

ParticleManager

ParticleManager の主な機能は以下の3つです.

- ・番号による ParticleSystem の参照
- ・[スクリプトや PlayMaker での Emit, Shot 関係の機能](#)
- ・スクリプトや PlayMaker での[ベクトル制御](#)

メソッドの呼び出し方

- ・Version1.03 からは, あなたのスクリプトの先頭に以下のコードを書いてください.

```
using ParticleBulletSystem;
```

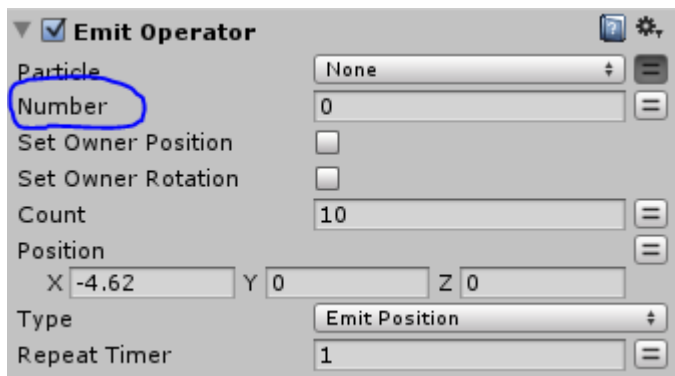
- ・スクリプトで使用する場合は以下のようにメソッドを呼び出して使用します.
- ・`ParticleManager.manager.使いたい機能()`;
- ・一つ目の引数は ParticleSystem を参照するための番号(int 型)です.
- ・一つ目の引数に直接 ParticleSystem を設定することも出来ます.
(ParticleManager で管理していないものを扱う場合に使用する)

例 OneShot を使用する場合

```
ParticleManager.manager.Emit_OneShot(0,transform,4,0.5f);
```

番号による ParticleSystem の参照の仕方.

- ・ParticleManager のほとんどの機能は, ParticleSystem を参照して使います.
- ・PlayMaker の場合, インスペクタで番号を指定できます. Number という変数です.
“Particle”が None (なし) ではない場合 Number は無視されます.



- ・C#の場合以下のように参照します.

```
ParticleSystem par = ParticleManager.manager.particle[0];
```

C#において通常, 上記のように参照することはありません.

メソッドを呼び出す際, 引数で番号を指定できるためです.

```
ParticleManager.manager.Vector_Rotate(0,20*Time.deltaTime);
```

スクリプトでも, 番号を使わず直接パーティクルを使用できます.

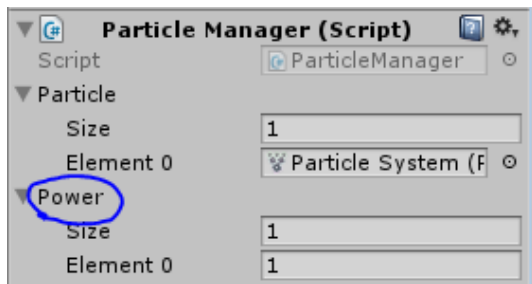
引数の名前はすべてのメソッドで一貫して, number としている.

ParticleSystem の引数名は par です.

この PDF において, パーティクルのことを弾と呼ぶことがあります.

威力の設定

変数”power”で設定します。



Particle(変数名)と同じ数だけ用意します。

パーティクルシステムごとに威力を設定します。

(攻撃以外の用途に使う場合も設定してください)

スクリプトや PlayMaker での Emit

Emit

```
public void Emit (int number,int count)
public void Emit (ParticleSystem par,int count)
public void Emit (int number,int count, Vector3 position)
public void Emit (ParticleSystem par,int count, Vector3 position)
public void Emit (int number,int count, Quaternion rotation)
public void Emit (ParticleSystem par,int count, Quaternion rotation)
public void Emit (int number,int count, Vector3 position, Quaternion rotation)
public void Emit (ParticleSystem par,int count, Vector3 position, Quaternion rotation)
```

説明

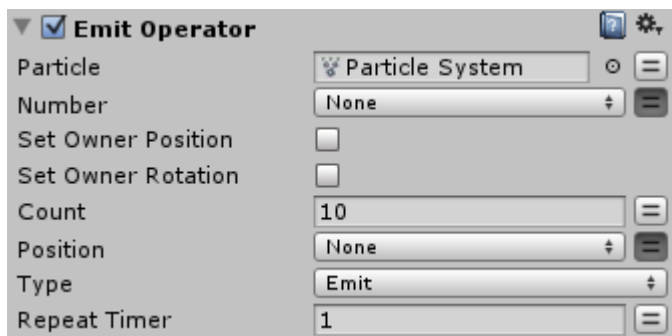
- 単純に ParticleSystem の設定を維持したまま Emit します.
- C#では ParticleSystem の座標と角度を設定することも出来ます.
- PlayMaker では EmitOperator で行います.
座標と角度を Owner に合わせるかを選択できます.RepeatTimer で繰り返しの時間間隔を指定します.

