Guide V 1.6

Table Of Contents

GAC (Great Animation Combos) Introduction

Thank you for purchasing the GAC (Great Animation Combos) System asset. This asset will help you get your animation attack combos up and running in no time for your game. The GAC system comes with multiple great features that go beyond just attack combos that will be explained later and there are more features planned in the near future to increase your development productivity.

GAC Step By Step Setup

So to jump in immediately and start using GAC, follow these steps:

a. Drag the GAC.cs onto your Player Game Object

b. ***As of Version 1.2***, GAC will set itself for ***Legacy*** or ***Mecanim*** 'Animation Controller Type' based on what type of Animation Component is on the Game Object (Animation or Animator). You can switch between each type depending on what you want to use later.

   NOTE: Keep in mind that switching between controller types resets all of GAC setups. If the ***Animation Component*** is attached, GAC will reference all those animations. If the ***Animator Component*** is attached, GAC will reference all the animation on the Animator Controller of your character, including all animations of every layer.

c. ***New in Version 1.6,*** GAC allows you to choose between using **3D** or **2D** 'Game Modes' depending on the type of game you want to build. Make sure to choose to proper Mode because certain modes disable/require specific components in Unity that correspond to 3D or 2D games.

d. Select the amount of 'Activators' the GAC System is able to use from the drop down. Activators are basically buttons that you want to use for your game like on a controller/keyboard etc. The more activators added, the more buttons can be used to create more complex combo setups. Max amount of activators available to use is 100 (will increase limit if necessary).

   NOTE: ***Version 1.3*** gives a warning when reducing the amount of activators already added just as a precaution to prevent deleting combos and activator inputs setup, which GAC will do if necessary attributes are removed.

e. Select the **'Animation Setup'** menu. To begin adding animation slots for use, click 'New Animation Slot' button. Subsequently, by adding at least 1 animation, each slot will have a "+" button to add more slots.

f. ***As of Version 1.2*** is the 'Play Mode' drop down to allow Normal or Cross-fading of animations for smooth blending.

g. You then need to set an animation to this slot for use. So select an animation from the dropdown menu then hit 'Apply' to set selected animation. If you change your mind you can hit 'Edit' to remove that animation from the slot.

   NOTE: GAC prevents the removal of animations from slots if animation is being used in a combo. Remove the animation from the combos first.

h. GAC requires at least 2 animations to be set to a slot so go ahead and add, then set another animation to a slot.

i. Now each combo needs to have a starter animation that initializes the combo, so go ahead and choose at least one of the animation slots to use to start a combo (you can use any and all animations set to a slot as starters if you want). Of course you can make that selection later on after you finished following the other steps below.

j. So linking a combo requires the player to successfully send an input event within an animation's frames. The 'Link Slider' allows you to set the frame range you want the player to be able to connect their animations in a combo. Set this up for the animation slots that you plan to use to link to other animation slots. ***As of Version 1.3*** you can set each animation slot to link into a 'Delayed Animation' that allows for even more complex combos specifically ones that are triggered when an input event is delayed. Eg. A, A…wait…A. To use this, turn on the trigger and set the 'Delay Frames' that counts down from 0.5 and the number you set in the slider is the range that is allowed in between for the input event to trigger the animation.

k. For optional use is the 'Move Slider'. Turn this on if you plan on moving the character during certain animation frames.

l. The 'Move Amount' option works in conjunction when the Move slider is turned on. Set this for the distance to move the character during animation. This option only allows movement on the Z axis, more options to come! Now we are almost finished setting up our animations for use, next is to begin setting up GACs hit detection; if not planning to use, let's go create combos!

m. The following four steps involve use of GACs basic custom hit detection, so if you have your own hit detection you can skip to ***Step Q***. For optional use is the Hit toggle. Turn this on if you plan on using GACs basic hit detection system then set the frame range on when to check for hits. If toggled off, ***Steps M through P*** will be grayed out.

n. Start with 'Affect Layers', choose what layers you want this animation slot to affect. Then make sure to add the GAC_TargetTracker.cs to that gameobject you want to affect, see **Page 26** that discuss the Target Tracker.

o. Set the max distance you want the animation slot to affect using the 'Affect Distance' option.

p. Set the max angle you want the animation slot to affect using the 'Affect Angle' option.

q. Use the 'Focus' Gizmo button to visually see the range your animation slot will affect in the Scene View. Customize the color for easier representation of all your animations.

r. Make sure you added activators if not, do so now. Select the **'Combos Setup'** menu. The Starter Animation dropdown will have all the animations you set to use as starter available. Add one to the combo system.

s. Now that a starter animation is added, ***as of Version 1.3*** click the 'Add Delay' button which adds a delay animation spot or click 'Add Animation' which adds an animation spot to begin customize your combo. Delayed animation spots are marked by the D on the left and colored Blue. Keep note that any animation under this starter animation group will ALWAYS start off with that starter animation of course.

t. In the animation spot you can select the activator and animation to use. Keep note that you cannot add another animation spot without first clicking the 'Set' button to place that animation on the sequence. You can go back and hit the 'Edit' button to change that slot if needed, GAC will keep a record of the animations that are in higher sequences.

u. After setting the spot for the first animation, continue to add animations slots to create your combo. You can add different combos under that Starter animation group.

