Universal Fighting Engine

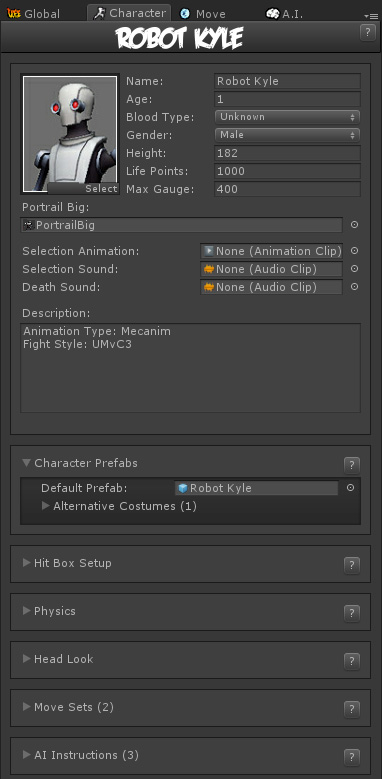
# Character Editor（编辑角色）

Here you can open an existing character info file or create a new one. Follow the instructions below to open Mike (character available on all versions):

Character Sample: .\UFE\Characters\Legacy\_Mike\Mike.asset.  
(SOURCE VERSION) Inherits from: .\UFE\Scripts\CharacterInfo.cs  
(OTHER VERSIONS) Inherits from: .\UFE\Plugins\UFE.dll\CharacterInfo

To create a new character, in the project window click on Create → U.F.E. → Character File.

Set all the major variables a character has as well as their move sets and most importantly, the hitboxes.



**Portrail (small)**: Can be used in your own GUI. You can find an usage example on .\Scripts\UI\Templates\DefaultCharacterSelectionScreen.cs

**Name**: The name of the character.

**Age**: Age of this character. This, along with other information can be later displayed in a custom GUI.

**Blood Type**: The character's blood type (You never know when someone might need blood transfusion).

**Gender**: The character's gender.

**Height**: The character's height.

**Life Points**: How many life points this character has. If you are using percentage as damage to your moves, changing this has no impact on gameplay.

**Max Gauge**: The maximum amount of gauge (meter) this character can retain.

**Portrail (big)**: Can be used in your own GUI. Code example at .\Scripts\UI\Templates\DefaultCharacterSelectionScreen.cs.

**Alternative Color**: In case of mirror match, the game will attempt on changing the rendered color of player 2 to this color.

**Character Selection Animation**: In case of using 3D character layout for the character selection screen, select the animation that triggers when this character is selected.

**Death Sound**: If defeated, this sound will be played.

**Description**: Character description.

**Sub-content:**

* [Character Prefabs](http://www.ufe3d.com/doku.php/character:prefabs)
* [Hit Box Setup](http://www.ufe3d.com/doku.php/character:hitbox)
* [Physics](http://www.ufe3d.com/doku.php/character:physics)
* [Head Look](http://www.ufe3d.com/doku.php/character:headlook)
* [Move Sets](http://www.ufe3d.com/doku.php/character:movesets)
* [AI Instructions (Fuzzy A.I.)](http://www.ufe3d.com/doku.php/character:aiinstructions)

## Character Prefabs

Here you will select the 3D model that represents this character. This model musts be encapsulated into a [Prefab](https://docs.unity3d.com/Manual/Prefabs.html) and it must have the following attached components:

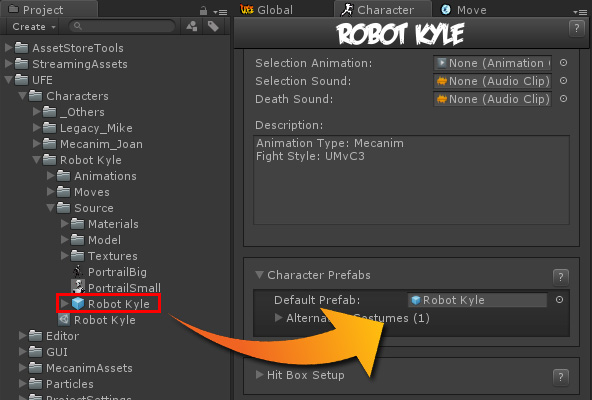
* (SOURCE VERSION) Hit Boxes Script (.\UFE\Scripts\HitBoxesScript.cs)
* (OTHER VERSIONS) Hit Boxes Script (.\UFE\Plugin\UFE.dll\HitBoxesScript)
* [Animation](https://docs.unity3d.com/Manual/class-Animation.html)/[Animator](https://docs.unity3d.com/Manual/class-Animator.html) (Unity Component)

Watch [this tutorial](https://www.youtube.com/watch?v=6tJ7rPAvt_w) to learn how to create a character prefab or follow the tutorial below.

### Create a character prefab（创建角色预制体）

Follow the steps below to create a character prefab for UFE using your recently imported 3D model:

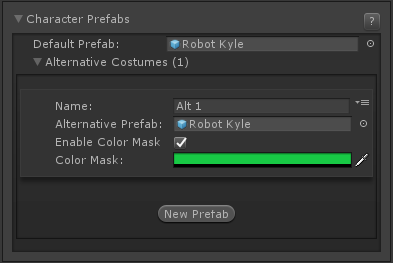
* Drag your character model from your project folder to the **Hierarchy**
* If it doesn't have the animation/animator component yet, click on Add Component and search for the component. Search for animation if you are using [Legacy](http://www.ufe3d.com/doku.php/animationtypes) or animator for [Mecanim](http://www.ufe3d.com/doku.php/animationtypes). No further configuration is necessary.（添加animation/animator component）
* Using the Project window, navigate to .\UFE\Scripts\HitBoxesScript.cs [1)](http://www.ufe3d.com/doku.php/character:prefabs#fn__1) and, with the character still selected, drag the script to the Inspector window. No further configuration is necessary.（添加脚本）
* Once you're done, drag the character back from Hierarchy to your desired folder in the Project window. Unity will automatically create a prefab.



Now drag your newly created prefab to the Default Prefab under the Character Prefabs panel. Once you have your character in, you can select the prefab under [HitBox Setup](http://www.ufe3d.com/doku.php/character:hitbox) to see and edit your character's hit boxes.

### Alternative costumes（服装选择）

To create alternative costumes just unfold the option and click on New Prefab. Repeat the process above to create a different prefab and drag it to the Alternative Prefab field. You can also enable color masks if you just want to have quick color switches to your prefab.



**Notes:**

* If you are using [Mecanim](http://www.ufe3d.com/doku.php/animationtypes), all prefabs **must** use the same [avatar](http://docs.unity3d.com/Documentation/Manual/ConfiguringtheAvatar.html) (set under [Move Sets](http://www.ufe3d.com/doku.php/character:movesets) option).
* UFE automatically attempts to load the first alternative costume listed if both players select the same character. Currently UFE does not have an UI template that allows for custom selection of alt costumes, but we hope to add that in the future. In the meantime feel free to create your own solutions using the open source GUI templates (.GUI\UI Prefabs\CharacterSelectionScreen.prefab and .Scripts\UI\Templates\DefaultCharacterSelectionScreen.cs) selectable under [GUI Options](http://www.ufe3d.com/doku.php/global:gui).
* I strongly recommend engaging with the [community](http://www.ufe3d.com/forum/viewforum.php?id=7) to find solutions on how to interact and apply your custom costumes. I'll even post the link here if someone makes a good tutorial about it!

## Moves (Move Sets)角色执行的动作集

### http://www.ufe3d.com/lib/exe/fetch.php/character:character_moveset.png

**Execution Timing:** How fast, in seconds, the keys listed in button sequence and execution have to be pressed in order to complete a special move.

**Possible Air moves:** How many moves can this character perform while jumping.

**Blending Duration:** How smooth the transition between basic moves are. Be careful as high numbers can cut down your animation and create small visual bugs as it trades response time for smoothness. The recommended (and default) value is 0.1.

**Animation Type:**

* Legacy: Tells UFE to read the animations from this character as Legacy.
* Mecanim: Tells UFE to read the animations from this character as Humanoid. [Click here to watch Mecanim tutorial](http://www.youtube.com/watch?v=8tLeUOL7lYM)

**Avatar:** If Mecanim was selected, drag the character's [avatar](http://docs.unity3d.com/Documentation/Manual/ConfiguringtheAvatar.html) here.

**Animation Control:** Tell the engine which animation control it should use.

* UFE Engine: The animation flow is fully controlled by UFE's own Update cycles. Due to limitations on the Unity engine, use this system to ensure there are no desynchronization when playing network games.
* Unity Engine: The animation flow is controlled by Unity's own update cycles. UFE will attempts to match the hitbox positions to control the moves AFTER Unity update the characters mesh position. Selecting this option may cause minor inconsistencies. If your game requires precise frame data use UFE Engine instead.

### Combat Stances（战斗准备）

Each character can have one or several move sets, all classified by a combat stance. Add a new Combat Stance by clicking the **New Combat Stance** button at the bottom of this panel's section.