パラメーター(スクリプト)

- count Emit するパーティクルの数
- position ParticleSystem の座標 (ParticleSystem のゲームオブジェクトを移動する)
- rotation ParticleSystem の角度 (ParticleSystem のゲームオブジェクトを回転する)

パラメーター(PlayMaker)

- SetOwnerPosition ParticleSystem の座標を Owner に合わせるか
- SetOwnerRotation ParticleSystem の角度を Owner に合わせるか
- Count Emit するパーティクルの数
- Position ParticleSystem の座標.
- Type Emit Position 変数を使用しない.
 EmitPosition Position 変数を使用する.
- RepeatTimer 繰り返しの時間間隔を指定します.



備考

SetOwnerPosition と EmitPosition を同時に使用した場合, SetOwnerPosition が優先されます.
ParticleManager の Emit と EmitOperator は別のスクリプトです.

Shot 関係の機能

説明

2D シューティングもしくは 2.5D シューティング用の Emit 機能.

シューティングゲーム(STG)らしい形でパーティクルを発射する.

パラメーター

- | | |
|---------------|--------------------------|
| 1. speed | Emit するパーティクルの移動スピード |
| 2. speedRate | Emit するパーティクルの移動スピードの上昇率 |
| 3. size | Emit するパーティクルの大きさ |
| 4. SizeRate | Emit するパーティクルの大きさの上昇率 |
| 5. figuarSize | 弾の描く図形の大きさ |
| 6. lifetime | Emit するパーティクルの寿命 |
| 7. rows | 弾の描く図形の行の数 |
| 8. cols | 弾の描く図形の列の数 |

ShotOperator

説明

- ・ PlayMaker で Shot 関係の機能を扱うためのカスタムアクションです.

パラメーター

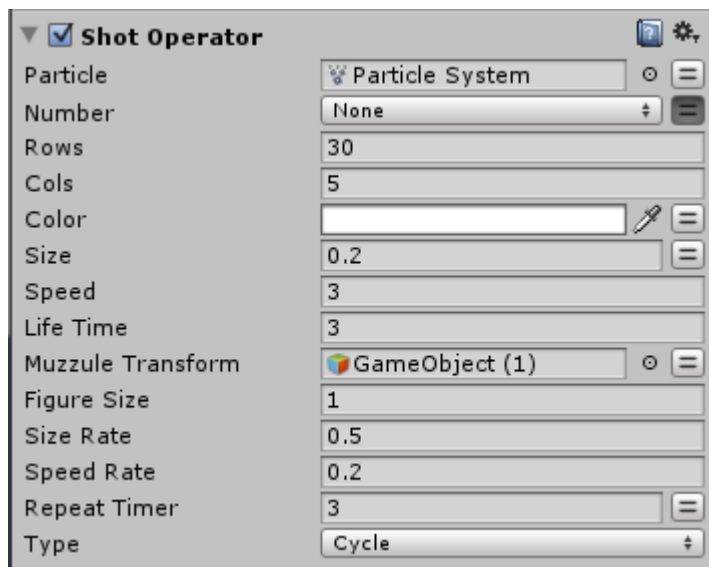
- ・ 各パラメーターの名称はスクリプトと同じ
- 詳細は後述の各 Shot 機能を参照してください.

使い方(スクリプト)

`ParticleManager.manager.Shot_使いたい機能`
でメソッドを呼び出してください.

使い方(PlayMaker)

1. PlayMakerEditor のアクションブラウザから,
カテゴリ ParticleBulletSystem の ShotOperator を選択して追加します.
2. Type で使いたい Shot の種類を選択します.
3. 各パラメーターを設定します.



OneShot

```
public void Emit_OneShot(int number, Transform tr, float speed, float size)
public void Emit_OneShot(ParticleSystem par, Transform tr, float speed, float size)
```

説明

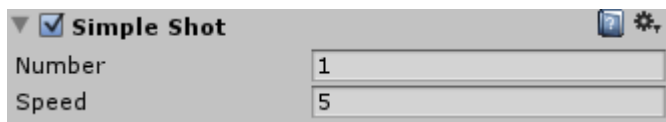
- Transform の正面に向けて Emit します.
- PlayMaker では SimpleShot もしくは、前述の ShotOperator の OneShot で行います.
- Shot 関係の機能では最もシンプルなもの.

パラメーター(スクリプト)

- tr Emit する座標と角度の Transform.
- speed Emit するパーティクルの移動スピード
- size Emit するパーティクルの大きさ

パラメーター(PlayMaker)

- speed Emit するパーティクルの移動スピード



備考

SimpleShot はリピート機能がありません。状態がアクティブになった瞬間だけ発射します。
ShotOperator では Size を指定できます。

ShotSelector

```
public void ShotSelector(shotType type, ShotParameter param, Transform tr)
public void ShotSelector(shotType type, ShotParameter param, Vector3 position, Vector3 forward, Vector3 upVec)
```

説明

- ショット機能を, Type で選び使用します.
 - ショット機能の中で, 最も汎用性に優れています.
- 特に理由がなければ, ShotSelector の使用を推奨します.

example スクリプトの ParticleShooter が参考になるでしょう.

パラメーター

1. type 使用するショットの種類
2. param ショットのパラメーター(前ページ参照)
3. tr Emit する座標と角度の Transform.

