Make it Flow - Version 1.1

Make It Flow is an **Editor tool and UGUI trigger framework** that allows you to add **extra life and action** to your game UI. Set how your objects will behave, link it to user actions and enjoy creating more.

This documentation contains the detailed explanations for the implemented features, classes, methods, variables, concepts and setting adjustments.

Contact

[Asset Store Page]

[support@meadowgames.com]

[MeadowGames.com]

Contents

1. Overview

Make it Flow is an editor tool and UGUI events framework that elevates the Canvas with handy events for pointer and system actions that facilitate and speed up the development of event based Canvas compositions by binding and configuring handy behaviors for the scene graphic objects.

From the editor window, it is possible to easily add behaviors to Graphic objects, adjust their settings and up and bind them up with pointer and system events.

What it is:

- An events framework that handles convenient events on canvas and Graphic objects to speed up your development
- A tool that adds a new layer of abstraction to the UGUI
- A framework to be used on Canvas objects
- A visual tool that allows adding behaviors to the UGUI objects without coding, while new behavior can be created by the user if needed
- A tool that can speed up and facilitate UI development, by making Graphic objects ready with handy events and allowing the easy binding with behaviors

What it is not:

- An all in one tool that will cover all the UI development process
- A tool that will replace all coding in a UI development
- An pure animation tool
- Just button events

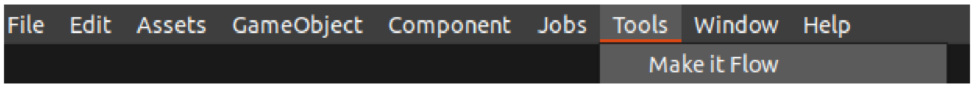
2. Getting Started

To have an idea of the possibilities, you can test the sample scenes available online, (1) and (2), made with Make It Flow.

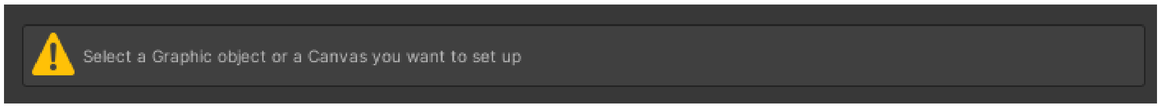
In the following sections you view the basic components of the system (MFSystemManager, CanvasManager, MFObject and MFBehavior), making the first steps to add behaviors to Graphic objects and bind to events.

2.1. Opening The MF Editor Window

The MF window is accessed from Unity's toolbar (Tools/Make it Flow). The visual interface helps on the configuration of behaviors as well as on the environment setup.



If you're starting from scratch, the window will present you a warning message, "Select a Graphic object or a Canvas you want to set up".


Select a Graphic object or a Canvas you want to set up

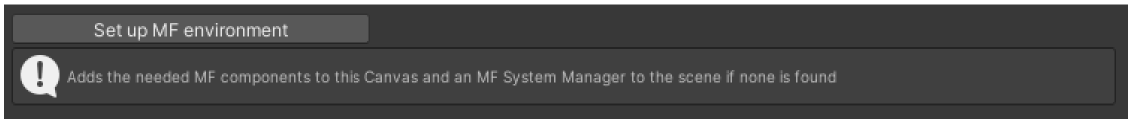
The next sections cover the environment setup, including the Canvas and later the MF objects. Details on the MF components are in section 3.

2.2. Setting Up the Environment

You can perform an automatic setup, or manually add the needed components.

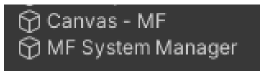
2.2.1. Automatic Setup

After selecting the Canvas in the scene that you want to have MF objects in, the MF window will display a button for you to set up the MF Environment. Click on the "**Setup MF environment**" button.



Adds the needed MF components to this Canvas and an MF System Manager to the scene if none is found

It will configure the Canvas GameObject with the needed components and add a mandatory MF System Manager to the scene if none is found.



2.2.2. Manual Setup

The basic environment components for the setup are Canvases and a single MF System Manager.

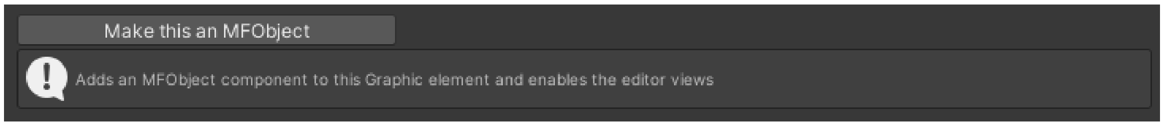
Canvas: A **CanvasManager** component is needed in a Canvas GameObject.

MF System Manager: A single **MFSystemManager** and a **MakeItFlow.InputManager** components must be added to the scene. Although it can be added to any object, it is recommended to have them in a dedicated GameObject.

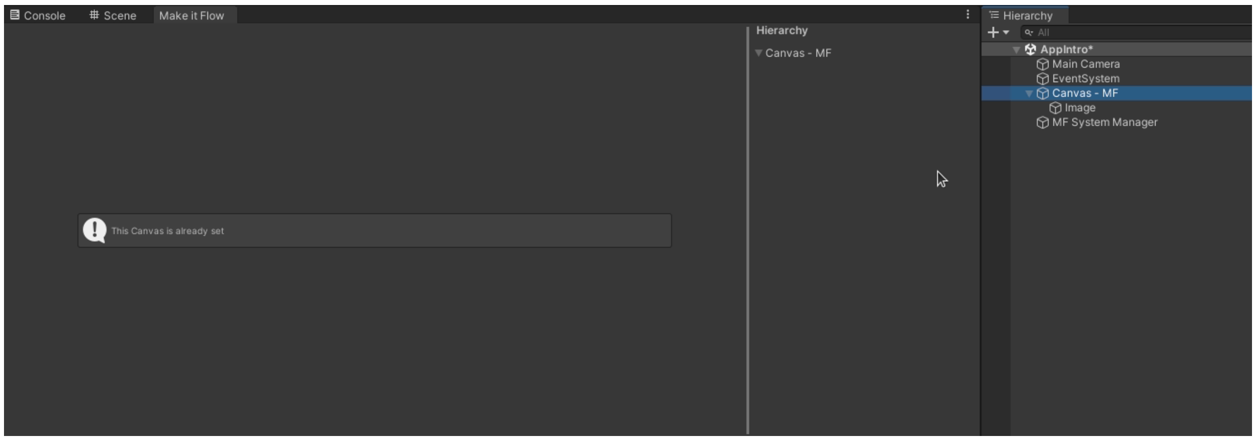
The package contains prefabs for both objects at "**Assets/MG_MakeItFlow/Prefabs**".

2.3. MF Object Setup

With the environment ready, you can set up a Graphic object by selecting it from the Hierarchy tab and clicking on the "**Make this an MF Object**" button from the MF Window. It will add the **MFObject** component to the object and the Object view will be enabled.



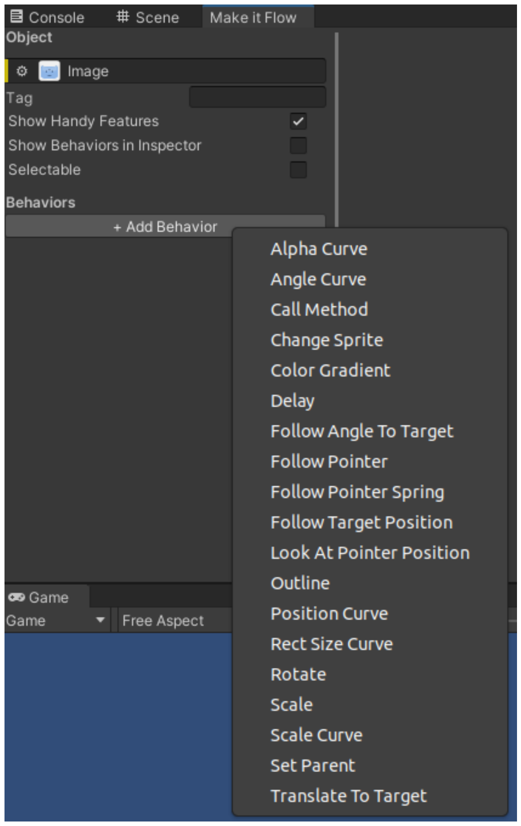
Adds an MFObject component to this Graphic element and enables the editor views



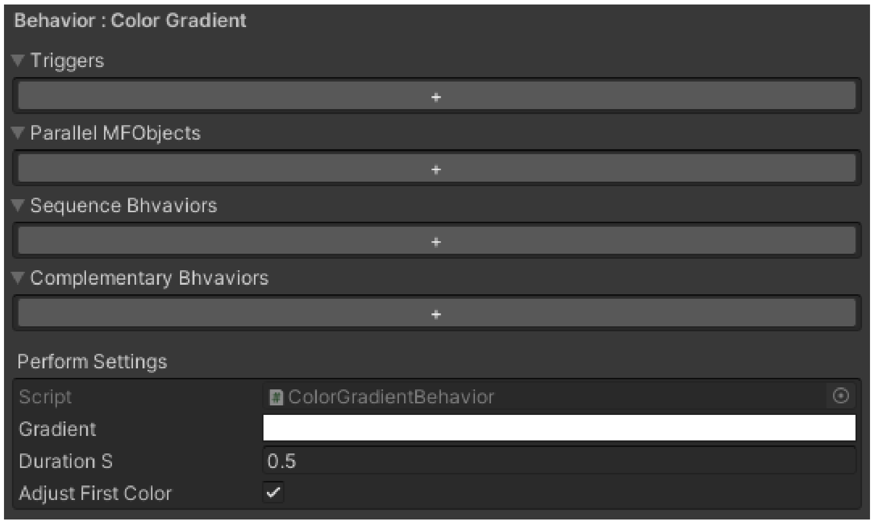
This Canvas is already set

2.4. Adding Behaviors to MF Objects

The Object view has an “**Add Behavior**” button that opens a list to choose from.