**Combat Stance:** A character can have several different fight styles, each with their own unique animations and special moves. In Street Fighter 4 for example we have Gen, a character that can swap between 2 different fight styles. Both fighting styles are completely different, from the way they move to the special moves you can execute. Combat Stances can also be used as “power up” or anything you like. Combat styles can be applied by simply changing the combat stance in the [move options](http://www.ufe3d.com/doku.php/move:start). (有的角色爆气后打斗动作完全不一样)

**Cinematic Intro:** (Optional, **PRO** and **SOURCE** only) a special move designed with just an intro animation and cinematics. UFE will check if you have this and play it at the beginning of each game. Check the example provided at .\UFE\Characters\Mike\Moves\Intro.asset （战斗开场动画）

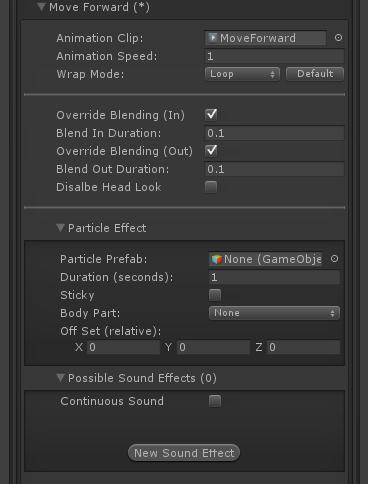
**Cinematic Outro:** (Optional, **PRO** and **SOURCE** only) a special move designed with just an outro animation and cinematics. UFE will check if you have this and play it at the end of each game if this character wins. Check the example provided at .\UFE\Characters\Mike\Moves\Outro.asset （战斗获胜后动画）

### Basic Moves（基本动作）

Set the basic animations and sounds for your character, as well as any particular effect you want to add during these events. Basic Moves are categorised into Standard Animations, Blocking Animations and Hit Animations. （基本动作集包含站立动画、防御动画、攻击动画，并可添加对应的声音及特效）

**Enabled Moves** Disable certain basic moves in case your game doesn't require it. You may disable Move, Jump, Crouch, Block or Parry. Disabling these options completely ignore any requests, including animation and input (from both player and AI). （动作忽略）

#### Basic Information（基本信息）



**Animation Clip:** The animation clip related to this move. Animation files are normally located under the file structure of an FBX file, or just as a single .anim file.

**Animation Speed:** Tune the animation speed to match the desired effect during the game. Some moves like Jumping, Falling and Landing are automatically calculated by UFE based on the jump height.

**Override Blending (In):** Overrides the previous out blending or default blending duration with the value below.

**Blend In Duration:** If override blending (in) is toggled on, this is the new blending in value for this animation.

**Override Blending (Out):** Overrides the blend in set for the next animation or default blending duration with the value below.

**Blend Out Duration:** If override blending (out) is toggled on, this is the new blending out value for this animation.

**Disable Head Look:** If [Head Look](http://www.ufe3d.com/doku.php/character:headlook) is enabled, it will be disabled during this move.

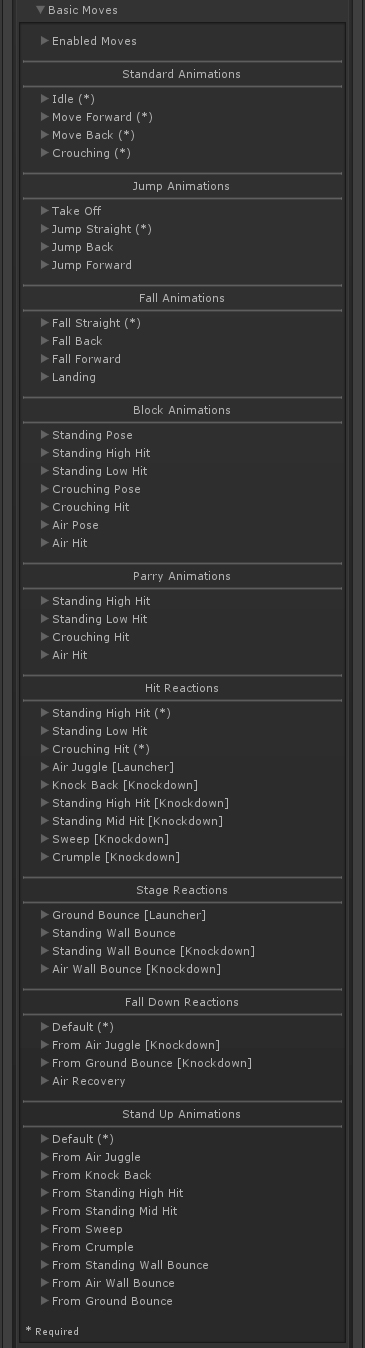
**Particle Effects:**

* Particle Prefab: A prefab with particle effects that is fired when this animation plays.
* Duration (seconds): How long this effect will last on the screen.
* Sticky: If toggled on, the particle will follow the body part as it emits.
* Body Part: The body part relative to the character (defined in the hitbox setup) where this particle will appear.
* Off Set (relative): The position (relative to the body part) this particle will appear when this move plays.

**Possible Sound Effect:** A list of possible sounds that plays at random when this move is executed. If only one sound is added it will only play that sound. UFE works with multi-channels, so don't worry about overlapping.

**Continuous Sound:** Should the selected sound effect keep playing while this move is in motion?

#### Specifics（详情）



**Standard Animations**: Used when character is not moving (Idle), moving back and forward, and crouching (if your game uses it). When choosing the Idle you can also set its resting animations:

* Resting Clips (1-5): When character isn't moving, you can have certain animations play at random.
* Resting Interval (seconds): Every interval cycle the engine will attempt to play one of the resting clips from above. If the randomly selected slot has no clip, it won't play anything. Example: If you want a clip to run every 6 seconds with 100% consistency, fill all the fields with that animation.

**Jump Animations**: These animations trigger when the character initiates a jump. If you don't have the respective Fall Animation setup, the engine will attempt to use these animation as single jump cycles.

* If Auto Speed is toggled the engine will attempt to synchronize the height of the jump with the animation length so both can end at the correct time to link with its next animation. Untoggle to adjust how your animation should play out.
* Take Off animation speeds are used in conjunction with the Jump Delay variable (located on the [physics option](http://www.ufe3d.com/doku.php/character:physics))

**Fall Animations**: These animations trigger when the character is falling from a jump.

* If Auto Speed is toggled the engine will attempt to synchronize the height of the fall with the animation length. Untoggle to adjust how your animation should play out.
* Landing animation speeds are used in conjunction with the Jump Delay variable (located on the [physics option](http://www.ufe3d.com/doku.php/character:physics))

**Block Animations**: Triggered when the character [blocks](http://www.ufe3d.com/doku.php/global:block) (or attempting to block) an attack.

* Standing High Hit is triggered when a an attack hits the upper part of a character while standing. Standing Low Hits is triggered on low hits.
* Blocking Poses triggers when the character is attempting to block as its under the blockable area, but not blocking (only happens when using hold back as the [block type](http://www.ufe3d.com/doku.php/global:block)).
* If Auto Speed is toggled the engine will use the [block stun](http://www.ufe3d.com/doku.php/move:activeframes) information to synchronize it with the animation length. Untoggle to run your own speed and [WrapMode](http://docs.unity3d.com/ScriptReference/WrapMode.html) values.

**Parry Animations**: Triggers when the character successfully [parries](http://www.ufe3d.com/doku.php/global:block) an attack.

* Standing High Hit is triggered when a an attack hits the upper part of a character while standing. Standing Low Hits is triggered on low hits.
* If Auto Speed is toggled the engine will use the parry stun information from [Block Options -> Parry](http://www.ufe3d.com/doku.php/global:block) to synchronize it with the animation length. Untoggle to run your own speed and [WrapMode](http://docs.unity3d.com/ScriptReference/WrapMode.html) values.

**Hit Reactions**: Triggers when the character gets hit.

* Standing High Hit is triggered when a an attack hits the upper part of a character while standing. Standing Low Hits is triggered on low hits.
* For regular standing and crouching attacks you can set different clips based on the Move's [Hit Type](http://www.ufe3d.com/doku.php/move:activeframes).
* Launcher animations are triggered when the character is launched up.
* Knockdown animations knocks the character down (which then triggers respective Stand Up animations).
* If Auto Speed is toggled the engine will use the [hit stun](http://www.ufe3d.com/doku.php/move:activeframes) applied to synchronize it with the animation length. Untoggle to run your own speed and [WrapMode](http://docs.unity3d.com/ScriptReference/WrapMode.html) values.

**Stage Reactions**: Triggers when the character getting hit interacts with the stage (ground and wall bounce).

* You must have the respective bounce options under [Bounce Options](http://www.ufe3d.com/doku.php/global:bounce).
* Unlike Air Hit, Ground Bounce does not operate using the auto-speed algorithm. Use WrapMode on Clamp Forever so the character stays in a single air pose.
* To adjust a stage reaction of your move go to [Active Frame](http://www.ufe3d.com/doku.php/move:activeframes) → Stage Reactions.
* Untoggle Grounded (Standing) under [Active Frame](http://www.ufe3d.com/doku.php/move:activeframes) → Stage Reactions → Wall Bounce Options to use Standing Wall Bounce (knockdown).

**Fall Down Reactions**: Triggered when the character hits the floor after being knocked down.

* The Default animation is triggered whenever the engine can't find the related knockdown animation.
* In certain animations you can set a Fall Clip and a Down Clip. Use this sequence when you want the Down Clip to be a loop (like breathing).
* If you don't want to have 2 clips it's important that the WrapMode is set to Clamp Forever under Fall Clip, so the character stays down (as the last frame should have the character laying down).
* Air Recovery is triggered if the character is recovered from Stun in mid-air. Make sure you have the proper option enabled under [Combo Options -> Air Recovery Type](http://www.ufe3d.com/doku.php/global:combo).
* Use Auto Speed to get the animation to follow the height path or have it end in a single pose with WrapMode set to Clamp Forever.

**Stand Up Animations**: Triggered after the knockdown timer runs out (Adjustable under [Global -> Knock Down Options](http://www.ufe3d.com/doku.php/global:knockdown)).

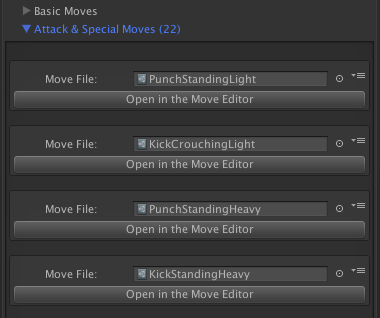
* The Default animation is triggered whenever the engine can't find the related stand up animation.
* All Knockdown animations can have its own follow up stand up animation.  
  Example: You may have a sweep animation that have your character facing down instead of facing up. You can set that animation to end at the down state and have a specific stand up for the sweep animation as a continuation of that clip.

