

Titel: FX Erstellung, Einführung ins Terrain

Schwierigkeit: Grundkenntnisse

Typ: Aufgaben, Aller Anfang hat einen Anfang (hä?)

Programm: Final BIG

Alien aka Infiltrator

### Wieder einmal: Willkommen zum Tutorial

Ich gehe davon aus das ihr das Tutorial FX-Grundbegriffe gelesen habt und euch bereits im klaren seit wofür die einzelnen Parameter gut sind. Am besten wäre es wohl, ihr habt das Grundbegriffe-Tutorial einfach zusätzlich zu diesem hier geöffnet, dann könnt ihr jederzeit nachlesen wenn ihr euch bei gewissen Werten nicht sicher seit (es verlangt ja keiner das ihr schon alle Parameter Auswendig wisst^^).

Dazu muss man jetzt sagen das es sehr schwer ist das ganze richtig anzugehen, daher werde ich einfach versuchen die Sache anzugehen und Schritt für Schritt die einzelnen Parameter dazu zu nehmen.

Soll heißen: Wir fangen klein (wirklich klein), mit möglichst wenig Werten an, und steigern uns dann langsam zu einer sich wild drehenden farbenfrohen FX mit der man sogar was anfangen kann :-P

### Inhaltsverzeichnis

Einführung und Inhaltsverzeichnis	Seite 1
Alles hat einen Anfang...	Seite 2 - 5
- Der starre Particle	Seite 2
- Der starre Particle – Lösung	Seite 3
- Der Starre wird farbig...	Seite 4
- Der Starre wird farbig – Lösung	Seite 5
Jetzt kommt Bewegung in die Sache...	Seite 6 - 8
- Das etwas... bunte WdM	Seite 6
- Das etwas... bunte WdM – Lösung	Seite 7 - 8
Größer, weiter, mehr und Zufall	Seite 9
- Twice is nice	Seite 9
- Twice is nice – Lösung	Seite 9
- Strukturell statt punktuell!	Seite 10

### Alles hat nen Anfang, nur die FX ist eigenartig...

Nun ja, als Erstes nehmen wir uns eine sehr simple FX vor:

Ein Particle der gespawnt wird und sich dabei nicht bewegt, dreht oder sonst was macht, sein einziger Lebenszweck wird in sein, in der Luft zu hängen und wieder zu verschwinden...

Ich würde vorschlagen ich gebe euch sehr wenig Werte vor, ich gebe euch also fast gar nichts...

Also die FX soll:

...sich kein bisschen bewegen

...einmal einen Particle spawnen

...welcher sich kein bisschen bewegt

...dieser Particle soll 5 Sekunden lang lebens

...der Particle sollte Rot sein und ohne speziellen Farbwechsel

...der Particle sollte zur Kamera, und nicht zum Boden, orientiert sein (böse Aufgabe :P )

...der Particel soll 50 groß sein...

Überlegt mal selbst was müsste hier statt den „?“ rein?

FXParticleSystem AlienSimpleParticle

System

Priority = ALWAYS\_RENDER ;;; Ich setze das hier einfach vorraus

ParticleName = EXRing01.tga ;;; für ein gleiches Ergebnis wäre es Vorteilhaft ihr nehmt das auch so...

Lifetime = ? ?

SystemLifetime = ?

SortLevel = 1

Size = ? ?

BurstCount = ? ?

;;;gehört hier noch was rein?

End

Color = DefaultColor

; brauch ich das überhaupt?

End

Update = DefaultUpdate

SizeRate = ? ?

SizeRateDamping = ? ?

AngleZ = ? ?

AngularDamping = ? ?

AngularDampingXY = ? ?

End

Physics = DefaultPhysics

;;; brauchen wir diesen Part?

End

EmissionVelocity = OutwardEmissionVelocity

;;; wow... erm... hier auch was rein?

End

EmissionVolume = PointEmissionVolume

IsHollow = Yes

;;; ? ist dieser Block auch vollständig?

