

Manual

概要

ParticleBulletSystem は簡単にパーティクルを扱うための機能です。

安くて、高速で、簡単です。

アセットストアにて 9.5 ドルで販売しています。

TPS と古風なシューティングゲーム、どちらにも使用できます。

弾や爆発エフェクトにお使いください。

- C# と PlayMaker で使用できます。
- 低スペック PC においても、数万発の弾を表示可能です。
- 指定した座標やベクトルでパーティクルを発射できます。
- パーティクルのベクトルを制御できます。
- 弾以外に、爆発エフェクトにも使用できます。
- 被弾してダメージを受けることができます。
- ParticleSystem ごとに、ダメージ量を設定できます。
- ParticleSystem を番号で参照できます。
- パーティクルの描画負荷を減らせます。

例えば爆発のような、

まったく同じ ParticleSystem が多数ある場合、1 つの ParticleSystem で描画できます。

要約すると、以下の方にお勧めです。

- とにかく動作の速い弾が欲しい。
- PlayMaker で ParticleSystem を扱いやすくしたい。
- 爆発の描画負荷を減らしたい。
- 好きなタイミングでパーティクルを使いたい。

簡単な使い方

1. シーンに、ParticleSystem と空のゲームオブジェクトを作成します。
2. 作成したゲームオブジェクトにコンポーネント”ParticleManager”を追加します。
3. ParticleManager の Particle(変数名)に、さきほど作成した ParticleSystem をドラッグ&ドロップします。
4. ParticleManager の Power(変数名)の size を 1 にします。
5. これで完了です。後述の各機能が使用できます。

注意

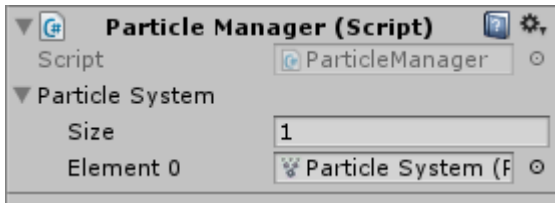
ParticleSystem の設定で、StretchedBillboard を用いる場合。

ParticleSystem の Render 設定の Pivot の Y で、位置を合わせてください。

これをしないと、衝突判定の位置がずれます。

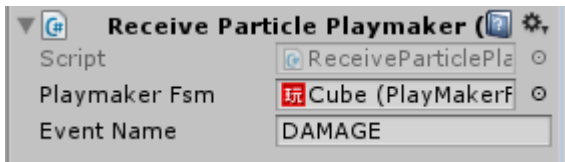
機能

ParticleManager



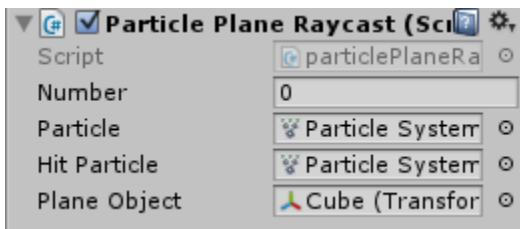
- これは静的なコンポーネントです. シーンに一つだけ存在させてください.
- ParticleManager は ParticleBulletSystem の大部分の機能を持っています.
これがパーティクルを発射, 制御します.
- これに, あらかじめ ParticleSystem を格納して, 使用します.
- これに格納した ParticleSystem は, 番号で参照できます.
- 詳細な使い方は ParticleManager.pdf を参照してください.

ReceiveParticle



- これは, パーティクルでダメージを受けるための機能です.
- 通常の C# と, PlayMaker 用の 2 種類を用意してあります.
- Unity 標準の ParticleSystem の Collision を使用します.
- 被弾すると, 指定した Message もしくは Event を,
指定したゲームオブジェクトもしくは Fsm に送ります.
- ParticleSystem の Collision は, ワールド座標をしようしてください.

ParticlePlaneRaycast



- これは, 弾の行き先にターゲットサイトを表示する機能です.

設定

- Number “Particle”を ParticleManager で参照するための番号.
- Particle この弾の行き先に, 照準を表示します. これを直接指定している場合, Number は無視される.
- HitParticle ターゲットサイトの ParticleSystem.
- PlaneObject PlaneRaycast で使用する. この Transform の right と up に広がる平面で判定.

ParticleBillboardRotation

- パーティクルの画像の回転を, 移動に合わせる機能です.
- 直接, パーティクルシステムにアタッチして使います.

設定

- Cam 使用するカメラ. これを設定しない場合, 自動で MainCamera が適用されます.

ParticlePaster



複数の Transform にパーティクルを貼り付けます。

ターゲットサイトや弾の描画に使用します。

RotationOverLifeTime 等のパーティクルシステムの標準機能を使用可能です。

これは Mesh Renderer のダイナミック・バッチングより高速です。

設定

Particle 使用するパーティクルシステム

AutoRemove 非アクティブなオブジェクトの描画をやめます。

使用方法

ゲームオブジェクトに ParticlePaster をアタッチ

ほかのスクリプトから ParticlePaster を参照, Add メソッドで Transform を追加します。

これで, 追加した Transform の座標にパーティクルを描画します。

Transform の削除は Remove メソッドで行います。

具体的な使用方法は, Demo4 のスクリプト examplePaster で確認できます。

ParticlePaster_LifeTime

基本的に ParticlePaster と同じです。

パーティクルの寿命が反映されます。

ParticleCheckSphere

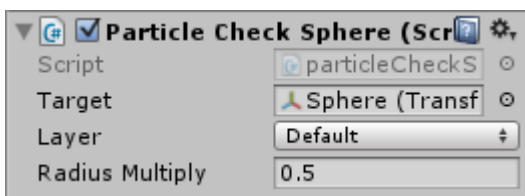
Physics.CheckSphere による衝突判定を行います。

弾の判定に太さが必要な場合に使用します。

ReceiveParticle は必要ありません。

あらかじめダメージを与える対象を決定する必要があります。

Demo3 のオレンジ弾に使用しています。



パラメーター

Target ダメージを与える対象です。

Layer 衝突判定のレイヤーです。

RadiusMultiply パーティクルに対する, 判定の大きさの倍率です。

使用方法(敵の弾に使用する場合)

弾のパーティクルシステムに ParticleCheckSphere をアタッチします。

Target をプレイヤーに設定します。

Layer をプレイヤーと同じレイヤーにします。(わからなければ Everything を選ぶ)