Selecting one behavior will enable the Behavior view and the settings of the new behavior will show up. As an example, the image below is a Color Gradient behavior.


Behavior : Color Gradient

Triggers

Parallel MFObjects

Sequence Bhvaviors

Complementary Bhvaviors

Perform Settings

Script	ColorGradientBehavior
Gradient	
Duration S	0.5
Adjust First Color	✓

2.5. Behavior Basics

The behavior is an “action” you want the graphic object to perform. This action is linked to a particular MF Object and can be related to additional ones (Parallel MFObjects) that will perform the same behavior. The action can be a visual or non-visual effect, as a color change, a transform movement, or a call for an external method.

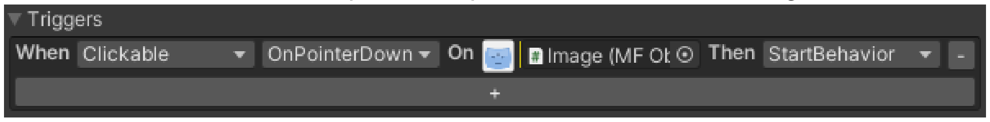
The main settings on a behavior are the “Triggers” and the “Perform Settings”, while the further fields are handy features to facilitate and speed up the workflow. All the settings are explained in the MF Editor Window section of this document.

The trigger lines are composed of a logic sentence with four settable fields, **“When [object event group] [object event] On [MF Object] Then [behavior call]”**.

The Perform Settings contains settings that are specific for each behavior.

Example: Configuring an object that slowly blinks orange when clicked.

The line below translates: “When the pointer is pressed down on the Image, start the behavior”.

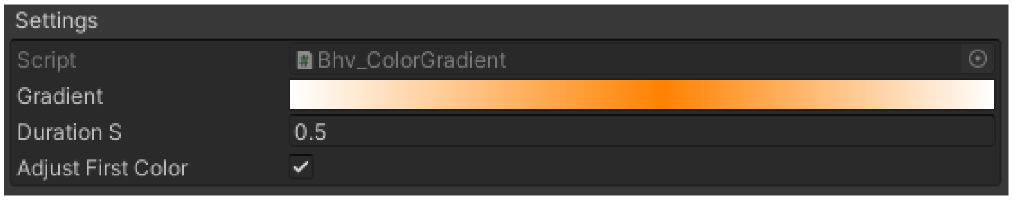


Triggers
When Clickable OnPointerDown On Image (MF Ob Then StartBehavior

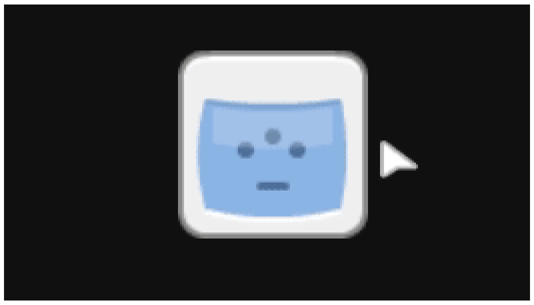
The behavior we will set up to perform is to change the object’s color by the gradient in 0.5 seconds.
Let’s change the color **Gradient** to go from the colors <u>white</u> to <u>orange</u> then <u>white</u> again, so we have a nice effect when the MF object is clicked.

The **Duration S** variable indicates that the gradient transition will take 0.5 seconds to complete.

The **Adjust First Color** toggle will make the first color of the gradient be the color of the MF object on the instant the behavior starts. Since the start and end color are the same in this example, it won't affect the behavior. This toggle is handy when you have complementary behaviors so the color changes smoothly.



Settings
Script Bhv_ColorGradient
Gradient
Duration S 0.5
Adjust First Color ✓



3. MF Editor Window

The editor window is accessed from Unity's toolbar “Tools/Make it Flow”.

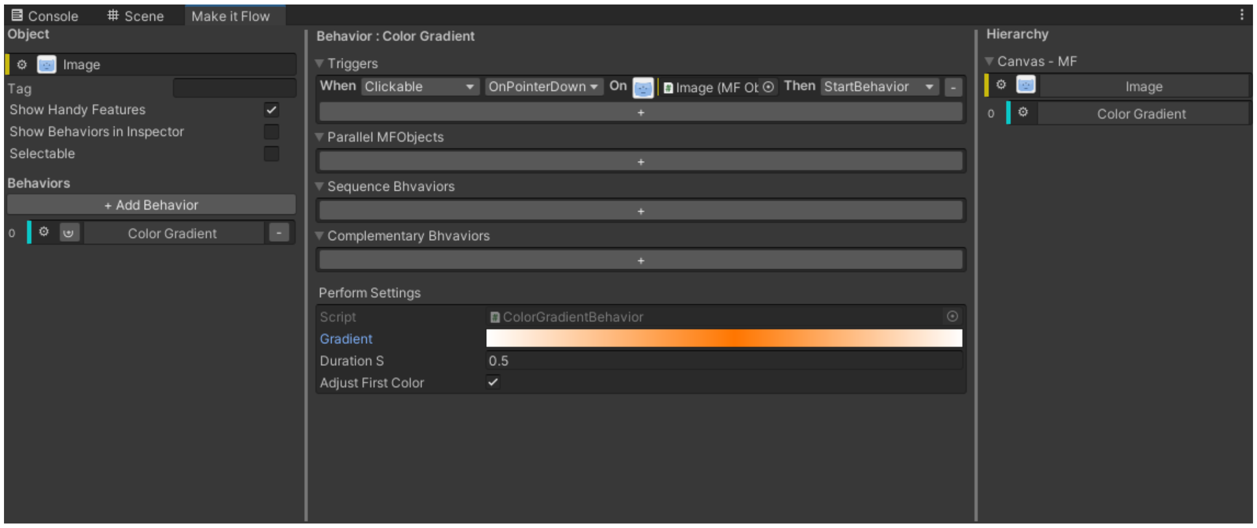


Image
Tag
Show Handy Features ✓
Show Behaviors in Inspector
Selectable

Behaviors
+ Add Behavior
0 Color Gradient

Triggers
When Clickable OnPointerDown On Image (MF Ot Then StartBehavior
Parallel MFObjects
Sequence Bhvaviors
Complementary Bhvaviors

Perform Settings
Script ColorGradientBehavior
Gradient
Duration S 0.5
Adjust First Color ✓

Canvas - MF
Image
0 Color Gradient

3.1.1. Interface Items

In the window, it is possible to select MF Object or Behavior and perform undo/redo.

There are four icons that identify actions you can perform in the MF window and two colors to identify objects, those are:

Gear or **Circled Asterisk** - Draggable spot for this item. From this icon you can drag and drop this object to places indicated with a thinner line of the same color.

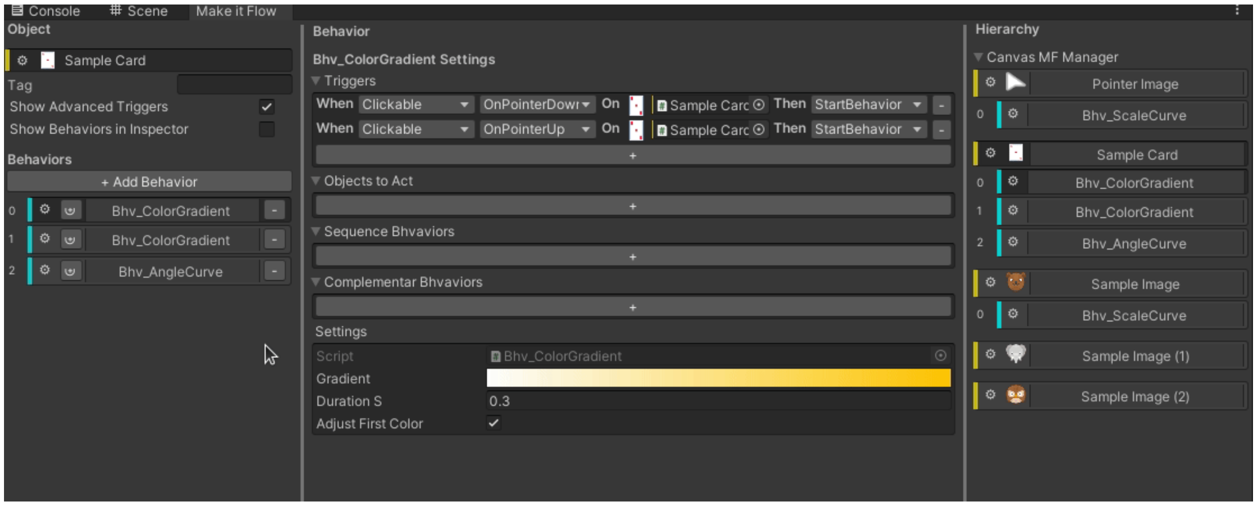
Plus - Add a new item.