End

Draw = DefaultDraw

End

End

## ACHTUNG: TUT EUCH EINEN GEFALLEN UND LÖST DIE AUFGABE ERST SELBST ;-)



Ihr seid fertig wenn euer Ergebnis so wie links im Bild aussieht...

Fertig? Gut dann geht's hier mit der Lösung weiter, hab die „?“ mit Leser-freundlichen Kommentaren und den Lösungen gefüllt :-P

FXParticleSystem AlienSimpleParticle

System

Priority = ALWAYS\_RENDER ;;; Ich setze das hier einfach vorraus

ParticleName = EXRing01.tga ;;; für ein gleiches Ergebnis wäre es Vorteilhaft ihr nehmt das auch so...

Lifetime = 150 150 ;;;; 150? Jap, wir wollten den Particle 5sek lang leben lassen

;;;  $\rightarrow 1 \text{ sec} = 30 \text{ Frames} \rightarrow 30 \text{ Frames} \times 5 = 150 \text{ Frames} \rightarrow \text{Lifetime} = 150 \text{ 150}$

SystemLifetime = 1 1 ;;; will ja nur 1 Particle spawnen, darum brauche ich kein System das mehr

;;; als einen Frame lebt....

SortLevel = 1

Size = 50 50 ;;; ich hab 50 gefordert, ich hab 50 rein geschrieben

BurstCount = 1 1 ;;;; Particle pro Welle, wir wollen ja nur einen..

;;hier muss noch BurstDelay rein:

BurstDelay = 5 5;;Der wert muss höher sein als die SystemLifetime, damit auch sicher nur einmal gespawnt wird...

End

Color = DefaultColor

;;; Hier brauchen wir natürlich die Farbangabe für Rot, ich ahbe hier einfach mal Maximales Rot gewählt

Color1 = R:255 G:0 B:0 0

End

Update = DefaultUpdate ;hier schreiben wir generell 1, wir wollen ja nicht das sich der Particle bewegt

SizeRate = 0 0

SizeRateDamping = 1 1

AngleZ = 0 0

AngularDamping = 1 1

AngularDampingXY = 1 1

End

Physics = DefaultPhysics

;;; auch hier bedarf es keines Eintrags, wir wollen den Particle unbeweglich

End

EmissionVelocity = OutwardEmissionVelocity

;;; Hier ist auch kein Eintrag notwendig

End

EmissionVolume = PointEmissionVolume

IsHollow = Yes

;;; Dieser Block braucht keine Änderung...

End

Draw = DefaultDraw

End

End

So, wir haben jetzt eigentlich alle wichtigen Werte in dem System- und in dem Color-Block gemacht, die anderen Blocks haben wir ignoriert oder mit 0 0 bzw. 1 1 besetzt ( $\rightarrow$  es bewegt sich nichts)

### Und weiter geht's:

Wenn ihr die Erste Aufgabe nicht 100% lösen konntet, dann wäre es jetzt angebracht die Lösung so lange zu studieren (indem ihr mit dem 1. Tutorial vergleicht), bis ihr wisst warum die Änderungen so sein müssen.

Solltet ihr beim besten Willen nicht nachvollziehen können warum das alles so ist, dann scheut euch nicht im Forum ( <http://www.modding-union.com> ) oder in ICQ zu fragen ;-)

Als nächstes werden wir den selben Particle einen Farbwechsel verpassen.

Er soll jetzt ganz am Anfang (Frame 0) rot gefärbt sein, und seine Farbe bis zu Sekunde 3 in Grün ändern.

Dazu müssen wir nur den Color Block ändern:

```
Color = DefaultColor
      Color1 = R:255 G:0 B:0 0
;;zweite Zeile bitte ergänzen ;-)
```

End

Ich gehe davon aus das ihr meine Lösung vom ersten Teil übernommen habt, mit unterschiedlichen Ergebnissen lässt es sich nämlich schlecht arbeiten.

