ParticleManager

Function of ParticleManager are three of the following.

1. Referance ParticleSystem by number.
2. Emit of a script or PlayMaker, Shot function
3. VectorControl of a script or PlayMaker

How to call Methods

- From Version 1.03, please write the following code at the beginning of your script.

```
using ParticleBulletSystem;
```

- If you want to use in a script, It calls the method as follows:.
- ParticleManager.manager.”UseFunction”();
- The first argument is a number (int type) used to refer to the ParticleSystem.
- You can also use the ParticleSystem that are not managed by ParticleManager .

  In this case, The first argument is ParticleSystem type.

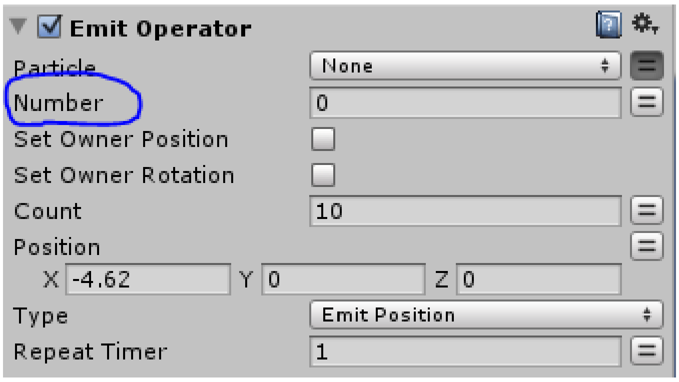
example OneShot

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

How to refer to ParticleSystem.

- ParticleManager of, use with reference to ParticleSystem.
- In the case of PlayMaker, you can specify the number in the Inspector. It is a variable called Number.

  Number it will be ignored if "Particle" is not None.



Emit Operator

Particle: None

Number: 0

Set Owner Position: □

Set Owner Rotation: □

Count: 10

Position: X -4.62 Y 0 Z 0

Type: Emit Position

Repeat Timer: 1

- In the case of C #, you will see in the following manner.

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

In C #, you need not be referenced, as described above.

When you call a method, it is because you can specify the number in the argument.

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

Even if you use the C #, You can use the direct particle without using numbers.

The name of the argument consistently in all the methods, it has a “number”.

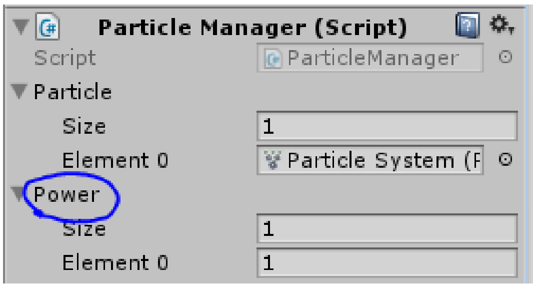
Argument names ParticleSystem is “par”.

A particle in this PDF, I may be rephrased as bullet.

Emit is sometimes referred to as a launch.

Setting of Power

It is set with the variable "power".



Particle Manager (Script)

Script: ParticleManager

Particle
- Size: 1
- Element 0: Particle System (F

Power
- Size: 1
- Element 0: 1

Prepare as many as Particle(ParticleManager's variable).

Set power every particle system.