MultiSpeed

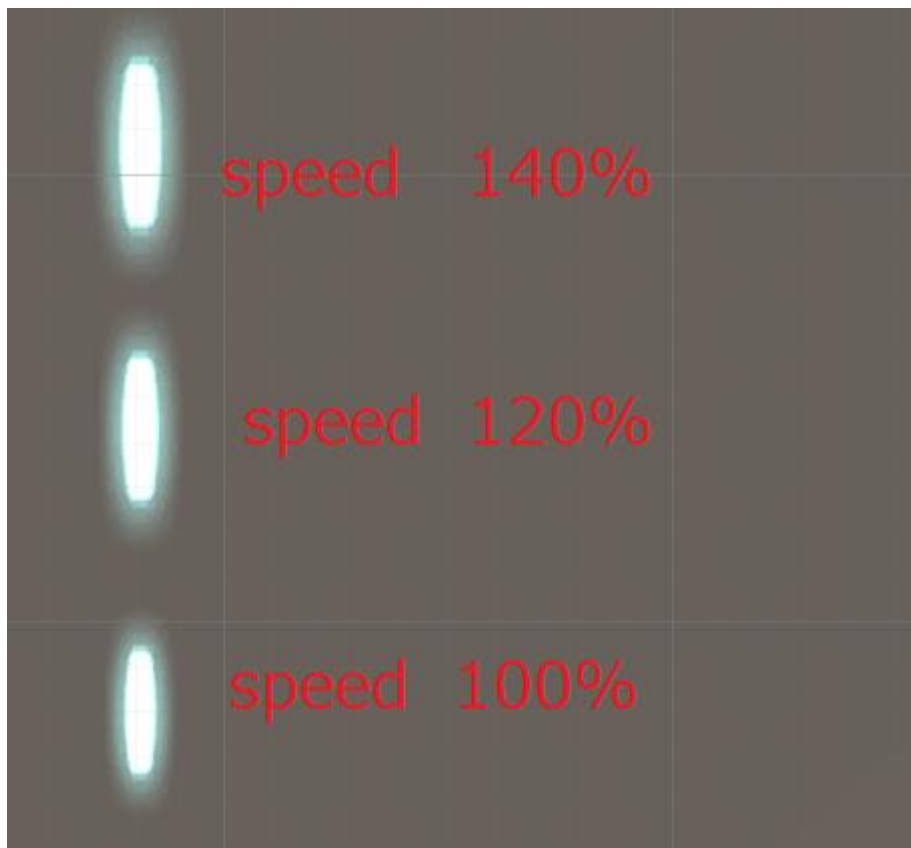
```
public void Emit_MultiSpeed(int number, Transform tr, float speed, float speedRate, float size, float sizeRate, float lifetime, int rows)
public void Emit_MultiSpeed(ParticleSystem par, Transform tr, float speed, float speedRate, float size, float sizeRate, float lifetime, int rows)
```

説明

- 複数のパーティクルを複数のスピードで **Emit** します.
- PlayMaker では ShotOperator の MultiSpeed で行います.

パラメーター

- | | |
|--------------|--|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | 1 発目のパーティクルの移動スピード |
| 3. speedRate | パーティクルの移動スピードの上昇率
0.2 なら, +20%ずつ増える |
| 4. size | パーティクルの大きさ |
| 5. sizeRate | パーティクルの大きさの上昇率
0.2 なら, +20%ずつ増える |
| 6. lifetime | パーティクルの寿命 |
| 7. rows | 弾の描く図形の行の数 |



