AniTMT Users Guide

Inhaltsverzeichnis

1 Einleitung

Filme wie Toy Story und A Bug's Life haben uns eindrucksvoll gezeigt, welche Qualität mit vollständig computer-generierten Filmen inzwischen erreicht werden kann. Die Erstellung solcher Filme – wenn auch in wesentlich bescheidenerer Form – ist für uns ein großer Traum seit wir uns mit Raytracing beschäftigen.

Für unsere ersten Versuche haben wir den Raytracer POV-Ray verwendet. Er hat uns im Bereich der Standbilder überzeugt, doch die darin integrierte Animationsfähigkeit ist in der Anwendung zu umständlich. Allerdings haben wir auch keine andere Software gefunden, die eine angemessene Lösung bietet.

Aus diesem Grund haben wir uns dazu entschlossen, ein eigenes Animationssystem zu entwickeln. Als Name hierfür haben wir AniTMT gewählt, wobei TMT für die Anfangsbuchstaben unserer Nachnamen steht.

2 Wie möchte ich die Szene animieren ?

Prinzipiell gibt es zwei Möglichkeiten eine 3-dimensionale Szene zu animieren:

2.1 Objekte animieren

Objekte können als eine Einheit auf einer Flugbahn bewegt werden. Zusätzlich kann die Richtung des Objektes in Abhängigkeit von der Flugbahn verändert werden.

Ein ganzes Objekt zu animieren ist immer sinnvoll, sobald sich das Objekt entlang einer zusammengesetzten Bahn bewegen soll.

2.2 Werte animieren

Bei der Interpolation von Werten wird immer eine in der Szenenbeschreibung definierte Variable benutzt. Diese Werte können nach belieben verwendet oder weiter verrechnet werden.

Werte können benutzt werden, wenn bestimme Eigenschaften (wie zum Beispiel Farbe) eines Objektes oder der Szene animiert werden sollen. Auch zum Verändern der Größe mit einer scale-Anweisung oder zum Rotieren von Objekten um nur eine Achse sind einzelne Werte sinnvoll. Die Benutzung der Variablen muß explizit in der Szenenbeschreibung angegeben werden.

3 Vorbereiten der Szenenbeschreibung

Eine Szene besteht aus mehreren Komponenten. Diese sind entweder ganze Objekte die bewegt und gedreht werden können oder einfache Werte. Alle Komponenten einer Szene, die animiert werden sollen müssen dort benannt werden.

3.1 POV-Dateien

Im folgenden wird für POV-Ray Szenen, wie man die Komponenten darin animierbar macht.

3.1.1 Kennzeichnung der Objekte

Um Objekten für eine Animation einen Namen zuzuweisen wird folgende Syntax verwendet:

```
sphere {                // My_Sphere        <-- Dies ist der Name
  < 0, 0, 0 >, 0.45
  pigment { OldGold }
  rotate < 0, 0, 0 >
  translate < -4, 1, 1 >
}
```

Hinter dem Objekttyp (sphere) und der geschweiften Klammer wird mit 2 Schrägstrichen ein Kommentar eingeleitet. Das erste darauf folgende Wort wird von `anitmt` als Name des Objektes übernommen. (Siehe auch: 4.2 Regeln für Namensgebung)

3.1.2 Variablen einfügen

Variablen muss nicht extra ein Name zugewiesen werden. Statt dessen wird der Variablenname verwendet. In der Szene muss wie folgt eine `#declare`-Anweisung stehen, die der Variable einen Standardwert zuweist:

```
#declare MyVariable = 12345;
```

