

2D MovieSzenen

Ein Handbuch zur App

Inhalt

2D MovieSzenen (<i>Vorwort zur App und Einführung</i>)	1
Verzeichnisstruktur (<i>Struktur, Module und Programmabläufe</i>)	1
	2
	3
	4
	5

2D Animationen für Sequenzen und Movie Szenen

Vorwort zur App:

Vieles von dem, was den nachfolgend beschriebenen Funktionsumfang der vorgestellten Anwendung anbelangt, können neuzeitliche Filmschnittprogramme nicht minder gut erledigen, einiges sogar schneller, übersichtlicher für den Anwender und vieles möglicherweise besser. Für andere Aufgaben würde hingegen zusätzliche Software für 3D-Animation benötigt, die dafür wesentlich umfangreicher ausgestattet wäre. Doch diese Anwendung wurde nicht entwickelt, um mit guten Schnittprogrammen mithalten oder 3D Anwendungen Konkurrenz zu machen. Sinn und Zweck bestand mehr darin, sich in Python weiter einzuarbeiten.

Heraus kam dabei eine Anwendung, die sich nicht als Filmschnittprogramm versteht, sondern als eine vorbereitende App, mit der Sequenzen für Szenen vorbereitet werden können, die dann von jedem Filmschnittprogramm einfach importiert werden können, welches durchnummerierte Images als Sequenzen akzeptiert.

Einführung in den Aufbau

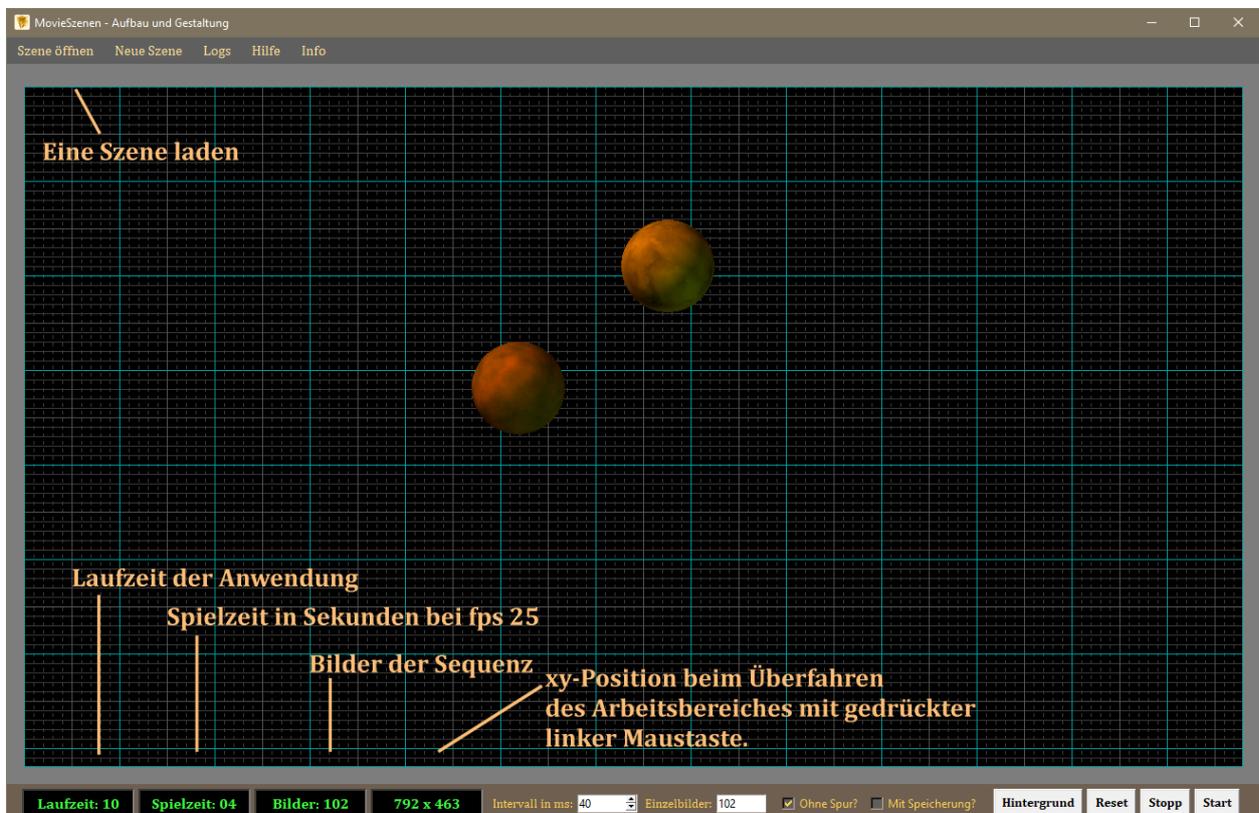
Die zur Anwendung gehörenden Scripts verteilen sich auf mehrere Verzeichnisse, wobei sich auf der obersten Ebene zwei „movie“-Module befinden, die sich aber nur durch ihre Endung *.py und *.pyw unterscheiden. Dabei ist die Datei movie.pyw zur Verknüpfung auf dem Desktop besser geeignet, es öffnet sich kein Konsolenfenster im Hintergrund.

