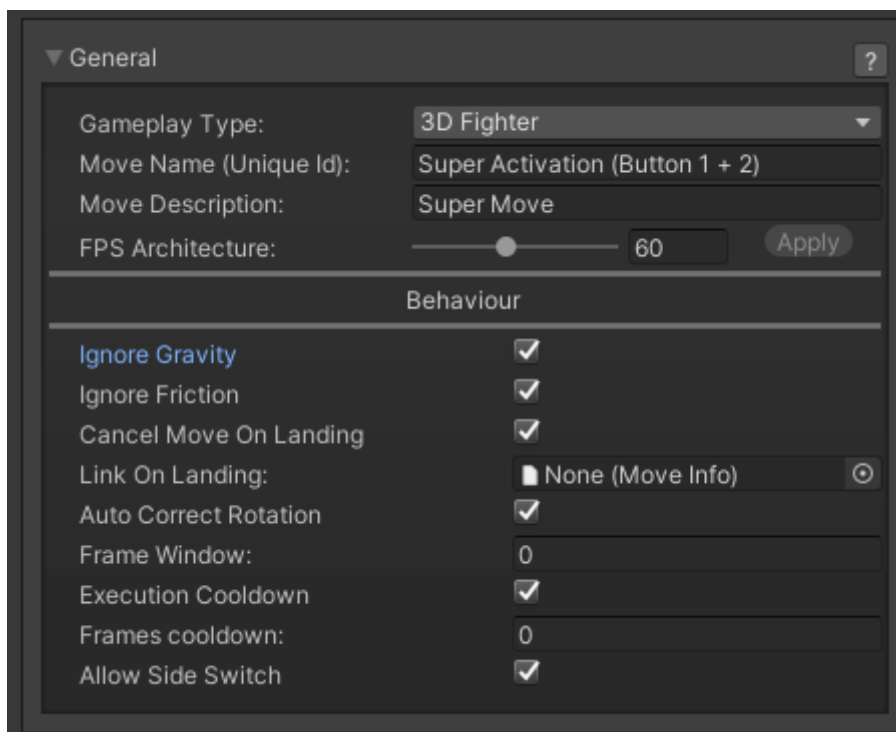


General Options

Set general options of your move here.



The screenshot shows a 'General' options window with a dark theme. At the top, there's a 'Gameplay Type' dropdown set to '3D Fighter', a 'Move Name (Unique Id)' text field containing 'Super Activation (Button 1 + 2)', and a 'Move Description' text field containing 'Super Move'. Below these is an 'FPS Architecture' slider set to 60, with an 'Apply' button. A horizontal separator line divides the 'General' section from the 'Behaviour' section. The 'Behaviour' section contains several toggleable options: 'Ignore Gravity' (checked), 'Ignore Friction' (checked), 'Cancel Move On Landing' (checked), 'Link On Landing' (set to 'None (Move Info)' with a circular icon), 'Auto Correct Rotation' (checked), 'Frame Window' (set to 0), 'Execution Cooldown' (checked), 'Frames cooldown' (set to 0), and 'Allow Side Switch' (checked).

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.

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.

Code example:

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

[< Back to Move Editor](#)

From:

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

Permanent link:

<http://www.ufe3d.com/doku.php/move:general?rev=1414663521>

Last update: **2014/10/30 06:05**