Das ihr fertig seit merkt ihr daran das sich die Farbe die ersten 3 Sekunden permanent ändert und in der 4 und 5 gleich bleibt.

## ACHTUNG: TUT EUCH EINEN GEFALLEN UND LÖST DIE AUFGABE ERST SELBST ;-)

(den Satz werdet ihr noch öfter lesen^^)

Also dieses Mal kein Screenshot, da der Code eigentlich reichen sollte:

```
Color = DefaultColor
    Color1 = R:255 G:0      B:0 0
    Color2 = R:0      G:255 B:0 90
End
```

Ich denke das ganze ist selbsterklärend, aber hier nochmal zur Erinnerung:

Wir können beliebig viele Farbbänderungen einbauen indem wir neue „Color“-Zeilen einfügen.

Dabei geben wir wie bei der Color1-Zeile einen RGB-Farbwert an.

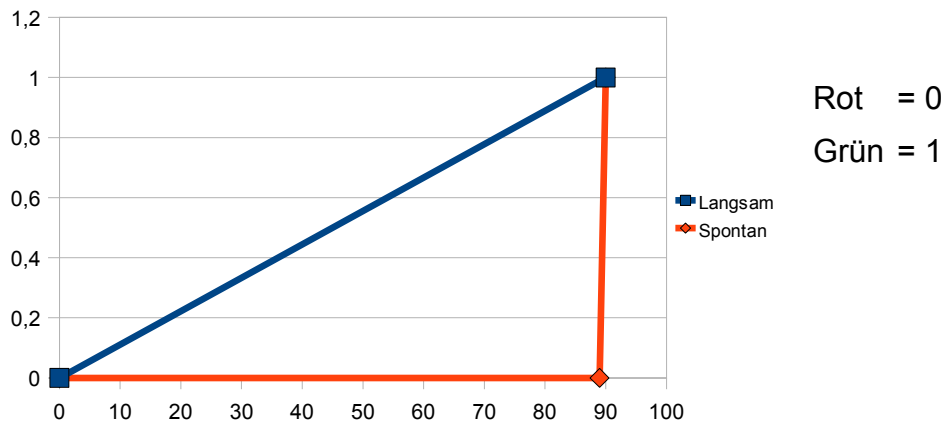
Die 90 hinter der Zeile sagt dem Spiel wann diese Farbe erreicht sein soll. Ich habe nach 3 Sekunden verlangt, also 30 Frames pro Sekunde x3 => 90 Frames

Der Farbwechsel läuft jetzt so ab, das von Frame 0 bis Frame 90 langsam die Farbe wechselt.

Wenn wir jetzt aber einen spontanen Farbwechsel wollen, also von einem Frame auf den nächsten, müssen wir das so angeben:

```
Color = DefaultColor
    Color1 = R:255 G:0      B:0 0
    Color2 = R:255 G:0      B:0 89
    Color3 = R:0      G:255 B:0 90
End
```

Hab das Ganze mal als Diagramm dargestellt:



In der langsamen Variante ist der Farbwechsel gleichmäßig über die Zeit 0-90 verteilt, dabei nimmt der Particle auch andere Farbwerte an welche zwischen den Farben liegen.

Bei der Spontanen Variante benutze ich 3 Color-Zeilen, wobei der Farbwechsel dann in einem Frame vollzogen wird. Ich hoffe man erkennt das gut im Diagramm^^

Wir haben für die Aufgabe vorhin allerdings die langsame Variante mit nur 2 Color-Zeilen gewählt, ich wollte nur einen einfachen Weg für einen spontanen Farbwechsel aufzeigen.

