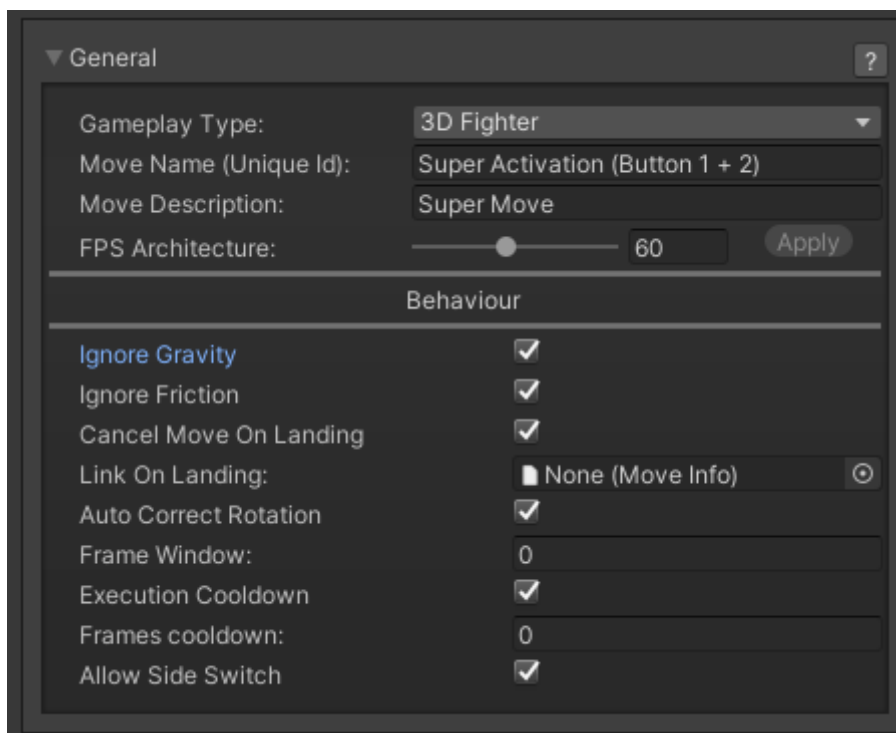


General Options

Set general options of your move here.



Gameplay Type: Set the move to be either for 2D or 3D gameplay. 3D gameplay allows you to access the z axis on certain options.

Move Name: The move name. This is the main identifier for this move.

Move Description: The move description. Can be used in handy guides inside your game.

FPS Architecture: The frame per second architecture this move is designed for. By default this value is set to 60. Be very careful when changing this value on already coded moves. The editor will recalculate the frames in your move which can cause already coded frame data to malfunction.

Behavior

Ignore Gravity: If this move is triggered in the air, should the character ignore the laws of physics for the duration of the move.

Cancel Move When Landing: If the move was executed in the air, should it automatically cancel into the landing animation once it hits the ground? Enable this for standard air moves. Disable this if you have moves that have their own landing animations or special moves that launches the character in the air but can be punished when they hit the ground. Make sure you apply the right amount of force so the whole animation can match the game's physics.

- **Link On Landing:** Select which move will play on landing.

Ignore Friction: If force is applied to this character while this move is active, ground friction will not slow it down. (Useful for moves like Blanka's Rolling Attack or Ryu's Tatsumaki Senpukyaku)

Auto-correct Rotation: If this is toggled on, this move will automatically correct its rotation in case it crosses the opponent over.

(if Auto-correct Rotation) **Frame Window:** From which frame this move will auto-correct rotation.

Execution Cooldown: Stops the move from being executed after being executed once.

(if Execution Cooldown) **Frames Cooldown:** How many frames ([FPS](#)) before the move can be executed again.

Code example:

```
void OnHit(HitBox strokeHitBox, MoveInfo move, ControlsScript hitter){  
    if (move.attackType == AttackType.EX) Debug.Log("EX Move fired!");  
}
```

[< Back to Move Editor](#)

From:

<http://ufe3d.com/> - **Universal Fighting Engine**

Permanent link:

<http://ufe3d.com/doku.php/move:general?rev=1670313186>

Last update: **2022/12/06 02:53**

