

Unityはじめるよ ~パーティクルについて~

統合開発環境を内蔵したゲームエンジン http://japan.unity3d.com/

※いろんな職業の方が見る資料なので説明を簡単にしてある部分があります。正確には本来の意味と違いますが上記理由のためです。ご了承ください。 この資料内の一部の画像、一部の文章はUnity公式サイトから引用しています。

今回は 「ちょっと進んだ使い方」 を紹介します。



■Unityのパーティクルシステム

Unityには「Shuriken」という、 パーティクルシステムが搭載されています。

パーティクルシステムはコンピュータグラフィック技術の ひとつで、従来型のレンダリング技術では再現が難しい、 ある種の曖昧さを持った事物をシミュレートするために使 われる。一般にパーティクル・システムを使って模写され る事物の例としては、炎、爆発、煙、流水、火花、落葉、 雲、霧、雪、埃、流星、毛髪、毛皮、草地、あるいは 光跡や呪文の視覚効果などが挙げられる。

ウィキペディアより引用

言わずもがなですが、 要は、

簡単にかっこいいエフェクトが作れますよ

ということです。

基本的な使い方は、今回は省略~。

公式ドキュメント http://docs.unity3d.com/ja/current/Manual/PartSysReference.html

今回はサブエミッタについて紹介します。

サブエミッタは、 **パーティクルからパーティクルを生み出す** 仕組みです。

よく使われるのが花火。

打ち上げのパーティクルから爆発のパーティクルを生み出し、そのパーティクルから更に キラキラのパーティクルを出すような感じ







パーティクルシステムのインスペクタビュー を見てみると、「SubEmitters」という項目 がある。その中の項目の意味は、

Birth

パーティクルが生成された時に、 新たなパーティクルシステムを生成

Collision

パーティクルが何かにぶつかった時に、 新たなパーティクルシステムを生成

Death

パーティクルの寿命が尽きたときに、 新たなパーティクルシステムを生成

🔻 😵 🔹 Particle System		🛐 🌣,
		Open Editor
Particle System (1)		
		0 +
Duration	5.00	
Looping	\checkmark	
Prewarm		
Start Delay	0	•
Start Lifetime	5	•
Start Speed	5	•
Start Size	1	•
3D Start Rotation		
Start Rotation	0	•
Randomize Rotation Direction	0	
Start Color	-	•
Gravity Modifier	0	
Simulation Space	Local	÷
Scaling Mode	Local	Ŧ
May Dautialan	¥.	
Max Particles	1000	
V Emission		
✓ Shape		
Velocity over Lifetime		
Limit Velocity over Lifetime		
Inherit Velocity		
Force over Lifetime		
Color over Lifetime		
Color by Speed		
Size over Lifetime		
Size by Speed		
Rotation over Lifetime		
Rotation by Speed		
External Forces		
Collision		
✓ Sub Emitters		
Birth None (Particle Sys	tem)	o +
None (Particle Sys	tem)	•
Collision None (Particle Sys	ollision None (Particle System)	
None (Particle Sys	tem)	· +
Death None (Particle Sys	tem)	⊙ +
None (Particle Sys	tem)	•
Tayture Sheet Animation		
✓ Renderer		

となる。

■サブエミッタの使い方②

Birth、Collision、Deathの項目に 生成したいパーティクルシステムを 指定してあげるだけでOK。

🗸 Sub Em	itters	
Birth	😵 Trails (Particle System)	0
	None (Particle System)	•
Collision	None (Particle System)	•
	None (Particle System)	•
Death	🗑 Smoke (Particle System)	0
	😵 Explosion SubEmitter (Particle System)	0

注意点

- SubEmitters/Deathを使う場合は、
 StartLifetimeの時間が
 経過したときである
- SubEmitters/Collisionを使う場合は、
 Collisionのを有効にする必要がある
 Collisionのを有効にする必要がある