Emit in C # and PlayMaker

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)
```

Description

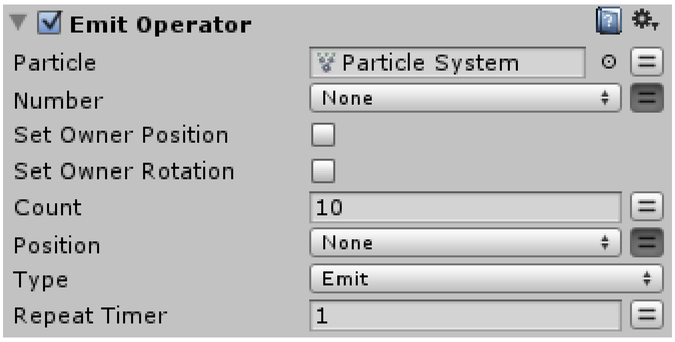
- Simply Emit while maintaining the setting of ParticleSystem.
- You can also set the positiion and rotation of in C # ParticleSystem.
- In the case of PlayMaker, you have done in the EmitOperator.
  The position and rotation to specify the time interval of repetition in .
  RepeatTimer that you can choose fit to Owner.

Parameters (script)

- count	The number of particles that Emit
- position	ParticleSystem of position (to move the game object of ParticleSystem)
- rotation	rotation of ParticleSystem (to rotate the game object of ParticleSystem)

Parameters (PlayMaker)

- SetOwnerPosition	Whether or not adjust the position of ParticleSystem to Owner
- SetOwnerRotation	Whether or not it matches the rotation of ParticleSystem to Owner
- Count	The number of particles that Emit
- Position	ParticleSystem position
- Type	Emit	not use the Position variable.
  EmitPosition	use the Position variable.
- RepeatTimer	It will specify the time interval of repetition.



☑ Emit Operator

Particle: Particle System

Number: None

Set Owner Position: □

Set Owner Rotation: □

Count: 10

Position: None

Type: Emit

Repeat Timer: 1

Remarks

If you use the SetOwnerPosition and EmitPosition at the same time, SetOwnerPosition takes precedence. EmitOperator and Emit of PaticleManager is another script.

Shot

Description

Emit functions for Shmup.

Emit in a form suitable for Shmup.

Parameters

1.	speed	Movement speed of Particle
2.	speedRate	Increase the magnification of the Movement speed of Particle
3.	size	The size of the particle
4.	SizeRate	Increase the magnification of the size of the particle
5.	figuarSize	The size of a graphic bullet draw
6.	lifetime	The life of the particle
7.	rows	The number of rows of figures drawn by the bullet
8.	cols	The number of cols of figures drawn by the bullet

ShotOperator

Description

- Using the Shot function in PlayMaker.

Parameters

- The name of each parameter is the same as the script

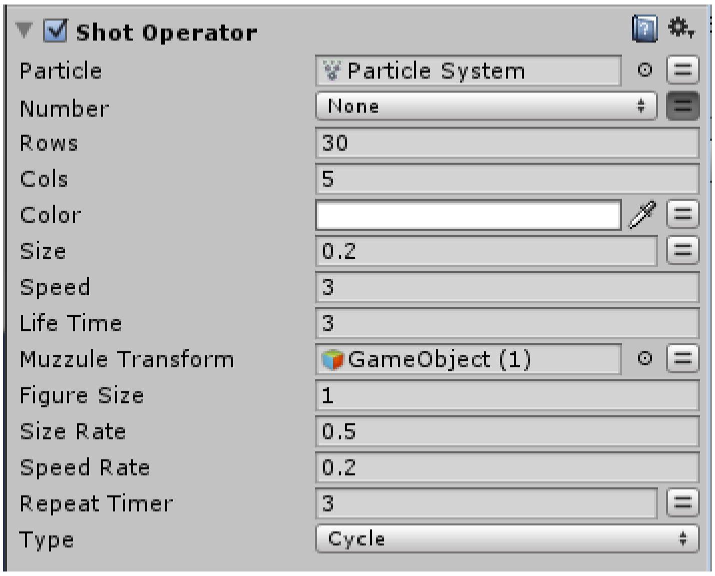
How to use (C#)

Please call the method as follows.

```
ParticleManager.manager.Shot_"Function"
```

How to use (PlayMaker)

1. From the action browser PlayMakerEditor,
   Add Select the ShotOperator categories ParticleBulletSystem.
2. In Type, select the one you want to use Shot.
3. You set the parameters.



☑ Shot Operator

Particle	Particle System
Number	None
Rows	30
Cols	5
Color	
Size	0.2
Speed	3
Life Time	3
Muzzule Transform	GameObject (1)
Figure Size	1
Size Rate	0.5
Speed Rate	0.2
Repeat Timer	3
Type	Cycle

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)
```

Description

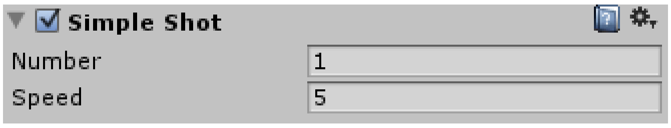
- This will Emit towards the front of the Transform.
- Done in the SimpleShot or ShotOperator Type OneShot in PlayMaker.
- Among the Shot function, it is the most simple things.

Parameters (script)

- tr This Transform will determine the position and rotation of Emit.
- speed Movement speed of Particle
- size The size of the particle

Parameters(PlayMaker)

- speed Movement speed of Particle



▼ ☑ Simple Shot

Number 1

Speed 5

Remarks

SimpleShot does not have a repeat function.

Only the moment when the state becomes active, it will emit.

You can specify the Size in ShotOperator.

ShotSelector

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

Description

- Choose the following shot function and use it.
- In a shot function, this is the most superior in versatility.
  Without a reason in particular, I recommend use of ShotSelector.
  ParticleShooter of the example script will serve as a reference.