Die Anweisung muss unbedingt mit einem Semikolon abgeschlossen werden, damit `anitmt` das Ende der Anweisung erkennt. Wenn diese Varible animiert werden soll wird `anitmt` für die Einzelbilder alle `#declare`-Anweisungen für die Varible aus der Szenebeschreibung löschen und statt dessen eine eigene einfügen. Aus diesem Grund sollte, wenn man die Variable weiter verwenden will, um beispielsweise ein Schleife abzuarbeiten, nicht der selbe Variablenname verwendet werden. Statt dessen müssen für weitere Verechnungen eine andere Variable benutzt werden, wie in dem folgendem Beispiel gezeigt:

```
#declare MyStartValue = 1;
#declare MyEndValue   = 9;
#declare i = MyStartValue;

#while (i <= MyEndValue)

  box {
    <0,0,0>, <1,1,1>
    pigment { color rgb < i * 0.1, 0, 0 > }
    translate i * y
  }

  #declare i = i + 1;
#end
```

Wenn man versuchen würde, anstelle von „i“ „MyStartValue“ inerhalb der Schleŋfe zu verwenden, würde dieses Beispiel nicht funktionieren, da `anitmt` auch die Zeile

```
#declare MyStartValue = MyStartValue + 1;
```

löschen würde.

4 Definieren der Animation in der ADL-Datei

Für Animationsskripte haben wir das Dateiformat „animation description language“ (ADL) entwickelt.

4.1 Aufbau und Syntax

Die ADL-Datei ist hierrachisch in verschiedene Ebenen eingeteilt: Szene, Komponente (Objekt/Variable), Funktion, Subfunktionen

```
povscene MeineSzene {        // POV-Ray-Szene
  filename "myscene.pov";     // Szenendatei
  my_obj {                    // Name des Objektes
    move {                      // Bewegung auf Flugbahn
      straight {                  // Gerade zum Ursprung
        startpos    <-2,0,0>;       // Startposition
        startdir    x;              // Startrichtung
        length      3;              // Laenge in LE
        startspeed  2;              // Geschwindigkeit in LEs
      }
      circle {                    // 180 Grad Kurve
        normal      y;              // X-Z Ebene
        radius      2;              // Radius in LE
        angle       180;            // Winkel in Grad
      }
      circle {                    // 130 Grad Kurve schief im Raum
        center      <1,1,-1>;       // Rotationszentrum
        angle       130;            // Winkel in Grad
      }
      straight {}                   // Gerade in die Unendlichkeit
    }
  }
}
```

Grundsätzlich werden zwei Angabearten unterschieden. Man kann Blöcke öffnen, deren Inhalt in geschweifte Klammern eingeschlossen wird, und man kann Eigenschaften eines Blockes definieren.

Blöcke werden durch einen Bezeichner (z. B. `povscene`) eingeleitet. Wenn später auf diesen Block verweisen können soll, muß man ein Name hinzufügen (im Beispiel `MeineSzene`), mit dem man diesen wieder ansprechen kann.

Eigenschaften bestehen aus einem Bezeichner (z. B. `filename`) und einem durch ein Leerzeichen abgetrennten Wert (z. B. `"myscene.pov"`) mit abschließendem Semikolon. Werte können sowohl Skalare als auch Vektoren, Zeichenketten oder komplette mathematische Ausdrücke sein. In jedem Block sind nur bestimmte Eigenschaften möglich.

Auf oberster Ebene werden Szeneblöcke geöffnet, bei POV-Ray durch den Bezeichner `povscene`. Alle Szenen benötigen die Angabe `filename`, die einen Dateinamen enthalten soll. Innerhalb der Szenen werden die Komponentenblöcke mit dem zuvor vergebene Namen aus der Szenenbeschreibung eingeleitet (siehe: 3 Vorbereiten der Szenenbeschreibung)

In den Komponenten können Funktionsblöcke aufgemacht werden. Die Funktion `change` animiert zum Beispiel Skalare, während die Funktion `move` Objekte durch Aneinanderreihung von Bahnelementen bewegt. Durch diese Funktionen wird der Typ einer Komponente bestimmt. Die Elemente einer Funktion werden durch Subfunktionsblöcke definiert. In ihnen werden die eigentlichen Eigenschaften festgelegt, die das Verhalten der Animation bestimmen.

