Advanced Playmaker Actions

Documentation

F. Rick Reich

20th of February, 2014

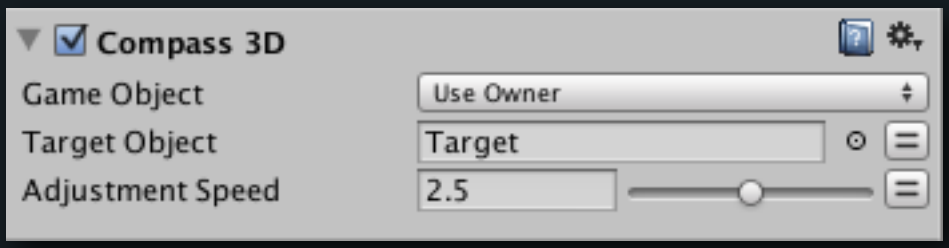
How To Install

Thank you for Purchasing Advanced Playmaker Actions, If you downloaded this Package with the Unity Asset Store, you can find all the new actions in your Action Browser under the „Advanced“ tab. Otherwise, drag the unitypackage file that was delivered with this documentation into your scene to use it.

Advanced Actions

Advanced Playmaker Actions is a package of assorted Playmaker actions that might come in handy for you, This Documentation will explain every Action to you.

3D Compass



Builds a compass to show the direction to a set target, bouncing a bit to add realism.

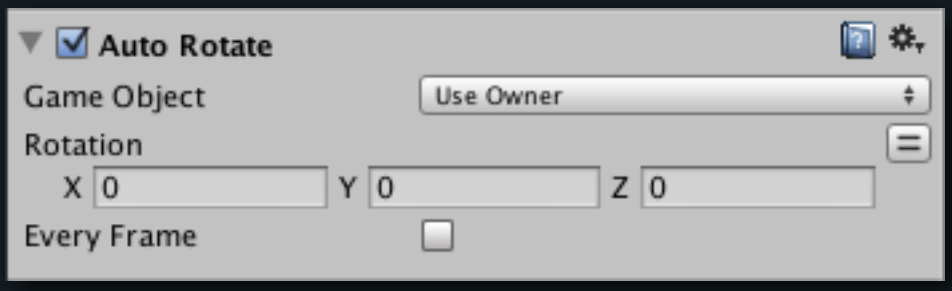
Game Object: The gameobject used as compass needle.
Target Object: The target of the compass needle.
Adjustment Speed: the speed the compass bounces/wiggles to the right direction.

Tip:

By using 2 compasses above each other and setting one to a game object far into the Z direction you can make a secondary needle always pointing to the north of the scene. (See example scene.)

Important: Attach the compass object to your Main Camera to avoid strange results.