## Das bunte Wort der Macht

Nun machen wir uns schon an eine kleine FX ran, die man sogar verwenden kann.  
Die Aufgabe ist wie folgt:

Ändere die momentane FX so ab, das von Sekunde 3-5 die Farbe auf Blau gewechselt wird.  
Der Particle soll seine Größe im Laufe der FX um das 2 ½-Fache seiner Anfangsgröße vergrößern  
(er ist dann also 3 ½ so groß).  
Außerdem soll sich der Particle ab jetzt am Boden und nicht mehr an der Kamera orientieren.  
Momentan arbeiten wir mit diesem Code:

```
FXParticleSystem AlienSimpleParticle
System
  Priority = ALWAYS_RENDER
  ParticleName = EXRing01.tga
  Lifetime = 150 150
  SystemLifetime = 1 1
  SortLevel = 1
  Size = 50 50
  BurstCount = 1 1
  BurstDelay = 5 5
End
Color = DefaultColor
  Color1 = R:255 G:0 B:0 0
  Color2 = R:0 G:255 B:0 90
End
Update = DefaultUpdate
  SizeRate = 0 0
  SizeRateDamping = 1 1
  AngleZ = 0 0
  AngularDamping = 1 1
  AngularDampingXY = 1 1
End
Physics = DefaultPhysics
End
EmissionVelocity = OutwardEmissionVelocity
End
EmissionVolume = PointEmissionVolume
  IsHollow = Yes
End
Draw = DefaultDraw
End
End
```

## ACHTUNG: TUT EUCH EINEN GEFALLEN UND LÖST DIE AUFGABE ERST SELBST ;-)

(hab's doch gesagt^^)

So den Color-Block erweitern wir erst mal um eine Zeile, das Prinzip sollte mittlerweile bekannt sein, wem das noch etwas unklar ist, der soll sich bitte Seite 5 nochmal ansehen und den dazugehörigen Teil des FX-Grundlagen-Tutorials lesen.

### Unser jetziger Color-Block

```
Color = DefaultColor
    Color1 = R:255 G:0    B:0    0
    Color2 = R:0    G:255 B:0    90
    Color3 = R:0    G:0    B:255 150
End
```

Die Size (spekulativ):

Die Größe wäre bei 3 ½-Facher Größe dann 175, und 50 ist die Anfangsgröße.

→  $175 - 50 = \text{Vergrößerung} = 125$

Das ganze geht über 5 Sekunden

→  $125 : 5 = \text{Vergrößerung pro Sekunde} = 25$

Unser SizeRate wäre demzufolge 2,5.

Da ich aber nie SizeRate mit Bruchzahlen angebe, gebe ich

SizeRate = 5 5 und

SizeRateDamping = 0.5 0.5

Was dann wieder 2.5 entspricht.

Dazu sei gesagt das sich diese Berechnung auf Schätzwerte meinerseits stützt (wie im FX-Grundlagen-Tutorial beschrieben), und ich selbst bei FX eigentlich so lange probiere bis die richtige Geschwindigkeit erreicht ist.

Soll heißen: Ich selbst habe eigentlich nie gerechnet was für ein Ergebnis ich erreichen möchte, ich habe bereits im Gefühl wie groß die Particle bei gewissen Werten werden. Die Theorie das sich bei SizeRate = 2 2 die Größe um 20 ändert ist eine Schätzung die ich für das Grundlagen-Tutorial aufgestellt habe, und hoffentlich stimmt (schließlich beruht diese Rechen-Lösung darauf).

Zur Erinnerung, der Auszug aus dem Grundlagen-Tutorial:

SizeRate \* SizeRateDamping = „Wahre Vergrößerungsgeschwindigkeit“

SizeRate = 2 2 und Damping = 1 1, bewirken das sich der Durchmesser des Particles um 20 20 in einer Sekunde (also 30 Frames) erweitert.

Der Particle soll jetzt auch noch parallel zum Boden laufen, dazu fügen wir diese Zeile im System-Block ein:

IsGroundAligned = Yes

Damit richten sich die Partikel horizontal zur Spielwelt aus (der oben gebrauchte Ausdruck „parallel zum Boden“ ist daher eigentlich genau genommen falsch).