Minus - Remove the respective item.

Multiset - Duplicate the respective item.

Color Blue - Behaviors. Thick lines are draggable, thin lines are spots to drop on.

Color Yellow - MF Objects. Thick lines are draggable, thin lines are spots to drop on.



Sample Card
Tag
Show Advanced Triggers ✓
Show Behaviors in Inspector

Behaviors
0 Bhv_ColorGradient
1 Bhv_ColorGradient
2 Bhv_AngleCurve

Bhv_ColorGradient Settings
Triggers
When Clickable OnPointerDowr On Sample Carc Then StartBehavior
When Clickable OnPointerUp On Sample Carc Then StartBehavior
Objects to Act
Sequence Bhvaviors
Complementar Bhvaviors
Settings
Script Bhv_ColorGradient
Gradient
Duration S 0.3
Adjust First Color ✓

Canvas MF Manager
Pointer Image
0 Bhv_ScaleCurve
Sample Card
0 Bhv_ColorGradient
1 Bhv_ColorGradient
2 Bhv_AngleCurve
Sample Image
0 Bhv_ScaleCurve
Sample Image (1)
Sample Image (2)

3.2. MF Object View

The MF Object view shows the selected object, it is where you add behaviors and see its general settings.

MFTag: this is an available string to be used if you need to identify MF objects.

Show Advanced Triggers: Toggle that enables/disables the view of the following behavior triggers: Parallel MF Objects, Sequence Behaviors and Complementary Behaviors.

Show Behaviors in Inspector: This toggle will enable the behavior components to be viewed in the object's inspector.

Selectable: Allows the MF object to invoke the Selectable events.

Behaviors: List of behaviors added to this MF object.

3.3. Behavior View

The selected behavior configurations are shown in the Behavior view, it contains triggers, handy features and the specific settings of the behavior.

Triggers: List of logic sentences that indicate to which events this behavior will be subscribed.

Parallel MF Objects: The MF objects added to this list will also behave the same way and based on the same triggers.

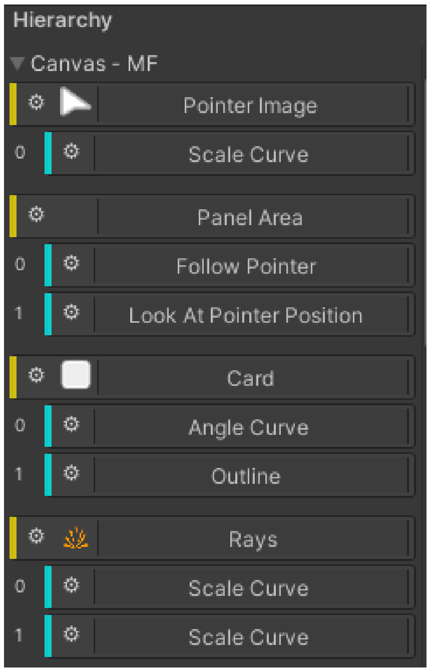
Sequence Behaviors: List of Behaviors that will start when the selected event is invoked.

Complementary Behaviors: The behaviors added to this list will have triggers setup automatically so they can be triggered complementary to each other (Start A will Interrupt B and Interrupt A will Start B).

3.4. Hierarchy View

The Hierarchy view is where all MF Objects in the scene and its behaviors are displayed, grouped by MF canvas. It also helps on the cross assignment of MF Objects to triggers and assigning sequential or complementary Behaviors.

All the active MF Objects and Behaviors (grouped by Canvas Managers) are displayed in the Hierarchy View. It facilitates the visualization and selection of MF Objects and Behaviors as well the the configuration of behaviors by allowing you to drag and drop items from the hierarchy.



4. MF Components

Make it Flow's main components.

4.1. Canvas Manager

The Canvas Manager is the component (CanvasManager class) used as reference so the Raycaster can find the child MF Objects. This component must be added to a Canvas object.

4.2. MF System Manager

The MF System Manager aggregates and executes all core update calls and Behavior.Execute() calls.

4.2.1. Events

Each Canvas Manager has its own events managed by an instance of EventsManager (EventsManager<MFObject> MFEvents).

The events listed below are called when the event occurs without specifying an MF Object.

All the logic to invoke the events is handled by an EventsHandler instance in the InputManager.

Event Group	**Event**	**Invoke condition**
Clickable	OnClick	Pointer goes down and up in under 0.2 seconds
	OnDoubleClick	Two clicks in under 0.2 seconds
	OnHoldClick	Pointer is down for more than 0.2 seconds
	OnPointerDown	Pointer changes from up to down
	OnPointerHold	Pointer is down
	OnPointerUp	Pointer changes from down to up
	OnSecondaryPointerDown	Secondary Pointer key changes from up to down
	OnSecondaryPointerHold	Secondary Pointer key is down

	OnSecondaryPointerUp	Secondary Pointer key changes from down to up
Draggable	OnDrag	Any MF Object is being dragged
	OnDragEnd	Any MF Object is released
	OnDragStart	Any MF Objects starts being dragged
Hover	OnPointerEnter	Pointer is up and enters any MF Object
	OnPointerEnterDrag	Pointer is down and enters any MF Object
	OnPointerExit	Pointer is up and leaves any MF Object
	OnPointerExitDrag	Pointer is down and leaves any MF Object
	OnPointerStay	Pointer is up and is over any MF Object
	OnPointerStayDrag	Pointer is down and is over any MF Object (pointer needs to move at least once)
Selectable	OnSelect	Any Selectable MF Object is clicked when not selected
	OnUnselect	Any Selectable MF Object is clicked when selected OR clicked at none when Any Selectable MF Object is selected
System Events	OnStart	System Manager Start method is called
	OnUpdate	MFSystemManager Update method is called
	OnFixedUpdate	MFSystemManager FixedUpdate method is called

4.2.2. Settings

Behavior Execution Times: Number of times the behaviors Execute method is called per update. This setting is a fine tune for the system that can be used if you need the system to respond faster to the trigger events.

Mind that it will increase the processing time and have a performance impact in scenes with a large number of behaviors.

Cache Graphic Raycasters: Cache the GraphicRaycasters in the scene at the Start to increase performance. GraphicRaycasters can be added to the cache by script if needed.

4.3. MF Object

The MF Objects are containers for the Behaviors, they have local events that are invoked based on the available user or system actions (ex.: click, drag, hover this object).

Any UI object with a Graphic component can be an MF Object.

4.3.1. Events

Each MF Object has its own events managed by an instance of the EventsManager (EventsManager MFEvents).

The events listed below are called when the event occurs to this specific MF Object.

All the logic to invoke the events is handled by an EventsHandler instance in the InputManager.

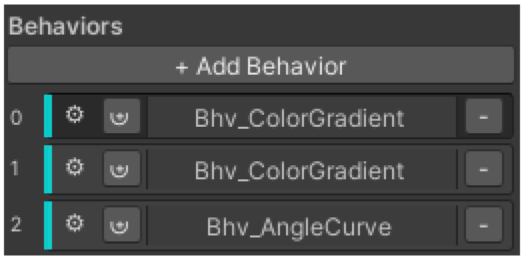
Event Group	**Event**	**Invoke condition**
Clickable	OnClick	Pointer goes down over this MF Object and up in under 0.2 seconds
	OnDoubleClick	Two clicks over the MF Object in under 0.2 seconds
	OnHoldClick	Pointer is down at this MF Object for more than 0.2 seconds
	OnPointerDown	Pointer changes from up to down over this MF Object
	OnPointerHold	Pointer goes down over this MF Object and keeps down

	OnPointerUp	Pointer changes from down to up over this MF Object
	OnSecondaryPointerDown	Secondary Pointer key changes from up to down over this MF Object
	OnSecondaryPointerHold	Secondary Pointer key goes down over this MF Object and keeps down
	OnSecondaryPointerUp	Secondary Pointer key changes from down to up over this MF Object
Draggable	OnDrag	This MF Object is being dragged
	OnDragEnd	This MF Object is released
	OnDragStart	This MF Objects starts being dragged
Hover	OnPointerEnter	Pointer is up and enters this MF Object
	OnPointerEnterDrag	Pointer is down and enters this MF Object
	OnPointerExit	Pointer is up and leaves this MF Object
	OnPointerExitDrag	Pointer is down and leaves this MF Object
	OnPointerStay	Pointer is up and is over this MF Object
	OnPointerStayDrag	Pointer is down and is over this MF Object (pointer needs to move at least once)
Selectable	OnSelect	This Selectable MF Object is clicked when not selected
	OnUnselect	This Selectable MF Object is clicked when selected OR clicked at none when This Selectable MF Object is selected
System Events	OnStart	This MF Object Start method is called

4.4. Behavior

Behaviors are adjustable predefined actions that the MF Object can perform based on triggers. You can choose from a dropdown list the behavior to add and, by selecting them it is possible to configure its settings (see next section).

The behavior line shows the behavior with an index to help identify the components during the behavior settings.