MultiWay

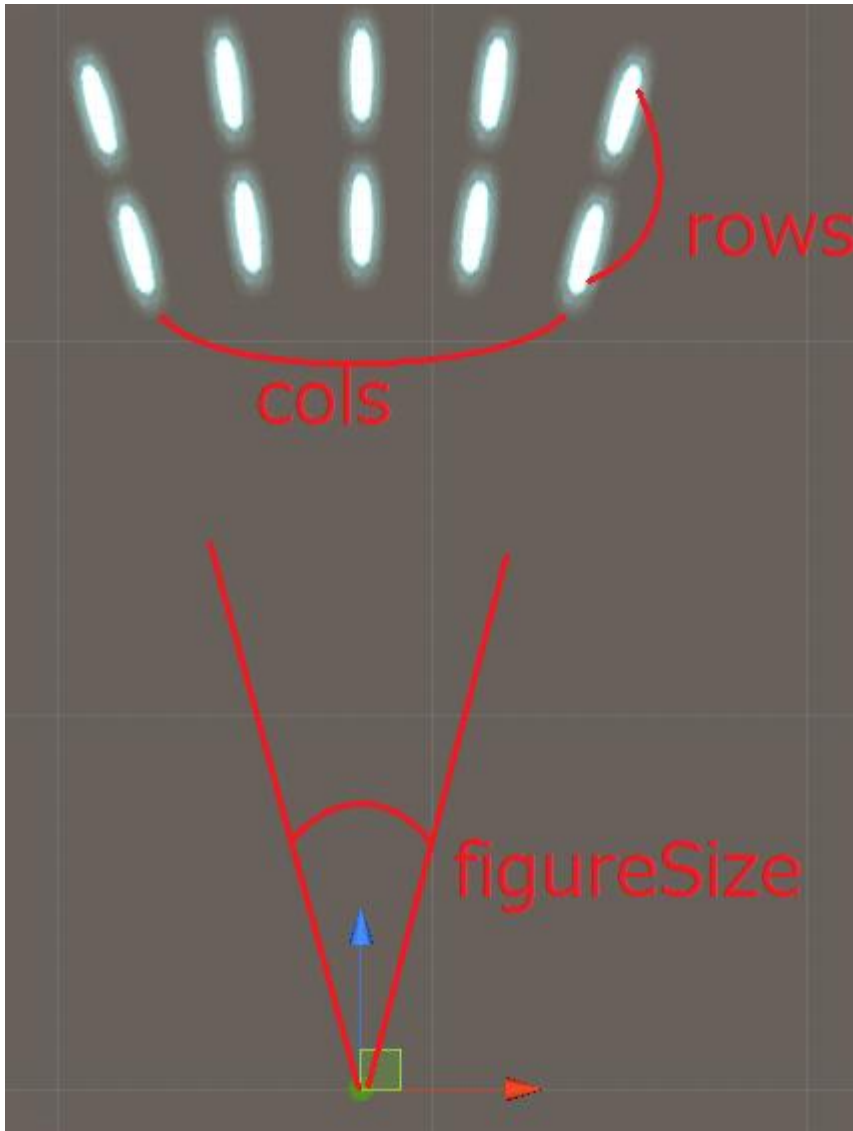
```
public void Emit_MultiWay(int number, Transform tr, float speed, float speedRate, float size,
float SizeRate, float figureSize, float lifetime, int rows, int cols)
public void Emit_MultiWay(ParticleSystem par, Transform tr, float speed, float speedRate,
float size, float SizeRate, float figureSize, float lifetime, int rows, int cols)
```

説明

- ・上記の MultiSpeed を複数の角度で発射します
- ・PlayMaker では ShotOperator の MultiWay で行います.

パラメーター

- | | |
|---------------|---------------------------------------|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | 1 行目のパーティクルの移動スピード |
| 3. speedRate | パーティクルの移動スピードの上昇率 (0.2 なら, +20%ずつ増える) |
| 4. size | パーティクルの大きさ |
| 5. sizeRate | パーティクルの大きさの上昇率 (0.2 なら, +20%ずつ増える) |
| 6. figureSize | 弾が描く弧の大きさ(30 なら 30 度) |
| 7. lifetime | パーティクルの寿命 |
| 8. rows | 弾の描く図形の行の数(MultiSpeed と同様) |
| 9. cols | 弾の描く図形の列の数(この数だけ MultiSpeed を発射する) |



CycleShot

```
public void Emit_CycleShot(int number, Transform tr, float speed, float speedRate, float size,
float sizeRate, float lifetime, int rows, int cols)
public void Emit_CycleShot(ParticleSystem par, Transform tr, float speed, float speedRate, float
at size, float sizeRate, float lifetime, int rows, int cols)
```

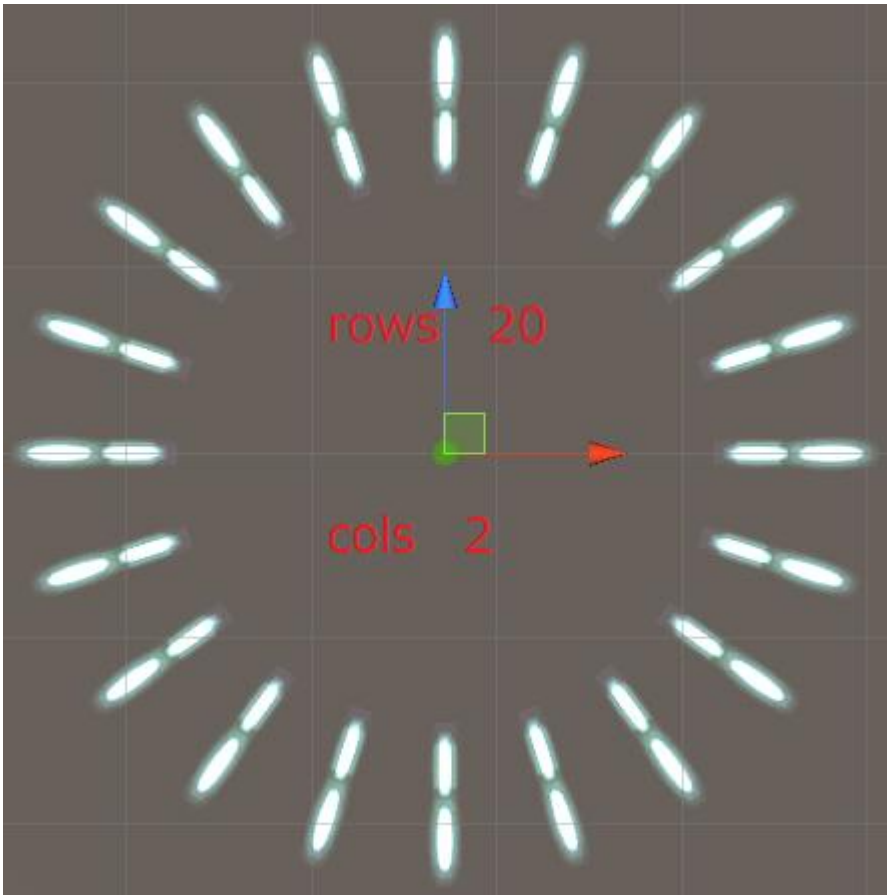
説明

- ・円形にパーティクルを発射します
- ・PlayMaker では ShotOperator の CycleShot で行います.