**Notes:**

* You can return the default speed and wrapMode of any Basic Move by clicking on the Default button next to WrapMode.
* Because of how fast a [hit stun](http://www.ufe3d.com/doku.php/move:activeframes) can be, it's recommended that you set the override blending (in) values from Hit, Block and Parry animations to 0.
* None of the Parry moves are required if you don't have [parry enabled](http://www.ufe3d.com/doku.php/global:block) in your game.
* None of the Block moves are required if you don't have [block enabled](http://www.ufe3d.com/doku.php/global:block) in your game.
* Bounce and Falling From Bounce animations are not required if you don't have bounce enabled in your game.
* Some animations allow you to hide the hitboxes. Can be useful if you want certain animations to leave the character invincible during a certain state. A good example is having Air Recovery invincible to emulate the Street Fighter style of air recovery (along with the option Cant Move under [Air Recovery Type](http://www.ufe3d.com/doku.php/global:combo)).

### Attack & Special Moves（攻击和特殊动作）

After creating a [new move](http://www.ufe3d.com/doku.php/move:start), add the move to the character by clicking **New Move** than dragging the file to the following field.



**Notes:**

* Moving a move file up/down with the panel options has no effect on priority of moves. It only affects the list order displayed in this panel.
* You can find several move samples under .\UFE\Characters\Mike\Moves

Code access:

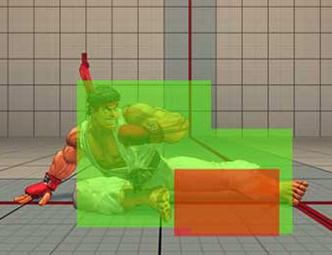
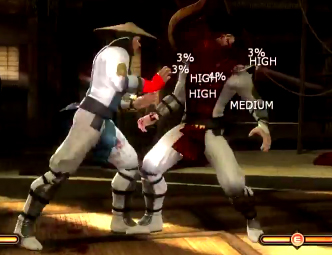
UFE.config.player1Character.moves

UFE.config.player2Character.moves

## Hit Box Setup（打击区域设置）

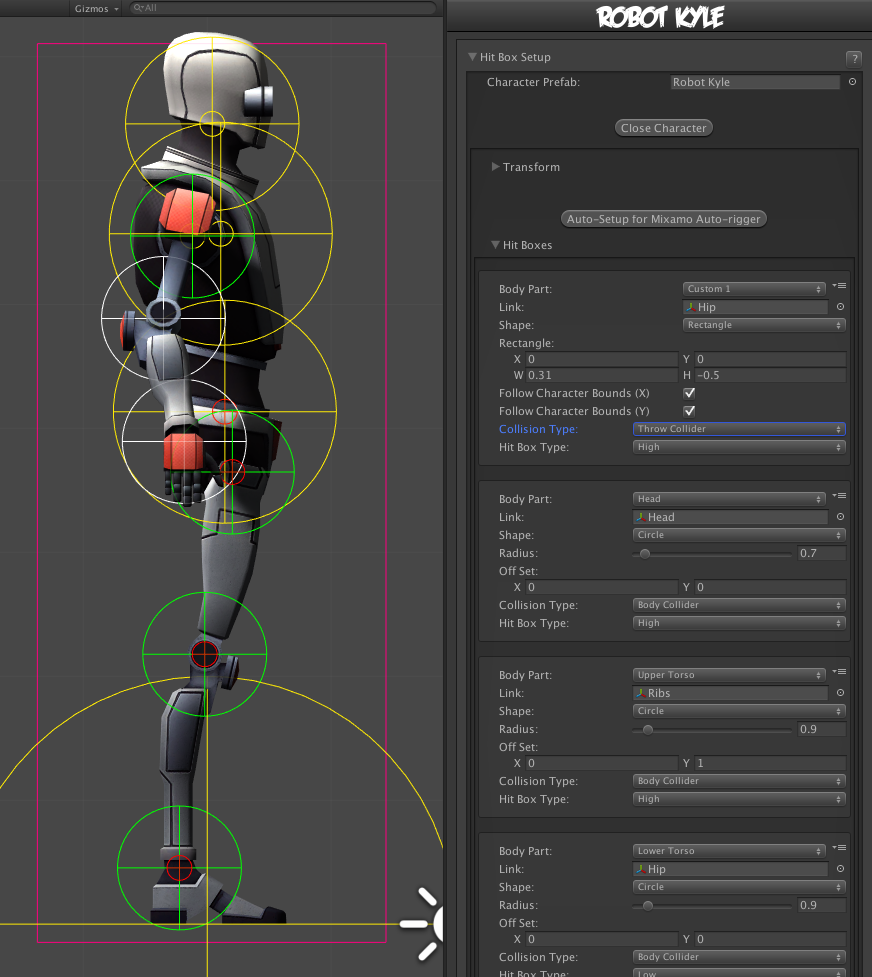
One of the key ingredients of any fighting game are hitboxes. Different than other action games, 2.5D fighting games rely on precision hits in a more “mechanical” way, as it emulates the sensation of playing a 2D game with the precision (and impact) of well adjusted invisible hitboxes.

There are many types of hitboxes, around, and thanks to the recent use of 3D technology to emulate 2D game play, hitboxes are commonly known for following body joints. Here are some examples:



UFE uses a similar technology: if a hitbox and a [*hurtbox*](http://www.ufe3d.com/doku.php/move:activeframes) are overlapping then a hit is confirmed. （打击区域和伤害区域重叠则确认攻击生效）

UFE uses circles and rectangles as hitboxes which gives you a wide variety of game play to work with. Circles and rectangles can be used simultaneously, UFE will detect hits between any combination of hitbox/hurtbox shapes. （可以共用圆形和矩形）



Make sure you first set your character model(s) under [Character Prefabs](http://www.ufe3d.com/doku.php/character:prefabs) before proceeding to the sections below.

### Transform

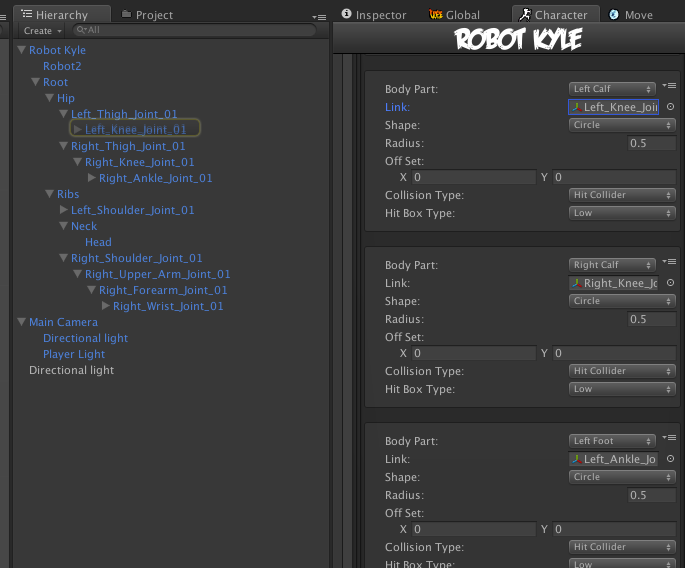
Works similar to Unity's default Transform component. Adjust your character's rotation so it looks like its standing on the left side of the screen.

**Notes:**

* Avoid changing the character's position as it will save its “local position” and it might cause some visual bugs in the future.
* Even though you can use the standard t-pose as the default animation when importing your character, its recommended that you use the idle animation as your character's default pose/animation. It makes it easier to work with hitboxes, specially if you adjust the character rotation.
* Make sure you adjust the characters rotation as you would like to see them on screen. 2.5D Fighting games tend to have more “theatrical” pose, in a sense that their body language is constantly facing the camera (or “audience”), while their heads are facing the opponent.
* Don't worry about mirroring your animations. UFE will take care of that for you.

### Hit Boxes

Tag your character's body parts to the hit boxes, set its shape, size, collision type and hitbox type.



* **Body Part:** A tag representing the body part inside UFE.
* **Link:** The body part related to the character on screen. Navigate through the mesh and select the bone that best cover the selected body part.
* **Shape:** Circle or Rectangle shape hitbox.
  + (Circle) **Radius:** How big the hitbox is. Its recommended that yellow hitboxes cover as much ground as possible to avoid unexpected cross-overs during battle.
  + (Circle) **Off Set:** Off set the hit boxes around the character body if needed. Be careful with high values, as different animations can cause unexpected positioning.
  + (Rect) **Rectangle**: X and Y make up the bottom left corner; W and H is the size calculated as distance from X and Y respectively.
  + (Rect) **Follow Bounds (X/Y):** Toggle this if the rectangle shape is to follow the bounds of the character. Note that this uses Unity's [Renderer Bounds](http://docs.unity3d.com/ScriptReference/Renderer-bounds.html). Adjust W and H to suit.
* **Collision Type:**
  + Body Collider (yellow hitboxes): Movement and hit collision. Opposing yellow hitboxes cannot overlap.
  + Hit Collider (green hitboxes): Hit collision only. Opposing green hitboxes can overlap.
  + No Collider (white hitboxes): No collision. Use these for body parts that are not considered hitboxes but can be a [hurtbox](http://www.ufe3d.com/doku.php/move:activeframes) if needed.
  + Throw Collider (red hitboxes): Used to detect attacks that have hittype of Throw. If the character doesn't have a Throw Collider hitbox, then throws will not work against them. See also: [Throw Creation Guide](http://www.ufe3d.com/doku.php/throwtutorial:throw_tutorial)
* **Hit Box Type:**
  + High: If a non-knockdown attack hits this hitbox, Get Hit High animation group is triggered.
  + Low: If a non-knockdown attack hits this hitbox, Get Hit Low animation group is triggered.
* **Default Visibility:** Set the default visibility for certain Game Objects under your character hierarchy. This option is useful for hiding or showing certain objects during gameplay (like weapons). This option only works for objects detached from the main [Mesh](http://docs.unity3d.com/Manual/class-Mesh.html). You can then toggle the visibility by using [Move Editor -> Body Parts Visibility Toggle](http://www.ufe3d.com/doku.php/move:start).