NOTE: You cannot add a different animation for a separate combo on the same sequence (unless the sequence is a delayed animation as of ***Version 1.3***, but the rule still stands with 2 delayed animations having different animations on same sequence). This conflicts with what GAC will allow to play for that sequence. GAC will notify you if this happens. To rectify this, you can change the activator used; GAC also prevents you from going into play mode if not fixed.

v. Our second to last step is to check out our combo setups in a visually readable format by using the GACPAC window located in Tools>Great Animation Combos (GAC) > Preview Animation Combos (PAC). This is great for also testing out the combos to see how they play into each combo link. New in Version 1.3, GACPAC has an 'Activators' button that shows or hides the activators used to start the animation in that specific combo. Also each animation box now has an activator label (marked with Red number) on bottom left and an delayed animation label (marked with a Blue D) on the bottom right. This provides more helpful information for your creations.

w. Our final step includes scripting our input events. Create a script that controls the input for your buttons and attach to Game Object. You can use Unity's Input System or other Asset Packages Events of course. See below on Page 18

Scripting Input Setup for more information on this. ***As of the release of Version 1.1, you can now set Activators through the inspector just follow the next steps.***

x. Depending on what asset you want to use, GAC has specific asset packages for different assets, one of which includes DFGUI support in this version 1.1. The default Editor script asset is called GAC_Editor.cs. Now if you want to use DFGUI, you must remove the default GAC_Editor.cs script and open the GAC_DFGUI_Editor asset package from the Input Sources Folder, which will load the GAC_Editor_DFGUI.cs file.

y. Depending on what editor file used will modify the amount of input source selections available. Click on the **'Activators Setup'** menu, and at the Activators dropdown select from list of input sources including Keyboard, Mouse, and Button selections. If the DFGUI Editor was loaded, a selection for DFGUI source will be available. Click the 'Add' button to add an activator to the slot.

z. Depending on the input source selected will provide different options. Page 13 ***Activator Setup***; explains all the options. Click the 'Set' button to start using this activator in the combo system.

aa. Now you are done, so go create **GREAT COMBOS**!

GAC Inspector

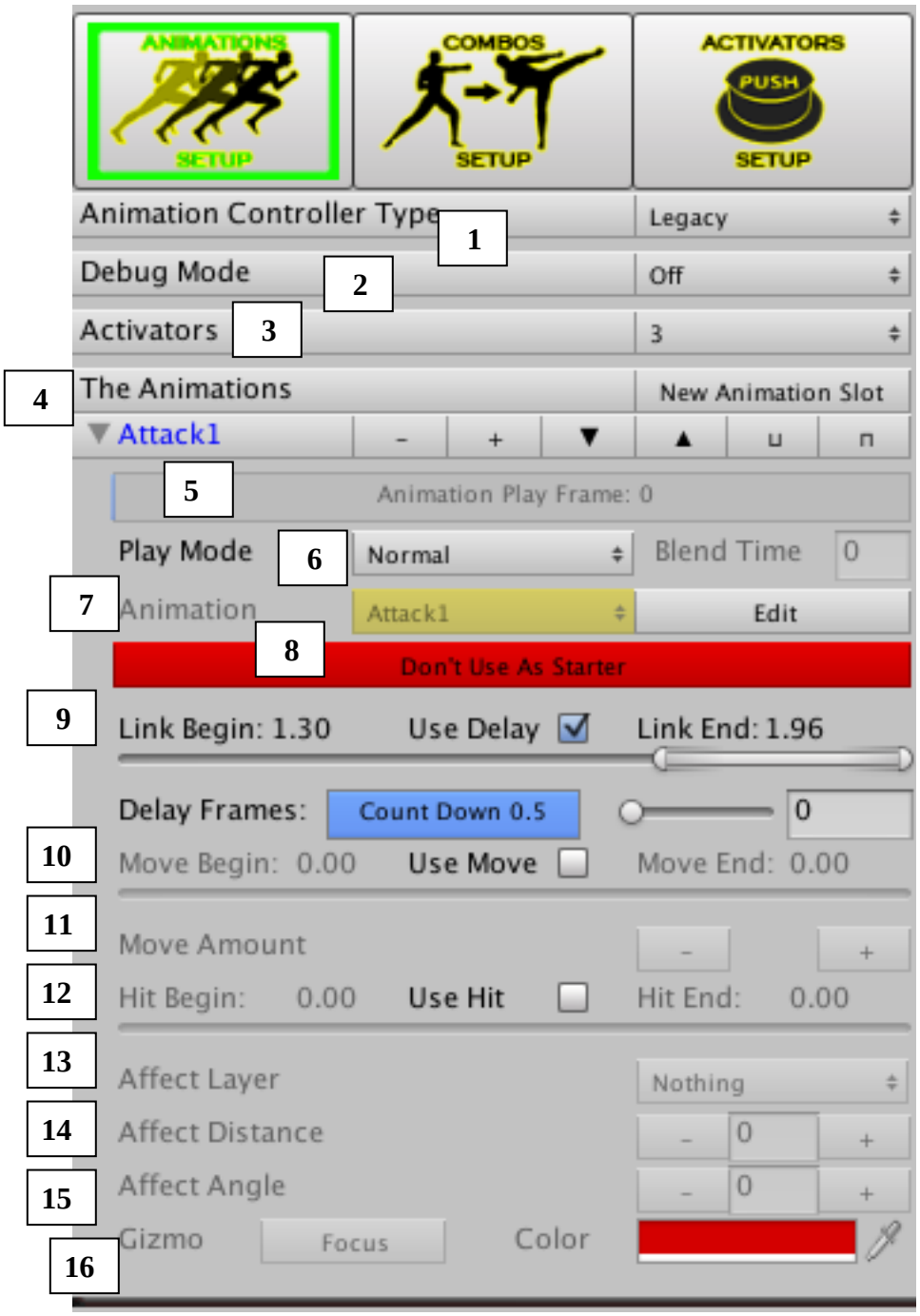
To begin using the GAC System, just like most things in Unity, it needs to be attached to a Game Object namely your main Player game object. Of course, the GAC System requires the game object to have animations to use to create the combos with.

TIP! When preparing animations for use in a combo, it's best to build animations around predetermined start and end spots to make everything look smooth. E.g. if one animation ends with the weapon above the character's head, it would be best to use animations that begin with the same weapon above the character's head; the transition will look natural instead of jerky.