🛛 😵 🛛 Particle System		Open Editor
_		
Fireworks		• +
Duration	5.00	
Looping	\checkmark	
Prewarm		
Start Delay	0	•
Start Lifetime	3	•
Start Speed	10	•
Start Size	3	•
3D Start Rotation		
Start Rotation	0	•
Randomize Rotation Direction	0	
Start Color		•
Gravity Modifier	0.2	
Simulation Space	Local	\$
Scaling Mode	Shape	\$
Play On Awake [≁]	~	
Max Particles	2	
V Emission		
✓ Shape		
Velocity over Lifetime		
 Limit Velocity over Lifetime 		
 Inherit Velocity 		
Force over Lifetime		
💿 Color over Lifetime		
Color by Speed		
✓ Size over Lifetime		
Size by Speed		
Rotation over Lifetime		
Rotation by Speed		
External Forces		
Collision		
✓ Sub Emitters		
Birth 😵 Trails (Particle)	System)	0
None (Particle Sys	None (Particle System)	
Collision None (Particle Sys	None (Particle System)	
None (Particle Sys	None (Particle System)	
eath Smoke (Particle System)		0
¥ Explosion SubEr	nitter (Particle System)	0
Texture Sheet Animetion		

ちょいと試してみよう 弾丸が壁にぶつかる様子を再現する

●弾っぽい見た目と動きを作る

- 1. パーティクルを作成
 - ・インスペクタービューで右クリック > ParticleSystem > 名前を「MachineGun」にする
- 2. 弾が一直線に飛ぶようにする
 ・「Shape」項目の「Angle」を0.1に、「Radius」を0.2に変更
- 3. 「StartSpeed」を上げて玉の速度を早くする
- 4. 弾が高速で飛んでるように見せる
 - ・「Renderer」項目の「RenderMode」を「Stretched Billboard」に変更
 - 「Material」を「matBullet」に変更
- 5. 壁を突き抜けてるので、「Collision」項目をONにする
 - ・「Planes」を「World」に変更
 - ・「Bounce」を0に変更
 - ・「LifeTime Loss」を1に変更

●銃弾痕をつくる

- 1. パーティクルを作成
 - ・インスペクタービューで右クリック > ParticleSystem > 名前を「BulletMark」にする
- 2. 動きと寿命を決める
 - ・動かないので「StartSpeed」を0にする。
 - ・弾が連射されてるので、たまりすぎないように「StartLifetime」を
 0.5くらいの短い値に変更
- 2. 見た目を変更
 - ・「Renderer」項目の「Material」を「matBulletMarks」に変更
- 3. パーティクルが一つしか出ないように変更
 - ・「Emission」項目の「Rate」を1にする。
 - ・「Bursts」の+ボタンを押し、TimeをMinを1 Maxを1にする
- 4.親子付け
 - ・インスペクタビューで「MachineGun」を選択
 - 「SubEmitters」項目の「Collision」の右の○を押して、
 さっき作ったパーティクル「BulletMark」を選択
 親子付けするか聞かれるので「Yes, Reparent」

●破片を作る①

- 1. パーティクルを作成
 - ・インスペクタービューで右クリック > ParticleSystem > 名前を「Debris」にする
- 2. 見た目を変更
 - ・「Renderer」項目の「RenderMode」を「Mesh」に変更(MeshはCube) 「Material」を「matDebri」に変更
- 3. 壁がはじけるような見た目を作るする
 - ・「Start Speed」を20にする
 - ・「Start Size」を「Random Between Two Constants」にし値を0.01、0.15にする
 - ・「Start Lifetime」を0.1に
 - 「3D Start Rotation」にチェックを入れ、
 「Start Rotation」を「Random Between Two Constants」にし、
 X, Y, Z値を0、360にする
 - 「Shape」項目を「Sphere」にする。

●破片を作る2

- 4. パーティクルがいっぺんに出るように変更
 - ・「Emission」項目の「Rate」を50にする。
 - ・「Bursts」の+ボタンを押し、TimeをMinを50 Maxを50にする
- 5.親子付け
 - ・インスペクタビューで「MachineGun」を選択
 - ・「SubEmitters」項目の「Collision」の右の○を押して、
 さっき作ったパーティクル「Debris」を選択
 親子付けするか聞かれるので「Yes, Reparent」

完成! 弾丸が壁にぶつかり、銃弾痕を残しながら、 壁がはじけ飛ぶ様子が再現できた!



■まとめ

サブエミッタを使うことで、 パーティクルシステムからパーティクルシステムの連携を 簡単に行うことができる。

スクリプトを使わずともエフェクトをスポーンできるため、 管理が非常に楽になる。

今回は紹介しなかったけど、 Collision周りはスクリプトをコールバックで呼ぶこともでき るため、より複雑な仕組みも組みこめる。

ご清聴ありがとうございました