**Note:**

* Each bodypart should be unique. If any bodypart doubles up, it can cause strange behaviour with hit detection. Use Custom 1-9 if you run out of defined ones.
* If you rigged your model using [Mixamo Auto-Rigger](https://www.mixamo.com/c/auto-rigger/) you can click on **Auto-Setup for Mixamo Auto-rigger** to automatically set the hit boxes of your character.
* After the last step if you can't see the hitbox in scene view, click on Gizmos (top right corner of scene view) and toggle “HitBoxesScript”.

### Video Tutorial

Code example:

void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){

if (strokeHitBox.type == HitBoxType.low) Debug.Log ("low hit!");

}

## Physics(物理系统)

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. （按压敏感度）

**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”. （角色重量，影响空中施力，包括跳跃）

**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 == "Dhalsim") {

UFE.config.player2Character.physics.weight = 175;

}

}

}

Code example 2:

void OnMove(MoveInfo move, CharacterInfo player){

if (move.moveName == "oil up"){

player.physics.friction = 10;

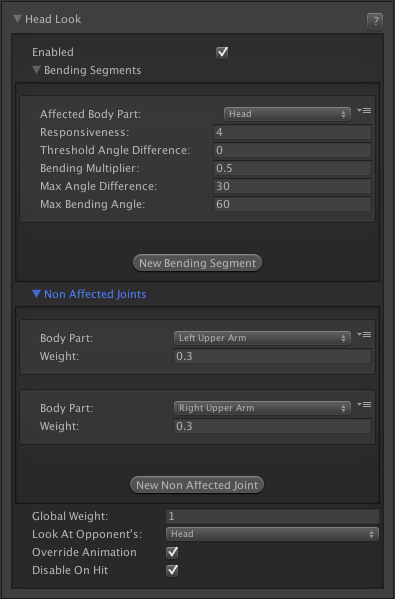
}

}

## Head Look（头部扭动，看起来生动）

Head Look is based on a script made by [Rune Skovbo Johansen](http://blogs.unity3d.com/2009/07/10/just-looking-around/). You can find the original script [here](http://wiki.unity3d.com/index.php?title=HeadLookController).

**Enabled**: You can disable Head Look if you don't want this character to use it.

[](http://www.ufe3d.com/lib/exe/detail.php/character:character_headlook.png?id=character:headlook)

### Bending Segments（融合部分）

Which segments of the rig will be used to “look at” the opponent.

* **Affected Body Part:** Which body part will be used to rotate. Its recommended that you assign the “Head” here (if hit boxes are set up)
* **Responsiveness:** How fast the head will move to follow the opponent
* **Threshold Angle Difference:** When selecting “head” as your body part, the engine will attempt to use its parent (normally the neck) as the bending angle control. This limits how far can the segments “bend” to account for the target direction.
* **Bending Multiplier:** Bending Multiplier control how much the segment bends or turns compared to how big the target angle is.
* **Max Angle Difference:** The segment will attempt to be no more than Max Angle Difference away from the target angle, though it won’t bend any further once it reaches the Max Bending Angle.
* **Max Bending Angle:** How far can the parent bone (the neck) bend.

### Non Affected Joints（不要影响的关节）

When attempting to bend and rotate certain body parts to accommodate the target direction, you can choose to make it so certain body parts and its children are not affected by the rotation. An example can be found under Robot Kyle, in which the Left Upper Arm and Right Upper Arm are unaffected by the head rotation.

* **Body Part:** Which body part should not be affected.
* **Weight:** By setting it to 0, no movement will be applied to this body part.
* **Global Weight:** You can set the overall effectiveness of this script using this value (0-1).
* **Look At Opponent's:** Which body part from the opponent should this character look at.
* **Override Animation:** If enabled the engine will attempt to override the current rotation of the character to account for the animation. This variable seems to have different results depending on the situation.
* **Disable On Hit:** If enabled, getting hit will disable head look (more realistic).

**Notes:**

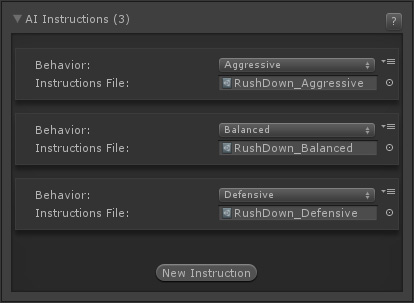
* You can find more information about this script [here](http://blogs.unity3d.com/2009/07/10/just-looking-around/).
* Some characters seems to behave differently depending on how they were originally setup. We recommend playing with all the values until you find the right tuning for each character.
* An example of this script in action can be found on Robot Kyle (Source version only).

Code access:

UFE.config.player1Character.headLook

## AI Instructions

Add AI instruction files here. See [AI Editor](http://www.ufe3d.com/doku.php/ai:start) and [Global Editor AI Options](http://www.ufe3d.com/doku.php/global:aioptions) for more information.



Click the **New Instruction** button to add a new instruction file.

**Behavior**: Select from Very Defensive, Defensive, Balanced, Aggressive and Very Aggressive. Choose one that best matches the instruction file you'll add below. This classification will let you dynamically change between files during a match by [switching behaviors](http://www.ufe3d.com/doku.php/ai:customrules?s%5b%5d=change%20behaviour#reaction).

**Instruction File**: Drag the AI Instruction file here (templates can be found at UFE Addons\AI Addon\Templates).

# Move Editor（编辑动作）

Create and edit all your attacks and special moves here. This is the kind of tool used by big companies like Capcom and NeatherRealms to create and balance their games.

Open an existing move file or create a new one. For this introduction we will be using several moves from the folder below. These moves are already targeted to the character Mike.

Folder Location: .\UFE\Characters\Mike\Moves  
(SOURCE VERSION) Inherits from: .\UFE\Scripts\MoveInfo.cs  
(OTHER VERSIONS) Inherits from: .\UFE\Plugins\UFE.dll\MoveInfo

To create a new move, in the project window click on Create → U.F.E. → Move File.

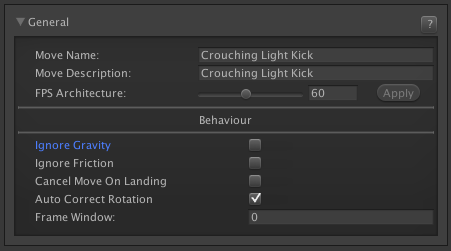
## Content



* [General](http://www.ufe3d.com/doku.php/move:general)
* [Gauge/Meter Options](http://www.ufe3d.com/doku.php/move:gauge)
* [Animation](http://www.ufe3d.com/doku.php/move:animation)
* [Active Frames](http://www.ufe3d.com/doku.php/move:activeframes)
* [Opponent Override](http://www.ufe3d.com/doku.php/move:opponentoverride)
* [Player Conditions](http://www.ufe3d.com/doku.php/move:playerconditions)
* [Input](http://www.ufe3d.com/doku.php/move:input)
* [Chain Moves](http://www.ufe3d.com/doku.php/move:chainmoves)
* [Cinematic Options](http://www.ufe3d.com/doku.php/move:cinematics)
* [Particle Effects](http://www.ufe3d.com/doku.php/move:particleeffects)
* [Sound Effects](http://www.ufe3d.com/doku.php/move:soundeffects)
* [Text Alerts](http://www.ufe3d.com/doku.php/move:textalerts)
* [Stance Changes](http://www.ufe3d.com/doku.php/move:stancechanges)
* [Self Applied Force](http://www.ufe3d.com/doku.php/move:selfappliedforce)
* [Body Parts Visibility Changes](http://www.ufe3d.com/doku.php/move:bodypartsvisibilitychanges)
* [Slow Motion Effects](http://www.ufe3d.com/doku.php/move:slowmotioneffects)
* [Armor Options](http://www.ufe3d.com/doku.php/move:armor)
* [Invincible Frames](http://www.ufe3d.com/doku.php/move:invincibleframes)
* [Projectiles](http://www.ufe3d.com/doku.php/move:projectiles)
* [A.I. Definitions](http://www.ufe3d.com/doku.php/move:aidefinitions) ([Fuzzy A.I Addon](http://forum.unity3d.com/threads/u-f-e-addon-fuzzy-a-i.277086/) Required)

## General Options（公共设置）

Set general options of your move here.



**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:

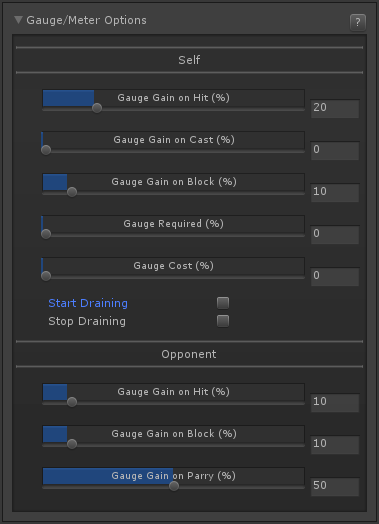
void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){

if (move.attackType == AttackType.EX) Debug.Log("EX Move fired!");

}

## Gauge Options（动作的能量设置，动作打击获取能量及需要的能量）

You can set the move's gauge gains and requirements here.