4.4.1. Events

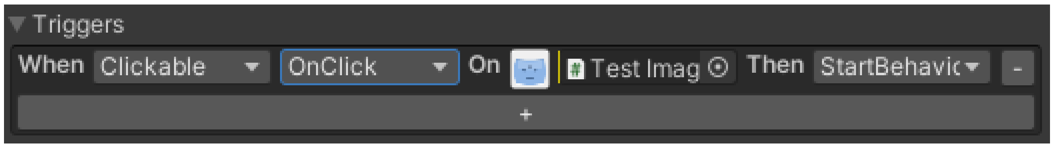
Each Behavior has its own events (BehaviorLocalEvents behaviorEvents).

The events listed below are called based on this specific behavior and it is used to create "Sequence Behaviors".

Event	**Invoke condition**
OnBehaviorStart	This Behavior starts
OnBehaviorEnd	This Behavior finishes its full cycle and stop
OnBehaviorInterrup	This Behavior is interrupted before completing its cycle

4.4.2. Triggers

List of triggers, in the form of a logic sentence, that indicate to which events this behavior will be subscribed. This is the main part of configuring a behavior, you can have triggers to make the behavior start, stop (when the behavior finishes) or interrupt it (as soon as the trigger is invoked) the behavior.



The sentence, "**When [object event group] [object event] On [MF Object] Then [behavior call]**" have four configurable fields:

Object event group: Interface that groups the MF object events by category facilitating the event selection.

Object event: Event that is invoked based on a user or system action that is done to the following MF object.

MF Object: The object that invokes the event. It is possible to select any MF object as a target or set none.

- Selecting an MF Object as a trigger will make this event be called only if the event is performed on the selected object
- If the field is cleared, none is selected, the event will be called for any occurrence of this event.
    - Clickable, Draggable and System Events will be called even if no object is under the pointer
    - Hover and Selectable events will be called if there is an object under the pointer.

Behavior call: You can choose from three types of behavior calls to invoke when the trigger happens, either to start or stop it:

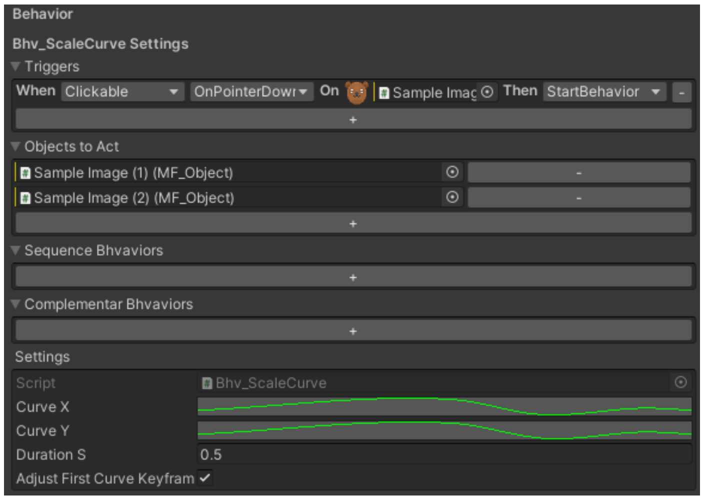
- StartBehavior: Start the behavior.
- StopOnBehaviorEnd: Different then behaviors that are fully executed in a single call, some behaviors will perform a smooth transition in a set time, for those, this call will mark the behavior to be stopped when the behavior is fully completed. Ex: if this call is made to the Color Gradient behavior, the object will finish the transition to the end color then stop.
- InterruptBehavior: This call interrupts the behavior immediately.

Obs.: Most of the behaviors, once started, cannot be started again until it is either finished or interrupted.

4.4.3. Parallel MF Objects

The MF Objects added to this list will perform the same behavior, in "parallel", based on the same triggers.

Example: Object A is set to change its scale when the user performs a click on A. Objects B and C, added to the Parallel MF Objects list, will also change the scale when the user clicks on A.



Behavior

Bhv_ScaleCurve Settings

Triggers

When Clickable OnPointerDown On Sample Imag Then StartBehavior -

+

Objects to Act

Sample Image (1) (MF_Object) -

Sample Image (2) (MF_Object) -

+

Sequence Bhvaviors

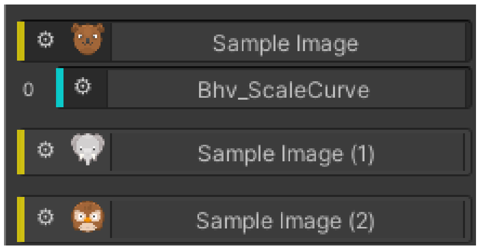
+

Complementar Bhvaviors

+

Settings

Script	Bhv_ScaleCurve
Curve X	
Curve Y	
Duration S	0.5
Adjust First Curve Keyfram	✓

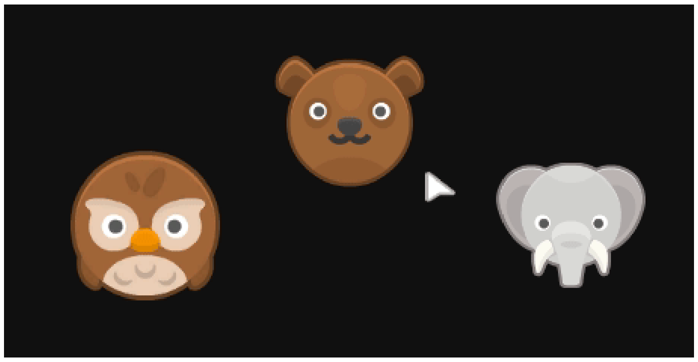


Sample Image

0 Bhv_ScaleCurve

Sample Image (1)

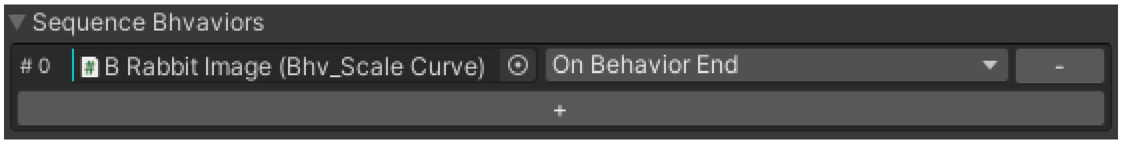
Sample Image (2)



4.4.4. Sequence Behaviors

You can make other behaviors start based on the selected behavior events by adding them to the Sequence Behaviors list.

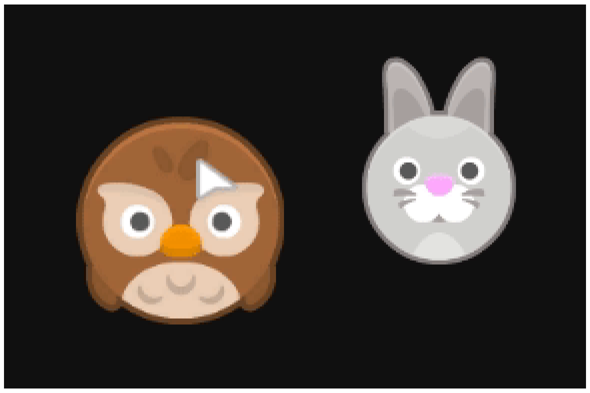
Example: Behavior of B starts right after Behavior of A ends



Sequence Bhvaviors

0 B Rabbit Image (Bhv_Scale Curve) On Behavior End -

+



4.4.5. Complementary Behaviors

The behaviors added to this list will have triggers setup automatically so they can be complementary to each other. When behavior A is started, behavior B is interrupted and when behavior A is interrupted, behavior B is started.

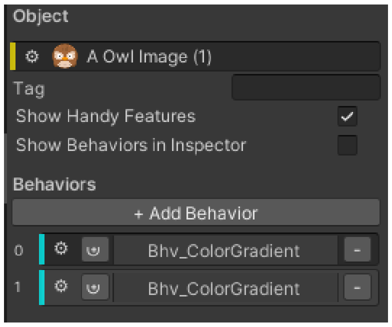
See the following examples:

Example 1: If you have a base behavior A and add a second behavior B to the Complementary list, every trigger of A set to call StartBehavior will automatically interrupt B (calling InterruptBehavior B) and every trigger of A set to call InterruptBehavior will automatically start B (calling StartBehavior B).

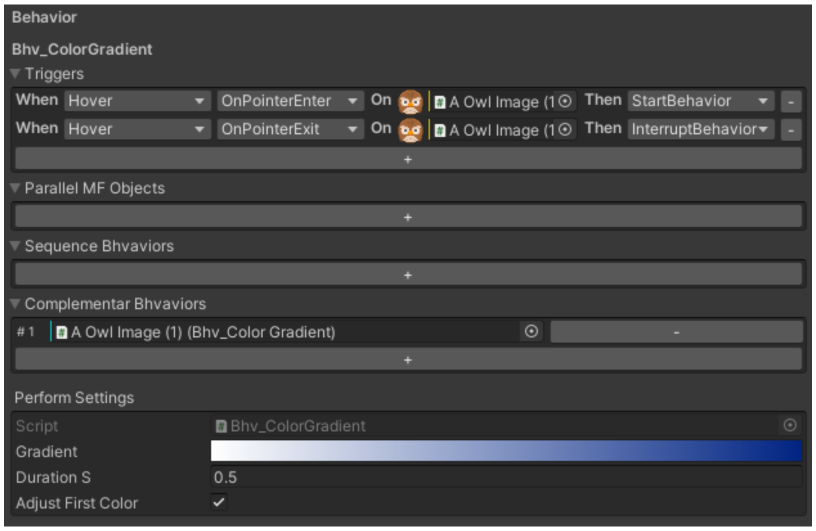
Obs1.: It is required that the base behavior A has the triggers to call StartBehavior, InterruptBehavior or both setup so this feature works.

