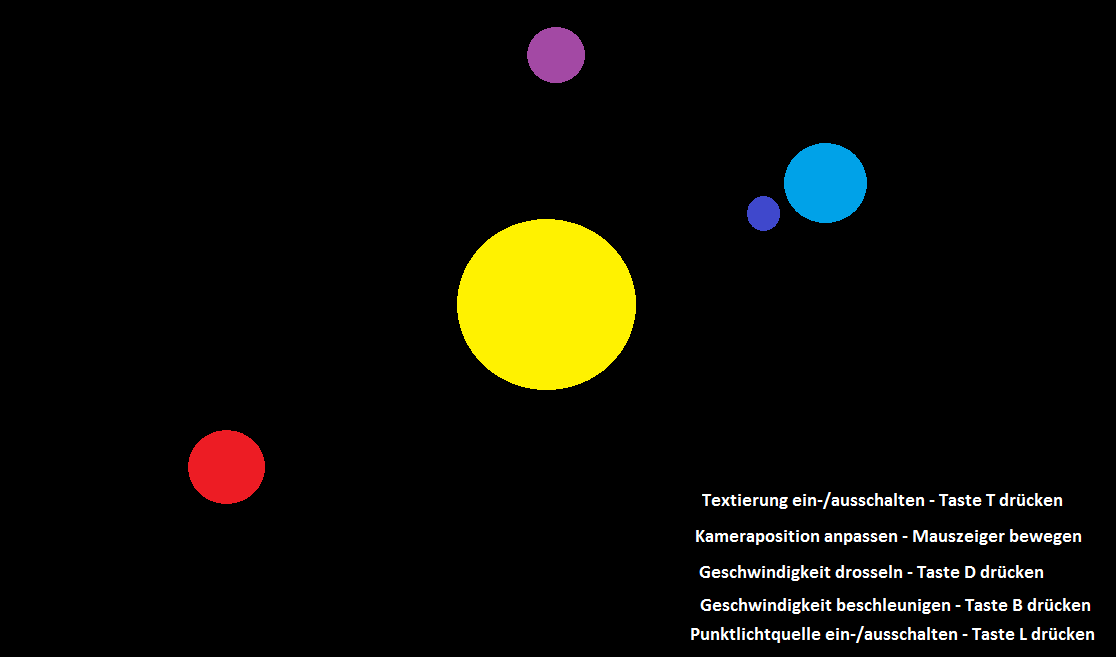
## nicht funktionale Anforderungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Arbeitspaket | Person | Schätzung  (min) | Tatsächlich  (min) | Erledigt |
| **3 verschiedene GUI-Tools in Python evaluieren** | **Erceg & Kritzl** | **180** | **165** | **** |
| **Grafische Übersicht erstellen** | **Erceg** | **15** | **20** | **** |
| **UML-Diagramm erzeugen** | **Erceg & Kritzl** | **60** | **75** | **** |
| **Struktur des UML-Diagramms beschreiben** | **Erceg** | **30** | **40** | **** |

