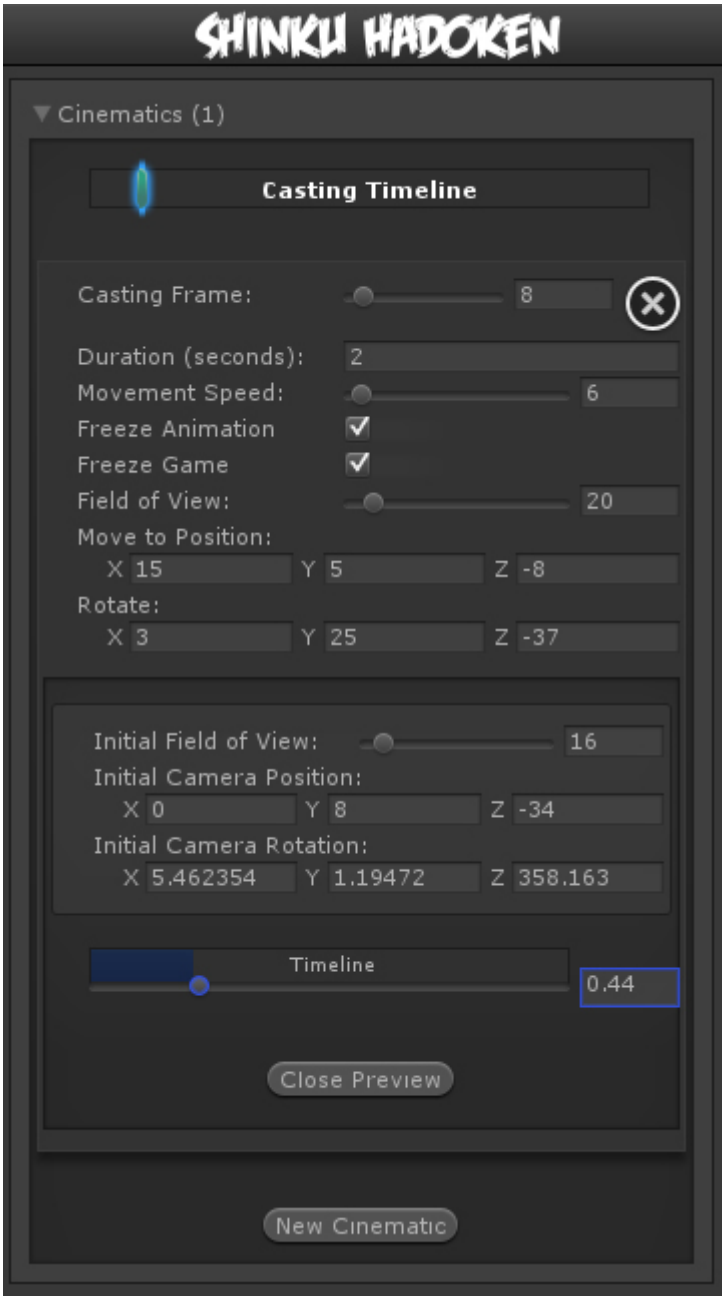
Cinematics

Cinematic Moves in a 2.5D fighting game was first introduced in 2008 with the Street Fighter 4 series. It refers to taking advantage of the 3D engine by moving the camera to different angles during a move for dramatic camera shot.

UFE currently offers this option as a pre-hit cinematic. Soon more options will be added to this feature including hit confirms and grappling.

Casting Frame: When during the move the cinematic should begin.



Duration (seconds): How long should the the camera control be on screen before returning to normal state.

Animation Speed: Slow down or stop the animation from playing during cinematics. Other move options will not trigger if the cast frame plays while in cinematic mode.

Freeze Game: Toggle rather or not you want the entire game to freeze during cinematic. Even though the game “freezes”, UFE is configured to have all animations still move in a very slow speed (.005 of regular speed) during cinematic to create a dramatic effect. This can be changed in `./Scripts/ControlsScript.cs` (Pro Source Only)

The options below refer to where and how you want your camera to end its move. A Key frame, if you will.

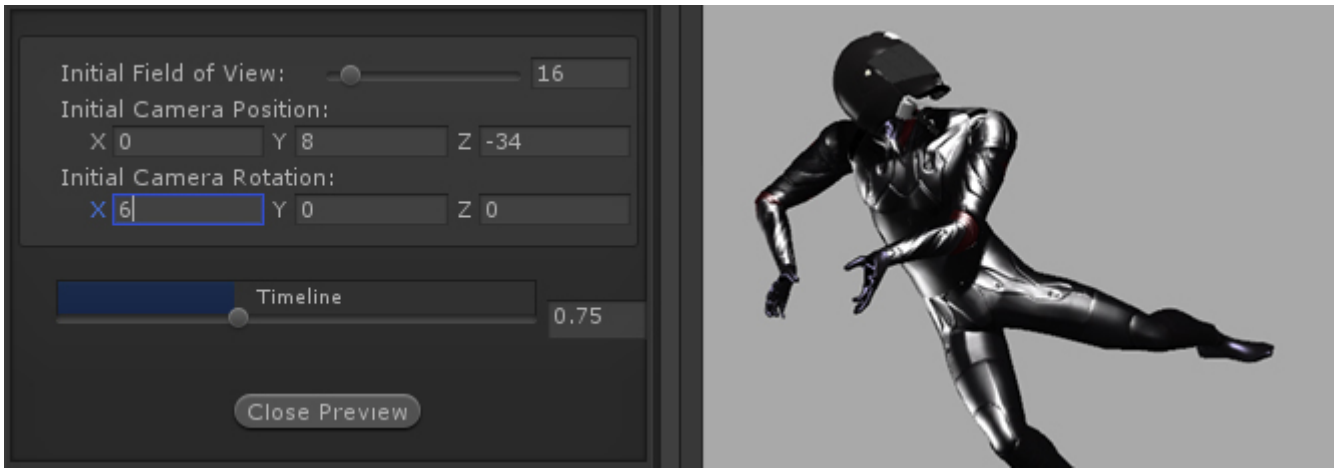
Field of View: The target field of view.

Move to Position: The target position (relative).

Rotate: The target rotation.

Camera Preview

Preview your cinematics before testing.



Initial Field of View: An emulated version of your camera's standard field of view. In this demo the initial field is set to 16.

Initial Camera Position: The initial camera position of your emulated camera.

Initial Camera Rotation: The initial camera rotation of your emulated camera.

Timeline: Use the slider to preview the camera. The timer is based on the duration set.

Snap Current Camera Info: Snaps the current camera transform into the *Move to Position* and *Rotate* values. The position converted is related to the character through a transform point.

Notes:

- If you want the camera to cut or camera shots, change the speed to 100.
- Multiple camera shots are possible, but they are a little tricky since the camera works in seconds while the animation works in frames. If you are working at 60 FPS, just remember that 60 frames = 1 second.
- To have your camera freeze after performing its move, make sure you set the speed to a value in witch it will reach its destination with time to spare.

Code example:

```
void OnHit(HitBox strokeHitBox, MoveInfo move, CharacterInfo hitter){
    foreach(CameraMovement cameraMovement in move.cameraMovements){
        if (cameraMovement.casted) Debug.Log("Cinematic has been cast.");
    }
```

```
}
```

< Back to Move Editor