### Self

**Gauge Gain on Hit:** How much gauge is gained when this move hits.

**Gauge Gain on Cast:** How much gauge is gained when this move is cast.

**Gauge Gain on Block:** How much gauge is gained when this move is blocked.

**Gauge Required:** How much gauge is needed to activate this move.

**Gauge Cost:** How much it costs to play this move. If the player does not have enough gauge, it will drain all of it.

**Start Draining:** Once triggered, the gauge will start draining using the values below.

* Inhibit Gauge Gain: If toggled the character will not gain gauge while draining is active.
* Drain Per Second (%): How much meter will be drained over the course of a second while drain is active.
* Total Drain (%): How much should be drained total (to drain the full bar just type in 100).
* Move (Drain Complete): Once the drain is over you can force the character to automatically play a move (like [Jason Voorhees](http://mortalkombat.wikia.com/wiki/Jason_Voorhees)' Pursuit move in Mortal Kombat X). This move will automatically overwrite any move the character might be playing.
* Stance (Drain Complete): Switch stances once drain is complete (see Robot Kyle's Stance Switch move for more information).

**Stop Draining:** If the meter is currently being drained, having this toggled will immediately stop the drain.

### Opponent

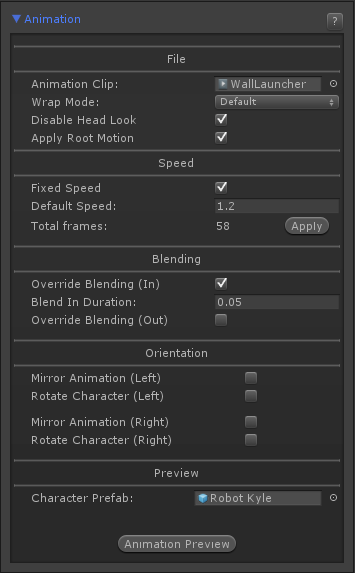
**Gauge Gain on Hit:** How much gauge will the opponent get if they get hit by this attack.

**Gauge Gain on Block:** How much gauge will the opponent get if they block this attack.

**Gauge Gain on Parry:** How much gauge will the opponent get if they parry this attack.

## Animation Options（动作的动画）

Set and test move animations here.



### File

**Animation Clip:** The animation file (it's the file under your imported FBX file structure).

**Wrap Mode:** The animation wrap mode. Only works if Animation Flow is set to Smoother in the [advanced options](http://www.ufe3d.com/doku.php/global:advanced) menu.

**Disable Head Look:** If enabled, will disable [Head Look](http://www.ufe3d.com/doku.php/character:headlook) during this move.

**Apply Root Motion:** If enabled, the animation will have total control of the characters position in the field during its duration. When the animation is over, the engine will automatically re-adjust to new position and realign the Z axis. If you are using Mecanim, make sure you have the proper [animation setup](http://docs.unity3d.com/Manual/RootMotion.html).

* Root Motion Node: If a body part is selected, after the animation ends the engine will attempt to force the character's position to the location of that part.
* Force Grounded: If toggled the character will automatically be brought back grounded on Standing state.

### Speed

**Fixed Speed**: Toggle this option to have a fixed animation speed throughout the entire move. Untoggle it to use speed keyframes.

**Default Speed:** The desired animation speed. Use negative numbers to make the animation play in reverse. When Fixed Speed is disabled, this the initial animation speed used in the keyframe timeline.

**Speed Key Frames:** Changes the animation speed at runtime, altering its frame data properties.

* Casting Frame: The frame in which the speed will change.
* New Speed: The new speed value.
* Frame Window (Aprox.): The amount of frames the animation will have in between this and the next key frame.

**Animation Speed:** The slider allows you to tune the speed of your animation, being negative or positive, while the blue bar shows how fast that animation will play out in the end. Negative speed will make your animation play in reverse.

**Total Frames:** Upon changing the speed of your animation (or using speed key frames), UFE will calculate (based on the FPS architecture of this move) how many frames it has. If you are satisfied with the results, click apply. Note: Make sure you set these before working with other frame options as changing these values will affect other frame data.

### Blending

**Override Blending (In):** If enabled, this value will override the default blending value set for the character. Setting this value too high may affect the frame data of your move.

**Override Blending (Out):** If enabled, this value will override the default blending value set for the character when the animation ends.

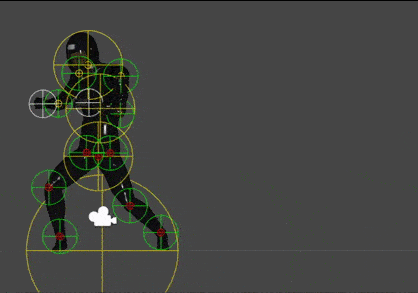
### Orientation

Set orientation and mirror settings for the move here.

**Mirror Animation (Left):** When character is on Left side, mirror the animation.  
**Rotate Character (Left):** When character is on Left side, rotate the character.  
NOTE: Setting both of the above effectively performs the [Auto Mirror](http://www.ufe3d.com/doku.php/global:rotation) but without requiring the character to be on the other side.

**Mirror Animation (Right):** When character is on Right side, mirror the animation.  
**Rotate Character (Right):** When character is on Right side, rotate the character.  
NOTE: Setting both of the above effectively performs the [Auto Mirror](http://www.ufe3d.com/doku.php/global:rotation) but without requiring the character to be on the other side.

### Preview

[](http://www.ufe3d.com/lib/exe/detail.php/move:movepreview.gif?id=move:animation)

**Character Prefab:** Drag your [character prefab](http://www.ufe3d.com/doku.php/character:hitbox) to this field in order to use the Animation Preview and Cinematics. This is only a preview reference for the move, the move does not require a prefab here.

**Animation Preview:** Preview the move animation as well as its hurtboxes. If you are using Mecanim animation, make sure the prefab has the [Animator component](https://docs.unity3d.com/Documentation/Components/class-Animator.html) attached with the correct avatar.

**Smooth Preview:** Play the animation in smooth motion instead of frame by frame.

**Notes:**

* Make sure the prefab's animation type matches the animation clip's animation type (Legacy vs Mecanim), or the preview will not work.

Code example:

void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){

Debug.Log("This move has a total of " + move.totalframes + " frames");

Debug.Log("Startup frames: " + move.startupFrames);

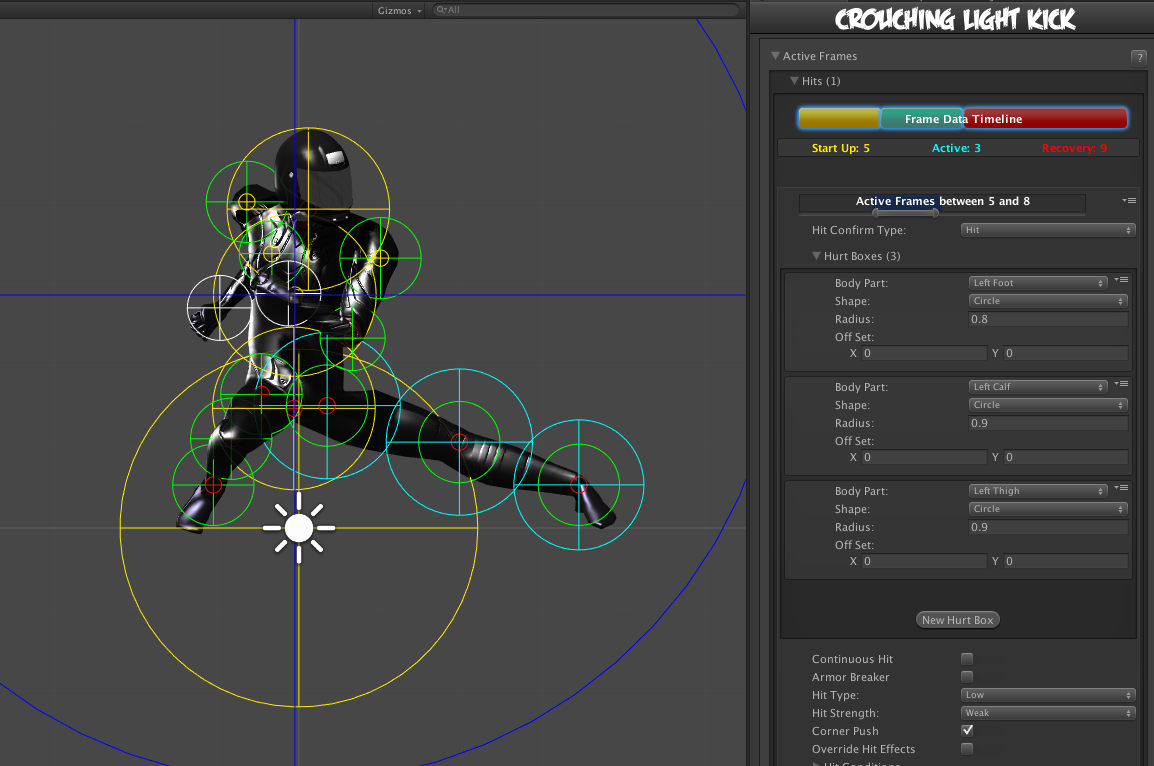
Debug.Log("Active frames: " + move.activeFrames);

Debug.Log("Recovery frames: " + move.recoveryFrames);

