Solarsystem

Dolacek | Özer

18.04.2015

Inhalt

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.

Arbeitsschritte

1. Gruppenbildung

2. Anforderungsanalyse (Funktionelle und nicht funktionelle Arbeitsanforderungen)
    a. Tabellarische Auflistung
    b. Rücksprache mit Auftraggeber

3. Designüberlegung
    a. Prototyp- Evaluation
    b. SW-Design
    c. GUI-Design
    d. Rücksprache mit Auftraggeber
    e. UAT- Überlegung

4. Implementierung & Tests

5. Abgabemodalitäten klären & Abgabe

Anforderungsanalyse

Anforderung	Priorität	Verantwortlicher	Zeit in min		Status				
			SOLL	IST	Design	Implentiert	Test	Dokumentiert	Fertig
Zentraler Stern	High	Dolacek	60	180	X	X	X	X	X
Planet 1	High	Dolacek	70	120	X	X	X	X	X
Planet 2	High	Dolacek	70	120	X	X	X	X	X
Drehung P1	High	Dolacek	120	240	X	X	X	X	X
Drehung P2	High	Dolacek	120	240	X	X	X	X	X
Mond	High	Dolacek	60	120	X	X	X	X	X
Drehung Mond	High	Dolacek	100	120	X	X	X	X	X
Textur	High	Dolacek	80	210	X	X	X	X	X
Licht&Schatten	High	Dolacek	120	120	X	X	X	X	X
Kameraposition	Middle	Dolacek	120	180	X	X		X	
Geschwindigkeit der Animation	Middle	Dolacek	110	90	X	X	X	X	X
Testen	High	Özer, Dolacek	180	260	X	~		X	~
Kreativität(Hintergründe, usw)	Low	Özer, Dolacek	120	70	X	X	NICHT TESTBAR	X	X
Performance	Middle	Özer	90	30	X	X	X	X	X
Splashscreen	Middle	Özer	80	90	X	X	X	X	X
PyGame lernen	High	Özer	120	100	X	X	NICHT TESTBAR	X	X
Libaries finden	High	Özer	60	90	X	X	NICHT TESTBAR	X	X
Gesamt			**28,0**	**39,7**					
			Stunden	**Stunden**					

Designüberlegung

Sonnensystem(Relationen)