4.2 Regeln für Namensgebung

In den Szenenbeschreibungen, wie zum Beispiel den POV-Ray-Dateien konnen den Objekten mit Kommentaren Namen zugewiesen werden. Ebenso kann in der ADL-Datei den einzelnen Blöcken Namen geben, um Verweise auf Eigenschaften von anderen Komponenten möglich zu machen.

Für einen solchen Namen gelten folgende Regeln:

- er darf nur aus Gross- und Kleinbuchstaben (a-z), Zahlen und Unterstrich bestehen
- er muss mit einem Buchstaben beginnen
- er darf höchstens 100 Zeichen lang sein.
- Gross und Kleinschreibung wird beachtet

5 Funktionen

Um Komponenten der Szene animieren zu können braucht man verschieden Funktionen. Eine Funktion bestimmt wie eine Komponente animiert wird. Für ein Objekt gibt es zum Beispiel die Funktion move. Diese Funktion, die nur bei Objekten zulässig ist, besagt, dass das Objekt sich entlang einer Flugbahn bewegen sollen, die aus mehreren Teilstücken definiert ist.

5.1 Allgemeines

Da bei `anitmt` die Definition einer Animation möglichst komfortabel sein soll gibt es für alle Funktion, also sowohl für Werte- und Vektorinterpolationen als auch für Flugbahnen folgende Möglichkeiten:

5.1.1 Eigenschaften Werte zuweisen

Um die Animation einer Komponente zu definieren weist man ihr eine Funktion zu. Diese enthält mehrere Subfunktionen, also zeitlich aufeinanderfolgende Abschnitte. In den Subfunktionen werden diese Abschnitte genauer Bestimmt mit Hilfe von Eigenschaften

Normalerweise werden Eigenschaften sofort ein fester Wert zugewiesen:

```
startpos    < 5, 2, 3.141 >;
```

Es jedoch auch möglich, Werte zu verrechnen und Konstanten zu verwenden:

```
startpos    (5 * x) + (2 * <0,1,0>) + <0,0,pi>;
```

Im Wesentlichen sind in AniTMT dieselben Rechenoperationen wie in POV-Ray möglich. Eine genauere Übersicht wird in Kürze folgen.

5.1.2 Bestimmung über beliebige Eigenschaften

Jede Subfunktion besitzt verschiedene Eigenschaften, wie zum Beispiel `starttime`, `endtime`, `startpos`, `endpos`, `duration`, ...

Wenn nun eine Eigenschaft benötigt wird, so sind alle Möglichkeiten bekannt, sie aus den anderen Eigenschaften zu berechnen. Bei einer geraden Bahn wird beispielsweise die Endposition (endpos) aus Startposition (startpos), Startrichtung (startdir) und Länge (length) berechnet. Falls diese Eigenschaften ebenfalls nicht angegeben wurden, bestehen wiederum mehrere Möglichkeiten, auch diese Werte zu ermitteln.

Es genügt also, einen Abschnitt eindeutig zu bestimmen, so dass alle anderen Werte ausgerechnet werden können.

5.1.3 Wertebestimmung über Nachbarelemente

Die Ermittlung der benötigten Eigenschaften erfolgt in mehreren Stufen. Zuerst werden in jedem Abschnitt die Unbekannten, wie gerade beschrieben, ermittelt. Wenn Subfunktionen nicht sämtliche ihrer Eigenschaften berechnen konnten, werden entsprechende Werte von Nachbarelementen übernommen.

Um zum Beispiel bei einer Flugbahn einen zeitlich zusammenhängenden Vorgang sicherzustellen, sollten die Start- und Endzeiten benachbarter Elemente übereinstimmen. Um Sprünge zu vermeiden, benötigt man gleiche Positionen; um Knicke zu verhindern, gleiche Richtungen; um die Animation flüssig zu gestalten, gleiche Geschwindigkeiten. Entsprechendes gilt natürlich auch für Werte und Vektorinterpolationen. Auf jeder Stufe wird eine Eigenschaft in obiger Reihenfolge zur Übergabe freigegeben.

Damit ist es möglich, eine Animation mit relativen Veränderungen, wie der Dauer, zu definieren und nur an einer Stelle absolute Werte, wie zum Beispiel die Startzeit, anzugeben.