When GAC script is attached to a game object, there will be 3 menus available. These are the Animation Setup, Combo Setup and Activator Setup menus.

Animations Setup

The Animation Setup menu provides what is needed to get your attack animations prepared for use in the GAC system.



1. Animation Controller Type

As of Version 1.2, Mecanim animations are supported. Use this to setup what the type of animation component you will be using. Select between using ‘Legacy’ or ‘Mecanim’ animations.

2. Debug Mode

This is used to get information feedback from the GAC System. It provides 2 types of information:
Animation Info: logs when an animation has been linked in a combo.
Hit Info: logs when a hit is detected from an animation.

3. Activators

Activators are like input methods (e.g. Key Press -A, Touch- slide, Mouse -drag). They can be used for any input method you setup for your games. Activators are needed to start creating combos.

4. Animation Group

Lists the animations’ name and holds the animations settings. Can be moved around for organization.

5. Animation Play Frame

Shows a display of the current playing frame of the animation.

6. Play Mode and Blend Time

Use this to make your animations play ‘Normal’ or blend them using the ‘Cross-fade’. Blend Time will be accessible when Cross-fade is selected. Set the time animations take to blend into them.

7. Animation

Set this animation for use in combos from the current animations attached to this game object. When using Mecanim animations, GAC will retrieve all animation states from every layer being used. This means each animation state name will have the layer index number it is currently on, for e.g. “Attack1 ‘L-2’” means the animation state is using layer index 2 (which is the third layer after the base layer).

8. Use/Don’t Use As Starter

Used to for setting this animation as a starter animation for combos.

9. Link Frame Ranges

Use the min/max sliders to decide what frames of the specific animation will be able to link into another animation. ***As of Version 1.3*** you can toggle on/off the use of this animation to link into a delayed animation. Set the ***Delay Frames*** that will be able to be triggered by input between what is set with the slider and 0.5f

10. Move Frame Ranges

Use the min/max sliders to decide what frames of the specific animation will be able to move the character. This can be toggled on/off

11. Move Amount

Used to decide at what amount to move the character forward/backwards

12. Hit Frame Ranges

Use the min/max sliders to decide what frames of the specific animation will be able to track hit detection from this animation. This can be toggled on/off. If toggled off the Affect Layer, Affect Distance, Affect Angle and Gizmo will be disabled.

13. Affect Layer

This allows you to add any amount of layers that his animation can affect. Each animation can have different layers to affect, which provides more freedom for customization based on what animations play.

14. Affect Distance

Provide the distance range in which the animation will be able to affect

15. Affect Angle

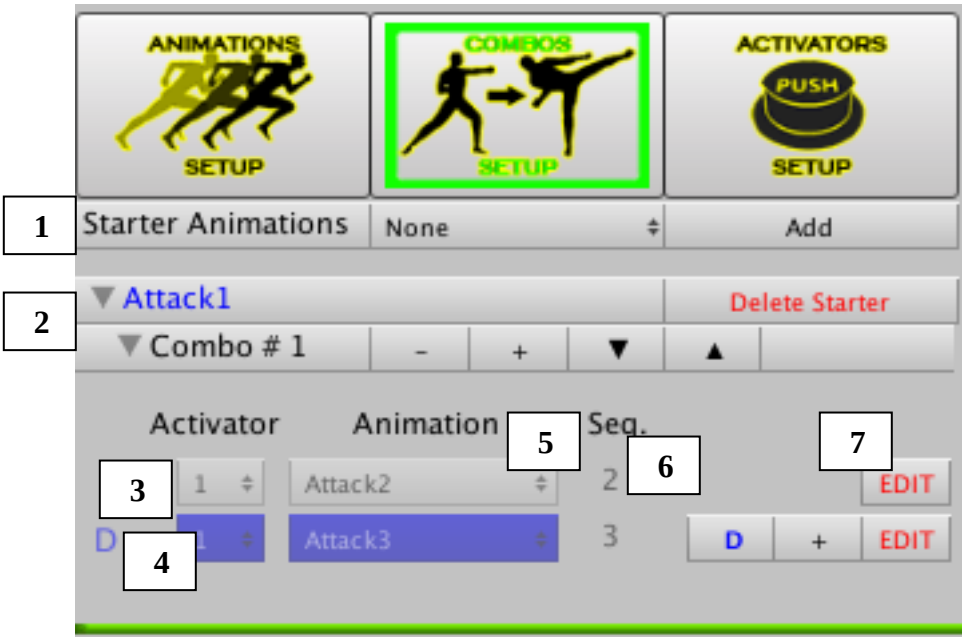
Prove the angle range in which the animation will be able to affect

16. Gizmo and Color

Use this as a visual feedback on the distance and angle that the animation will be able to affect. Change the color on each animation gizmo for customization and readability purposes.

Combos Setup

The Combo Setup menu is where you get to create the combos themselves from all the animations you have setup in the Animation Setup menu.



1. Starter Animations

This gives a drop down list of the animations that are set to be used as starters. Any starter animation that has not been added to the combos will still be available here; otherwise if they were added to the system already, they won't show.

2. Starter Animation Name

This is the starter animations combo settings. All combos that begin with this starter animation will be in its drop down.

