Thomas Stedronsky, Simon Wortha

tstedronsky@student.tgm.ac.at, swortha@student.tgm.ac.at

Solarsystem

SEW-TGM

Inhalt

[1. Aufgabenstellung 2](#_Toc436122662)

[2. Projektbeschreibung 2](#_Toc436122663)

[2.1. Anforderungen 2](#_Toc436122664)

[2.2. Team 2](#_Toc436122665)

[3. Evaluierung 2](#_Toc436122666)

[4. GUI-Skizzen 3](#_Toc436122667)

[5. Bedienkonzept 3](#_Toc436122668)

[6. Technische Dokumentation 3](#_Toc436122669)

[7. Bedienungsanleitung 3](#_Toc436122670)

## Aufgabenstellung

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.

## Projektbeschreibung

Es soll ein Solarsystem mit einer Sonne und mindestens 2 Planeten implementiert werden. Außerdem soll mindestens ein Planet einen Mond haben. Diese Planeten und Monde bewegen sich in realistischer Umlaufbahnen im Solarsystem.

## Anforderungen

Es muss eine IDE zur Implementierung von Python vorhanden sein, in unserem Fall verwenden wir PyCharm 4.5. Wir verwenden die Frameworks PyGlet und PyWavefront, diese müssen korrekt installiert sein.

## Team

Unser Team besteht aus 2 Mitgliedern.

* Thomas Stedronsky
* Simon Wortha

## Evaluierung

* PyGame
* PyGlet

Es wurden alle Frameworks auf Dokumentation, Community, Prototypen und Bedienung getestet. In allen diesen Punkten hat PyGlet gut abgeschnitten. Durch die besonders große Dokumentation fällt es uns leicht sich in das Framework einzulesen.

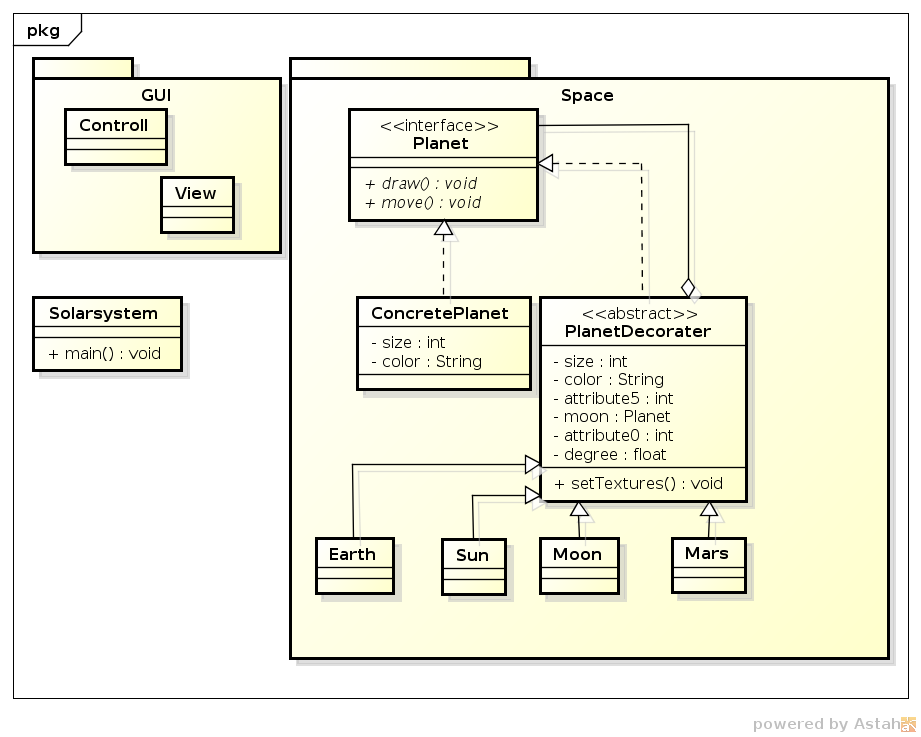
Außerdem haben wir ein Framework zur Dekoration für die Planeten gefunden, dies heißt PyWavefront. Durch dieses Framework ist es leicht Texturen(sofern .obj Files verfügbar sind) an Planeten weiter zu geben. PyGlet und PyWavefront arbeiten sehr eng zusammen, deswegen sprach ein weiterer Punkt für PyGlet. Außerdem hat uns das Example von PyGlet mehr überzeugt und war leichter zu verstehen als das Example von PyGame.

Wir haben uns für **PyGlet** entschieden.

## GUI-Skizzen

## Bedienkonzept

## Technische Dokumentation



## Bedienungsanleitung