Obs2.: Only the triggers from A will be mirrored to the behavior B, therefore, triggers set on B are not mirrored to A.

Example 2: Object that changes from white to blue when the pointer is on top of it and changes back to white when the pointer leaves.	
Object View	Behavior A

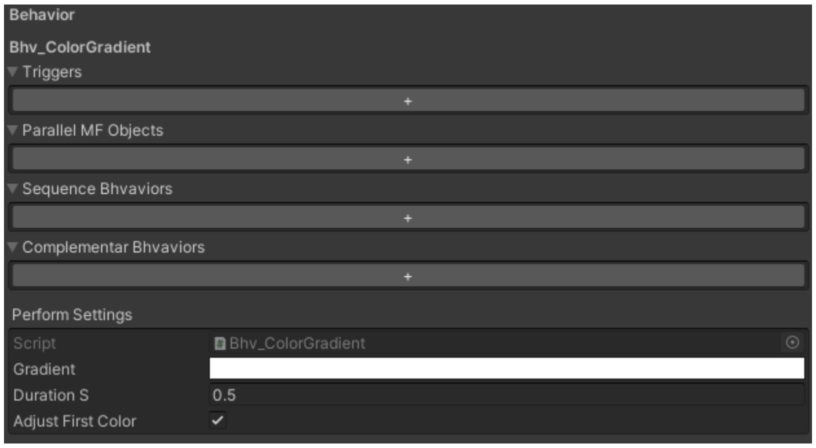


Object
A Owl Image (1)
Tag
Show Handy Features ✓
Show Behaviors in Inspector
Behaviors
+ Add Behavior
0 Bhv_ColorGradient -
1 Bhv_ColorGradient -



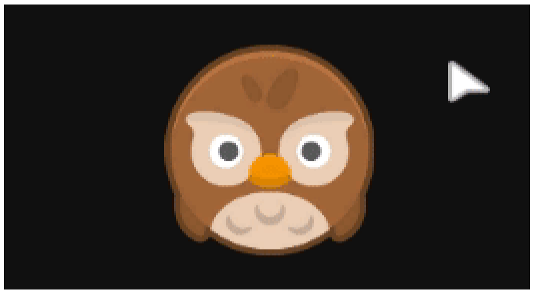
Behavior
Bhv_ColorGradient
Triggers
When Hover OnPointerEnter On A Owl Image (1 Then StartBehavior -
When Hover OnPointerExit On A Owl Image (1 Then InterruptBehavior -
Parallel MF Objects
Sequence Bhvaviors
Complementar Bhvaviors
1 A Owl Image (1) (Bhv_Color Gradient) -
Perform Settings
Script Bhv_ColorGradient
Gradient
Duration S 0.5
Adjust First Color ✓

Behavior B



Behavior
Bhv_ColorGradient
Triggers
Parallel MF Objects
Sequence Bhvaviors
Complementar Bhvaviors
Perform Settings
Script Bhv_ColorGradient
Gradient
Duration S 0.5
Adjust First Color ✓

Obs.: Since the checkbox “Adjust First Color” is marked, the Behavior B doesn’t need to start from blue, the first color will be the color of the object at the moment the behavior starts, it makes the color change smoothly.



4.4.6. Behaviors and Perform Settings

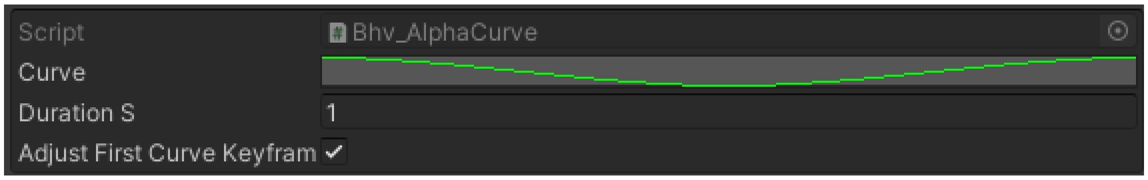
Each behavior has settings to configure how it will be performed, the shown fields are based on the behavior type.

> Alpha Curve: Interpolates the alpha value of the CanvasGroup from time 0 to 1. It adds a CanvasGroup component to the object, if none is found.

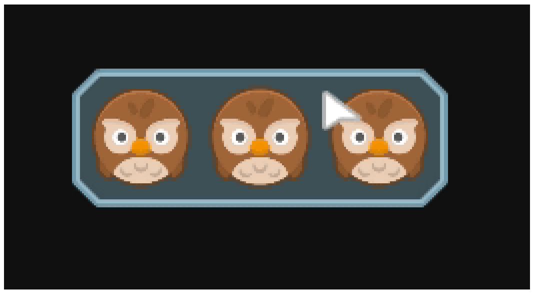
Curve - Alpha value curve. Alpha value is ranged 0-1.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first keyframe of the curve is set as the CanvasGroup alpha at this frame.



Script	Bhv_AlphaCurve
Curve	
Duration S	1
Adjust First Curve Keyfram	✓

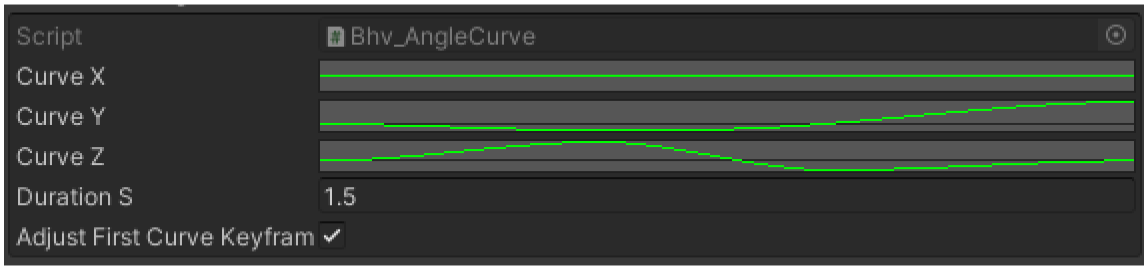


> Angle Curve: Interpolates the RectTransform euler angles from time 0 to 1.

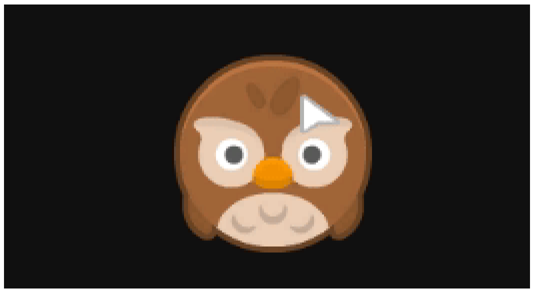
Curve X, Y, Z - Angle value curves in degrees.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first keyframe of the curves are set as the RectTransform angle at this frame.

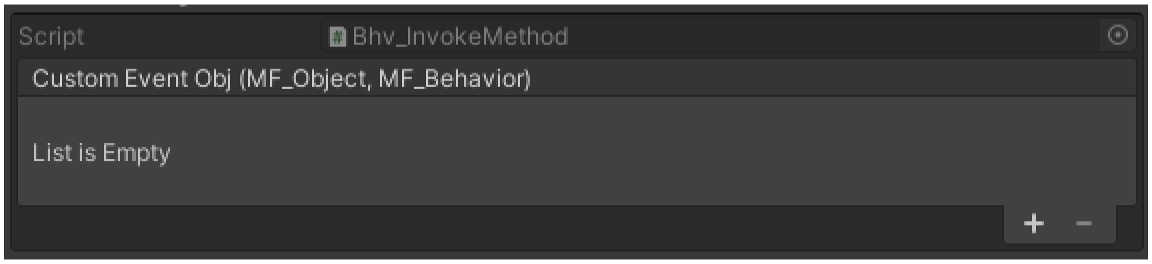


Script	Bhv_AngleCurve
Curve X	
Curve Y	
Curve Z	
Duration S	1.5
Adjust First Curve Keyfram	✓



> Call Method: Invokes a UnityEvent<MF_Object, MF_Behavior>. You can register a callback with the same arguments (MF_Object, MF_Behavior), the passed values are the object that called the StartBehavior and this behavior.

Listeners - Delegates that will be called when this behavior starts (event is invoked).



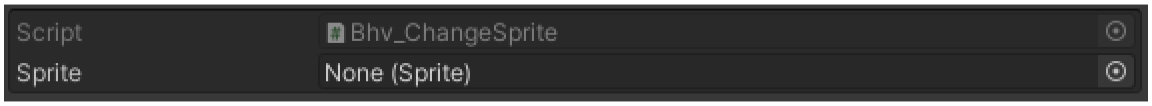
Script Bhv_InvokeMethod

Custom Event Obj (MF_Object, MF_Behavior)

List is Empty

> Change Sprite: Immediately changes the sprite of the Image component to the selected one.