3. Activator

This provides a dropdown list of all the activators available to be used with this starter animation

4. Delayed Animation Spot

As of Version 1.3 is the delayed animations feature to add animations that are delayed in the combo. These spots are colored in blue for easy distinction

5. Animation

This provides a drop down list of all the animations that were set in the Animation Setup menu excluding the starter animation itself

6. Sequence

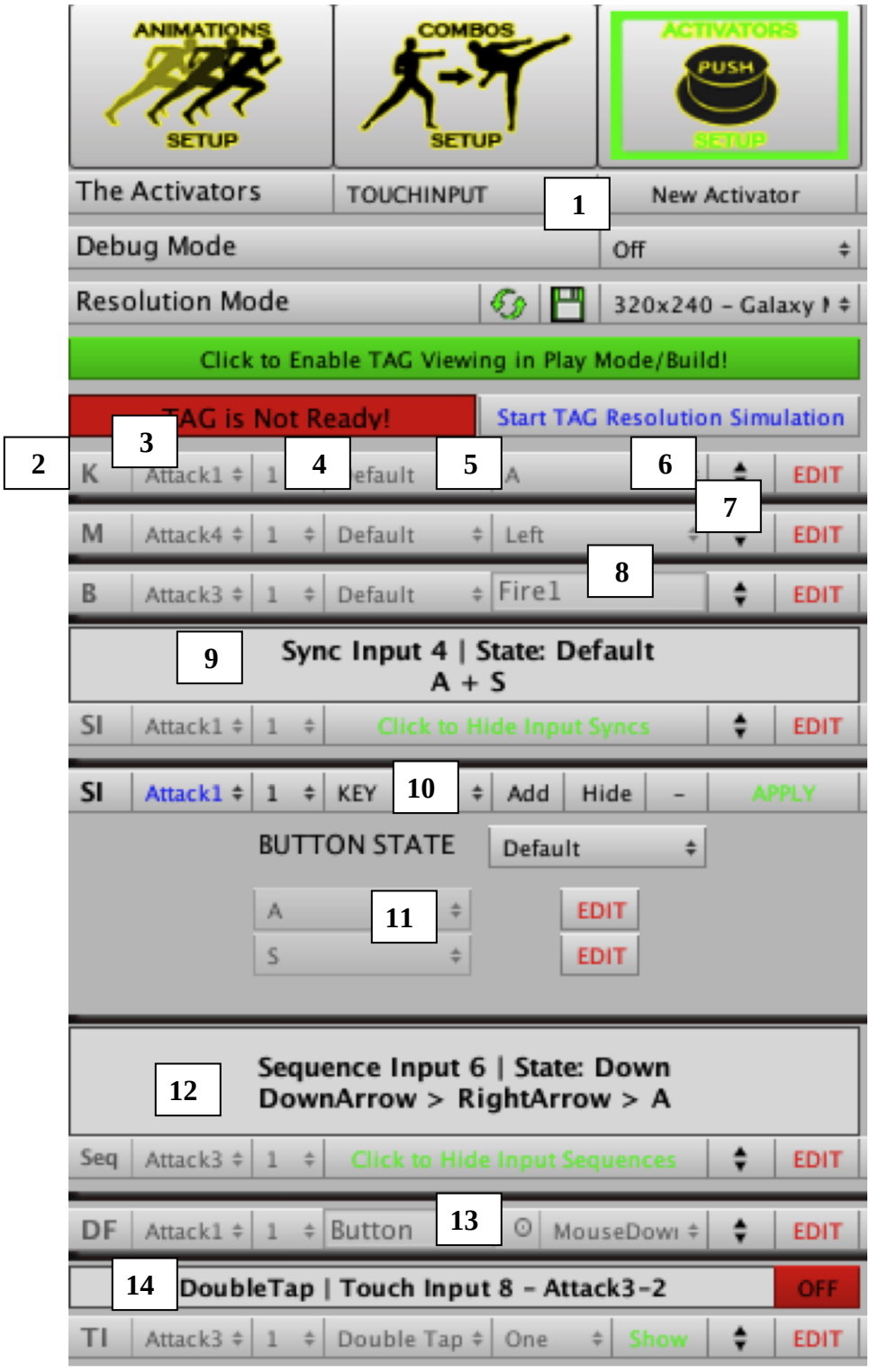
The sequence that each animation in the combo will be called on

7. Set/Edit

These buttons allow you to set what animation you want to use for that combo in that sequence with the specific activators and make changes to this setup whenever needed

Activators Setup

As of Version 1.1, the Activator Setup menu is where you can create all the activators that are going to be used in the system. This of course means you are not required to write a single line of code to set up your activator events unless a specific event is not provided.



1. Input Source

This is used to set the type of input you want to use. The different types available are Keyboard, Mouse, Button (Unity Input Manager), Synchro-Input (Combinations) and DFGUI support.

2. Input Type

This is the slot for each input type. The following initials distinguish the input types: M – Mouse, K – Keyboard, B – Button, SI – Synchro-Input, Seq – Sequence Input, DF – Daikon Forge GUI, DSy – DFGUI Synchro, DSq – DFGUI Sequence

3. Starter Name

Select the starter name to use for this activator

4. Activator Number

Register this activator number to this activator slot to use

5. Event State

Unity provides different states for its events. This means for e.g. Keyboard Event: Default – GetKey(), Down – GetKeyDown(), Up – GetKeyUp()

6. The Events

The dropdown provides the list of events available for use based on the input source selected

7. Move Activator Dropdown

New in Version 1.6 is the Move drop down selection for moving the activator up or down to reorganize them.

8. Button Text Field

If the Button input source was chosen for this activator, the text field is used for the name of the button that needs to be accessed from Unity's Input Manager

9. Synchro-Activator Input Source

If the Synchro input source was chosen for this activator, this allows you to select what input source (Keyboard, Mouse, Unity Input Manager Buttons) to add to the synchronizing pool to combine

10. Input Sources To Combine

These are the input source values to combine (a mixture of all inputs provided can be used) in the synchro-activator

11. Synchro-Activator Input Result

This is the result of all the input source values combined for the Synchro-activator

12. Sequence-Activator Input Result

This is the result of all the input source values combined for the Sequence-activator