}

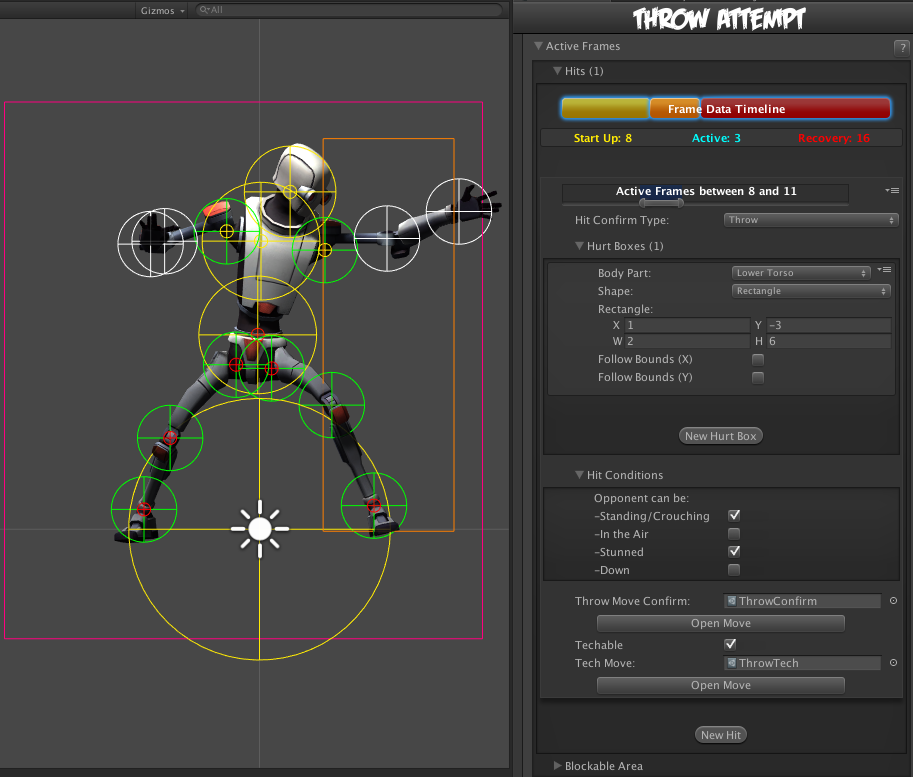
## Active Frames（激活帧）

Here you can set when (and where) your move will become hit active, as well as define every aspect of the hit, from damage to frame advantage.   
In this documentation sometimes you will read “frames” as a reference to time. In fighting game terminology, a frame is, based on the standard 60 fps, 1/60 of a second. For example, a move that has 30 frames lasts half a second. Don't worry though, as this tutorial makes no requirement of such frame rate.



In this example we will be using .\UFE\Characters\Mike\KickCrouchingLight.asset or .\UFE\Characters\Robot Kyle\Moves\ThrowAttempt.asset

### Hits



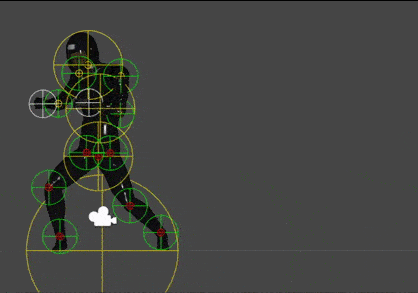
**Active Frames:** Set when this hit becomes active during the move. Moves can have multiple hits, and you can define the timing of each hit by carefully spacing each group of active frames and adjusting frame advantage. To read more about start up, active and recovery frames, follow this [link](http://wiki.shoryuken.com/Super_Street_Fighter_IV/Controls_and_Terminology/Startup_Active_and_Recovery_Frames).

* Start Up (Frames): The amount of frames that this move takes before having active frames.（攻击生效前的帧）
* Active (Frames): The amount of frames this move has during its active state. （攻击生效的帧）
* Recovery (Frames): The amount of frames this move has in order for this character to be safe (move again).（攻击生效后的帧，收招）

**Hit Confirm Type:**

* **Throw**: When the hit confirms, the character will immediately play the move listed on Throw Move Confirm field. If the opponent also executes a throw, both characters will immediately execute the move listed on Tech Move. This is not to be confused with the [Chain Moves](http://www.ufe3d.com/doku.php/move:chainmoves) links. See also: [Throw Creation Guide](http://www.ufe3d.com/doku.php/throwtutorial:throw_tutorial).
* **Hit**: A Regular attack. Allows for the standard hit options such as damage, hit stun, force, etc.

### Hurt Boxes（伤害区域）

[](http://www.ufe3d.com/lib/exe/detail.php/move:movepreview.gif?id=move:activeframes)Hurt boxes are the selected body parts from your character (previously set on [Hit Box Setup](http://www.ufe3d.com/doku.php/character:hitbox)) that become active during your move. When they collide with the opponent, a hit is detected. Each hit can be filled with several different sets of hurt boxes.

* **Body Part:** The selected body part that will hold the hurt box.
* **Shape:** Select whether it's a circle or rectangle. When rectangle is selected, X and Y are the offset for the related body part.
* **Radius:** The size of this hurt box. Sometimes, for balancing purposes, you might want to have bigger hurt boxes than the animation seems to have. This can be adjusted here.
* **Off Set:** Off sets the hurt box positions.
* (Rectangle Only) **Follow Bounds (X/Y):** Toggle this if the rectangle shape is to follow the bounds of the character. Note that this uses Unity's [Renderer Bounds](http://docs.unity3d.com/ScriptReference/Renderer-bounds.html). Adjust W and H to suit.

### Hit Conditions（攻击条件，对手处在什么状态可以被击中）

#### Basic Filters

For this attack to hit, the opponent can be in any of the below conditions. You must have at least one of these on for the move to be able to hit the opponent.

* **Standing/Crouching:** In standing or crouching, including if they're attacking while in this state.
* **In the Air:** In the air, including if they're attacking in air.
* **Stunned:** In hit stun.
* **Down:** In a knocked down state. Make sure you have [knockdown](http://www.ufe3d.com/doku.php/global:knockdown) hitboxes toggled on.

NOTE: If Hit Confirm Type is set to Throw and Basic Filters → Stunned enabled, the throw can connect if the opponent is in hit stun, allowing this Throw to be part of combos.

#### Advanced Filters

Make it so the attack will only hit confirm if the opponent is playing one of the following basic moves or in one of the states below. Leave both options at 0 if you don't want to use any filter.

### Hit Confirm Type: Hit（攻击类型）

**Continuous Hit:** If enabled, as long as the active hurtboxes are in contact with the opponent, a new hit will be confirmed, spaced out by its Hit Strength speed. Useful for making moves like Chun-li's Lightning Legs or Blanka's Electricity. （连续攻击）

(Continuous Hit) **Space Between Hits:** How much interval between each hit should it hit again. Selecting High for example will have it cause less hits. （连续攻击的攻击间隔）

**Armor Breaker:** If enabled, this move will ignore any armor an opposing move has.

**Unblockable:** If enabled, this move will ignore any attempt at blocking.

**Hit Type:** Determine the hit conditions for this move to be blockable or trigger a different animation.

* Mid: Can be blocked high (standing) or low (crouching).
* Low: Can only be blocked low.
* Overhead: Can only be blocked high. Its recommended to always use this for air moves.
* Launcher: Can be blocked high or low. If hit, sets the opposing character direct into juggle animation.
* High Knockdown: Instantly send the opponent into Get Hit High Knockdown animation. This hit can only be blocked high.
* Mid Knockdown: Instantly send the opponent into Get Hit High Knockdown animation. This hit can be blocked either high or low.
* Knock Back: Instantly send the opponent into Get Hit Knock Back animation. Must apply vertical force for it to work.
* Sweep: Instantly send the opponent into Get Hit Sweep animation. This hit can only be blocked low.

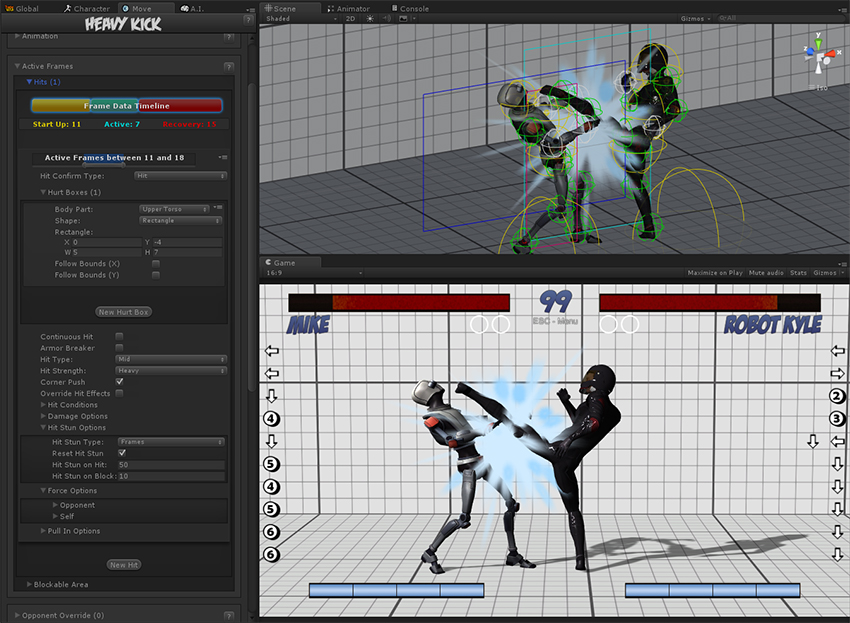
**Hit Strength:** Set what kind of hit this is based on your [hit effect](http://www.ufe3d.com/doku.php/global:hiteffects) options. Crumple hit can also be used as a [knockdown](http://www.ufe3d.com/doku.php/global:knockdown).

**Reset Hit Animation:** If toggled on, every consecutive hit after the first one will restart the hit animation.

**Force Stand:** If toggled the opponent will automatically stand up if they are crouching.

#### Damage Options

* **Damage Type:** Set if the damage is in points or percentage.
* **Damage on Hit:** Damage caused on hit.
* **Damage on Block:** Damage caused on block.
* **Damage Scaling:** Whether or not should this damage be scaled down based on the hits in a combo. More options available at [Combo Options](http://www.ufe3d.com/doku.php/global:combo).
* **Hit Doesn't Kill:** If toggled, this attack will deal damage, but it won't kill the opponent. Useful for cinematic moves that can end a match in one specific attack, or poison like attacks.