Sprite - Target sprite.



Script Bhv_ChangeSprite

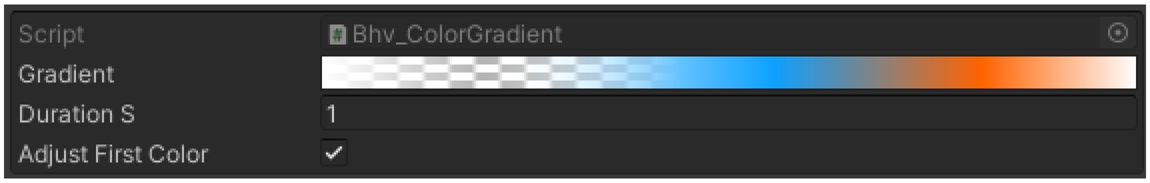
Sprite None (Sprite)

> Color Gradient: Interpolates the Image color.

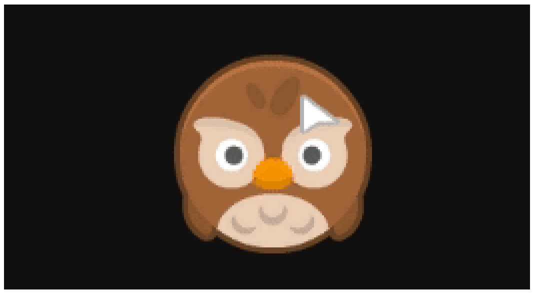
Gradient - Color values to interpolate.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first color of the gradient is set as the Image color at this frame.

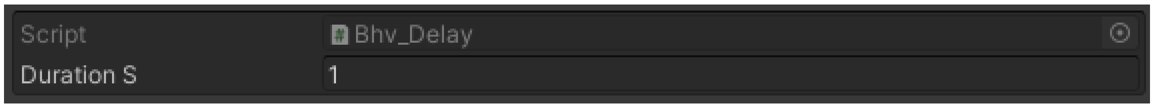


Script	Bhv_ColorGradient
Gradient	
Duration S	1
Adjust First Color	✓



> Delay: Waits the set duration to invoke the "BehaviorEnd" event. Meant to be used to start Sequence Behaviors after a set time.

Duration S - Duration of the delay in seconds.



Script	Bhv_Delay
Duration S	1

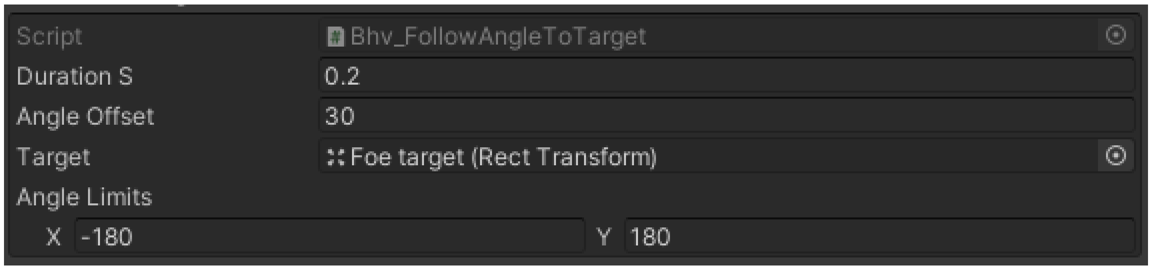
> Follow Angle to Target: Rotates the object on the Z axis so it points to the Target.

Duration S - Duration of the rotation until reach the target angle in seconds.

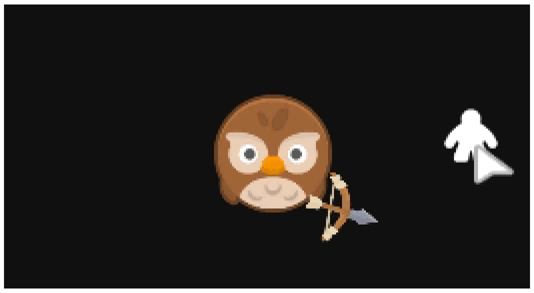
Angle Offset - Z axis angle offset.

Target - Position target for the object to point at.

Angle Limits - Z axis rotation min (X) and max (Y) limits.



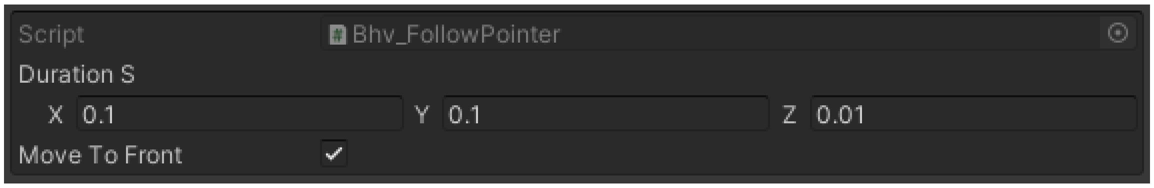
Script	Bhv_FollowAngleToTarget
Duration S	0.2
Angle Offset	30
Target	Foe target (Rect Transform)
Angle Limits	X -180 Y 180



> Follow Pointer: Make the object follow the pointer position.

Duration S - Duration of the interpolation in seconds.

Move to Front - Set object as last sibling.



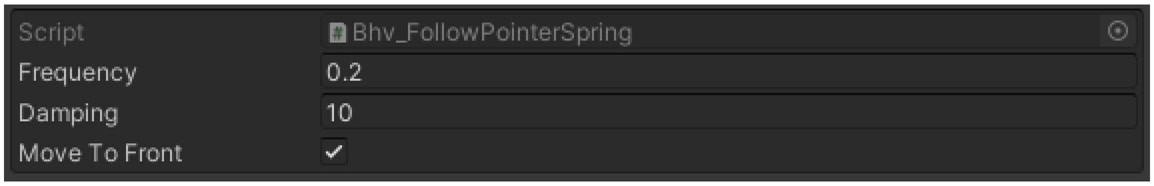
Script	Bhv_FollowPointer
Duration S	
X 0.1	Y 0.1 Z 0.01
Move To Front	✓

> Follow Pointer Spring: Makes the object follow the pointer position with a spring effect.

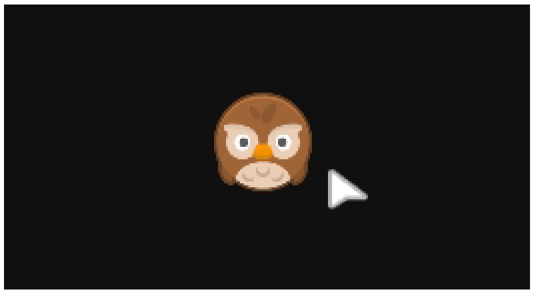
Frequency - Movement oscillation frequency.

Damping - Oscillation decay rate.

Move to Front - Set object as last sibling.



Script	Bhv_FollowPointerSpring
Frequency	0.2
Damping	10
Move To Front	✓

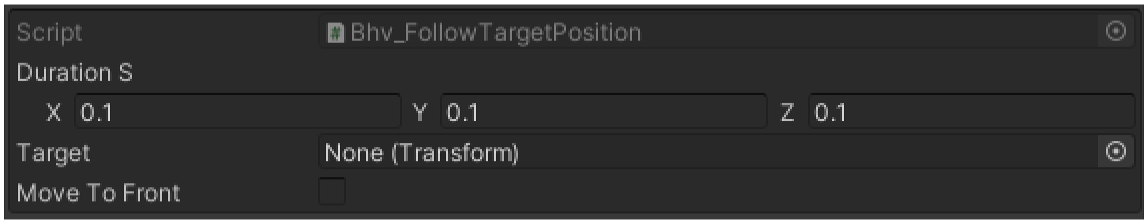


> Follow Target Position: Makes the object follow the target object position.

Duration S - Duration of the interpolation in seconds.

Target - Target position.

Move to Front - Set object as last sibling.



Script	Bhv_FollowTargetPosition
Duration S	
X 0.1	Y 0.1 Z 0.1
Target	None (Transform)
Move To Front	☐

> Look at Pointer Position: Angles the object so it faces the pointer position based on the distance.

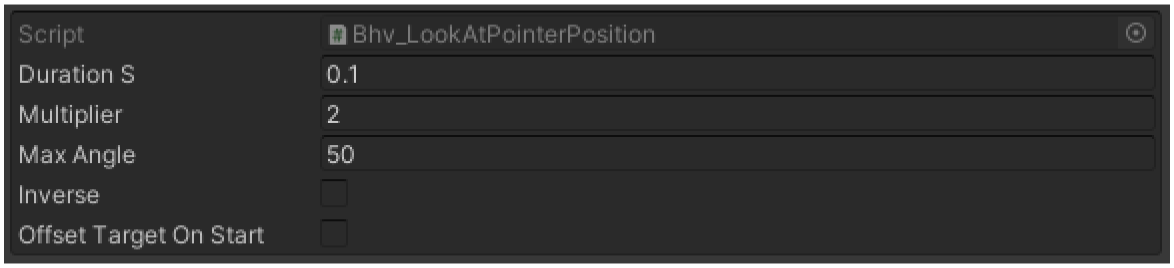
Duration S - Duration of the interpolation in seconds.