パラメーター

- | | |
|--------------|--|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | 1 行目のパーティクルの移動スピード |
| 3. speedRate | パーティクルの移動スピードの上昇率
0.2 なら, +20%ずつ増える |
| 4. size | パーティクルの大きさ |
| 5. sizeRate | パーティクルの大きさの上昇率
0.2 なら, +20%ずつ増える |

- | | |
|-------------|------------|
| 6. lifetime | パーティクルの寿命 |
| 7. rows | 弾の描く図形の行の数 |
| 8. cols | 弾の描く図形の列の数 |



RingShot

```
public void Emit_RingShot(int number, Transform tr, float speed, float speedRate, float size, float SizeRate, float figureSize, float lifetime, int rows, int cols)
```

```
public void Emit_RingShot(ParticleSystem par, Transform tr, float speed, float speedRate, float size, float SizeRate, float figureSize, float lifetime, int rows, int cols)
```

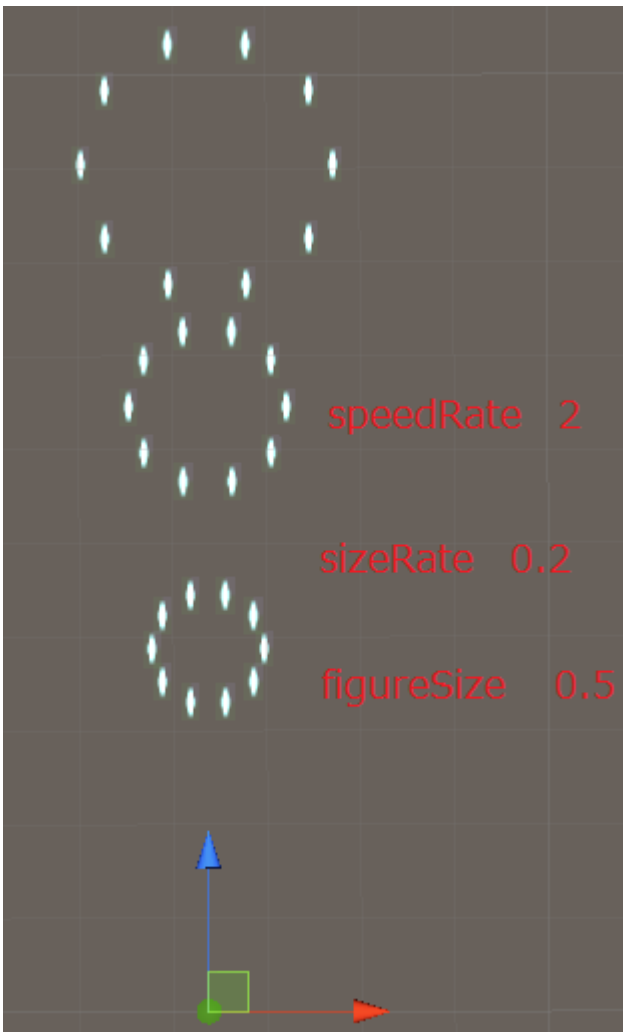
説明

- ・円形に配置したパーティクルを、複数の速度で発射します
- ・CycleShot と MultiSpeed を組み合わせたようなものです.
- ・リングの大きさは発射時点で固定です. CycleShot とは違い、拡散しません.
- ・PlayMaker では ShotOperator の RingShot で行います.

パラメーター

- | | |
|---------------|-------------------------------------|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | 1 個目のリングの移動スピード |
| 3. speedRate | リングの移動スピードの上昇率
0.2 なら, +20%ずつ増える |
| 4. size | パーティクルの大きさ |
| 5. sizeRate | リングの大きさの上昇率(0.2 なら, +20%ずつ増える) |
| 6. figureSize | 1 発目のリングの大きさ |

- | | |
|-------------|-----------------------------------|
| 7. lifetime | パーティクルの寿命 |
| 8. rows | 弾の描く円の行の数 |
| 9. cols | 発射する円の個数(MultiSpeed の rows に該当する) |