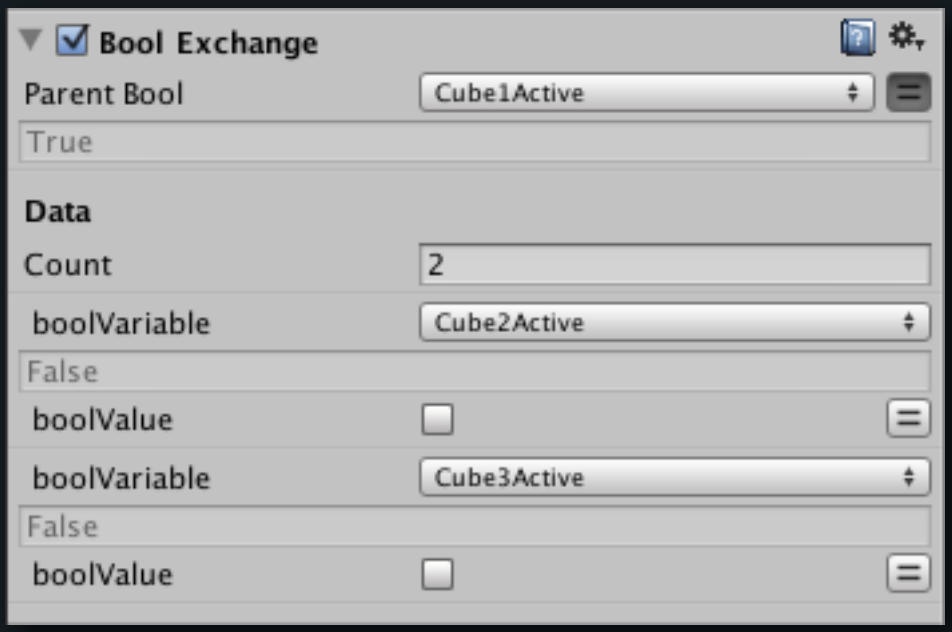
Auto Rotate



Constantly Rotates an object along an axis, using a speed value.

Game Object: Lets you select a GameObject to be rotated, defaults to Owner.
Rotation: Sets the speed of the rotation per axis by using a float value.
Every Frame: Repeats the action every frame.

Bool Exchange



Sets an Array of bools to the opposite of the parent bool.

Parent Bool: The parent bool Variable.
Count: The amount of bools to set.
boolVariable: the variable of the array bool.
boolValue: the value of the array bool.

Tip:
Best for use to deactivate an array of gameobjects or gui objects by bool.

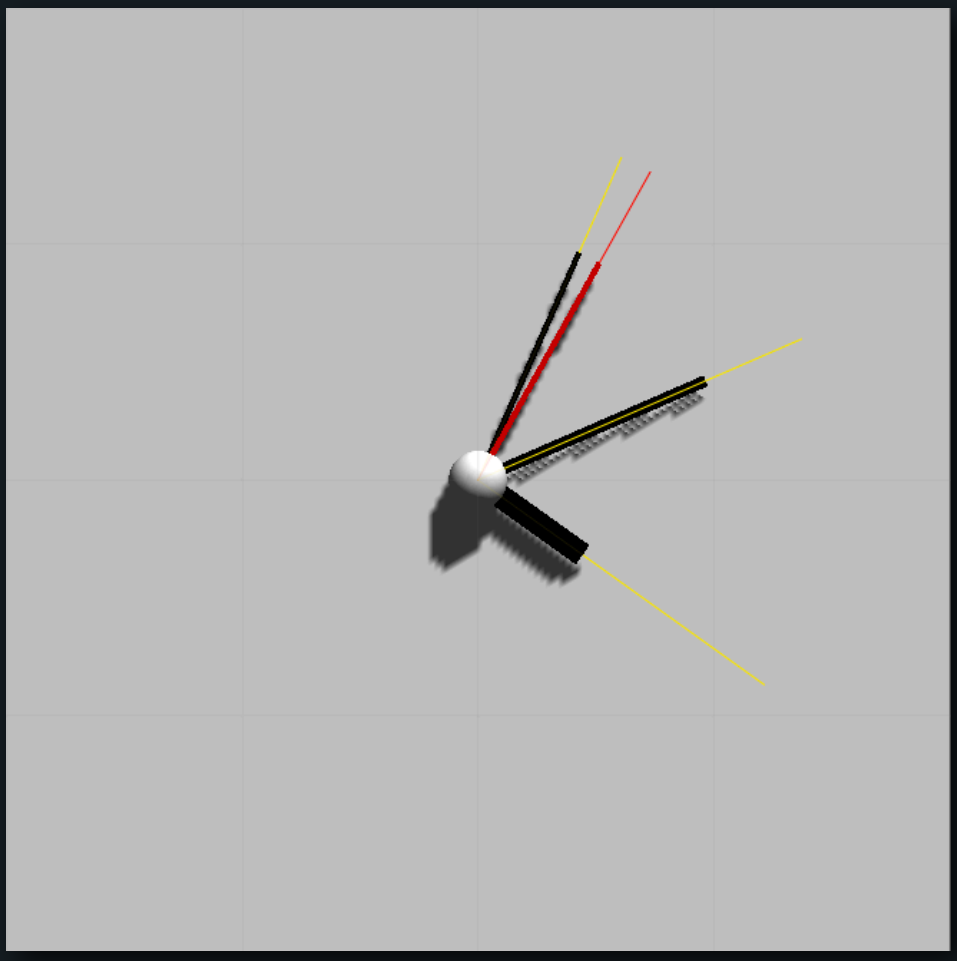
Clock Timer

Creates an analog clock using system date time.