#### Hit Stun Options

* **Reset Hit Stun:** If you are using [hit stun deterioration](http://www.ufe3d.com/doku.php/global:combo), enable this to reset the stun and its deterioration value.

**Hit Stun Type - Frame Advantage**  
You can choose to set your hit stun as a static frame advantage value. In UFE, frame advantage is calculated by accounting the remaining frames of an animation when it hits plus the defined values below. Frame advantage does not take into consideration move cancels, just the total animation frames.

* **Frame Advantage on Hit:** If positive, after this move animation ends, the opposing character will remain in hit stun for this amount of frames. For more information about frame advantage, check out this [link](http://www.eventhubs.com/guides/2009/feb/17/how-read-frame-data-street-*%20fighter-4/).
* **Frame Advantage on Block:** If positive, after this move animation ends, the opposing character will remain in block stun for this amount of frames. For more information about frame advantage, check out this [link](http://www.eventhubs.com/guides/2009/feb/17/how-read-frame-data-street-fighter-4/).

To quote Ashn0d from Eventhubs on frame advantage:

A positive number means how many frames faster than your

opponent you will recover after successfully hitting them

with this attack. Negative numbers mean how much faster

your opponent will recover than you after being hit.

**Hit Stun Type - Frames**  
Set the hit stun as a raw frame count format.

* **Hit Stun on Hit:** The amount of frames the opponent will remain stunned for after a hit.
* **Hit Stun on Block:** The amount of frames the opponent will remain stunned for after a block.

**Hit Stun Type - Seconds**  
Set the hit stun time in seconds.

* **Hit Stun on Hit:** The amount of seconds the opponent will remain stunned for after a hit.
* **Hit Stun on Block:** The amount of seconds the opponent will remain stunned for after a block.

#### Force Options

**Opponent/Self:**

* **Reset X Forces:** Reset any previous horizontal forces applied to the target before applying new forces.
* **Reset Y Forces:** Reset any previous vertical forces applied to the target before applying new forces.
* **Applied Force:** The horizontal (x) and vertical (y) forces that will be applied to the target on hit. These forces may vary according to friction, character weight and gravity.
* Apply Different Air Force: If toggled, you can have a different set of force values applied to the opponent if they are air borne.

#### Stage Reactions

**Corner Push:** Toggle if this character should be pushed away when attacking opponent in corner. Helps prevent lockdown situations. NOTE: If [Never Corner Push](http://www.ufe3d.com/doku.php/global:combo) is on, this is ignored.

**Ground Bounce:** Enable this to make it so this attack forces a ground bounce (if you have [Ground Bounce](http://www.ufe3d.com/doku.php/global:bounce) enabled).

* **Override Bounce Forces**: Override the standard [force options](http://www.ufe3d.com/doku.php/global:bounce) and use the options below instead.
* **Reset X Forces:** Reset any previous horizontal forces applied to the target before applying new forces.
* **Reset Y Forces:** Reset any previous vertical forces applied to the target before applying new forces.
* **Applied Force:** The horizontal (x) and vertical (y) forces that will be applied to the target on hit.

**Wall Bounce:** Enable this to make it so this attack triggers a wall bounce (if you have [Wall Bounce](http://www.ufe3d.com/doku.php/global:bounce) enabled).

* **Knockdown**: Toggle this to knock the opponent down when it bounces off the wall.
* **Bounce on Camera Edge**: Makes it so the opponent bounces off the edge of the screen regardless of stage position (similar to Juri's EX Dive Kick).
* **Override Bounce Forces**: Override the standard [force options](http://www.ufe3d.com/doku.php/global:bounce) and use the options below instead.
* **Reset X Forces:** Reset any previous horizontal forces applied to the target before applying new forces.
* **Reset Y Forces:** Reset any previous vertical forces applied to the target before applying new forces.
* **Applied Force:** The horizontal (x) and vertical (y) forces that will be applied to the target on hit.

NOTE: Check out Robot Kyle's WallLauncher for an example of a Wall Bounce move.

#### Pull In Options

Pulls characters close to one another until the determined body parts collide. Useful for moves that needs to adjust the opponent in a fixed position before applying a hit. Much like Ibuki's Raida attack.  
If you apply a low speed, this can also be used to pull enemies in as it follows an animation. Without going into much detail, something like Scorpion's Spear hook could be coded using this.

**Opponent Towards Self:** Pulls the opponent towards the character.

**Self Towards Opponent:** Pulls the character towards the opponent.

* Speed: Speed in witch the body parts will run into one another.
* Body Part (self): The character's body part that will magnetize towards the opponent's body part.
* Body Part (enemy): The opponent's body part that will magnetize towards the character's body part.

**Speed:** How fast should the character be pulled to target

**Distance:** How close should it get to target

**Force Ground Stand:** If enabled, if this attack hits while the opponent is in the air, they will immediately be pulled back to the ground in stand-stunned pose.

#### Override Events (On Hit)

**Override Hit Effects:** Override the [default hit effects](http://www.ufe3d.com/doku.php/global:hiteffects) (based on Hit Strength) for a custom one.

**Override Hit Animation:** Overrides the hit animation that would otherwise play with the one on a selected basic move.

**Override Hit Acceleration:** When toggled the animation will start at 1.5 of its original speed and decelerate according to the stun applied. Disable to have full control of your hit animation speed.

**Override Effect Spawn Point:** Select another position for the hit effect to spawn.

**Override Hit Animation Blend-in:** Select a different value from the one used under [Basic Move](http://www.ufe3d.com/doku.php/character:movesets).

**Override Juggle Weight:** Toggle it to change the opponent's weight during a juggle (useful to deal with characters like Dhalsim).

**Override Air Recovery Type:** For just this hit confirm, overrides the current [air recovery type](http://www.ufe3d.com/doku.php/global:combo).

**Override Camera Speed:** Overrides the current camera movement and rotation for a determined amount of time (useful when combined with Wall Bounce options).

### Hit Confirm Type: Throw（摔）

See [Throw Tutorial](http://www.ufe3d.com/doku.php/throwtutorial:throw_tutorial) for more information on creating throw moves.

**Throw Move Confirm:** The throw move to cast when this throw attempt successfully hits.

**Techable:** Can this throw attempt be countered with a “Tech”?

**Tech Move:** If Techable is toggled on, this is the move the player casts for the tech.

### Hit Conditions

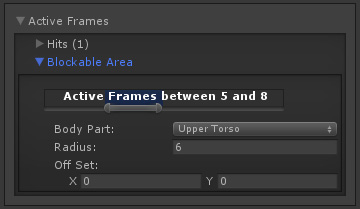
For this move to hit, the opponent can be in any of the below conditions. You must have at least one of these on for the move to be able to hit the opponent.

* **Standing/Crouching:** In standing or crouching, including if they're attacking while in this state.
* **In the Air:** In the air, including if they're attacking in air.
* **Stunned:** In hit stun.
* **Down:** In a knocked down state. Make sure you have [knockdown](http://www.ufe3d.com/doku.php/global:knockdown) hitboxes toggled on.

NOTE: If Hit Confirm Type is Throw, setting Stunned to on will mean the throw can connect if the opponent is in Hit Stun, this can allow throws to be part of combos.

### Blockable Area（防御区域）

The area that will trigger a blocking pose from the opponent if they are holding the block button. Leave the radius at 0 if you want this move to be unblockable. （当触发防御区域时，如果对手按下了防御按钮，将触发防御动作）



* **Active Frames:** The frames in which the opponent can block this move. It's recommended to have it covering all the active hit frames.（哪些帧能被防御，建议包含所有打击帧）
* **Body Part:** The body part in witch the block area will spawn and follow.
* **Shape:** Select whether it's a circle or rectangle. When rectangle is selected, X and Y are the offset for the related body part.
* **Off Set:** If necessary, off set the area.

**Note:** By default the block area is already set to move about half a unit forward.

Code example:

void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){

foreach(Hit hit in move.hits){

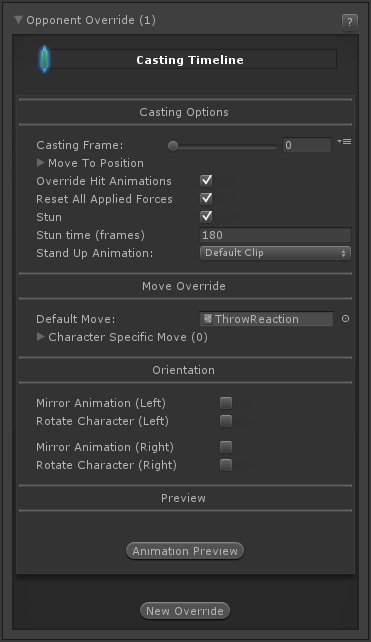
Debug.Log ("Damage: "+ hit.damageOnHit);

}

}

## Opponent Override(对手动作撤销)

Here you can set an override move for your opponent. This is mainly used for throw reactions, or any move that requires the opponent to react in a specific way.（对手动作撤销主要用于摔的反应或对手的其他特殊方式）



When the casting frame is reached, it will immediately override the opponent position and move with the one listed here. This move will act like it's being cast by the opponent. To override their controls, make sure you also set the stun options. （当抛投的帧已经到达，将会终止对手的动作并覆盖其位置，并撤销对手的控制，也可以用于击晕选项）

### Casting Options（投掷选项）

**Casting Frame:** Frame to cast opponent override move.

**Move To Position:** You can force the opponent to a specific position to better sync up the override move with this move. (强制改变对手位置)

* **Position:** The position, relative to this character, to move the opponent to.
* **Move Speed:** How fast/smooth to move the opponent to this position (as percentage).

