

# MAGICAL

## THE MAGIC EFFECTS

### Basic Instruction

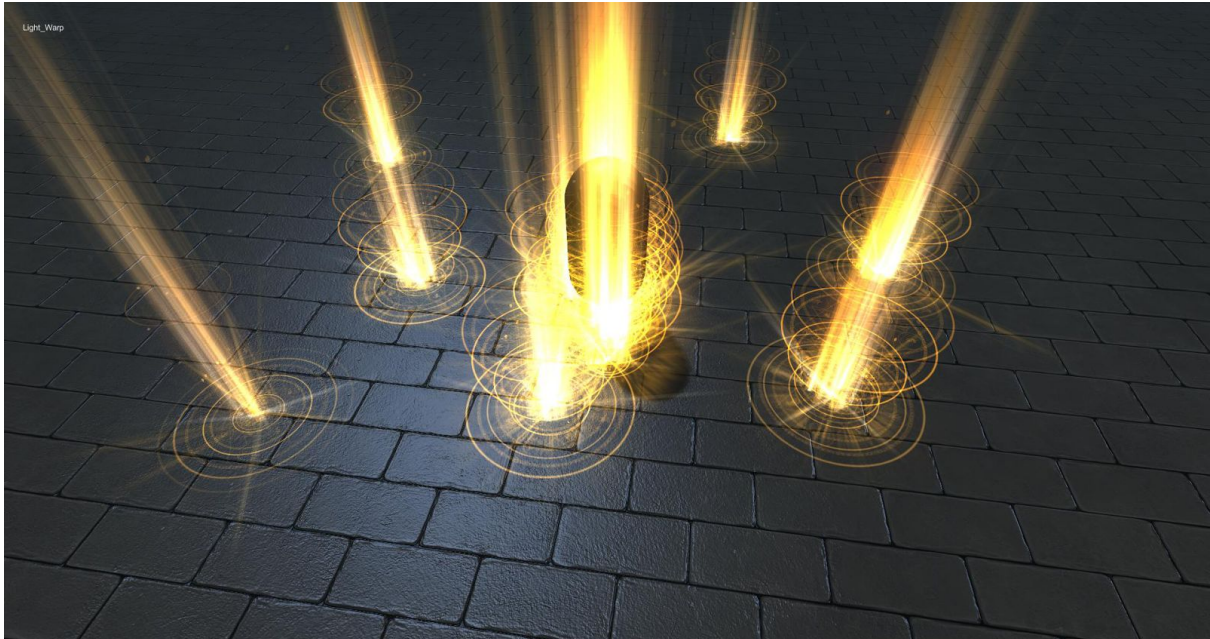
This instruction is a basic how to use the Magical particle effect in your game, Magical effect comes with a complete particles system prefabs, in a several styles such as Fire , Ice , Water , and more, you can found them in **Magical/FX/**, so all you have to do is just place it into the scene by hand or spawn it in difference way by scripts in a following spawn type.



Asset store [Hardworker Studio by Rachan N.](#)

# Static spawn

You can spawn a magic effect directly into the specific position, this type should apply with a *static* magic effect or something that's spawn without a direction or moving. such as [Warp](#), [Friewall](#), [Lighting strike](#). by using `Game.Instantiate` to spawn in specific **position**.



## Code Example.

we have to create a spwan function and add magic effect prefab to a **skill** parameter.

```
// declare skill parameter
Public GameObject skill;

// create direction spawn function. required position.
void StaticSpawn (Vector3 position)
{
    GameObject.Instantiate (skill, position, skill.transform.rotation);
}
```