Parameters

1. type Type of shot functiion.
2. param Parameter (cf. previous page) of the shot
3. tr This Transform will determine the position and rotation of Emit

MultiSpeed

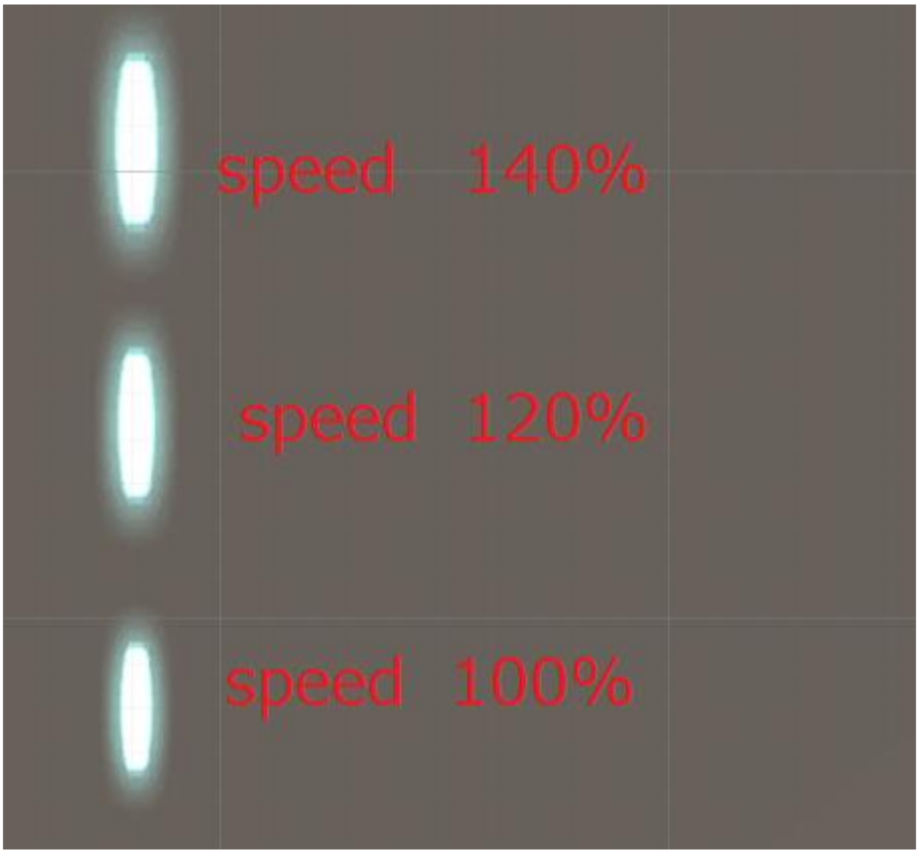
```
public void Emit_MultiSpeed(int number,Transform tr,float speed,float speedRate,float size,float sizeRate,float lifetime,int rows)
public void Emit_MultiSpeed(ParticleSystem par,Transform tr,float speed,float speedRate,float size,float sizeRate,float lifetime,int rows)
```

Description

- It will Emit multiple particles in multiple speed.
- In PlayMaker, you have done in the ShotOperator(Type MultiSpeed).

Parameters

4.	tr	This Transform will determine the position and rotation of Emit.
5.	speed	Movement speed of 1th Particle
6.	speedRate	The rate of increase in movement speed of the particles Example 0.2, by 20% increase
7.	size	The size of the particle
8.	sizeRate	The size of the rate of increase in the particle
9.	lifetime	The life of the particle
10.	rows	The number of rows of figures drawn by the bullet