SquareShot

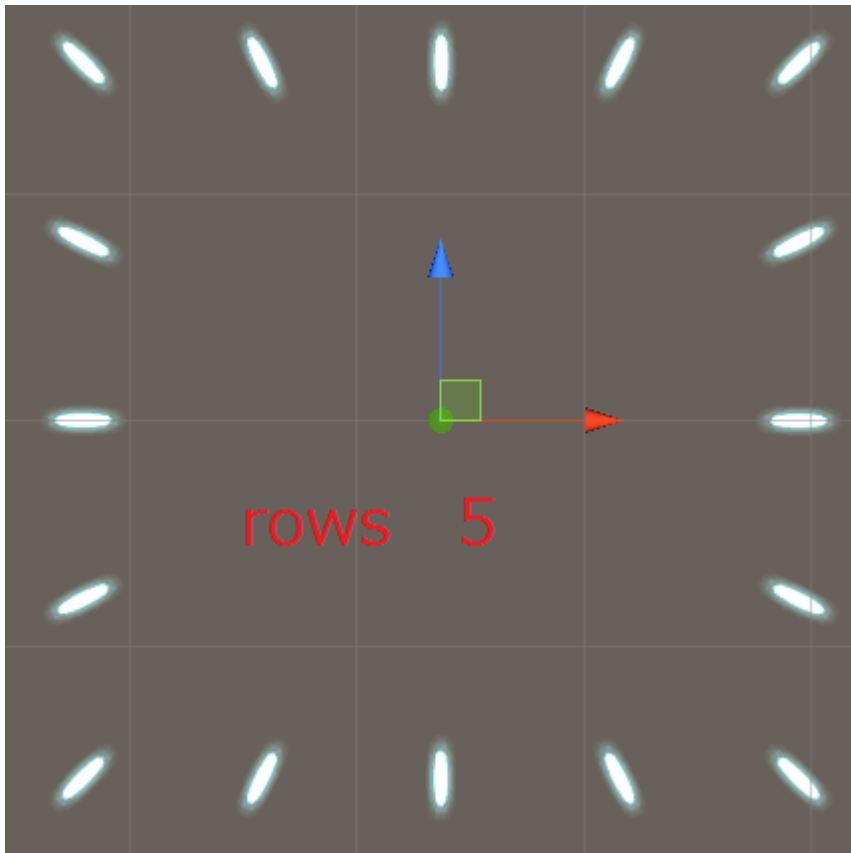
```
public void Emit_SquareShot(int number, Transform tr, float speed, float size, float lifetime,
int rows)
public void Emit_SquareShot(ParticleSystem par, Transform tr, float speed, float size, float lifetime, int rows)
```

説明

- ・トランスフォームを中心に、四角形にパーティクルを発射します
- ・PlayMaker では ShotOperator の SquareShot で行います.

パラメーター

- | | |
|-------------|-----------------------------|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | 1 行目のパーティクルの移動スピード |
| 3. size | パーティクルの大きさ |
| 4. lifetime | パーティクルの寿命 |
| 5. rows | 弾の描く四角形の、1 辺の弾の数
最低 3 必要 |



TriangleShot

```
public void Emit_TriangleShot(int number, Transform tr, float speed, float speedRate, float size, float figureSize, float lifetime, int rows)
public void Emit_TriangleShot(ParticleSystem par, Transform tr, float speed, float speedRate, float size, float figureSize, float lifetime, int rows)
```

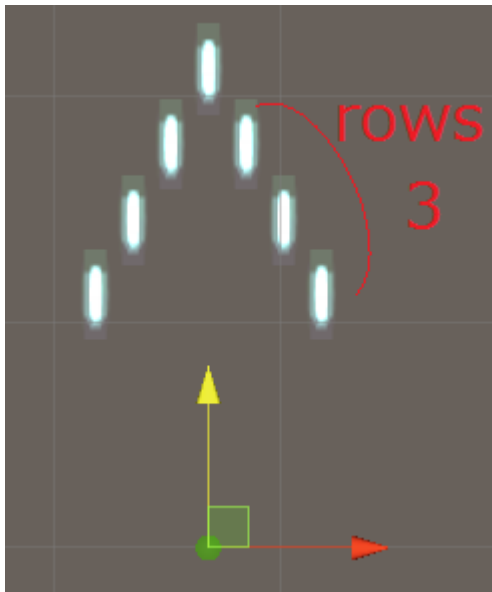
説明

- ・ <型にパーティクルを発射します
- ・ PlayMaker では ShotOperator の TriangleShot で行います.
- ・ 発射時点で図形の大きさが決まっています.

パラメーター

- | | |
|--------------|-----------------------------|
| 1. tr | Emit する座標と角度の Transform. |
| 2. speed | パーティクルの移動スピード |
| 3. speedRate | 図形を発射する位置に影響. Tr の Z 軸での座標. |
| 4. size | パーティクルの大きさ |

- | | |
|----------------------------|---------------------------|
| 5. <code>figureSize</code> | 図形の大きさに影響する.マイナスで図形が反転する. |
| 6. <code>lifetime</code> | パーティクルの寿命 |
| 7. <code>rows</code> | 弾の描く図形の弾の数 |



TriangleSpread

```
public void Emit_TriangleSpread(int number, Transform tr, float speed, float speedRate, float size,
float figureSize, float lifetime, int rows, int cols)
```

```
public void Emit_TriangleSpread(ParticleSystem par, Transform tr, float speed, float speedRate,
float size, float figureSize, float lifetime, int rows, int cols)
```

