Thomas Stedronsky, Simon Wortha

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

Solarsystem

SEW-TGM

Inhalt

[1. Aufgabenstellung 2](#_Toc436134560)

[2. Projektbeschreibung 2](#_Toc436134561)

[2.1. Anforderungen 2](#_Toc436134562)

[2.2. Team 2](#_Toc436134563)

[3. Evaluierung 3](#_Toc436134564)

[3.1. Dokumentation 3](#_Toc436134565)

[3.2. Installation 3](#_Toc436134566)

[3.3. Community 3](#_Toc436134567)

[3.4. Prototyp 3](#_Toc436134568)

[3.5. Conclusio 3](#_Toc436134569)

[4. GUI-Skizzen 4](#_Toc436134570)

[5. Bedienkonzept 4](#_Toc436134571)

[6. Technische Dokumentation 4](#_Toc436134572)

[7. Bedienungsanleitung 4](#_Toc436134573)

## 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
* Panda3D

Es wurden alle Frameworks auf Dokumentation, Community, Prototypen und Installation getestet.

## Dokumentation

PyGlet: 1

PyGame: 0

Panda3D: 1

Gute Dokumentation bei Panda3D und PyGlet

## Installation

Die Installation war bei PyGlet sehr einfach und war ohne große Probleme zum Installieren. PyGame konnte auf einer Linux-Distribution nicht installiert werden.

Panda3D war sehr leicht zu installieren. Bei der Installation wurden außerdem reichlich Examples beigefügt um Panda gut zu verstehen.

PyGlet: 1

PyGame: 0

Panda3D: 2

## Community

Die Community ist bei allen ca. gleich, ähnliche Stackoverflow Fragen, …

PyGlet: 2

PyGame: 1

Panda3D: 3

## Prototyp

Der Prototyp wurde nur bei PyGlet zum Laufen gebracht, dieser Prototyp war sehr hilfreich für das Projekt. Bei PyGame konnte kein vollständiges Example zum Laufen gebracht. Bei Panda3D wurden diverse Examples zum Laufen gebracht, durch diese ist Panda leicht verständlich.

PyGlet: 3

PyGame: 1

Panda3D: 4

## Conclusio

PyGlet: 3

PyGame: 1

Panda3D: 4

Durch das Ergebnis haben wir uns für **Panda3D** entschieden.

## GUI-Skizzen

## Bedienkonzept

## Technische Dokumentation

## Bedienungsanleitung