Die Relationen des Sonnensystems, siehe Link **[5]**

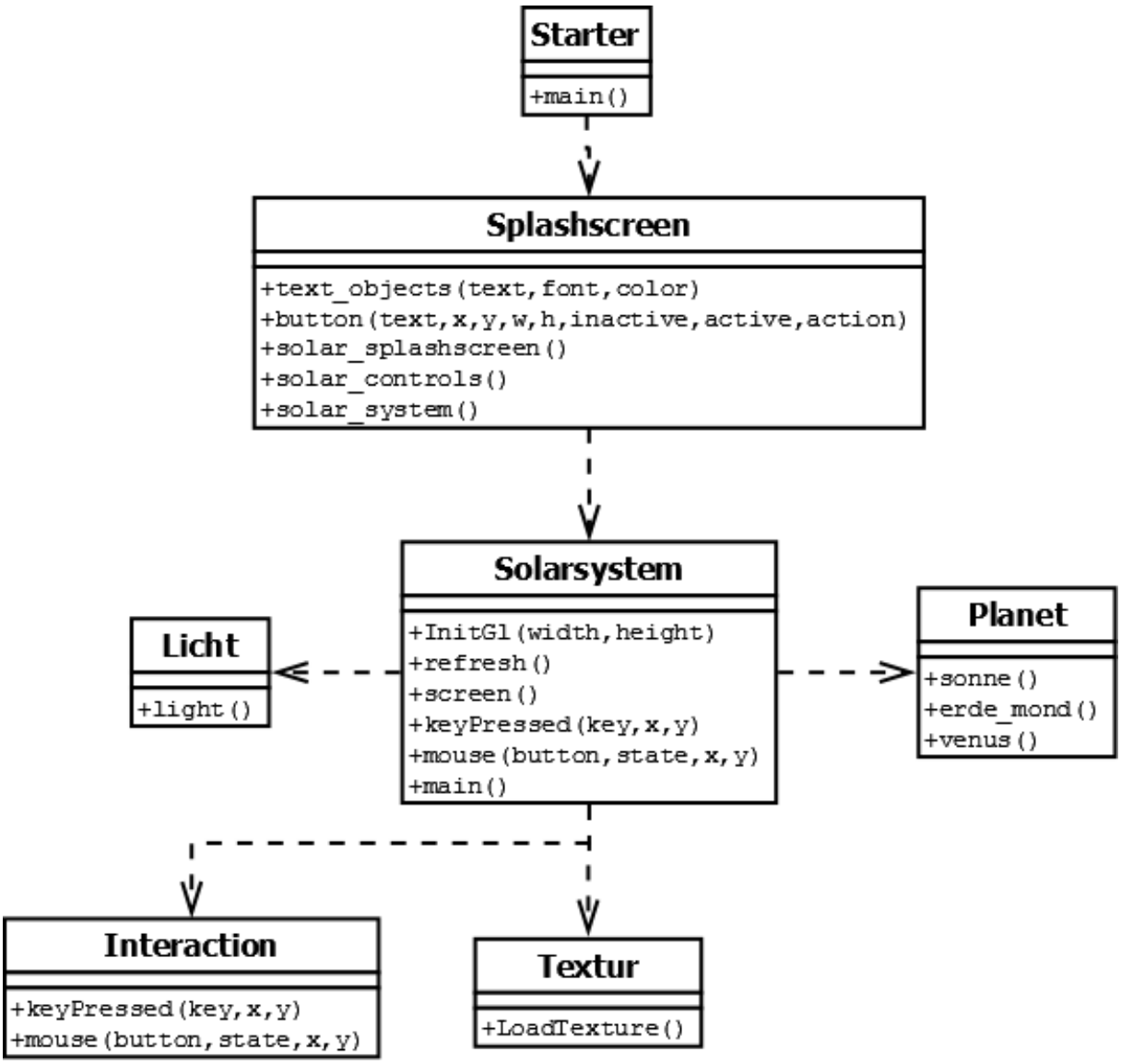
Es wurden die Planeten Erde und Venus gewählt.

Planet	Durchmesser	Umlaufzeit	Python Durchmesser	Python Umlaufzeit
Erde	12 800 km	365 Tage	0.1	0.03
Venus	12 100 km	224 Tage	0.1	0.02

```
#Erde
planet.DrawGLScene_erde(self, 0.1, 0.8, 0.0, 0.0,0.03, self.erde)
planet.DrawGLScene_erde(self, 0.03, 1, 0.0, 0.0, 0, self.mond)
```

```
#Venus
Planet.DrawGlScene_venus(self, 0.1, 1.5, -0.3, 0.5, 0.02, self.rosa)
```