**Override Hit Animations:** When toggled the override animation will not be interrupted by regular [hits](http://www.ufe3d.com/doku.php/move:activeframes). You will want this for predefined throw reactions. （不能被正常攻击打断）

**Reset All Applied Forces:** Toggle to reset all applied forces to the opponent to 0. If the override animation moves the character realistically, you should also override applied forces to prevent affecting the opponent with physics. （重置对手身上施加的力）

**Stun:** Force the opponent into a stunned state, thus unable to move. （强制对手进入眩晕状态）

* **Stun time (frames):** How long will the stun last, in frames. Manually controlling the stun time is also useful if you want the opponent to be able to partially react to a certain move sequence (such as Killer Instinct combo breakers).
* **Stand Up Animation:** If the opponent is still stunned after the move ends, you can opt to have the animation end with a knock down (such as throws/grappling moves). When that happen you can select which of the predefined Stand Up animations the character should trigger after the stun ends. Select None if you do not wish to use this option.

### Move Override（动作覆盖）

**Default Move:** The default move file the opponent will play for this override. Every character should have this move file in their move set, if applicable. If you have characters that cannot use the default, you can set Character Specific Moves below.

**Character Specific Move:** If the default move is not available for certain characters (i.e. if the default is a Legacy animation type and a character is Mecanim type), then you can set character specific moves here. Click the **New Character Specific Move** button to add a character specific move.

* **Character Name:** Character's name for this override. Must match the Name in [Character Editor](http://www.ufe3d.com/doku.php/character:start).
* **Move:** The move file for this character. The character must also have this move file in their move set.

### Orientation（适应）

Similar to regular moves, you can set orientation and mirror settings for the move here.

**Mirror Animation (Left):** When character is on Left side, mirror the animation.

**Rotate Character (Left):** When character is on Left side, rotate the character.

NOTE: Setting both of the above effectively performs the [Auto Mirror](http://www.ufe3d.com/doku.php/global:rotation) but without requiring the character to be on the other side.

**Mirror Animation (Right):** When character is on Right side, mirror the animation.

**Rotate Character (Right):** When character is on Right side, rotate the character.

NOTE: Setting both of the above effectively performs the [Auto Mirror](http://www.ufe3d.com/doku.php/global:rotation) but without requiring the character to be on the other side.

### Preview

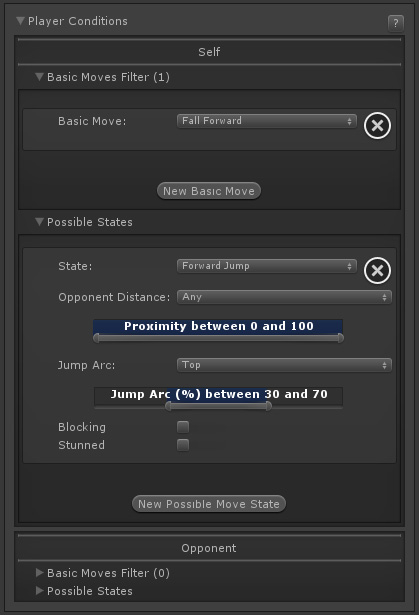
Click the **Preview** button to preview the override move.

* **Animation Frames:** Drag the slider to preview the animation.
* **Smooth Preview:** Toggle for smooth preview instead of by frame.
* **Reset Scene View:** Click this button to reset the Scene view. This helps with positioning.
* **Close Preview:** Closes the preview.

NOTE: **Move To Position** and **Orientation** is also previewed here to help with alignment.

Click the **New Override** to add a new override move.

## Player Conditions（执行动作的条件）



Select which conditions player must be in order for this move to be executable. （执行该动作需要先按顺序先执行某些条件）

**Self / Opponent**

* **Basic Move Filter:** This move is only playable if the character/opponent is executing one of the listed basic moves. To ignore this filter, leave this list with 0 elements.
* **Possible States:**
  + **State:** Select whether the character must be standing (idle, moving back, moving forward), crouching, jumping straight, jumping forward, jumping back or down.
  + **Opponent Distance:** On a proximity range, how far is the other character for this move to be allowed. 0 being very close and 100 very far.
  + **Jump Arc:** (Jump states only) Where in the jump should the character be to be able to play this move, being 0 taking off, 50 top of the jump, 100 at landing.
  + **Idle, Moving Forward, Moving Back:** (toggles, Stand state only) If standing, when can this move be executed. Example: Forward + Button moves can be created by having only Moving Forward toggled.
  + **Blocking:** Can this move be executed while the player is blocking?
  + **Stunned:** Can this move be executed while the player is stunned?

NOTE: With the new Player Conditions system introduced in v1.5, your previous moves (from v1.0.x) will need to run a small automatic update. To do so, simply select each move from your move list. Once the Move Editor loads it, it will automatically update them to the new system.

# 开发日志

## 动画在切换后就不动了，Animation Speed 没设置

## 跑动作实现

Input manager中取消勾选Snap，当正负两个按键同时按下时Input.GetAxisRaw还是会归零。

要想类似DNF一样移动控制，左跑动时同时按下右方向键会往右跑的效果。

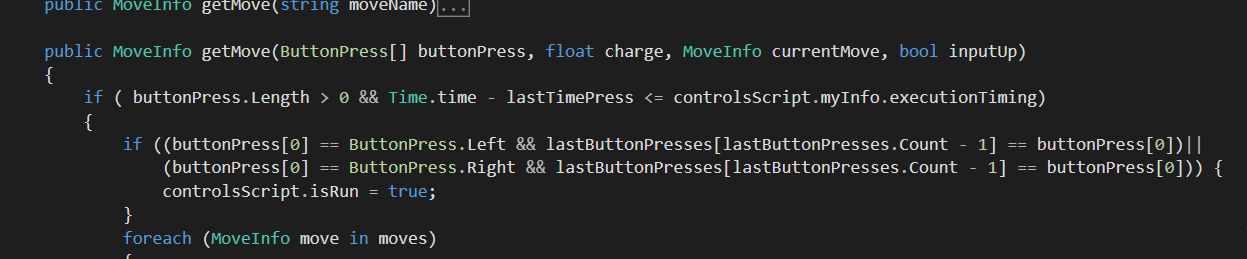
痛点：

* 1. 按两下左或右键执行跑动作
  2. 跑动时，同时按下其他方向键会向其他方向移动，送开任意其他方向键，还是跑动动作。

实现方式：

1.Vector2 moveDirection保存现在的移动方向；

2. MoveSetScript 判断设置isRun标识



4.当某帧没有方向键按下，isRun = false；

5.在循环每个方向键按下情况时，单独记录按下时间，在循环完后根据按下时间判断移动的方向和玩家朝向，这样避免循环方向键顺序问题而覆盖前面也按下的情况；

6.上下左右按键分开，避免同时按下标识正负键时归零的情况；

7. PhysicsScript添加力，在myPhysicsScript中判断力的大小执行walk动画或run动画

float force = isRun ? myInfo.physics.runSpeed : myInfo.physics.walkSpeed;

## walk、run、jump人物移动位置更新

效果：

Idle时jump，原地垂直跳跃；

walk时jump，向walk方向跳跃，walk方向移动的距离等于walkDistance + jumpDistance；

run时jump，向run方向跳跃，run方向移动的距离等于runDistance + jumpDistance；

jump后松开方向键，停止方向移动；

jump后按下任意方向键，会向方向键移动，延续起跳时walk或run的速度；

实现方式：

* 1. walkDistance和runDistance在FixedUPdate中由walkSpeed和runSpeed乘以Time.deltaTime得到。
  2. jumpDistance在FixedUpdate中由faceDirection \*jumpForwardSpeed\* Time.deltaTime得到。
  3. 当执行jump动作时只更新垂直向上的力，方向速度仍然由walk或run控制。
  4. faceDirection控制跳跃的方向，0为原地跳跃。
  5. 执行jump后会更新y和verticalTotalForce，PhysicsScript.isGrounded判断是否在地面上，
     + 地上执行idle、walk、run动画，
     + 离地后执行jumping动画，
     + 着地（isGrounded && sverticalTotalForce !=0）执行landing动画
  6. 跳跃参数
     + float maxHeight = Mathf.Pow(appliedForce, 2) / (appliedGravity \* 2);// 跳跃的最大高度
     + maxHeight += transform.position.y;
     + airTime = Mathf.Sqrt(maxHeight \* 2 / appliedGravity);
     + verticalTotalForce = appliedGravity \* airTime;

参考：

上升最大高度： https://gss1.bdstatic.com/-vo3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D63/sign=8eabd29a0db30f24319aef00c995439d/0df3d7ca7bcb0a46073df2c26963f6246b60af2a.jpg（抛出点算起）

上升与下落过程具有对称性，如在同点速度等值反向等。△s=https://gss2.bdstatic.com/9fo3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D22/sign=70849a43d63f8794d7ff4f2cd31b7a3f/f603918fa0ec08fab3fc6a5c5bee3d6d54fbda49.jpg

## 在地上站立判断

PhysicsScript挂的transform点到ground Layer的垂直距离判断。

注意：

设置的距离大于一次更新跳跃的距离时会返回true。

人物模型的脚没正好在地面上时，可以调整人物模型transform的y。

PhysicsScript：

public bool isGrounded()

{

Vector3 p = transform.position + Vector3.up + new Vector3(0, 0f, 0);

if (Physics.RaycastAll(p, Vector3.down, 1.5f, groundMask).Length > 0)

{

if (transform.position.y != 0) transform.Translate(new Vector3(0, -transform.position.y, 0));

return true;

}

return false;

}

## 硬直时间

## 拉动对手的动作

## 打击时镜头拉近

## 场景晃动