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)
```

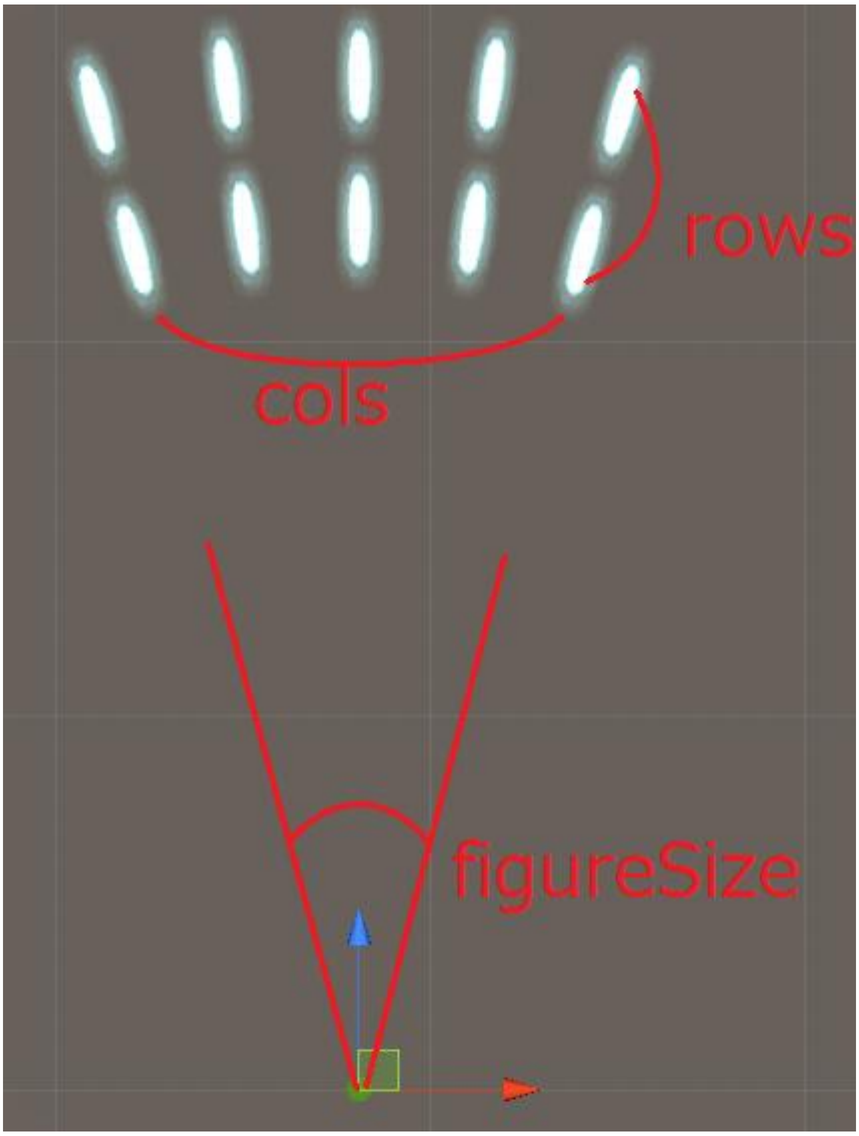
Description

・It will launch the above MultiSpeed at a plurality of angles

・In PlayMaker, you have done in the ShotOperator(Type MultiWay).

Parameters

1. tr　　This Transform will determine the position and rotation of Emit.
2. speed　　Movement speed of The first line Particle
3. speedRate　　The rate of increase in movement speed of the particles
   In the case of 0.2, increasing by 20
4. size　　The size of the particle
5. sizeRate　　The size of the rate of increase in the particle（In the case of 0.2, increasing by 20）
6. figureSize　　The magnitude of the arc (figureSize 30 = 30°　)
7. lifetime　　The life of the particle
8. rows　　The number of rows in the graphic bullet draw (MultiSpeed と同様)
9. cols　　The number of columns of a graphic bullet draw



CycleShot

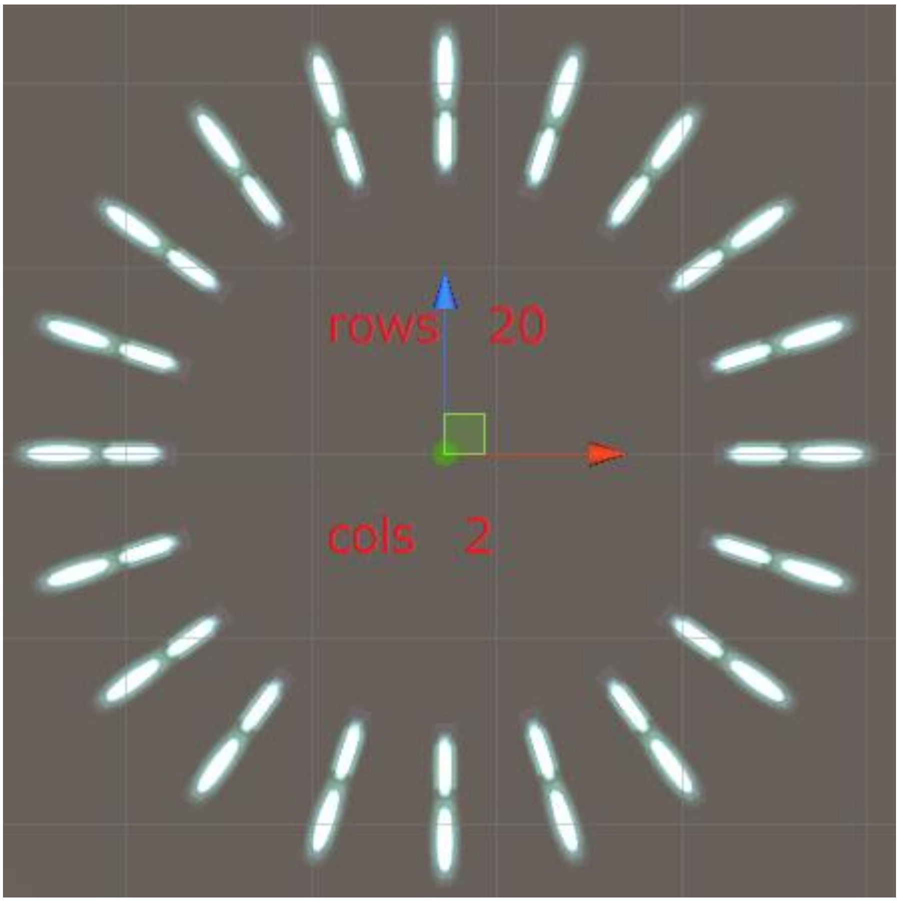
```
public void Emit_CycleShot(int number,Transform tr,float speed,float speedRate,float size,float sizeRate,float lifetime,int rows,int cols)
public void Emit_CycleShot(ParticleSystem par,Transform tr,float speed,float speedRate,float size,float sizeRate,float lifetime,int rows,int cols)