13. DFGUI Object Field (DFGUI-Synchro and Sequence Activators)

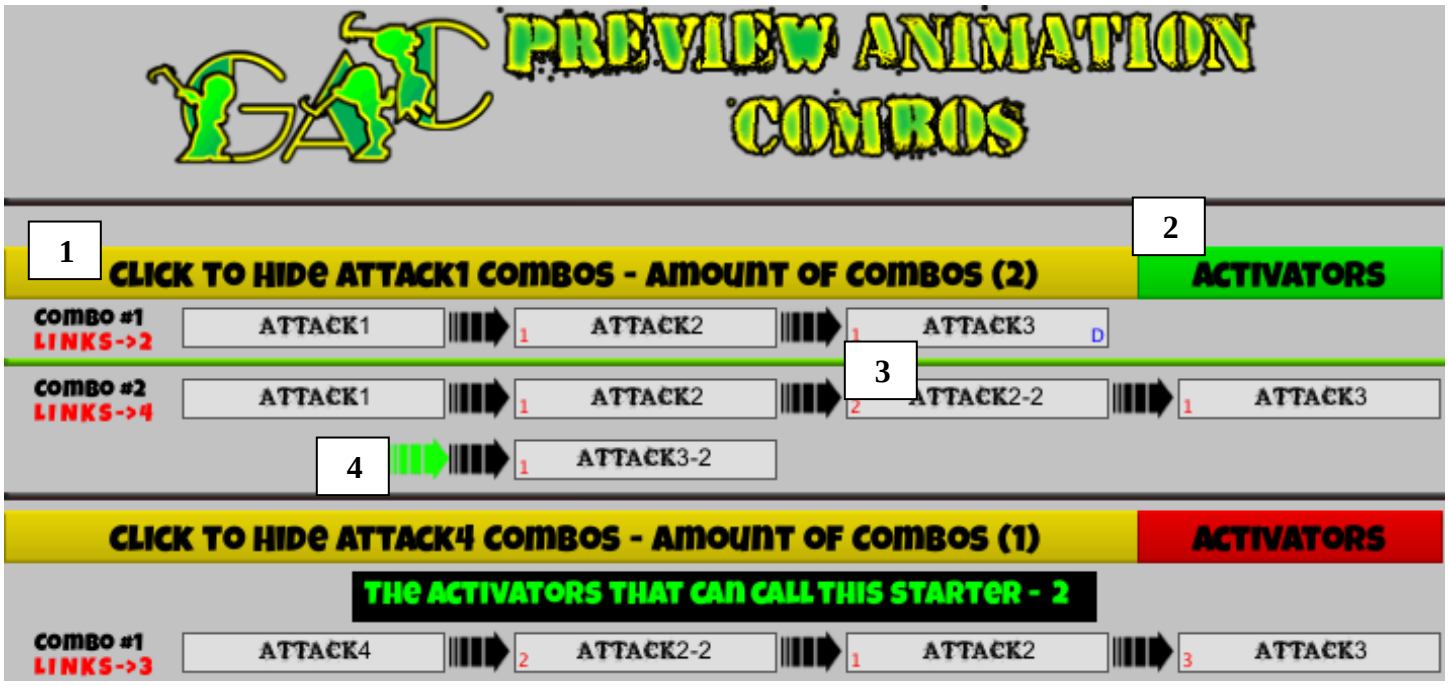
If the DFGUI input source was chosen for this activator, the object field is used for the game object that has DFGUI setup for it to access its events. ***As of Version 1.5***, you can use DFGUI to set Synchro and Sequence activators. Just select it from the Activator Source dropdown – SYNCHRO_DFINPUT and SEQUENCE_DFINPUT

14. Touch Activators

New in ***Version 1.6*** is the Touch Activators that allow input from gestures with touch screen or mouse. **See page 18**.

GAC-PAC (Preview Animation Combos)

The GAC-PAC window is the intuitive information tool that gives visual feedback of all your combos that you have setup in the **Combo Setup** menu.



1. Hide/Show Button

This is used to hide or show the combos of that specific starter animation

2. Activators Button

This is used to hide or show the activators that are used to call this starter animation. Multiple activators can be listed of multiple activators are setup to call this from the Activator Setup menu.

3. Animation Box

This shows the animation that is in the combo in sequence. Bottom left red number is the activator used to call this in combo and bottom right blue D signifies that the animation is delayed in the combo.

4. Link Arrows

These arrows show the progression of the combo. Black arrows are normal and green arrows tell that the combo is being continued on a different line when no more space is available in the window to list it.

Synchro And Sequence Step Guide

As of version 1.4 and 1.5, you are able to setup advanced animation activators in the form of Synchro and Sequence Activators. Synchro-Activators basically allows the player to trigger inputs simultaneously to play an animation. The Sequence-Activators are used to link inputs in a sequence like triggering Ryu's Hadouken which is done by inputting Forward>Down>Forward Punch on the controller. The following steps of how to setup both Synchro and Sequence activators work the same:

1. After adding the Activator source (Synchro or Sequence), select from the animation list to call

2. Set the activator number to use

3. Choose a source (Key, Buttons, Mouse) to add a slot to use

4. Based on the source, select the type of event to trigger with (eg. Key Source – A key)

5. Set each slot (At least 2 slots set is required, max is 5)

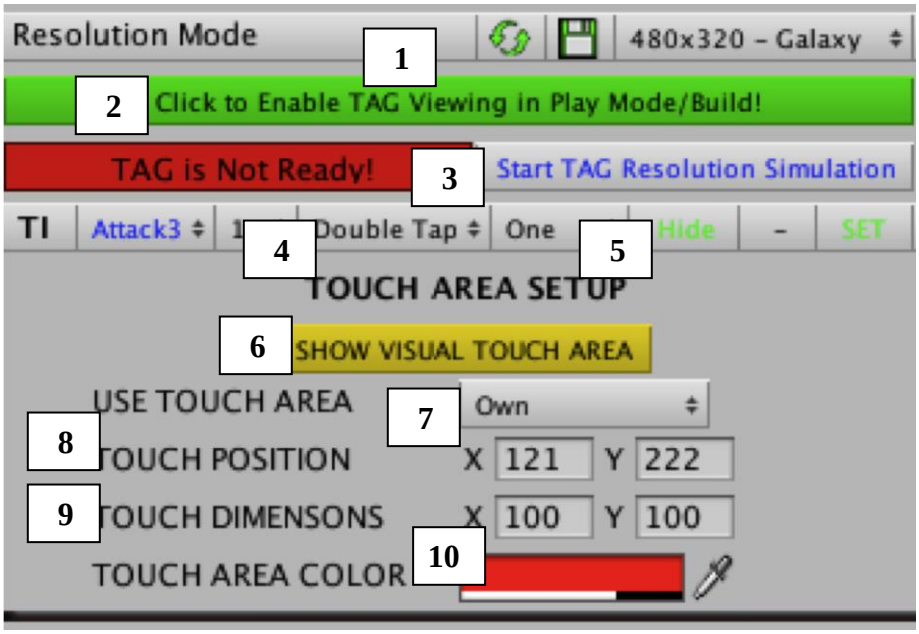
6. Then when finished, hit 'Apply' to see the resulting setup

7. When working with Sequence Activators, you can use Synchro Activators that were 'Set' to use for this character. You will see 'Sync' in the source dropdown. This allows you to maximize the usage of both together. E.g. Forward>Down>A+S is a combination of Sequence and Synchro activators. **NOTE:** Removing a Synchro Activator that was set will remove it from the Sequence Activator if it was being used

8. Check out upcoming video tutorials on the JrdevArts Page on Youtube: GAC Tutorial Playlist for more help

Touch Activators Settings

Version 1.6 brings the all-new Touch Activators. This is a great new feature that allows the use of gestures from touch devices and mouse inputs. Below is a Diagram of what it looks like and will be explained in detail. Touch activators provides support for Standalone, iOS and Android platforms. More platforms coming soon!



1. Resolution Mode

This drop down becomes active when there is at least one Touch Activator added. This provides all the most used Resolutions for multiple devices from specific platforms currently supported (i.e. PC/MAC, iOS, Android). Next to the drop down are the following icons:
Refresh – Clears all the touch activators settings saved on this resolution
Save – Saves all the touch activators settings for this resolution

2. Enable TAG Viewing

This is used to enable/disable the viewing of Touch Area's when the game is in play mode or built to the device. Used for debugging.

3. TAG Info and Resolution Simulation Button

The Tag info bar provides feedback to when the TAG window is not ready for use. The Resolution Simulation Button next to it is used to open the TAG Window

4. Touch Gestures

These are the gestures that are available to use for the activators

5. Finger Amounts

These are the amount of fingers to set to correspond with the touch gesture

6. Show Visual Touch Area

This turns on the Touch Area to be able to see it in the TAG Window and Game View. The TAG window won't show anything without at least one Touch Area being turned on.

7. Use Touch Area

This provides a list of all the Touch Areas that are currently **'Set'** for GAC. You can you either use the Touch Area's own settings by leaving the default selection to 'Own' or use another Touch Area's settings from the drop down. This is perfect for when you want to have multiple gestures using the same area of the screen.

8. Touch Position

This is the current position of the touch area and can use in conjunction with the TAG Window**.** **See page 20**.

9. Touch Dimensions

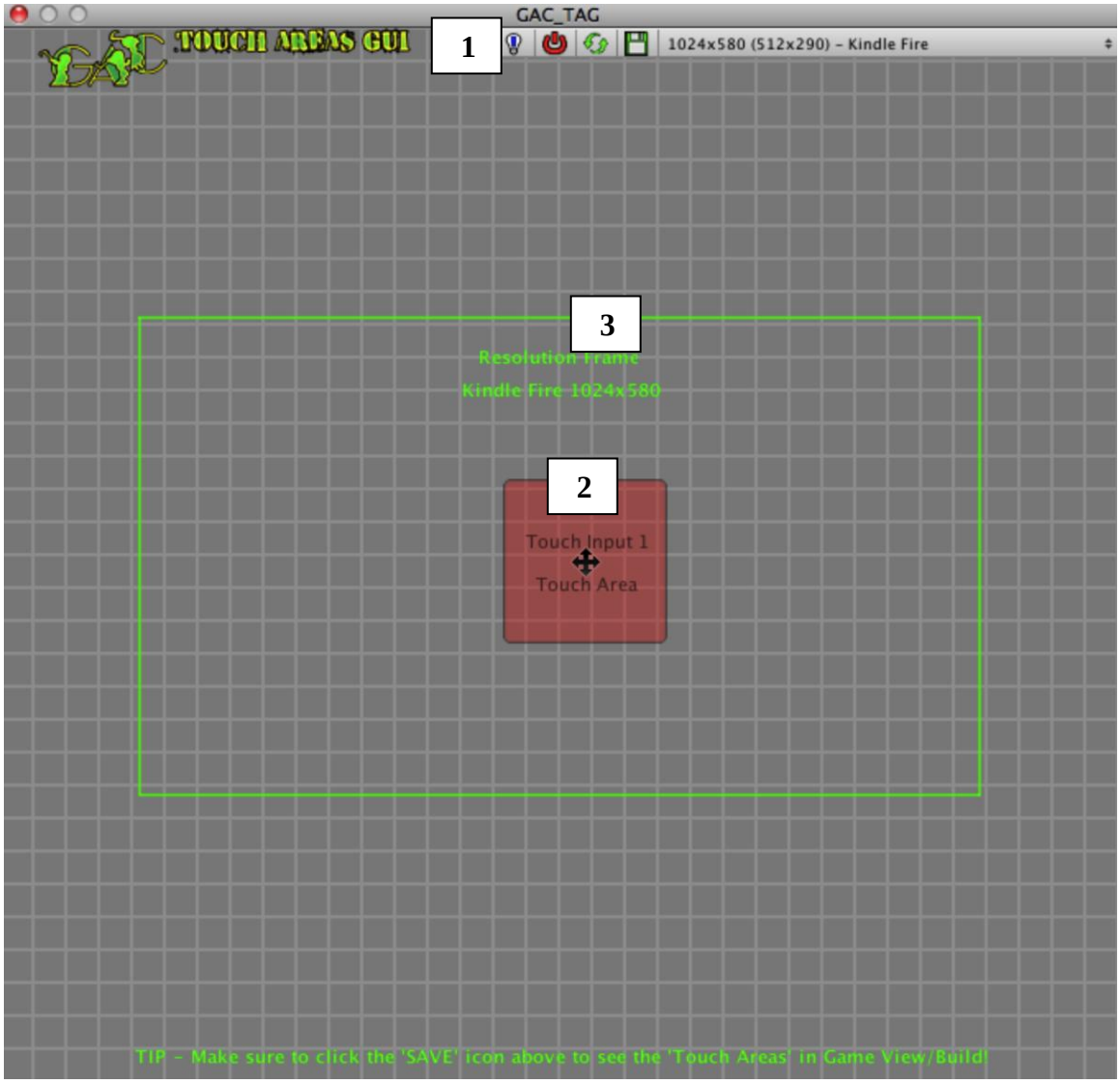
This is the current dimensions of the touch area and can use in conjunction with the TAG Window. **See page 20**.