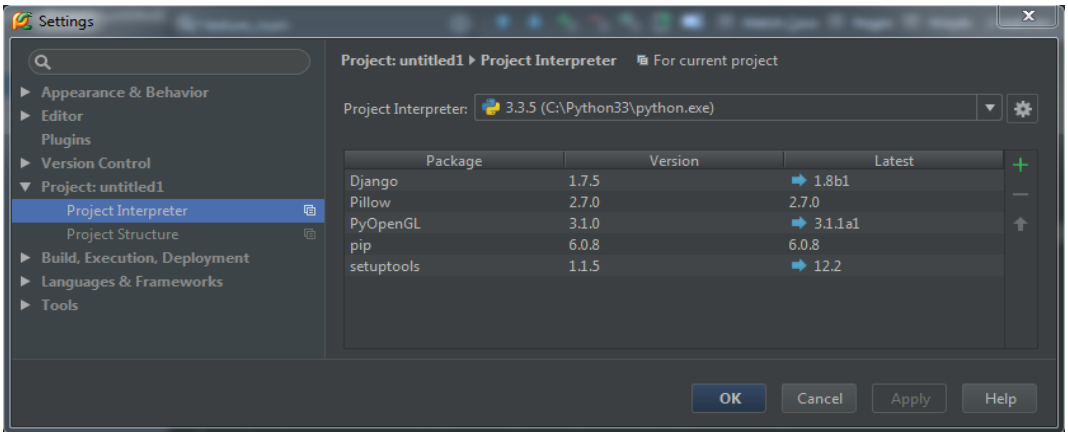
SW Design


Starter
+main()
Splashscreen
+text_objects(text,font,color)
+button(text,x,y,w,h,inactive,active,action)
+solar_splashscreen()
+solar_controls()
+solar_system()
Solarsystem
+InitGl(width,height)
+refresh()
+screen()
+keyPressed(key,x,y)
+mouse(button,state,x,y)
+main()
Licht
+light()
Planet
+sonne()
+erde_mond()
+venus()
Interaction
+keyPressed(key,x,y)
+mouse(button,state,x,y)
Textur
+LoadTexture()

Libaries

- Pygame **[3]**
- PyOpenGl
- Numpy
- Pillow

PyCharm -> File -> Settings -> Project Interpreter



Mögliche Fehler:

1. Für Glut wird eine zusätzliche .dll Datei aus dem Internet benötigt. Diese muss dann in dem Package eingebunden werden. Es gibt eine 32 und 64 Bit Version! Wenn Python 64bit installiert ist, muss die 64bit Version heruntergeladen werden.

   http://sourceforge.net/projects/freeglut/

2. Bei dem Import muss man darauf achten, dass man die unterliegenden Packages richtig anspricht. D.h. Groß- und Kleinschreibung beachten.

GUI Design

Splashscreen:

Der Menüpunkt "About Us" wurde durch Controls ersetzt. Dem User wird beim Anklicken auf Controls eine Übersicht der Steuerung des Programms angezeigt.

Entwicklung

Planeten

```
gluSphere(quadratic,radius, 32,32)
```

Texturen einbinden

für beliebige Objekte:

```
glBindTexture(GL_TEXTURE_2D,txt)
quadratic = gluNewQuadric()
gluQuadricNormals(quadratic, GLU_SMOOTH)
gluQuadricTexture(quadratic, GL_TRUE)
```

Textur ein/aus

Die Textur wird mit der rechten Maustaste ein und ausgeschalten.

```
if (button==GLUT_RIGHT_BUTTON):
    if state == GLUT_DOWN:
```

Die Textur wird dann mit glEnable(GL_TEXTURE_2D) eingeschalten und mit glDisable(GL_TEXTURE_2D) ausgeschalten.

Licht ein/ aus

Die Licht wird mit der linken Maustaste ein und ausgeschalten.
Die Taste wird wie folg angesprochen:

```
if (button==GLUT_LEFT_BUTTON):
    if state == GLUT_DOWN:
```

[7]

Das Licht wird dann mit glEnable(GL_LIGHTING) eingeschalten und mit glDisable(GL_LIGHTING) ausgeschalten.

Kamera

Die Kamera wird mit gluLookAt definiert.
Die Kamera hat in diesem Programm eine Entfernung von 4 und der Vektor vom Objekt zum Aug ist 1/1/1.

```
gluLookAt(0, 0, 4, 0, 0, 0, 1, 1, 1)
```

Kameraposition

Die Kameraposition wird mit der Taste C geändert.
Es gibt nur eine Überkopf-Ansicht.

```
if key == b'c':
```

[8]

Die neue Kameraposition hat folgende Werte.

```
gluLookAt(0, 4, 4, 0, 0, 0, 1, 1, 1)
```