Dass eine Python-Anwendung nur laufen wird, wenn Python bereits als Umgebung auf dem Rechner installiert wurde, wird als bekannt vorausgesetzt, denn Windows fragt zumindest nach, womit sie diese Anwendung ausführen möchten. Sie können für die movie.pyw die pythonw.exe auswählen. Eine richtige Installation ist hingegen nicht erforderlich, so dass die App praktisch in einem beliebigen Verzeichnis abgelegt werden kann.

Beim Start der Anwendung öffnet sich die movie.pyw (falls nicht *.py ausgewählt wurde) und mit ihr das eigentliche Arbeitsfenster. Diese enthält die Klasse Movie mit einer Vielzahl an Methoden, darunter die Methode Layout. Die Größe des aktuellen Layouts wurde auf 1280 x 720 Pixel zugeschnitten, was für YouTube Videos bislang noch genügen sollte. (Stand: 2021)

Es sei angemerkt, die Größe der Fläche lässt sich am einfachsten durch das Laden eines Hintergrundbildes verändern. Eine Anpassung ist aber auch in Zeile 96 der movie.py(w) möglich. Soll eine dauerhafte Anpassung erfolgen, muss weiterhin der Wert in Zeile 149 und 150 der Datei movie_funktionen.py (Verzeichnis ressourcen) entsprechend angepasst werden. Für Testzwecke ist eine kleinere mini-matrix.png beiliegend, auf eine größere wurde bisher verzichtet, weil dann weitere Änderungen erforderlich wäre.

Ansicht des Arbeitsfensters:



Der Anwendung liegen einige Muster für Testzwecke bei, die im weiteren Verlauf noch mit den zugehörigen Funktionen besprochen werden. Über den Tab „Szene öffnen“

lassen sich diese laden. Für den Anfang wäre es wichtig zu wissen, bei jeder Änderung innerhalb einer Szene-Datei ist selbige neu zu laden und der Reset Button zu benutzen.

Nach dem Start einer Szene werden einige laufende Werte angezeigt, wobei die Laufzeit sich nicht auf die Spielzeit bezieht, sondern lediglich anzeigt, wie lange das Script bisher läuft. Die Spielzeit gibt hingegen wieder, wie viele Sekunden Filmzeit die bisherige Anzahl der Bilder bei fps 25 ergeben würde.

Im Beispiel vom Screenshot lag die Laufzeit des Scripts bei 10 Sekunden, in denen 102 Einzelbilder gefertigt wurden, die eine Spielzeit von rund 4 Sekunden ergeben. Die Laufzeit sollte mehr im Zusammenhang mit Intervall in Millisekunden (ms) beachtet werden, eine Einstellung von 40 ms ist nur ein mittlerer Richtwert. Für einfache Objekte ohne Speicherung kann dieser Wert etwas tiefer angesetzt werden, für größere Objekte mit Speicherung sollte der Anwendung etwas mehr Zeit gegeben werden, eine höhere Anzahl von Einzelbildern zu erstellen. Ein Wert von 60 bis 80 ms könnte dann eher geeignet sein, was letztendlich aber auch von der Leistungsfähigkeit des Rechners abhängt.

Der Wert für „Einzelbilder“ sollte immer wenigsten dem in einer Szene definierten nim-Wert entsprechen. Nur so ist sichergestellt, dass die letzte xy-Koordinaten auch in den Logs erfasst werden kann. Dieser Wert aus den Logs kann dann dafür genutzt werden, um mit einer weiteren Bildsequenz nahtlos an die letzte Position einer bereits erstellten Sequenz anzuknüpfen. Liegt die Eingabe im Feld „Einzelbilder“ etwas höher, so hat dies keine negativen Auswirkungen, da die jeweilige Funktion dennoch nur bis zum durch „nim“ vorgegebenen Wert ihren Dienst verrichten und die letzte Position speichern wird.

Es sei angemerkt, eine Speicherung der letzten Position bzw. der letzten Koordinaten wurde nur für die Funktionen „berechne_verschiebung()“ und „berechne_zoom()“ umgesetzt, um die Logs nicht unübersichtlich werden zu lassen. Wer von den anderen Funktionen die Werte erfassen möchte, kann die Anwendung über IDLE starten und mit einer einzufügenden Print-Anweisung die laufenden Werte erfassen. Ein auskommentiertes Beispiel für eine Printanweisung befindet sich in Zeile 342 der Datei „movie_funktionen.py“, welches bei Bedarf einkommentiert werden kann.

Struktur, Module und Programmabläufe

Verzeichnisstruktur:

Ansicht der Verzeichnisstruktur:

