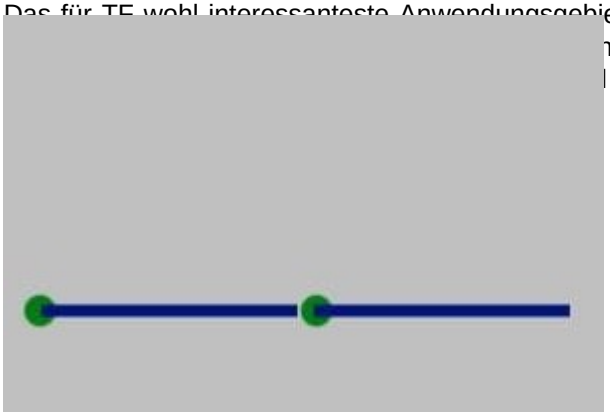
Animation: Gruppe

Inhaltsverzeichnis

- 1 Einleitung
- 2 Eine ganze Gruppe animieren!
- 3 Praktische Umsetzung
- 4 Thinkin' outside the box

1 Einleitung

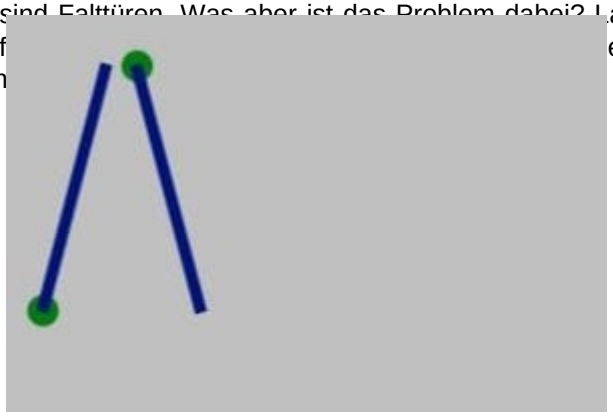
Das für TF wohl interessanteste Anwendungsgebiet sind Falttüren. Was aber ist das Problem dabei? Lasst



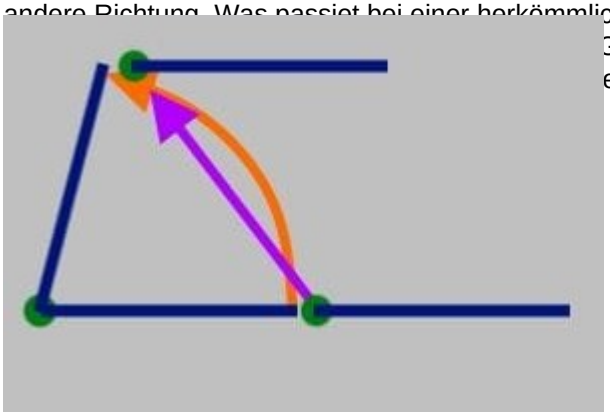
hoff eine

l un

->



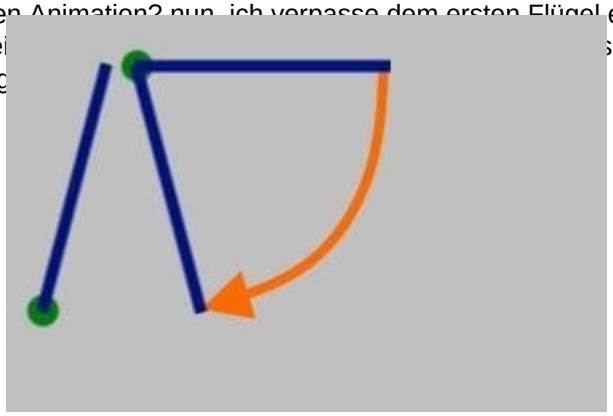
Das Blaue sind die beiden Flügel und die grünen Punkte sind die Drehpunkte. Was passiert nun beim Öffnen? Der erste Flügel dreht sich und der zweite macht diese Drehung mit, dreht sich selber aber in die andere Richtung. Was passiet bei einer herkömmlichen Animation? nun, ich verpasse dem ersten Flügel eine



Glei ss er

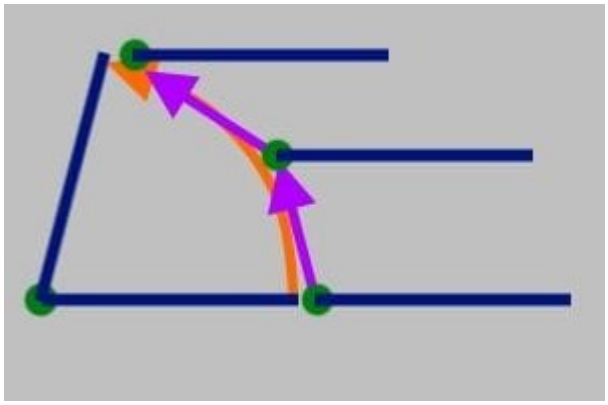
eng

->



Der erste Flügel wird wie gesagt einfach gedreht, so wie das der orange Pfeil darstellt. Der zweite Flügel aber wird wird verschoben und dies passiert geradlinig. Das wäre dann der pinke Pfeil. In ein zweites Bild hab ich die Drehung in die entgegen gesetzte Richtung ausgelagert, weil das sonst etwas zu unübersichtlich geworden wäre. Diese Drehung macht uns aber auch keine Probleme, es ist diese Diskrepanz zwischen der Kurvenbewegung der "Flügelspitze" der ersten Tür und der geradlinigen Bewegung vom Ansatzpunkt der zweiten Tür. Das sieht im Ergebnis einfach nicht gut aus. Was kann man also tun?

Man könnte diese Translation der zweiten Türe in Zwischenschritte unterteilen:



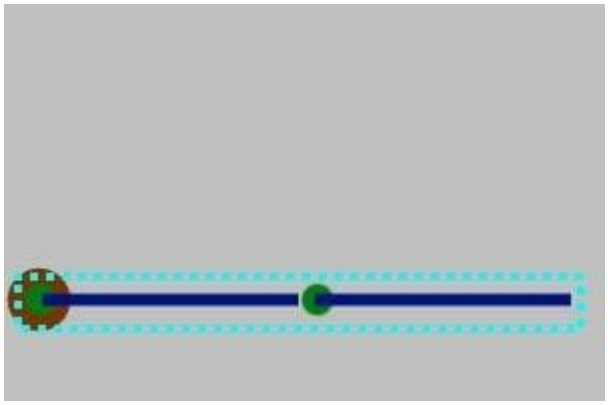
Die gleichzeitige Drehung hab ich hier mal wieder gekonnt aussen vor gelassen. Diese Unterteilung kann man nun beliebig oft vornehmen und irgendwann wird man ein passables Resultat erzielen. Dennoch ist diese Lösung unbefriedigend, da schlicht und ergreifend verflucht aufwändig. Entweder greift man für jeden Zwischenschritt in Blender (oder Max oder...) die Koordinaten ab, was einerseits tippel/kopier-Arbeit ist und andererseits auch die laufende Anpassung der Drehungen und Positionen der Einzelmeshes bedarf. Und dann schiebt man den zweiten Flügel eher nach Augenmaß und so richtig perfekt isses auch wieder nich. Andererseits kann man sich auch ein Skript bauen das einem die Positionen errechnet und gleich als fertig formatierten text ausgibt - aber da jetzt noch Programmieren lernen? Selbst wenn mans kann steht da Aufwand und nutzen nicht wirklich im verhältnis !P Page not found or type unknown

Was wäre also richtig toll? Man bräuchte zwei Drehpunkte. Einmal statt einer Translation eine Drehung um den Drehpunkt der ersten Türe vollführen und dann die eigentliche Drehung der zweiten Türe um den eigenen Drehpunkt. Aber... woher nehmen, wenn nicht stehlen? Ein Mesh hat nunmal nur einen Drehpunkt.

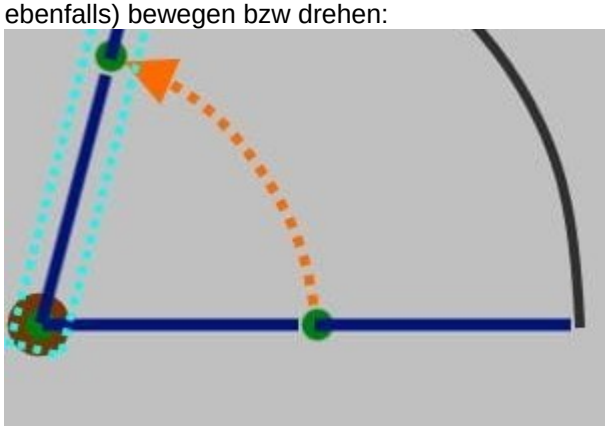
2 Eine ganze Gruppe animieren!

Hier kommt nun die Gruppe ins Spiel. Lasst und also wieder vorne beginnen, diesmal packen wir aber unsere beiden Türflügel in eine Gruppe und in der mdl wird dann die Gruppe eingebunden statt die Meshes

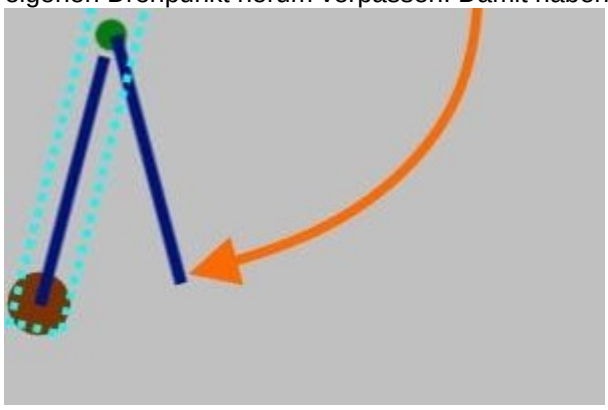
direkt:



Der große dünnpfiffbraune Flatschen ist der Drehpunkt der hier hellblau gestrichelten Gruppe. Die Idee ist nun, alles in der Gruppe auf einmal zu animieren - eben indem wir die Gruppe (und somit alles darin ebenfalls) bewegen bzw drehen:



Nun hat sich alles in der Gruppe um den Gruppendrehpunkt herum gedreht - also auch unsere zweite Tür. Diese vollführt also auf simpelste Art und Weise eine perfekte Kreisbewegung - genau was wir wollten. Und das Beste: Wir können nun dem Mesh der zweiten Türe immernoch eine eigenständige Animation um den eigenen Drehpunkt herum verpassen! Damit haben wir genau das erreicht, was wir wollten.



Und schon sind wir fertig - theoretisch.

3 Praktische Umsetzung

Gut, wie es theoretisch klappt, wissen wir nun, aber wie funktioniert es praktisch? Eigentlich ist es total simpel, aber ich kam auch nicht von alleine drauf >< Ich war halt so gefangen davon, dass bei TF alles bei 1 mit Zählen anfängt... Aber starten wir erstmal einen kleinen Versuchsaufbau:

Zuerst mal eine simple Modell-Datei:

Code: beispiel.mdl

```
function                                                                    data()
return                                                                           {
 boundingInfo                                    =                                {
  bbMax          =          {                        1,                      1,
  bbMin          =          {           -1,           -1,           -1,           },
 },
 collider                                      =                                  {
  params                                       =                                   {
  },
  type                                    =                                  "MESH",
 },
 lods                                      =                                      {
  {
   children                                      =                                   {
    {
     id                        =                        "<path>/beispiel_gruppe.grp",
     transf                                      =                                    {
      1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0,
     },
     type                                    =                                 "MESH",
    },
   },
   matConfigs                                    =                                   {
    {
     0,
    },
   },
   static                                    =                                 false,
   visibleFrom                                   =                                 0,
   visibleTo                                   =                                1500,
  },
 },
 metadata                                      =                                  {
 },
}
end
```

Alles anzeigen

https://www.transportfever.net/lexicon/entry/141-animation-gruppe/

Und natürlich eine Gruppe:

Code: beispiel_gruppe.grp

```
function                                                                    data()
return                                                                          {
 children                                    =                                   {
  {
   id                              =                              "<path>/door1.msh",
   transf                                    =                                    {
    1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0,
   },
   type                                  =                                  "MESH",
  },
  {
   id                              =                              "<path>/door2.msh",
   transf                                    =                                    {
    1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.5, 0.0, 0.0, 1.0,
   },
   type                                  =                                  "MESH",
  },
 },
 matConfigs                                  =                                  {
  {
   0,                                                                           0,
  },
 },
}
end
```

Alles anzeigen

Die mdl verweist also auf die Gruppe, und diese wiederrum auf die Meshes der Türen. Nun möchten wir die Türen um sagen wir mal 85° drehen. Nun müssen wir bedenken, dass die zweite Türe sich doppelt so weit in entgegen gesetzter Richtung drehen muss. Sie muss ja einmal die Drehung der Gruppe ausgleichen und dann zusätzlich noch die eigene Drehung vollführen. Also in unserem simplen Beispiel eben nicht -85° sondern -170°. Wir verpassen unserem Mesh also einen Animationsblock mit einer `turn`-Animation (also mit dem Namen turn), die dies realisiert. Falls ihr nicht wisst, wie das geht, schaut nochmal hier vorbei: Animationen: Aufbau.

Diese Animation können wir nun über den events-Block der Gruppe ansteuern.

Code: biespiel_gruppe.grp

```
function data()
return {
 children = {
  ...
 },
 events = {
  close_all_doors = {
   [2] = { forward = false, name = "turn", },
  },
  open_all_doors = {
   [2] = { forward = true, name = "turn", },
  },
 },
 matConfigs = {
  ...
 },
}
end
```

Alles anzeigen

Schon bekanntes hab ich hier mal weggelassen. Mit [2] greifen wir auf das zweite Element der children-Liste zu, welches unsere zweite Türe ist. In gleicher Weise fügen wir der .mdl einen solchen Block hinzu, indem wir die dort eingetragene Gruppe ansprechen. Somit wird das Auslöseevent durchgeschleift bis zu unserer zweiten Türe. Soweit so gut, was ist nun aber mit der ersten Türe bzw der Gruppe? Weil die Tür selbst braucht keine Animation, da sie ja mit der Gruppe animiert wird. Wie stellen wir das an?

Nunja, da wir die Gruppe animieren wollen, bekommt die Gruppe auch den Animationsblock!

Code: beispiel:gruppe.grp

```
function data()
return {
 animations = {
  turn_grp = {
   -- Animationsdaten
  },
 },
 children = {
  ...
 },
 events = {
  ...
 },
 matConfigs = {
  ...
 },
}
end
```

Alles anzeigen

Soweit so gut - bleibt die letzte und wichtigste Frage: Wie greifen wir darauf zu...

Wie gesagt ist es eigentlich total simpel, aber nunja, gerade da kommt man irgendwie nur selten drauf xD Man spricht ganz einfach die ID0 an - fertig ^^ Image not found or type unknown

Code: beispiel_gruppe.grp

```
function data()
return {
 animations = {
  turn_grp = {
   -- Animationsdaten
  },
 },
 children = {
  ...
 },
 events = {
  close_all_doors = {
   [0] = {forward = false, name = "turn_grp"},
   [2] = {forward = false, name = "turn"},
  },
  open_all_doors = {
   [0] = {forward = true, name = "turn_grp"},
   [2] = {forward = true, name = "turn"},
  },
 },
 matConfigs = {
  ...
 },
}
end
```

Alles anzeigen

Und das war es dann auch schon. Nun haben wir eine wunderbar animierte Faltentür. Das lässt sich jetzt natürlich noch beliebig weiter schachteln, indem man weitere Türen hinzufügt. Bei 3 Türen hätte man dann eben diese Struktur:

Code

```
mdl
  grp1
    tür1
    grp2
      tür2
      tür3
```

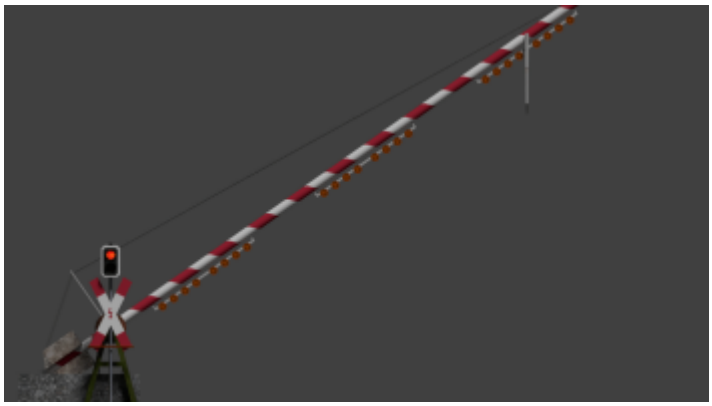
... und so weiter und so fort !) Image not found or type unknown