Hours: The amount of Hours to be shown.
Minutes: The amount of Minutes.
Seconds: The amount of Seconds.
Milliseconds: The amount of Milliseconds.
Hours Game Object: The Game Object to be used as Hour Clock Hand.
Minutes Game Object: The Game Object to be used as Minutes Clock Hand.
Seconds Game Object: The Game Object to be used as Seconds Clock Hand.
Milliseconds Game Objects: The Game Object to be used as Milliseconds Clock hand.
Every Frame: Repeats the action every frame.
Debug: Shows the Clocks hands as lines in the Scene View.

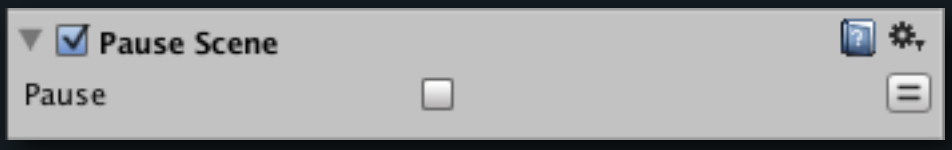
Tip:
Make sure that all the Hands pivots are in the same Y and X position.

Example:



Using the system date we display the current time in the scene using 4 Game Objects as Hands that share the same pivot point.

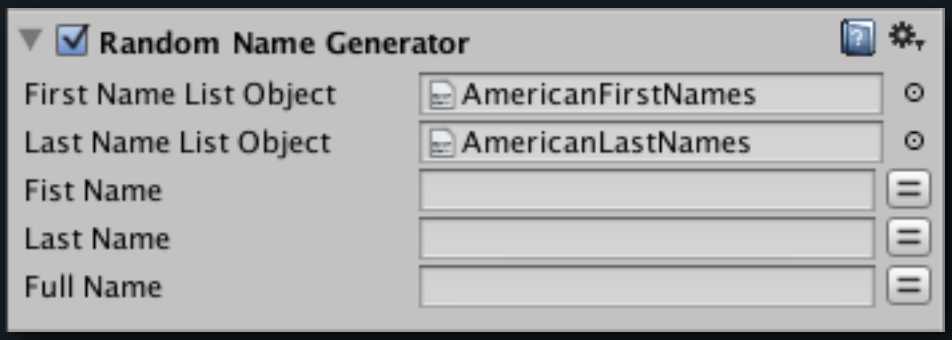
Pause Scene



Pauses/Unpauses the current scene using a bool value.

Pause: Pauses or Unpauses the current scene.

Random Name Generator

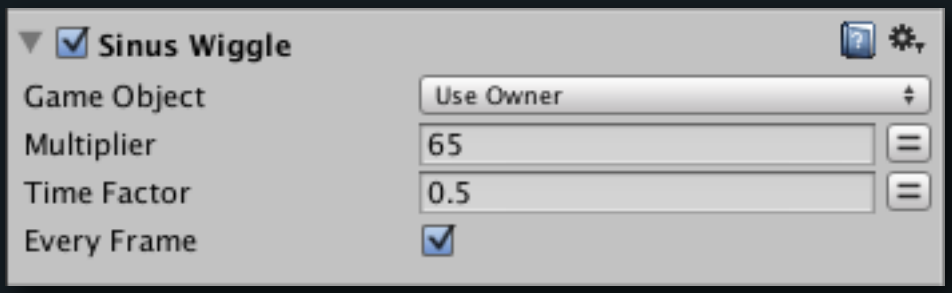


Generates a random name from lists of first and last names.