```

Description

- It will launch a particle in a circular
- In PlayMaker, you have done in the ShotOperator(Type CycleShot).

Parameters

1. tr — This Transform will determine the position and rotation of Emit.
2. speed — Movement speed of The first line Particle
3. speedRate — The rate of increase in movement speed of the particles
   In the case of 0.2, increasing by 20
4. size — The size of the particle
5. sizeRate — The size of the rate of increase in the particle
   In the case of 0.2, increasing by 20
6. lifetime — The life of the particle
7. rows — The number of rows in the graphic bullet draw
8. cols — The number of columns of a graphic bullet draw



RingShot

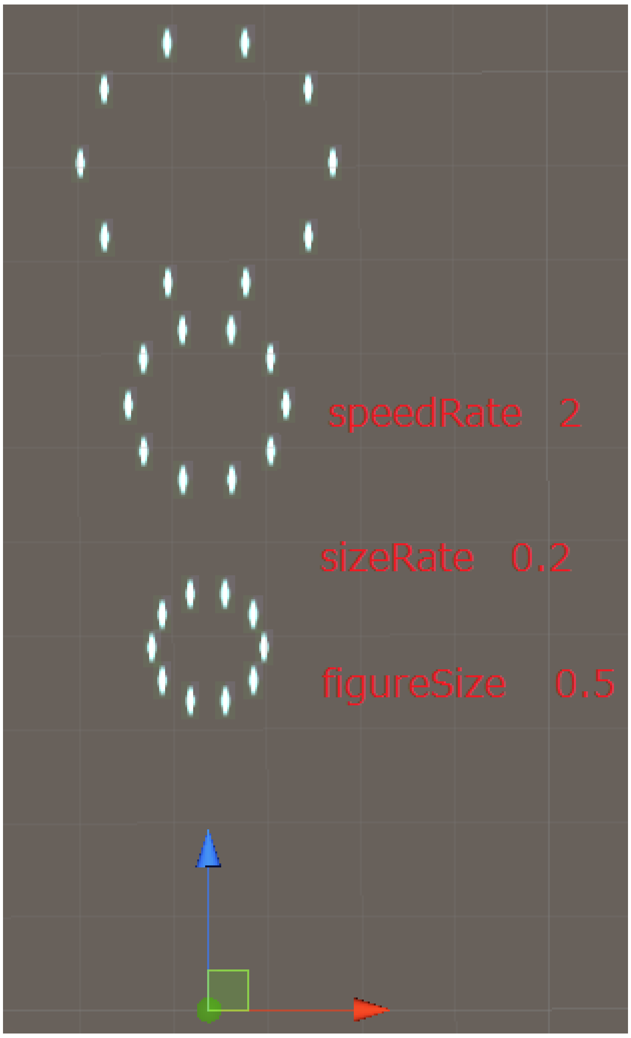
```
public void Emit_RingShot(int number,Transform tr,float speed,float speedRate,float size,float SizeRate,float figureSize,float lifetime,int rows,int cols)
public void Emit_RingShot(ParticleSystem par,Transform tr,float speed,float speedRate,float size,float SizeRate,float figureSize,float lifetime,int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