Geschwindigkeit schneller/ langsamer/ stoppen

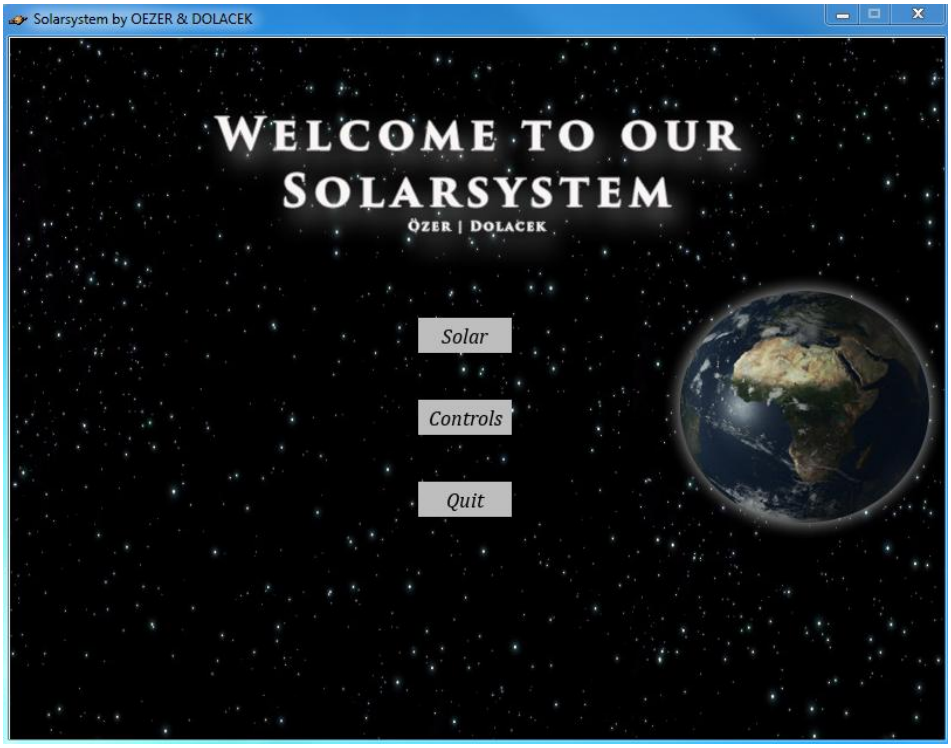
Die Animation wird mit d schneller gemacht, mit a langsamer und mit s gestoppt.

Je nachdem welche Taste gedrückt wurde, wird ein Wert dazu gezählt oder abgezogen.
Bei der gestoppten Animation ist alles auf 0.

Splashscreen

```
def solar_intro():
    """
    Solarsystem Splashscreen
    :return:
    """
    intro = True

    while intro:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_c:
                    intro = False
                if event.key == pygame.K_q:
```



Testen

Getestet wurde das Programm mit PyUnit. Es wurden alle Methoden zur Generierung von Planeten auf falsche Parameter eingaben überprüft.

Beispiel:

```
def test_sonne(self):
    """
    Überprüft die Eingaben --> Datentypen
    """
    test = Planet()
    glutInit(sys.argv)

    window = glutCreateWindow(b'Testing')
    sonne = Textur.LoadTextures(self,
            "Resources/textures/sun.gif")
            #Parameters like this: Planet.sonne(self, 0.2,
            0.0, 0.0, 0.0, 0.01 + self.geschws, self.sonne)
    self.assertRaises(Exception, test.sonne("falsche Eingabe",
                      1.0, 2.0, 3.0, 5.0, sonne))
```

Ausgabe:

```
F:\Python34\python.exe "F:\Program Files (x86)\J
Testing started at 01:22 ...
Eingaben überprüfen

Process finished with exit code 0
```

Testfälle wurden auch mit Jenkins durchgeführt!

Quellen

[1] pydoc.net,
http://pydoc.net/Python/PyOpenGL-Demo/3.0.1b1/PyOpenGL-Demo.NeHe.lesson18/,
04.03.2015

[2] StackOverflow-User: eryksun,
http://stackoverflow.com/questions/6756820/python-pil-image-tostring,
08.03.2015

[3] Chistoph Gohlke,
http://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame,
06.03.2015

[4] *"John F. Fay, John Tsiombikas, and Diederick C. Niehorster are the current maintainers of the FreeGLUT project".*,
http://freeglut.sourceforge.net,
10.03.2015

[5] Astronomie.de,
http://www.astronomie.de/astronomie-fuer-kinder/interessantes-fuer-lehrer-eltern/in-der-schule/groessenvergleich-der-planeten/,
16.03.2015

[6] deepsky.at,
http://www.deepsky.at/tabellen/planeten.shtml,
15.03.2015

[7] opengl.org,
https://www.opengl.org/documentation/specs/glut/spec3/node50.html,
18.03.2015

[8] swiftless.com,
http://www.swiftless.com/tutorials/opengl/keyboard.html,
18.03.2015