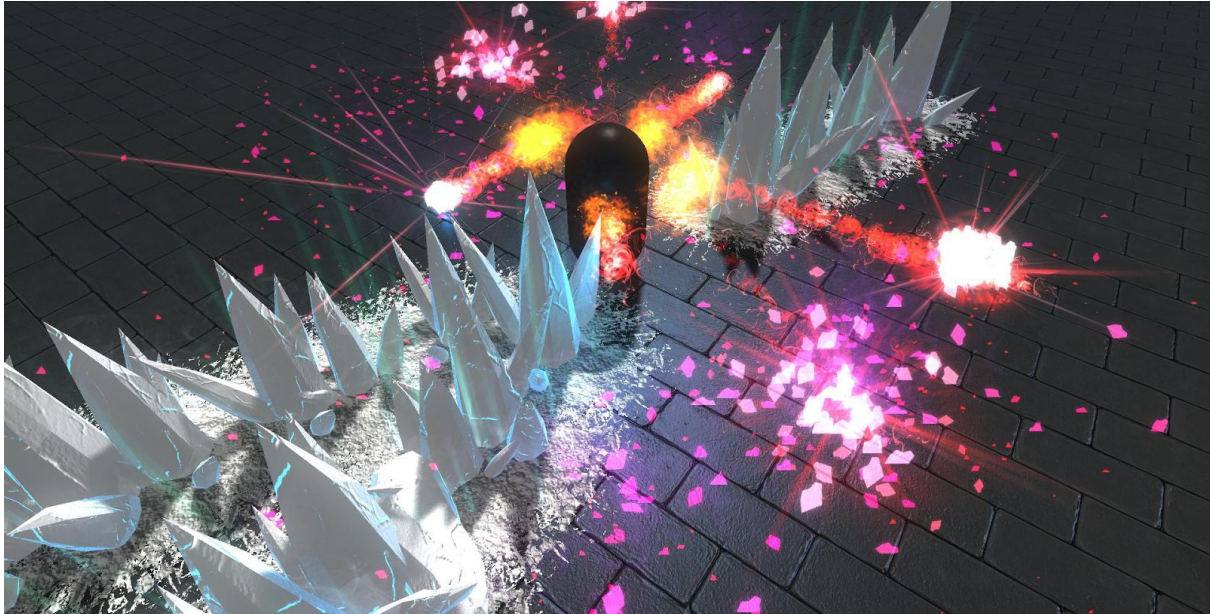
## How to use.

```
// instance a variable
Vector3 position = your specific position.

// call a function
StaticSpawn (position);
```

## Direction spawn

You can spawn a magic effect directly into the specific position and let it moving facing forward to a direction, this type should apply with a *throwing* magic effect such as [Fireball](#) , [Missile](#) , [Ice Hammer](#). by using `Game.Instantiate` to spawn in specific **position** and **direction**.



### Code Example.

we have to create a spawn function and add magic effect prefab to a **skill** parameter.

```
// declare skill parameter
Public GameObject skill;

// create direction spawn function. required position and direction
void DirectionSpawn(Vector3 position , Vector3 direction)
{
    GameObject sk = (GameObject)GameObject.Instantiate (skill, position,
skill.transform.rotation);
    sk.transform.forward = direction;
}
```



## How to use.

```
// instance a variables
Vector3 position = add your character position here.
Vector3 direction = add your character direction, so you must use forward vector of your
character transform.

// call a function
DirectionSpawn(position,direction.normalized);
```

If your game is **Topview** style like Diablo, or Warcraft, so you need to throwing it to the target position.

```
// instance a variables
Vector3 position = your character position.
Vector3 direction = your character position - target position

// call a function
DirectionSpawn(position,direction.normalized);
```

# Thank you!

If you have anyquestion or suggestion please let me know.

Rachan Neamprasert

email : [hwrstudio@gmail.com](mailto:hwrstudio@gmail.com)

<http://www.hardworkerstudio.com>

# The Magic FX Series you should have.



## Magical

<https://assetstore.unity.com/packages/vfx/particles/spells/magical-39561>



## Magical : Dark Edition

<https://assetstore.unity.com/packages/vfx/particles/spells/magical-dark-wizard-edition-53905>



## Magical : Pro

<https://assetstore.unity.com/packages/vfx/particles/spells/magical-pro-edition-55713>



## Magical : Effects

<https://assetstore.unity.com/packages/vfx/particles/spells/magical-effects-63956>



## Magical : Ultimate

<https://assetstore.unity.com/packages/vfx/particles/spells/magical-ultimate-91406>