Description

- The particles are arranged in a circle, and then fired at multiple speeds
- It's like a combination of CycleShot and MultiSpeed.
- At launch time, the size of the figure has been determined._Unlike CycleShot, it does not spread.
- In PlayMaker, you have done in the ShotOperator(Type RingShot).

Parameters

1. tr This Transform will determine the position and rotation of Emit.
2. speed Movement speed of 1th ring.
3. speedRate The rate of increase in movement speed of the ring
   In the case of 0.2, increasing by 20
4. size The size of the particle
5. sizeRate The rate of increase in the size of the ring
6. figureSize The size of 1th Ring
7. lifetime The life of the particle
8. rows The number of rows in the circle drawn by the bullet
9. cols The number of the circle to fire (It corresponds to a row of MultiSpeed)



SquareShot

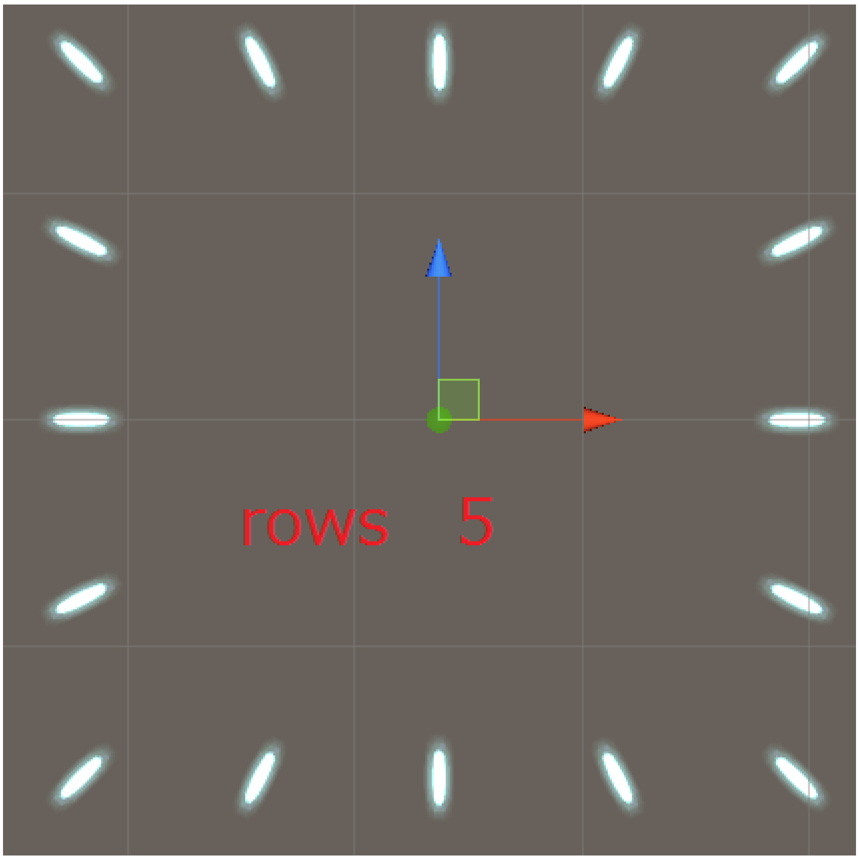
```
public void Emit_SquareShot(int number,Transform tr,float speed,float size,float lifetime,int rows)
public void Emit_SquareShot(ParticleSystem par,Transform tr,float speed,float size,float lifetime,int rows)
```

Description

- Mainly Tr, we will launch a particle in a square
- In PlayMaker, you have done in the ShotOperator(Type SquareShot).

Parameters

1. tr — This Transform will determine the position and rotation of Emit.
2. speed — It affects the particles move speed
3. size — The size of the particle
4. lifetime — The life of the particle
5. rows — Rectangle drawn by the bullet, the number of one side of the bullet
   A minimum of three required



TriangleShot

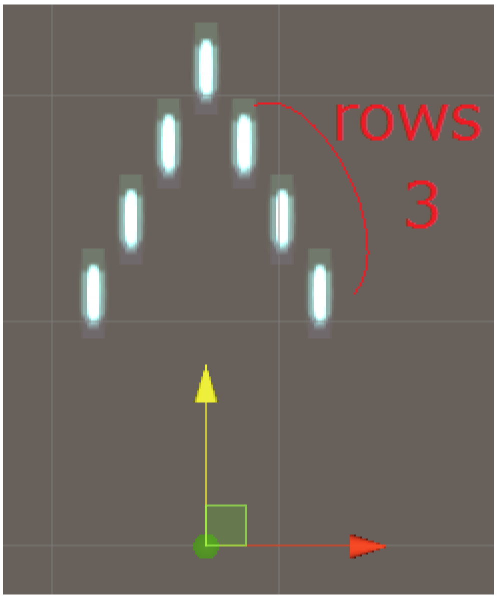
```
public void Emit_TriangleShot(int number,Transform tr,float speed,float speedRa
te,float size,float figureSize,float lifetime,int rows)
public void Emit_TriangleShot(ParticleSystem par,Transform tr,float speed,float speedRate,
float size,float figureSize,float lifetime,int rows)
```

Description

- "<" Shape even though I will fire the particle
- In PlayMaker, you have done in the ShotOperator(Type TriangleShot).
- At launch time, the size of the figure has been determined.