Also unser Ergebnis würde dann so aussehen:

```
FXParticleSystem AlienSimpleParticle
System
  Priority = ALWAYS_RENDER
  ParticleName = EXRing01.tga
  Lifetime = 150 150
  SystemLifetime = 1 1
  SortLevel = 1
  Size = 50 50
  BurstCount = 1 1
  BurstDelay = 5 5
  IsGroundAligned = Yes
End
Color = DefaultColor
  Color1 = R:255 G:0 B:0 0
  Color2 = R:0 G:255 B:0 90
  Color3 = R:0 G:0 B:255 150
End
Update = DefaultUpdate
  SizeRate = 5 5
  SizeRateDamping = 0.5 0.5
  AngleZ = 0 0
  AngularDamping = 1 1
  AngularDampingXY = 1 1
End
Physics = DefaultPhysics
End
EmissionVelocity = OutwardEmissionVelocity
End
EmissionVolume = PointEmissionVolume
  IsHollow = Yes
End
Draw = DefaultDraw
End
End
```

Noch so ein Hinweis nebenbei:

Achtet möglichst darauf euren Code ebenso zu sortieren (Abstände, Reihenfolge, etc.) wie es EA Games getan hat, das schafft ein gewisses Gesamtbild, Übersicht, und andere Teammitglieder haben es leichter das Ganze zu deuten.



## Twice is nice

(aber 5 mal wär auch nicht schlecht...)

Eine einfach Aufgabe:

Ab jetzt sollen zufällig zwischen 2 und 5 Particle gespawnt werden.

Da das aber langweilig wäre wenn die gleich schnell wären, müsst ihr auch noch einen Zufallswert für die Vergrößerung bestimmen, der Halbwegs sinnvoll ist.

Aja... verschieden groß sollen sie auch sein (bleibt aber im realistischen Rahmen^^).

Den Code von dem ich ausgehe findet ihr in der Lösung der vorangegangenen Aufgabe.

//LÖSUNG//

## **ACHTUNG: TUT EUCH EINEN GEFALLEN UND LÖST DIE AUFGABE ERST SELBST ;-)**

Ja, hier einfach mal gleich noch die Lösung:

FXParticleSystem AlienSimpleParticle

System

Priority = ALWAYS\_RENDER

ParticleName = EXRing01.tga

Lifetime = 150 150

SystemLifetime = 1 1

SortLevel = 1

Size = 30 50

BurstCount = 1 1

//Größe geändert auf 30 50

//Auf 2 5 geändert, wurde ja gefordert

//bei Fragen seht im „Grundbegriffe“-Tut nach

BurstDelay = 5 5

End

Color = DefaultColor

Color1 = R:255 G:0 B:0 0

Color2 = R:0 G:255 B:0 90

Color3 = R:0 G:0 B:255 150

End

Update = DefaultUpdate

SizeRate = 8 3

//ich habe nur die SizeRate geändert, ist

//überschaubarer

SizeRateDamping = 0.5 0.5

AngleZ = 0 0

AngularDamping = 1 1

AngularDampingXY = 1 1

End

Physics = DefaultPhysics

End

EmissionVelocity = OutwardEmissionVelocity

End

EmissionVolume = PointEmissionVolume

IsHollow = Yes

End

Draw = DefaultDraw

End

End

### Strukturell statt punktuell!

Warum den eigentlich alles nur auf einen Punkt beschränken?

Das haben sich schlaue Neandertaler schon vor Jahrtausenden gedacht und angefangen ihre Steine als Kreis zu legen anstatt auf einen Haufen.

Soll heißen:

Wir beschäftigen uns jetzt damit die Particle jetzt nicht mehr auf einem Punkt sondern in einem Quadratischem Bereich spawnen zu lassen.

Dazu ändern wir das Emission Volume, seht in den Grundlagen nach wenn ihr euch nicht sicher seit was ihr reinschreiben müsst wenn ihr einen Würfel mit 40 Seitenlänge spawnen wollt.