説明

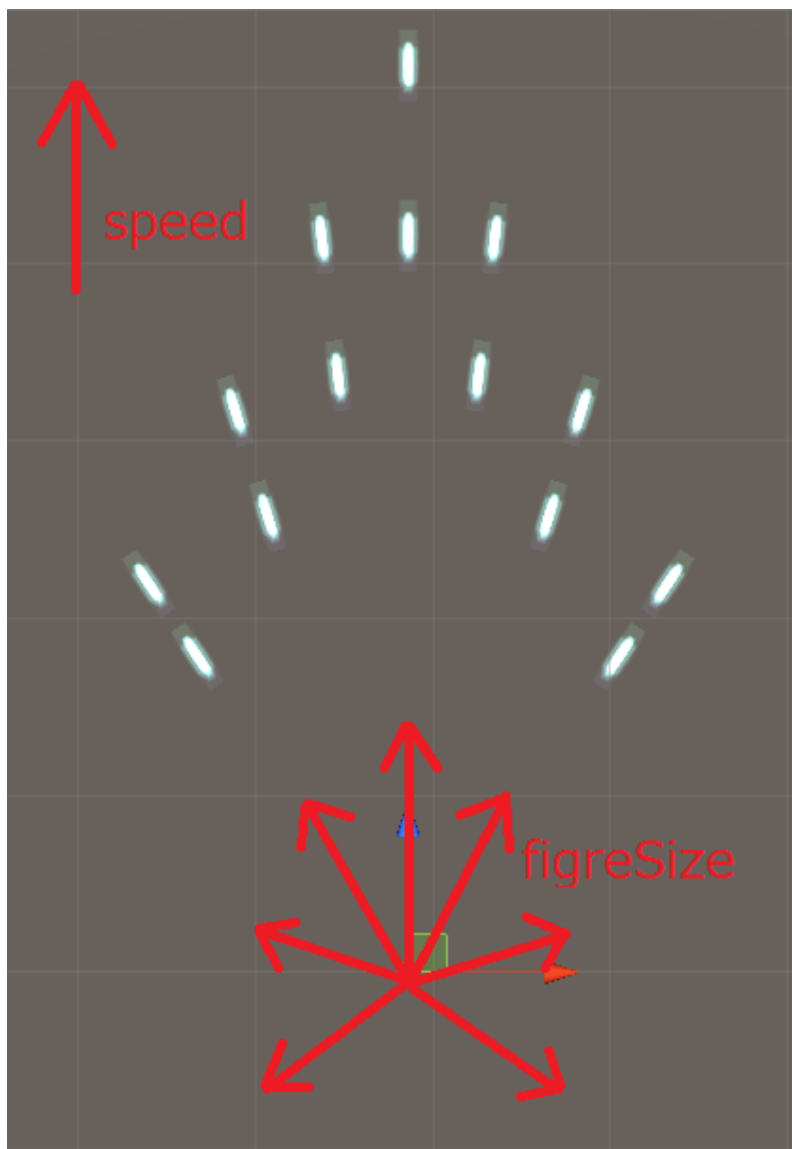
- ・ <型にパーティクルを発射します
- ・ PlayMaker では ShotOperator の TriangleShot で行います.
- ・ 図形の大きさが拡散します.

パラメーター

- | | |
|----------------------------|------------------------------|
| 1. <code>tr</code> | Emit する座標と角度の Transform. |
| 2. <code>speed</code> | 前方への移動スピード |
| 3. <code>speedRate</code> | 移動スピードの上昇倍率 |
| 4. <code>size</code> | パーティクルの大きさ |
| 5. <code>figureSize</code> | 図形が拡散するスピード. |
| 6. <code>lifetime</code> | パーティクルの寿命 |
| 7. <code>rows</code> | 弾の描く図形の弾の数(TriangleShot と同様) |

8. cols

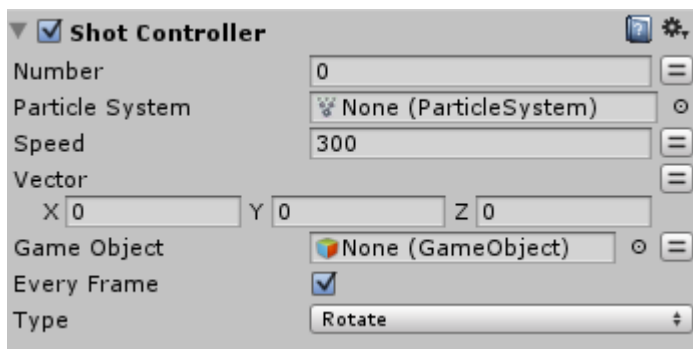
発射する図形の数.



ベクトル制御

説明

- ParticleSystem の Particle のベクトルを制御します.
- ベクトルを回転, 加速, 標的に向けることができます.
- PlayMaker では, ShotController で行います.



パラメーター

Number	ParticleManager で参照するための番号
ParticleSystem	制御する ParticleSystem

	Number が None(なし)でなければ, 実行時に自動で設定される.
Vector	制御に使う Vector3. (Type によっては必要ない)
GameObject	制御に使うゲームオブジェクト (Type によっては必要ない)
EveryFrame	制御を毎フレーム実行するか
Type	実行する制御の種類

使い方(スクリプト)

`ParticleSystem.manager.Vector_`使いたい機能
でメソッドを呼び出してください.