Parameters

1.	tr	This Transform will determine the position and rotation of Emit.
2.	speed	Movement speed of Particle
3.	speedRate	It affects in a position to launch the graphic. Coordinates in the Z-axis of veriable Tr.
4.	size	The size of the particle
5.	figureSize	It will affect the size of the figure. Graphic is inverted in the negative.
6.	lifetime	The life of the particle
7.	rows	The number of bullets to draw shapes



TriangleSpread

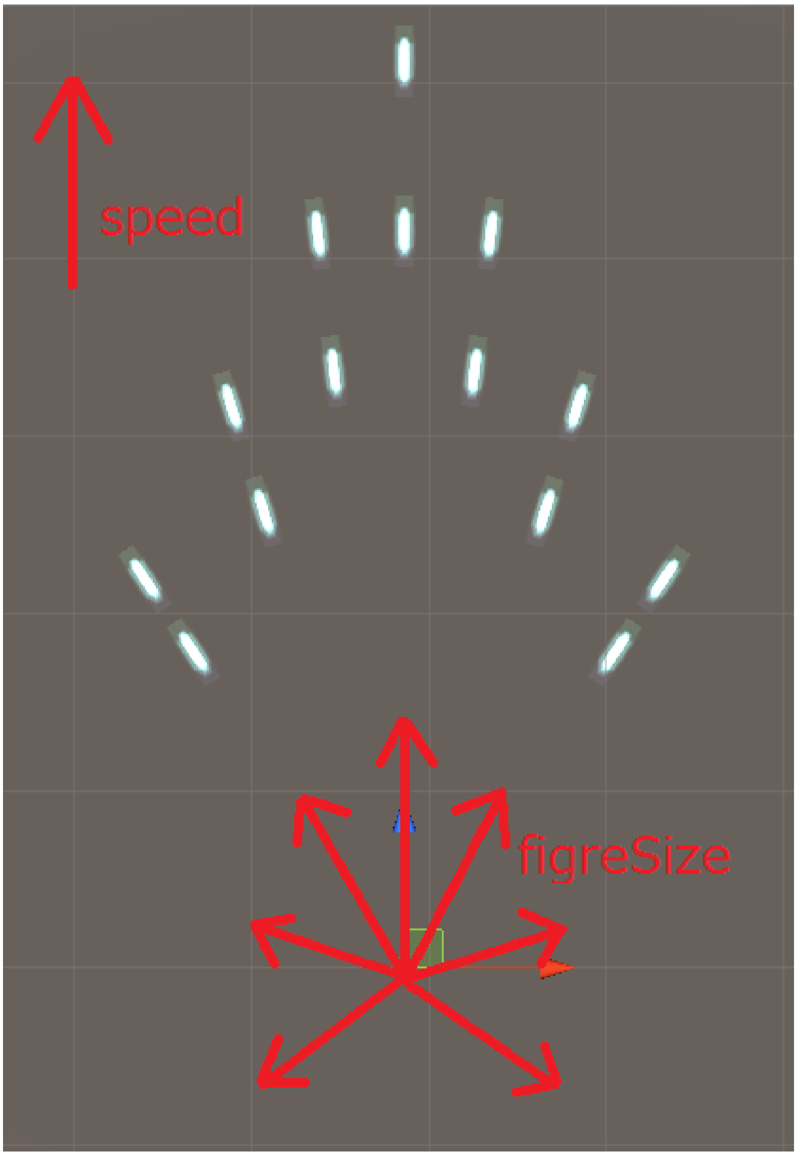
```
public void Emit_TriangleSpread(int number,Transform tr,float speed,float speedRate,float size,
float figureSize,float lifetime,int rows,int cols)
public void Emit_TriangleSpread(ParticleSystem par,Transform tr,float speed,float speedRate,
float size,float figureSize,float lifetime,int rows,int cols)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

Description

- "<" Shape even though I will fire the particle
- In PlayMaker, you have done in the ShotOperator(Type TriangleSpread).
- The size of the graphic is to spread.

Parameters

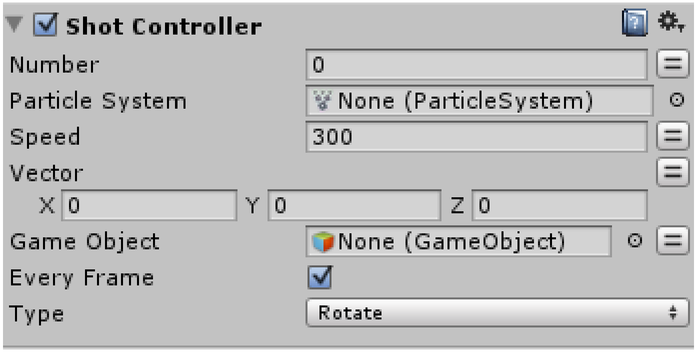
1.	tr	This Transform will determine the position and rotation of Emit.
2.	speed	Movement speed forward
3.	speedRate	The rate of increase in movement speed of the particles
4.	size	The size of the particle
5.	figureSize	Speed to diffuse the figure shape.
6.	lifetime	The life of the particle
7.	rows	The number of bullets to draw shapes (As with TriangleShot)
8.	cols	The number of shapes to fire.



Vector Control

Description

- It will control the vector of Particle of ParticleSystem.
- Rotate vector, acceleration, it can be directed to target.
- In PlayMaker, you have done in the ShotController.



☑ Shot Controller

Number 0

Particle System None (ParticleSystem)

Speed 300

Vector

X 0 Y 0 Z 0

Game Object None (GameObject)

Every Frame ☑

Type Rotate

Parameters

Number	Number for reference in ParticleManager
ParticleSystem	The control to ParticleSystem Number is not equal to None, and is set automatically at runtime.
Vector	Vector3 to use for control. (Not required by Type)
GameObject	Game objects that you use to control. (Not required by Type)
EveryFrame	Or the control to run every frame
Type	Types of control

How to use (C#)