5.1.4 Vergabe von Standardwerten

Da wir sowohl mächtige als auch komfortable Funktionen anbieten wollen, werden bei Abschnitten, die noch nicht eindeutig definiert sind, Standardwerte gesetzt. Damit wird beispielsweise statt einer beschleunigten Bewegung eine gleichförmige Bewegung angenommen. Einige Anfangswerte des ersten Abschnitts werden standardmäßig gesetzt. Auch die Endzeit des letzten Abschnittes kann mit Hilfe der Endzeit der gesamten Animation ermittelt werden.

5.1.5 Verweise auf Werte anderer Elemente

Desweiteren ist es möglich, Beziehungen zwischen mehreren Bewegungen herzustellen.

Beispielsweise kann man eine Rakete durch Verweise auf die Eigenschaften eines Flugzeuges so ausrichten, dass sie dieses zum richtigen Zeitpunkt trifft, ohne die exakte Position und Zeit zu bestimmen.

Generell ist es ja auch möglich, die Werte weiter zu verrechnen, um beispielsweise die Rakete das Flugzeug verfehlen zu lassen, indem sie die Bahn erst einige Sekunden später kreuzt.

5.2 Liste aller möglichen Funktionen

5.2.1 Werteinterpolation (change)

Diese Funktion dient zum Animieren einer Variablen, die der Anwender beliebig im Programm weiter verwenden kann, um zum Beispiel Texturen oder Objekte wie Türen (Öffnungswinkel) und ähnliches zu verändern.

Jede Subfunktion hat Standardeigenschaften, die bei allen möglich sind, und weitere spezielle. Die folgenden Eigenschaften sind für alle Subfunktionen gleich:

`starttime`	Startzeitpunkt
`endtime`	Endzeitpunkt
`duration`	Dauer
`startvalue`	Startwert
`endvalue`	Endwert
`difference`	Unterschied zwischen Start- und Endwert

Für die Werteinterpolationen gibt es 2 Subfunktionen:

- `linear`
- `accelerated`

linear Die Funktion linear entspricht einer konstanten Zunahme eines Wertes. Als weitere Eigenschaften für linear gibt es:

`slope`	Zunahme in Einheiten pro Sekunde

accelerated Die Funktion accelerated entspricht einer konstant beschleunigten Zunahme eines Wertes. Als weitere Eigenschaften für linear gibt es:

`startslope`	Zunahme in Einheiten pro Sekunde am Anfang
`endslope`	Zunahme in Einheiten pro Sekunde am Ende

5.2.2 Bewegung von Objekten mittels einer Flugbahn (move)

Für alle Objekte wird eine „Vorne“-Richtung und eine „Oben“-Richtung bestimmt, die ihre ursprüngliche Lage in der Szene festlegt. Diese werden zusammen mit dem Drehzentrum `center` im Objekt wie folgt angegeben:

```
povscene{
  my_object{
    center      <1,1,1>;
    front       <1,0,1>;
    up          y;

    move{...}
  }
}
```

Der Standardwert für das Drehzentrum is `<0,0,0>`, der für `front` ist `x` und der für `up` ist z.

Um die Lage eines Körpers im dreidimensionalen Raum eindeutig zu bestimmen verwenden wir ein System, das eine Blickrichtung durch zwei senkrecht aufeinander stehende Winkel definiert, und die Rotation des Objektes um die Blickrichtung angibt.

Ein Flugzeug, zum Beispiel, könnte man als erstes um seine Blickrichtung oder Längsachse drehen. Zusätzlich ist eine Drehung um die Querachse möglich oder auch eine Ausrichtung des Cockpits nach oben oder unten. Beide Achsen werden durch die „Oben“-Richtung festgelegt, nachdem diese in Abhängigkeit zur Flugbahn gedreht wurde. Zuletzt wird es dann an die richtige Position geschoben. Dadurch werden alle erdenkbaren Lagen abgedeckt.