使い方(PlayMaker)

1. PlayMakerEditor のアクションブラウザから,
カテゴリ ParticleBulletSystem の ShotController を選択して追加します.
2. Type で使いたい制御の種類を選択します.
3. 各パラメーターを設定します.

Rotate

```
public void Vector_Rotate(int number,float speed)
public void Vector_Rotate(ParticleSystem par,float speed,ParticleSystem.Particle[] pars)
public void Vector_Rotate(int number,float speed, Vector3 axis)
public void Vector_Rotate(ParticleSystem par,float speed, Vector3 axis, ParticleSystem.Particle[] pars)
```

説明

- ・パーティクルのベクトルを回転させます.
- ・PlayMaker の場合は, 自動で Time.deltatime が適用されます.

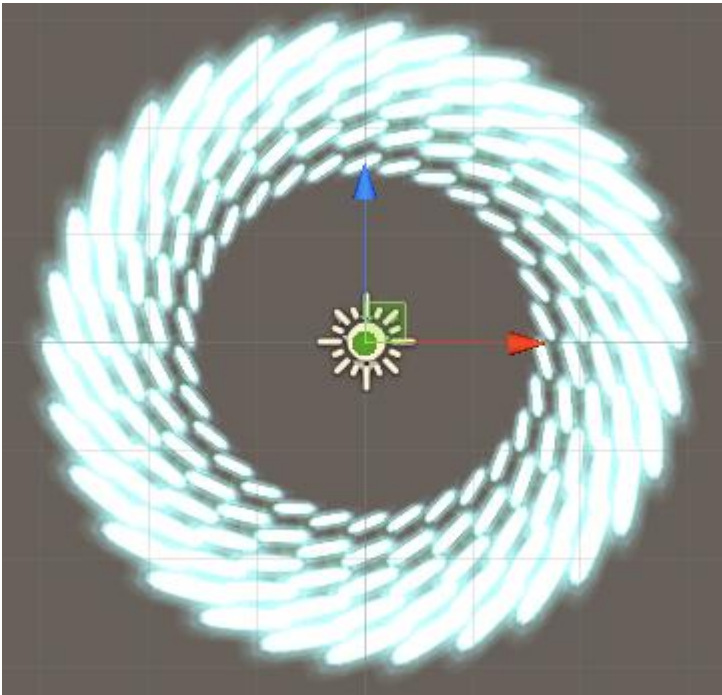
パラメーター

number	制御する ParticleSystem を ParticleManager で参照するための番号.
par	制御する ParticleSystem
speed	回転速度

スクリプトでは `Time.deltaTime` は自動で適用されません.

axis 回転の軸. `PlayMaker` の場合は, `vector` で設定します.

pars `par` が `None` でない場合, これも設定しなければなりません
`ParticleManager` で管理していない `ParticleSystem` を使用するためです.



この画像は `CycleShot` を回転させたものです.

AddVelocity

```
public void Vector_AddVelocity(int number, Vector3 vector)
```

```
public void Vector_AddVelocity(ParticleSystem par, Vector3 vector, ParticleSystem.Particle[] pars)
```

説明

パーティクルの移動ベクトルに, 新たなベクトルを加えます.

パラメーター

Vector パーティクルに加えるベクトル
1 フレームで加えるベクトルです. `Time.deltaTime` は適用されていません.

AddSpeed

```
public void Vector_AddSpeed(int number, float speed)
```

```
public void Vector_AddSpeed(ParticleSystem par, float speed, ParticleSystem.Particle[] pars)
```

説明

パーティクルの移動ベクトルを加速します.

パラメーター

speed パーティクルに加えるスピード

SetVelocity

```
public void Vector_SetVelocity(int number, Vector3 vector){  
public void Vector_SetVelocity(ParticleSystem par, Vector3 vector, ParticleSystem.Particle[] pars)
```

説明

パーティクルの移動ベクトルを、新たなベクトルで上書きします。

パラメーター

vector 新たに設定する、パーティクルの移動ベクトル。

LookPosition

```
public void Vector_LookPosition(int number, Vector3 position)  
public void Vector_LookPosition(ParticleSystem par, Vector3 position, ParticleSystem.Particle[] pars)
```

説明

パーティクルの移動ベクトルを、指定した座標に向けます。

パラメーター

position 移動ベクトルのターゲット座標(PlayMakerではVectorで設定します.)

RotateTowards

```
public void Vector_RotateTowards(int number, float speed, Vector3 position)  
public void Vector_RotateTowards(ParticleSystem par, float speed, Vector3 position  
, ParticleSystem.Particle[] pars)
```

説明

パーティクルの移動ベクトルを、指定した座標に、時間をかけて向けます。

パラメーター

speed 移動ベクトルが **position** に向くまでの時間に影響する。

position 移動ベクトルのターゲットの座標。

その他の機能

ClearAllParticle

```
public void ClearAllParticle()
```

説明

ParticleManagerで管理しているすべてのParticleSystemのパーティクルを消します。

パラメーター

なし