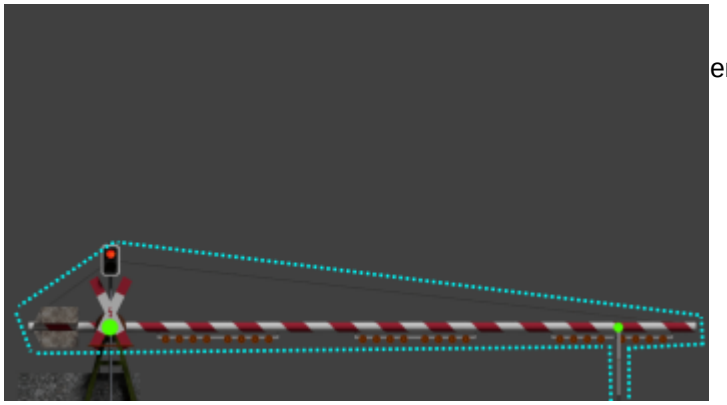
4 Thinkin' outside the box

Fein, wir können nun also schön Falttüren auf simple und zugleich sehr ansprechende Art und Weise animieren. War es das? Oh nein! Allein mir sind schon zig Anwendungsbeispiele eingefallen. Denkt nur einmal an einen Jahrmarkt mit all den Karousells, Schiffs Schaukeln und Riesenrädern! Da ließe sich soviel

ie letzten zwei Tage mal entwickelt, einfach weils Stützfuß:

->

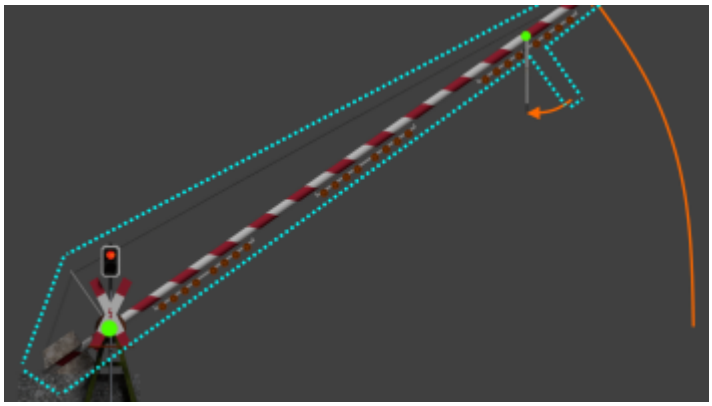




emalt:

->

https://www.transportfever.net/lexicon/entry/141-animation-gruppe/



Der Schlagbaum und der Fuß sind in einer Gruppe. Der Schlagbaum entspräche unserer Tür1 von eben und der Fuß der zweiten Türe. Beim Öffnen wird nun also die ganze Gruppe gedreht und damit auch der Fuß, welcher also seperat wieder zurück gedreht werden muss. Da er hier einfach nur die Drehung der Gruppe ausgleichen soll, reicht eine Drehung um das Negative (nicht das doppelt Negative wie bei den Falttüren), damit er immer schön nach unten baumelt :woot:

Dass dies nicht unbedingt immer eine simple Animation sein muss, könnt ihr euch bald in der Webdisk selber anschauen, wenn ich euch diesen Bahnübergang hochgeladen habe :D Ich aktualisiere dann den Link noch hier herein :) Edit: So, hier ist das gute Stück für euch: (Ost)Deutscher Bahnübergang :)

Image not found or type unknown

Image not found or type unknown

Image not found or type unknown

Also, lasst eurer Kreativität freien Lauf und baut was schönes mit dieser eigentlich doch simplen Technik.