10. Touch Area Color

You can set a custom color for each touch area

Touch Activators GUI (TAG)

With the new Touch Activators of ***Version 1.6*** comes the Touch Activators GUI (TAG) window used to setup Touch Areas visually. Using TAG you will be able to get a visual simulation of the resolution mode you selected from the dropdown. Most of the resolutions are scaled of course to fit the window but they represent the resolution perfectly.



1. Toolbar Icons and Resolution Mode

The TAG window provides toolbar icons for quick use and also provides the same Resolution Mode dropdown as the GAC Inspector. These include:

Tips - This provides a simple tip at the bottom of the window, clicking changes the tip shown

Refresh - Clears all the touch activators settings saved on this resolution
Save - Saves all the touch activators settings for this resolution

2. The Touch Area

This is the visual representation of the Touch Area's position, dimensions and color. The Move and Scale icons can modify the Touch Area. Move icon is in the middle of Touch Area and scale icon shows up for the edge of a Touch Area. The move button is used to, of course, move the Touch Area around on the TAG canvas. To scale the Touch Area, just place the mouse cursor at any edge of the Touch Area you want to scale and naturally drag to adjust it's dimensions.

3. Resolution Frame

This is the visual representation of the Resolution selected from the dropdown. Whatever is inside the frame is what is naturally seen and used for the position and dimensions of each Touch Activator Area.

Touch Activators Setup Guide

Setting up touch activators require multiple steps to get going. The following will walk through the steps to setting Touch Activators.

1. Choose Touch Input and Add it from the **'Input Source'** dropdown

2. With at least one Touch Activator added, The TAG info section will show at the top of GAC menu. Click the "Start TAG Resolution Simulation" button. This will open the TAG window.

3. With the TAG window opened, the canvas will be blank. A TIP at the bottom of window will tell you to click on the '**Show Visual Touch Area**' button of one of the Touch Activators to get started.

4. Now a Frame for the Resolution will show up based on the Resolution selected from the dropdown menu; choose the Resolution you would like to setup. You will also see the Touch Area for the activator; signified by its name.

5. Place and scale the Touch Area for the activator until it fits your needs using the move and scale options explained on **Page 21** for TAG window.

6. When you are satisfied with the setup of Touch Area(s), you need to select what Touch Area to use from the dropdown in the GAC Inspector. Using 'Own' allows the Touch Area to use its settings; otherwise, choosing from another Touch Activator will use that activator's settings. Now **'Set'** the Touch Area and then click the Save icon to record your settings for use.

   NOTE: A Touch Activator needs to be 'SET' before you can record the Resolution Settings. GAC will log a warning message if not set.

   NOTE: Always save any changes you want immediately! If you change to a different resolution that has saved settings before saving on current resolution, all savings will be lost.

   NOTE: If a Touch Area uses another Touch Areas Settings, then **'Edit'** was clicked on for that other activator, GAC will let you know that the Touch Areas using the settings will default back to it's own if you continue.

7. Now open the Game View window to view your setup. If you chose a resolution that is not a **Preset of Game View** resolution, your Touch Area will not be visible or position will be off. TAG requires the matching Game View Preset to be selected to show a correct representation for the Resolution.

8. So what you need to do is create a **Custom Preset Resolution** in the Game View dropdown menu. To do this click the "+" icon at the bottom of dropdown menu. A mini setting box will open.

9. In the Label box, give it the name of the resolution from the Resolution Mode e.g. 1024x580 (512x290) selected from TAG window dropdown, so name of Label should be first resolution i.e. 1024x580.

10. Make sure Fixed Resolution is selected from the drop down. Now you need to type the second resolution that was in the brackets i.e. 512x290 in the corresponding Width and Height boxes. As you can see, this resolution in scaled in half as a representation for the Editor, but GAC will scale it properly on the device. Repeat the steps as necessary of course to view other Resolution Settings not in the Game View presets.

    TIP! You can also add the name of the resolution provided by the TAG in the first Label section for better readability. i.e. "1024x580 – Kindle Fire".

    NOTE: If you did not save your Resolution Settings before, Touch Areas **WILL NOT** show up in the Game View even after the Preset Resolution is created and selected. **MAKE SURE TO SAVE!**

11. With the Touch Areas setup and saved on the specific resolution, you are freely able to Move around and scale the Touch Area and see it get updated in the Game View. If the Game View does not mirror what is being done in the TAG window, click the 'Save" icon again the reinitiate. You can close the TAG Window when finished with the 'Power' icon up top.