Multiplier - Distance multiplier.

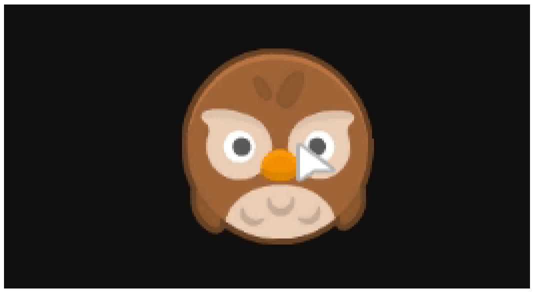
Max Angle - Maximum angle the object turns.

Inverse - Invert the face direction.

Offset Target On Start - Offsets the look target position by the distance between the object and the pointer on the start of the behavior (Tip: Set true if using alongside Follow Pointer behavior. Set it false if using in a static object).



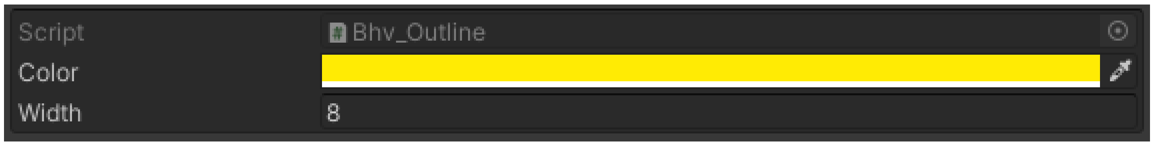
Script	Bhv_LookAtPointerPosition
Duration S	0.1
Multiplier	2
Max Angle	50
Inverse	☐
Offset Target On Start	☐



> Outline: Enables the Outline component on the object. Adds an Outline component if none is found.

Color - Outline color.

Width - Outline width.



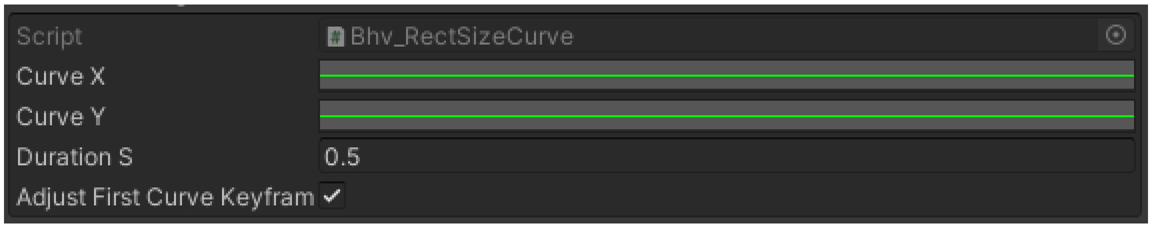
Script	Bhv_Outline
Color	🟨
Width	8

> Rect Size Curve: Interpolates the RectTransform size from time 0 to 1.

Curve X, Y - Width (X) and Height (Y) value curves.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first keyframe of the curves are set as the RectTransform size.

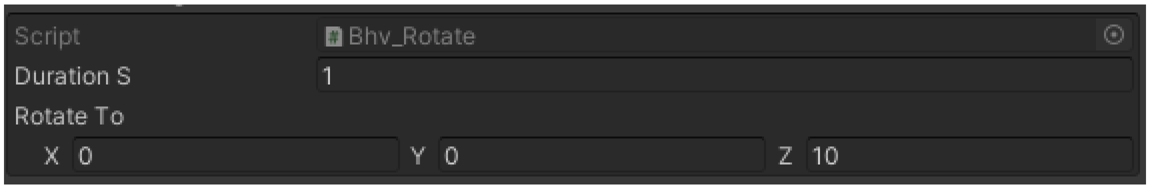


Script	Bhv_RectSizeCurve
Curve X	
Curve Y	
Duration S	0.5
Adjust First Curve Keyfram	✓

> Rotate: Interpolates the RectTransform euler angle.

Duration S - Duration of the interpolation in seconds.

Rotate to - Target angle in degrees.

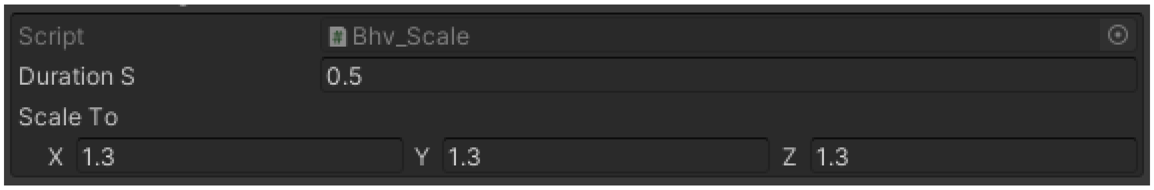


Script	Bhv_Rotate
Duration S	1
Rotate To	
X 0	Y 0 Z 10

> Scale: Interpolates the RectTransofmr scale.

Duration S - Duration of the interpolation in seconds.

Stale to - Target scale.



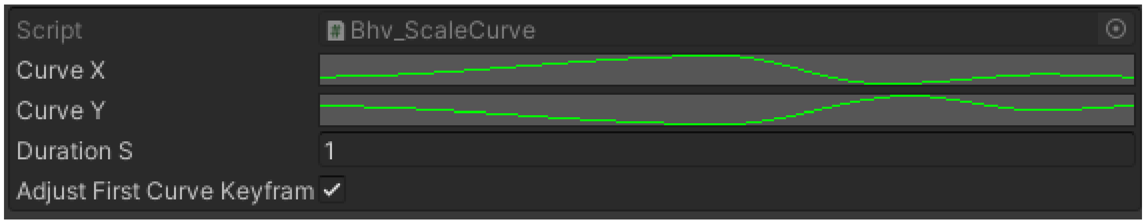
Script	Bhv_Scale
Duration S	0.5
Scale To	
X 1.3	Y 1.3 Z 1.3

> Scale Curve: Interpolates the RectTransform scale from time 0 to 1.

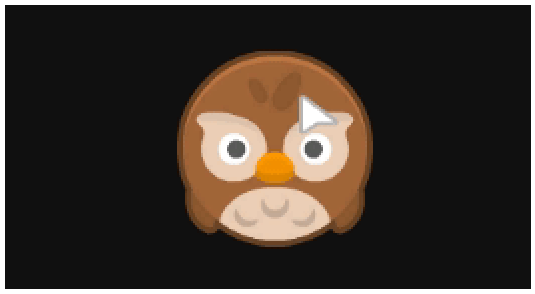
Curve X, Y - Scale value curves.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first keyframe of the curves are set as the RectTransform scale at this frame.

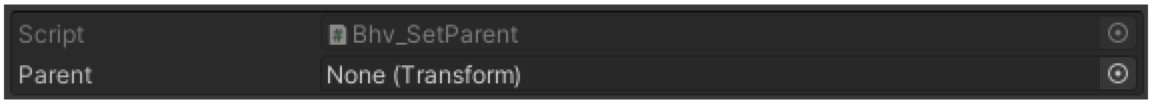


Script	Bhv_ScaleCurve
Curve X	
Curve Y	
Duration S	1
Adjust First Curve Keyfram	✓



> Set Parent: Make the object child of the target.

Parent - Target parent.



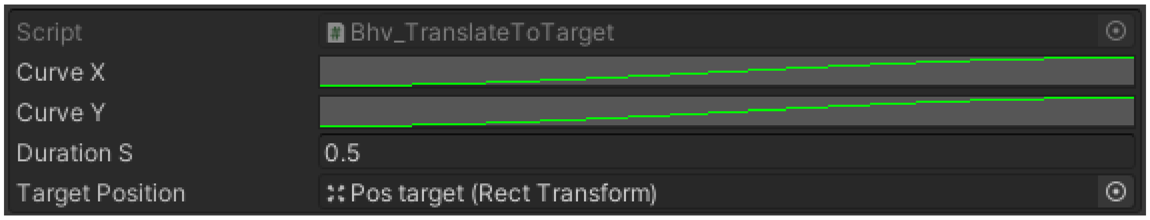
Script	Bhv_SetParent
Parent	None (Transform)

> Translate to Target: Interpolates the RectTransform position from time 0 to 1.

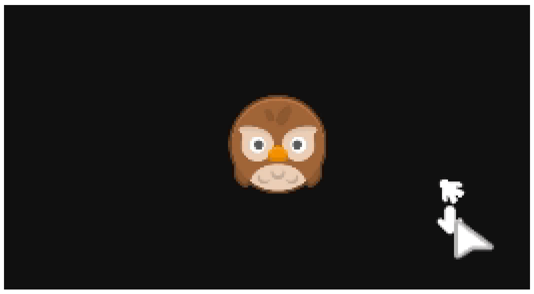
Curve X, Y - Position value curves.

Duration S - Duration of the interpolation in seconds.

Adjust First Curve Keyframe - On the behavior Start, the first keyframe of the curves are set as the RectTransform position at this frame.