# Evaluierung der grafischen Tools in Python

# Designüberlegung

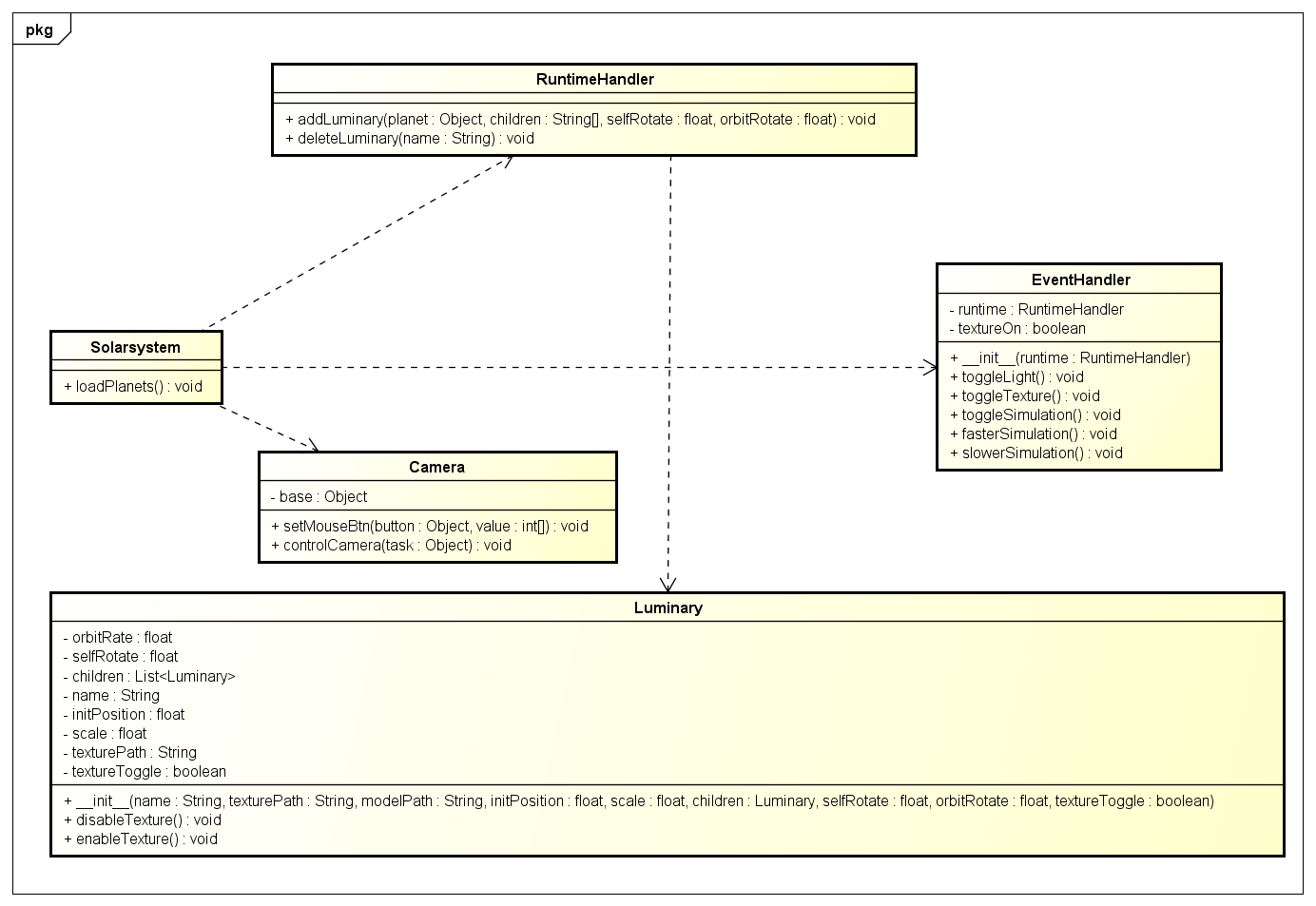
## grafische Übersicht



Wie an der grafischen Übersicht zu erkennen ist, haben wir uns überlegt, in der Mitte die Sonne abzubilden. Um die Sonne herum rotieren die Planeten (vorgenommen haben wir uns 3 zu implementieren) und zur Erde ebenfalls der Mond. Im unteren rechten Abschnitt der grafischen Oberfläche soll eine Legende angezeigt werden, um auf Events nach bestimmten Tastendrücken bzw. nach Bewegen des Mauszeigers hinzuweisen.

## Abbildung des UML-Diagramms

Das UML-Diagramm wurde mit dem Programm „Astah“ erstellt.



## Überlegungen zur Struktur

Wir haben uns überlegt, unser Programm in 5 Klassen aufzuteilen:

### Klasse „Luminary“

Diese Klasse nimmt im Konstruktor Parameter zum Laden der jeweiligen Texturen und Körper zu den Himmelskörpern (engl.: „Luminary“) entgegen. Ebenfalls wird hier deren Startposition und Größe gesetzt. Zum Ein- bzw. Ausblenden der Texturen existieren ebenfalls Methoden. Zurückgegeben wird zum Schluss ein Himmelskörper.

### Klasse „RuntimeHandler“

Der „RuntimeHandler“ existiert, um einen bestimmten Himmelskörper zum Gesamtsystem bzw. zur grafischen Oberfläche hinzuzufügen. Als Parameter können ebenfalls Kinder definiert werden, wie z.B., dass der Mond ein Kind von der Erde ist. Die jeweilige Selbstrotation bzw. die Rotation um den Mittelpunkt, nämlich der Sonne, werden ebenfalls angegeben. Eine Methode zum Entfernen eines bestimmten Himmelskörpers (als Parameter wird dessen Name angegeben) steht ebenfalls zur Verfügung.

### Klasse „Camera“

Die Klasse „Camera“ ist für das Verwalten der aktuellen Kameraposition zuständig. Diese Klasse wird dann vom „EventHandler“ aufgerufen, welcher im nächsten Unterkapitel beschrieben wird.

### Klasse „EventHandler“

Wie der Name der Klasse bereits sagt, werden hier bestimmte Events (Tastendrücke und/oder Mausklicks) entgegengenommen. Nachdem eine bestimmte Taste gedrückt bzw. die Maus bewegt wurde, werden entsprechende Methoden aufgerufen (z.B.: beim Drücken der Taste „T“ wird die Textierung ein- bzw. ausgeschaltet).

### Klasse „Solarsystem“

Diese Klasse stellt die Main-Klasse des Programms dar. Hier werden die jeweiligen Himmelskörper initialisiert.

# Arbeitsdurchführung

# Lessons learned