Die Funktion `move` ist an dieses Konzept angelehnt. Mit einzelnen Abschnitten, wie Gerade oder Kreisbogen, wird eine Flugbahn oder Position bestimmt. Das Objekt schaut mit seiner Blick- oder Vornerichtung zunächst immer tangential zur Flugbahn. Wenn das Objekt durch eine Kurve fliegt, wird die Obenrichtung entsprechend dieser gedreht.

Wenn man jedoch ein Objekt haben möchte, dessen Lage immer gleich ist (Zum Beispiel ein Ufo) gibt es die Möglichkeit die automatische Rotation in Abhängigkeit zur Bahn auszuschalten. Dazu kann man die Eigenschaft der Funktion move wie folgt festlegen:

```
autorotate false;
```

Alle Subfunktionen unterstützen beschleunigte Bewegungen und Rotation um die drei Achsen. Deswegen besitzen alle Abschnitte einer Flugbahn folgende Eigenschaften:

Allgemein:

`starttime`	Startzeitpunkt
`endtime`	Endzeitpunkt
`duration`	Dauer
`startup`	ÖbenVector am Anfang
`endup`	ÖbenVector am Ende
`up_roll`	Diese Zahl gibt die Rotation des ÖbenVectors um die Bahn in Grad an

Bewegung:

`startpos`	Startposition
`endpos`	Endosition
`startdir`	Startrichtung
`enddir`	Endrichtung
`startspeed`	Startgeschwindigkeit
`endspeed`	Endgeschwindigkeit
`acceleration`	Beschleunigung
Rotation um die Blickrichtung:	
`startrot_roll`	Startwert in Grad
`endrot_roll`	Endwert in Grad
`diffrot_roll`	Unterschied in Grad
`startrotspeed_roll`	Rotationsgeschwindigkeit am Anfang in Grad pro Sekunde
`endrotspeed_roll`	Rotationsgeschwindigkeit am Ende in Grad pro Sekunde
`rotacceleration_roll`	Beschleunigung der Rotation
Rotation in der Horizontalen:	
`startrotspeed_horizontal`	Startwert in Grad
`endrotspeed_horizontal`	Endwert in Grad
`diffrot_horizontal`	Unterschied in Grad
`rotacceleration_horizontal`	Rotationsgeschwindigkeit am Anfang in Grad pro Sekunde
`startrot_horizontal`	Rotationsgeschwindigkeit am Ende in Grad pro Sekunde
`endrot_horizontal`	Beschleunigung der Rotation
Rotation in der Vertikalen:	
`startrot_vertical`	Startwert in Grad
`endrot_vertical`	Endwert in Grad
`diffrot_vertical`	Unterschied in Grad
`startrotspeed_vertical`	Rotationsgeschwindigkeit am Anfang in Grad pro Sekunde
`endrotspeed_vertical`	Rotationsgeschwindigkeit am Ende in Grad pro Sekunde
`rotacceleration_vertical`	Beschleunigung der Rotation

Natürlich muß für eine Flugbahn meist nur ein sehr kleiner Teil dieser Eigenschaften angegeben werden. Wenn zum Beispiel keine Rotationen stattfinden sollen, muss keine dieser Eigenschaften angegeben werden. Die drei Rotationswinkel werden dann automatisch auf 0 gesetzt und das Objekt dreht sich nur mit der Flugbahn. (Siehe auch 5.1.4 Vergabe von Standardwerten)

Für die Bewegung von Objekten gibt es 2 Subfunktionen:

- `straight`
- `circle`

straight Die Subfunktion `straight` bewegt das Objekt auf einer Geraden. als einzige weitere Angabe kommt die Streckenlänge hinzu:

`length`	Streckenlänge

circle Die Subfunktion `circle` bewegt das Objekt auf einer Kurve im Raum. Mögliche weitere Angaben sind:

`length`	Streckenlänge
`normal`	Normalenvektor der Ebene in der die Rotation stattfindet
`radius`	Radius des Kreises
`center`	Mittelpunkt des Kreises
`angle`	Winkel, um den gedreht werden soll