Erhöht außerdem noch die Anzahl der Partikel auf 300 oder 400.

Beachtet: Der Würfel soll hohl sein.

Wir nehmen das hier als Grundlage (aus der Lösung von Seite 4), damit man die Spawn Position schöner sieht:

```
FXParticleSystem AlienSimpleParticle
System
  Priority = ALWAYS_RENDER
  ParticleName = EXRing01.tga
  Lifetime = 150 150
  SystemLifetime = 1 1
  SortLevel = 1
  Size = 50 50
  BurstCount = 1 1
  BurstDelay = 5 5
End
Color = DefaultColor
  Color1 = R:255 G:0 B:0 0
  Color2 = R:0 G:255 B:0 90
End
Update = DefaultUpdate
  SizeRate = 0 0
  SizeRateDamping = 1 1
  AngleZ = 0 0
  AngularDamping = 1 1
  AngularDampingXY = 1 1
End
Physics = DefaultPhysics
End
EmissionVelocity = OutwardEmissionVelocity
End
EmissionVolume = PointEmissionVolume
  IsHollow = Yes
End
Draw = DefaultDraw
End
End
```

## ACHTUNG: TUT EUCH EINEN GEFALLEN UND LÖST DIE AUFGABE ERST SELBST ;-)

Hier ist dann auch schon die Lösung:

```
FXParticleSystem AlienSimpleParticle
System
    Priority = ALWAYS_RENDER
    ParticleName = EXRing01.tga
    Lifetime = 150 150
    SystemLifetime = 1 1
    SortLevel = 1
    Size = 50 50
    BurstCount = 1 1
    BurstDelay = 5 5
End
Color = DefaultColor
    Color1 = R:255 G:0    B:0 0
    Color2 = R:0    G:255 B:0 90
End
Update = DefaultUpdate
    SizeRate = 0 0
    SizeRateDamping = 1 1
    AngleZ = 0 0
    AngularDamping = 1 1
    AngularDampingXY = 1 1
End
Physics = DefaultPhysics
End
EmissionVelocity = OutwardEmissionVelocity
End
EmissionVolume = BoxEmissionVolume
    Halfsize = X:20 Y:20 Z:20
    IsHollow = Yes
End
Draw = DefaultDraw
End
End
```

Denkt daran das ihr nur den Halben Würfel seht, die andere Hälfte ist unter der Erde, diese Höhe könnt ihr allerdings nicht im Particlesystem ändern, sondern dies muss in der FXList geschehen.

Dies könnt ihr in „FXList Grundlagen und Einbau“ nachlesen.

Wir können in dem System nur mit der PointEmissionVolume einen „Ort“ angeben, dann wäre aber nicht mehr der Würfel als Spawn-Grundform möglich. Diese PointEmissionVolume-Variante macht zB bei manchen Heldenaren oder an Bones geheftete FX Sinn (zB. Die kleine Gold FX die den Ring an der Hand der Ringhelden in der Edain Mod darstellt; da man die FX an den Hand-Bone hängen kann, aber nicht an den Finger, kann man die Position der FX mit der PointEmissionVolume korrigieren)

Anmerkung:

Gebt ruhig mal andere Werte für die Box an und seht sie euch damit nochmal an, so entwickelt ihr etwas Gefühl für die Größe der Fx.

Das Schlusswort (... eigtl die Schlussätze)

Generell hoffe ich doch mal das Ganze war verständlich, es handelte sich um FX an sich, die Anwendung und Einbindung in Helden/Gebäuden/usw. werde ich wohl wann anders behandeln, sollte aber für jeden halbwegs erfahreneren Coder eh kein Problem sein.

Greez  
Alien aka Infiltrator

(ACHTUNG: ES BESTEHT KEINERLEI GARANTIE AUF OBRIGEN INHALT, DER VERFASSEN STAND ZUM ZEITPUNKT DES VERFASSEN UNTER PERMANENTER METAL-BESCHALLUNG)