```
ParticleManager.manager.Vector_”Function”
```

Please call the method as described above.

How to use (PlayMaker)

1. From the action browser PlayMakerEditor,
2. Add Select the ShotController categories ParticleBulletSystem.
3. Select the type of control that you want to use in Type.
4. you set the parameters.

Rotate

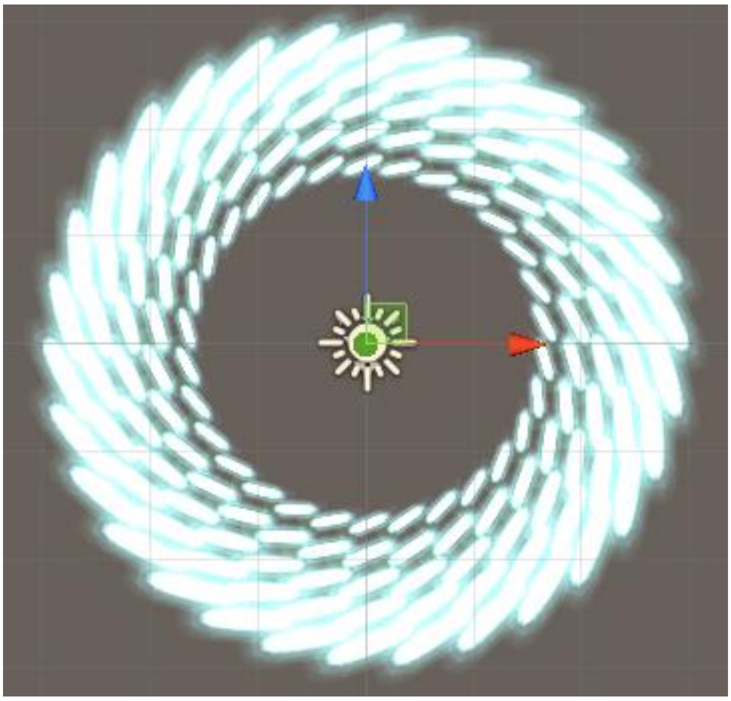
```
public void Vector_Rotate(int number,float speed)
public void Vector_Rotate(ParticleSystem par,float speed,ParticleSystem.Particle[] pars)
public void Vector_Rotate(int number,float speed, Vector3 axis)
public void Vector_Rotate(ParticleSystem par,float speed, Vector3 axis, ParticleSystem.Particle[]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

Description

- It rotates the vector of the particle.
- In the case of PlayMaker, Time.deltatime will be applied automatically.

Parameters

number	Number for reference in ParticleManager
par	The control to ParticleSystem
speed	Rotational speed Time.deltatime does not apply automatically in the script.
axis	The axis of rotation. In the case of PlayMaker, you set in the vector.
pars	If you want to set the variable par, this must also be set.



This image is intended to rotate the CycleShot.

AddVelocity

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

Description

The movement vector of the particle, and add a new vector.

Parameters

Vector	Vector added to the particle A vector added in one frame. Time.deltatime is not applicable.

AddSpeed

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

Description

This will accelerate the movement vector of the particle.

Parameters

speed	Speed to be applied to the particle

SetVelocity

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

Description

The movement vector of the particle, it will overwritten with new vector.

Parameters

vector	Newly set, move vector of the particle.

LookPosition

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

Description

The movement vector of the particle, and directed to the specified coordinates.

Parameters

position	The coordinates of the target of the movement vector. (You set at Vector in PlayMaker.)

RotateTowards

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

Description

The movement vector of the particle, and directed to the specified coordinates. It will receive over time.

Parameters

speed	Affect the time it takes to rotate.
position	The coordinates of the target of the movement vector.

Other Features

ClearAllParticle

```
public void ClearAllParticle()
```

Description

Turn off all the ParticleSystem of particles that are managed by ParticleManager.

Parameters

nothing special