Script	Bhv_TranslateToTarget
Curve X	
Curve Y	
Duration S	0.5
Target Position	Pos target (Rect Transform)



4.5. Input Manager

The Input Manager indicates the input sources for the system.

The default Input System used is based on the legacy Unity's input system.

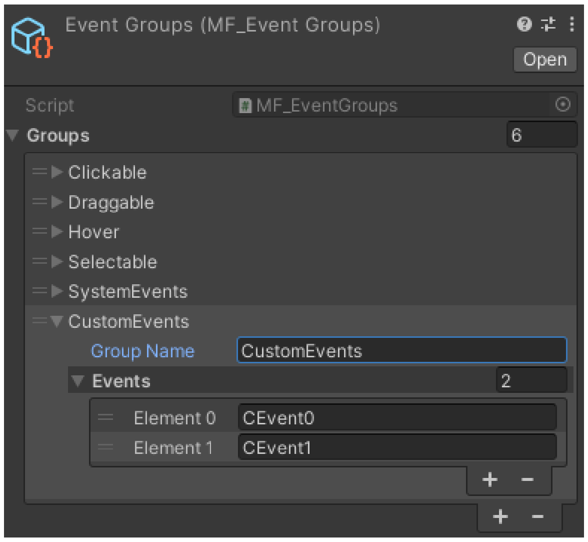
5. Customization

It is possible to enable custom Events and new Behaviors to the system.

5.1. Enabling Custom Events

To use custom events in the MF Window (select them to create triggers) you must add them to the “Event Groups” scriptable object. The object is located at:

“Assets/Make It Flow/Scripts/Core/Event System/Event Groups/Event Groups.asset”.



Script	MF_EventGroups
Groups	6
Clickable	
Draggable	
Hover	
Selectable	
SystemEvents	
CustomEvents	
Group Name	CustomEvents
Events	2
Element 0	CEvent0
Element 1	CEvent1

Each group added must have an unique name and each event must also have an unique name regarding all the events from all the event groups.

You have to implement the logic for invoking the event and, for consistency, invoke it by calling the two possible cases:

Triggered from an MF Object:

```
mfObject.MFEvents.TriggerEvent("CEvent0")
```

Triggered from any or no MF Object:

```
mfCanvasManager.MFEvents.TriggerEvent("CEvent0")
```

5.2. Adding Behaviors

New Behaviors scripts must inherit from Behavior and be placed at:

“Assets/Make It Flow/Scripts/Behaviors”.

For consistency, the three local events should be invoked accordingly:

When the behavior start:

```
localEvents.OnBehaviorStart.Invoke();
```

When the behavior finishes its full cycle and stop:

```
localEvents.OnBehaviorEnd.Invoke();
```

When the behavior is interrupted before completing the full cycle:

```
localEvents.OnBehaviorInterrupt.Invoke();
```

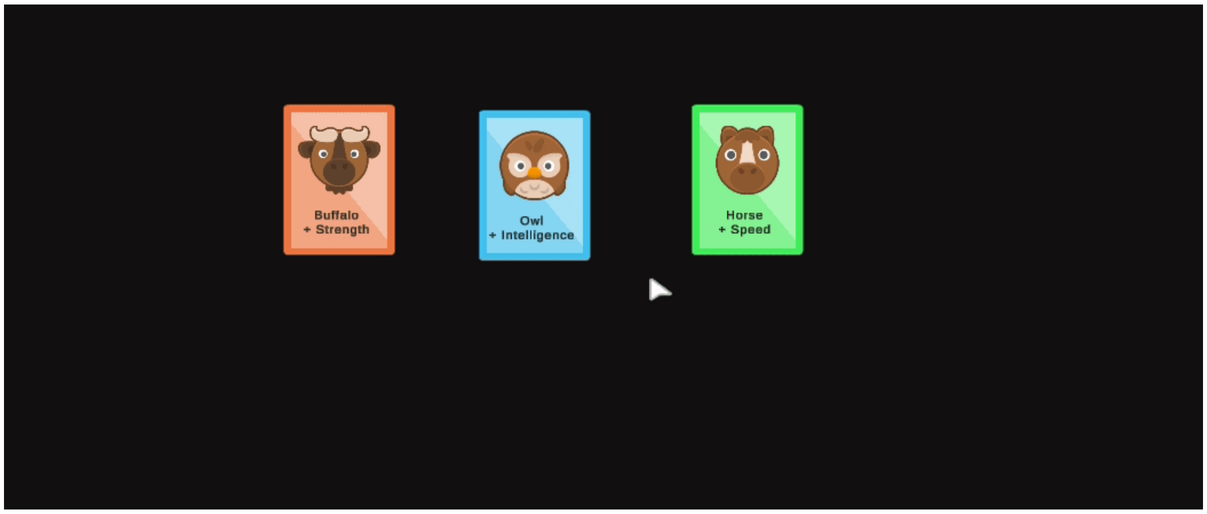
Use one of the default behaviors as a template (a simple example is the DelayBehavior.cs).

6. Sample Scenes

There are two sample scenes in the project with multiple examples showcasing the usage of Make it Flow. Both scenes are available online for testing in WebGL builds.

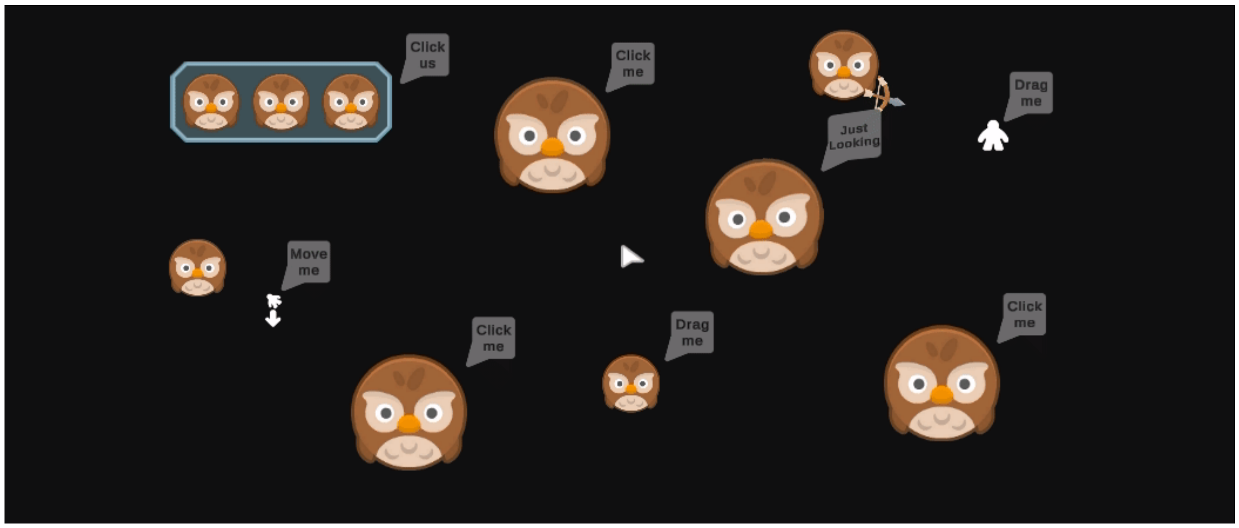
6.1. Showcase Scene

This scene contains a composition used to showcase the system.



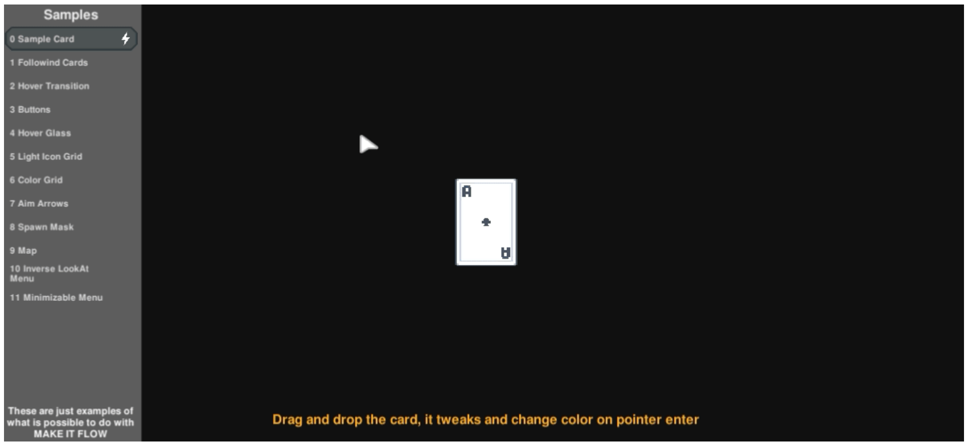
6.2. Single Behavior Samples

This scene contains multiple objects, each of them making use of a single behavior configured with a single trigger.



6.3. Composition Samples

This scene contains 12 compositions, showcasing the use of multiple behaviors, mixed triggers and the handy features.


Samples
0 Sample Card
1 Followind Cards
2 Hover Transition
3 Buttons
4 Hover Glass
5 Light Icon Grid
6 Color Grid
7 Aim Arrows
8 Spawn Mask
9 Map
10 Inverse LookAt Menu
11 Minimizable Menu
These are just examples of what is possible to do with MAKE IT FLOW
Drag and drop the card, it tweaks and change color on pointer enter