Reason for making the Tool?

The Purpose of this tool is to be able to get the play animations based on a specific frame of an animation/or get variables of an animation, like getting the current frame of an animation, or the name of the animation.

Without this tool, getting to the current frame of an animation can be very confusing and frustrating to work with. Getting accurate frames of an animation instead of a float value will make it a lot easier to call functions or have different functionalities be called at a specific frame without having to make an animation event.

This tool helps circumvent having to make an animation event for animation in the animation panel.

What are the Restraints and Limitations?

The Frame Aid Tool has a few restraints that you must keep in mind.

Low FPS - If your machine has low FPS, and you have functions called at a specific frame, then they may not be called or completely skipped.

Sync up the FixedTimeInterval & Sample Rate - The Frame Aide tool uses the sample rate of an animation to calculate how many frames the animation has dynamically; however, if your update mode on the animator is Fixed, this may lead to inconsistent frame count and issues. What you need to do is set the **sample rate** of the animation to the same interval as the **FixedTimeInterval**. You can do this by dividing 1 by whatever the samplerate of your animations is **(1 / 30) = 0.033**.

The outcome would be what you want to put as your **fixedTimeInterval (0.033).** This way, the FixedUpdate is called the same time as when the animator is rendered (if the update mode is fixed).

Use FixedUpdate for animator.UpdateMode -
I strongly suggest you change the updateMode of your animator to **FixedUpdate** because the normal updateMode is dependent on the update loop.

If any sudden frame drops occur, then this could affect how accurate the tool can be. It may work if your project is light and doesn't require much processing power.

However, for major projects, it may be safer to set the Update mode to Fixed so that it runs consistently.

This tool is focused on 2D animation.
This tool is mostly used for basic 2D animation. So, you will run into problems and issues with any animation that uses Animation Layers. I may add future additional changes to incorporate animation grouping, but as of now, it's for basic 2D animation.

Core Functions

Frame-Aide takes the animations on the attached animator and breaks them into frame data. That data is stored and then used to make accurate, frame-based function calls.

IEnumerator PlayTillFrame(string animationName, int startFrameIndex, int endFrameIndex**)**

```
case AnimState.Idle:
    StartCoroutine(frameAideTool.PlayTillFrame("MainChar", 0, 9));
    animPlaying = true;
    break;
```

PlayTillFrame is a function that will play an animation at a specific start and end. This can be useful if you want to play a specific part of an animation without having to play to skip through or without needing to make a separate animation state just for that specific part you want.

(I may modify it to ask if you want it to reset the animator, going back to the default layer state)

Void PlayAtFrame(string animName, int frameIndex**)**

PlayAtFrame will simply play an animation at a specific frame. This is useful if you want to skip through parts to play something that is important.

```
    frameAideTool.PlayAtFrame("PUNCH", 4);
}
```

Int GetCurrentFrame(Animator animator**)**
GetCurrentFrame will get the currentFrame of the animation when it takes in a valid animator in its parameter.

```
if (frameAideTool.GetCurrentFrame(animator) == 1)
{
    playerMoveScript.isPunch = true;
    playerMoveScript.isMoving = false;
}
```

GetCurrentClipName(Animator animator**)**
GetCurrentClipName will simply get the current name of the animation that is playing when called. This is an easier way to ask for an animation name without needing to call the stateinfo, then the animation group, clip, and lastly the name.

```
frameAideTool.GetCurrentClipName(animator);
```

Showcase Scene Descriptions?

There are 4 scenes that show the number of ways the tool can be used.

AnimEvent Trigger
The AnimEvent scene showcases how the tool can be used as an animation event, allowing certain functions to be called at a specific frame without going to the Unity panel and manually making an animation event.

ComboOnFrame
ComboOnFrame showcases how the tool can be used for a fighting game input system. When F is initially pressed and is pressed at a specific frame, it will go into a different animation, like a combo system.

FrameStop
FrameStop showcases how accurately the tool can stop an animation that is already playing. A random number between 0 and 20 is generated. When it reaches a random number, the animator will stop at that specific frame.

ManyInOne
Lastly, the ManyInOne showcases the use of the PlayTillFrame function, which can allow multiple animations to be in one single animation state, calling specific frames to be played based on the start and the end (IDLE 0 - 8, RUN 11 - 34)

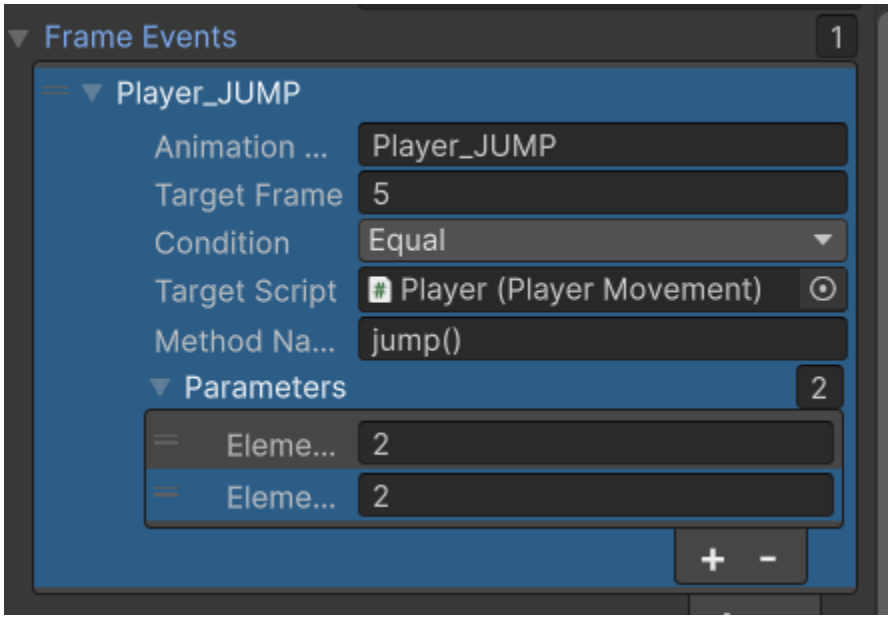
FrameTool Animation Event

Lastly, the tool also has a pseudo style of animation events that adds different condition types like less than, greater than, less than or equal, and greater than or equal.

This is for people who do not want to go into C# script files and write code to call functions.

If the function has parameters int, float, and string, they can be input into it. However, that is all the supported types for now. Fair warning, the pseudo animation event manager is used to call multiple functions at a specific frame, making it more readable for users.

(If it seems too odd or inconsistent, feel free to go back to using the animation event or making calls within a script file)



How to use the tool

The tool is used by attaching the script to the GameObject you want to get the frame information on.

It's important to note that this script needs to have a valid animator for it to work properly.

Where to contact for issues?

If there are any issues with the tool, feel free to email me at boswelldane37@gmail.com. I will soon get a proper GitHub or Discord to put in requests or issues.