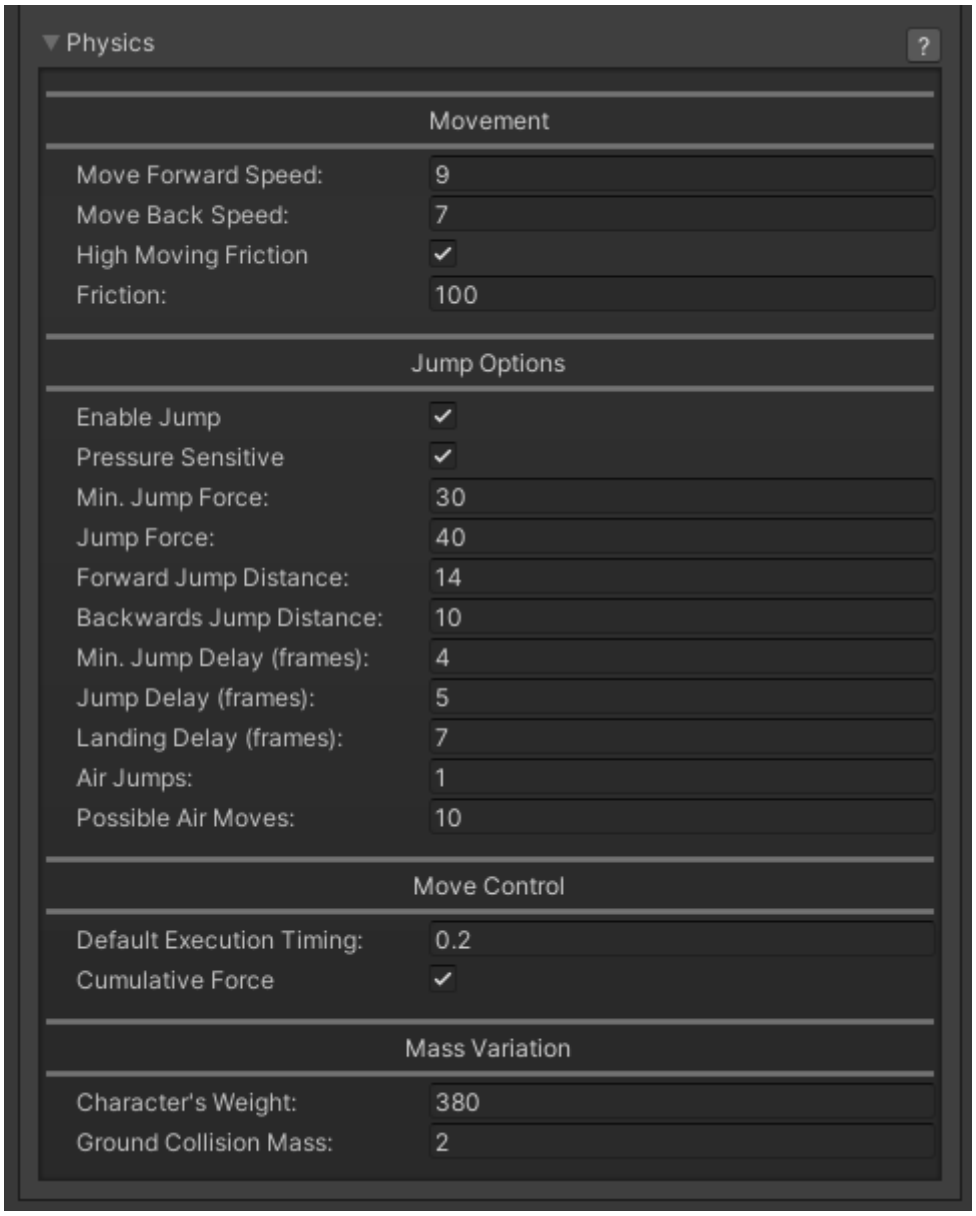
Physics

Define here how this character moves and reacts to applied forces in your game.



Horizontal Movement

Move Forward Speed: How fast this character moves when walking forward.

Move Back Speed: How fast this character moves when walking back.

High Moving Friction: When this character stops moving, should it stops immediately or slide slightly (based on the friction)?

Friction: When forces are applied to this character while on the ground, how far will it “slide”. If High Moving Friction is off, this friction will also be applied to the character when its walking. Characters like Hakan (SF4) while oiled up, has a very low friction.

Jump Options

Enable Jump: Toggle off to disable jumping for this character.

Pressure Sensitive: Toggle to enable pressure sensitivity on the jump input's press/release states.

Min. Jump Force: If Pressure sensitive is enable, when the minimum input is applied, this is the lowest force that will be applied to the jump.

Jump Force: How much power is applied by this character when it jumps. The higher the force, the higher the jump. If pressure sensitive is enabled, this is the maximum force applied if the input is pressed to its maximum frames.

Jump Distance: When making an angled jump, how far will this character move while in the air.

Min. Jump Delay (frames): If pressure sensitive is toggled, this is the minimum amount of frames that will be accounted for the take off.

Jump Delay (frames): How many frames should the character wait before it jumps after you press the button. The `take off animation` will play during these frames. If pressure sensitive is toggled this is the maximum amount of frames allowed before the jump automatically releases (thus applying the maximum force).

Landing Delay (frames): How many frames before the character is allowed to move after landing from a jump (character can still block and cancel landing into moves).

Air Jumps: Set double/triple jumps here. This works as any game with double jump, aka, pressing up while in the air.

Mass Variation

Character's Weight: The weight of the character related to air resistance. Weight directly affects every air force applied to this character, including jumps. Example: A character like Dhalsim can have weight set to a very low value, making him almost “float”.

Ground Collision Mass: The density of the character's ground collision box. A lower value makes the character stronger when pushing into another character assuming both have same movement speed and weight. Best to leave this at default of 2 for consistent behaviour.

Cumulative Force: Anytime force is added, do we accumulate with current forces? Leave this on if you want new applied forces to stack up.

Code access:

```
UFE.config.player1Character.physics
```

Code example 1:

```
void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){
    if (hitter == UFE.config.player1Character){
        if (UFE.config.player2Character.characterName == "Shozo Iizuka") {
            UFE.config.player2Character.physics.weight = 175;
        }
    }
}
```

Code example 2:

```
void OnMove(MoveInfo move, CharacterInfo player){
    if (move.moveName == "oil up"){
        player.physics.friction = 10;
    }
}
```

< Back to Character Editor