    NOTE: The 'Power' icon needs to be used for closing the TAG window or the window will stay opened if trying to close with default window close 'X' button.

12. The final thing of note is the **'Enable TAG Viewing'** button at the top of GAC inspector. Make sure this is enabled (it is red when it is enabled/green if not) if you want to debug and see the Touch Area GUIs in Play mode of editor or on the Device.

Using Other Animations

Using ***Legacy*** animations, when you need to use animations such as idle and running, all you need to do is regularly play them in the Update function of your script.
NOTE: It is best to put animations on different layers based on what priority you want them to take. GAC will take care of the rest. Using ***Mecanim*** animations, just set these animations how you normally would. For e.g. setting the Idle animation to be the default state, and use a parameter for changing to the run state.
NOTE: Do not attach transitions from Other animations to your animations being used in GAC combos, GAC will handle transitions.

Playmaker Setup

As of Version 1.4, Playmaker Actions are provided to those that choose to use FSMs to work with GAC. Check the JrDevArts video page for an upcoming tutorial of using GAC with Playmaker. See the Conclusion on **Page 28** of this guide for more information.

Scripting Input Setup

After setting up all the combos you want for your game, you will need a way to trigger these animations of course. The GAC System comes with a call method that you can add for any of Unity's supported input methods (Keyboard, Mouse, Touch, Controller buttons etc).

The call method is: JrDevAssets.GAC.PlayTheAnimation(starter animation name, activator number);

For example: Using the Keyboard Input A button

```
if (Input.GetKeyUp (KeyCode.A)){
        JrDevAssets.GAC.PlayTheAnimation("Attack1", 1);
}
```

NOTE: Read **Page 26** if you want to support multiple characters for GAC!
NOTE: The called animation must be setup as a starter animation to begin any combos.

The A button from the keyboard will call the starter animation "Attack1" and any successive push of this button will be read as using Activator 1; so any animation in the combo that has following sequences that are set to use activator 1 will be linked in the combo.

NOTE: ***GAC has namespace support***; just add "using JrDevAssets" to the top of your script so all you would need to do is type GAC.PlayTheAnimation(). You can do it like this:

```
using UnityEngine;
using System;
using JrDevAssets,
```

Specific Object Animation (For Multiplayer)

As of version 1.4, GAC now allows you to specify what gameobject to call the GAC animations on; no more Singletons! This opens up the gate to create multiplayer games. So instead of using the above script on the previous **Page 24,** you can use:

For example: Using the Keyboard Input A button

```
if (Input.GetKeyUp (KeyCode.A)){
        JrDevAssets.GAC.PlayTheAnimation(player1, "Attack1", 1);
}
```

The difference there is the 'player1' variable reference for that specific gameobject.

Target Tracker

New in version 1.2.1, GAC_AttackHitEvents.cs has been replaced with the new more optimized way using the GAC_TargetTracker.cs.
NOTE: ***Remove the old GAC_AttackHitEvents.cs file if still in project***. As with previous script, it needs to be attached to gameobjects that are supposed to be affected by the player character's attack animations.
NOTE: ***Hit detection will not work without this script!***
Then in your own script, you can check if the GameObject has been hit and respond to that hit:

```
if (JrDevAssets.GAC.TargetHit(GameObject theObject)){
      //Damage the gameobject
      objectHealth = objectHealth – 1;
}
```

API Reference

As of version 1.3, access to some more of GACs API code has been added. More API access is planned for the future. ***New in version 1.4***, most of the API has been updated to require a specific gameobject; this change is made to work better with the new multiplayer update.

void PlayTheAnimation(string StarterName, int Activator)

Description
Play the animation in the combo

```
if (Input.GetKeyUp (KeyCode.A)){
        JrDevAssets.GAC.PlayTheAnimation("Attack1", 1);
}
```

---

void PlayTheAnimation(GameObject theObject, string StarterName, int Activator)

Description
Play the animation in the combo for a specific gameobject

```
if (Input.GetKeyUp (KeyCode.A)){
        JrDevAssets.GAC.PlayTheAnimation(player1, "Attack1", 1);
}
```

---

bool TargetHit(GameObject target)

Description
Check if the GameObject that the player is attacking has been hit.

```
if (JrDevAssets.GAC.TargetHit(GameObject theObject)){
        //Damage the gameobject
        objectHealth = objectHealth – 1;
}
```

---

void AddTarget(GameObject theObject)

Description
Add a Gameobject to be tracked by the target tracker for hit detections

```
JrDevAssets.GAC.AddTarget(enemy1);
```

---

void RemoveTarget(GameObject theObject)

Description
Remove a target from hit detection

```
JrDevAssets.GAC.RemoveTarget(enemy1);
```

---

bool IsPlaying(GameObject target, string animName)

Description
Check if a specific GAC animation is playing

```
if (JrDevAssets.GAC.IsPlaying(player1, "Attack1")){
        Debug.Log("Atttack1 is playing");
}
```

---

bool ArePlaying(GameObject target)

Description
Check if GAC has any animation playing

```
if (JrDevAssets.GAC.ArePlaying(player1){
        Debug.Log("An animation is playing");
}
```

---

string AnimationPlaying(GameObject target)

Description
Returns the name of the GAC animation playing

```
//Logs the name of the animation
Debug.Log((JrDevAssets.GAC.AnimationPlaying(player1));
```

Conclusion

Thanks again for choosing to purchase the GAC System. I am looking forward to adding new features as time goes on. If anyone has any requests or bug problems just contact me at jrdevarts@comcast.net

GAC Official Website: Here

For Video Tutorials, check JrdevArts Page on Youtube: GAC Tutorial Playlist

Don't forget to also check the Official Forum Thread for video tutorials, post and ask questions:
GAC-Great Animation Combos System Thread