First Name List Object: Selected file of First Names.
Last Name List Object: Selected file of Last Names.
First Name: Randomly selected First Name.
Last Name: Randomly selected Last Name.
Full Name: Randomly selected First and Last name together.

Tip:
Comes with 2 lists of the most common american First and Last Names, use whatever list you want by putting a different name in each line.

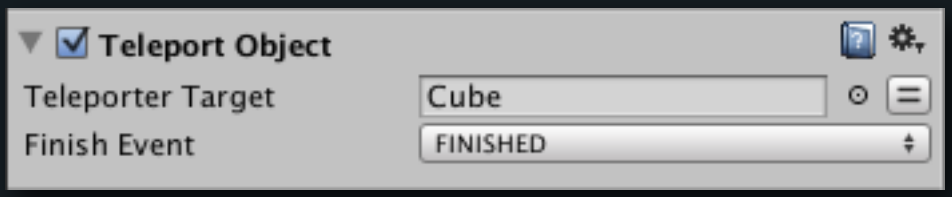
Sinus Wiggle



Wiggles an object around the Y axis using a time factor.

Game Object: The selected object to be wiggled around.
Multiplier: The time factor multiplier
Time Factor: The time the object needs to wiggle around itself.
Every Frame: Repeats the action every frame.

Teleport Object

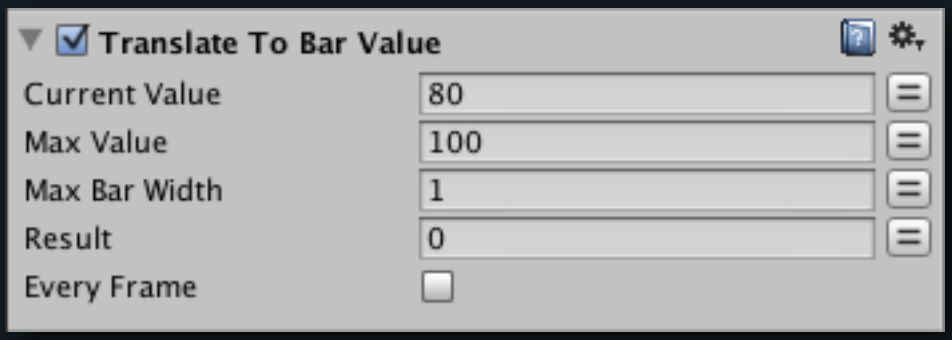


Teleports triggering object to a set target and fires an event.

Teleporter Target: Target of the teleporter action.
Finish Event: Event fired after teleportation succeeded.

Tip:
Dont forget that your triggering-objects pivot point may not be on the same level with the target object, so better use a freely positioned Empty Game Object as Target.

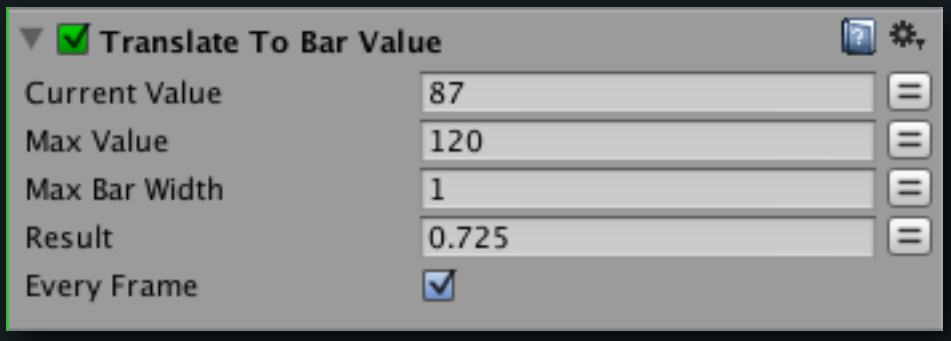
Translate To Bar Value



Translates a float to the width of a Progressbar (Usable for eg. Healthbars in NGUI).

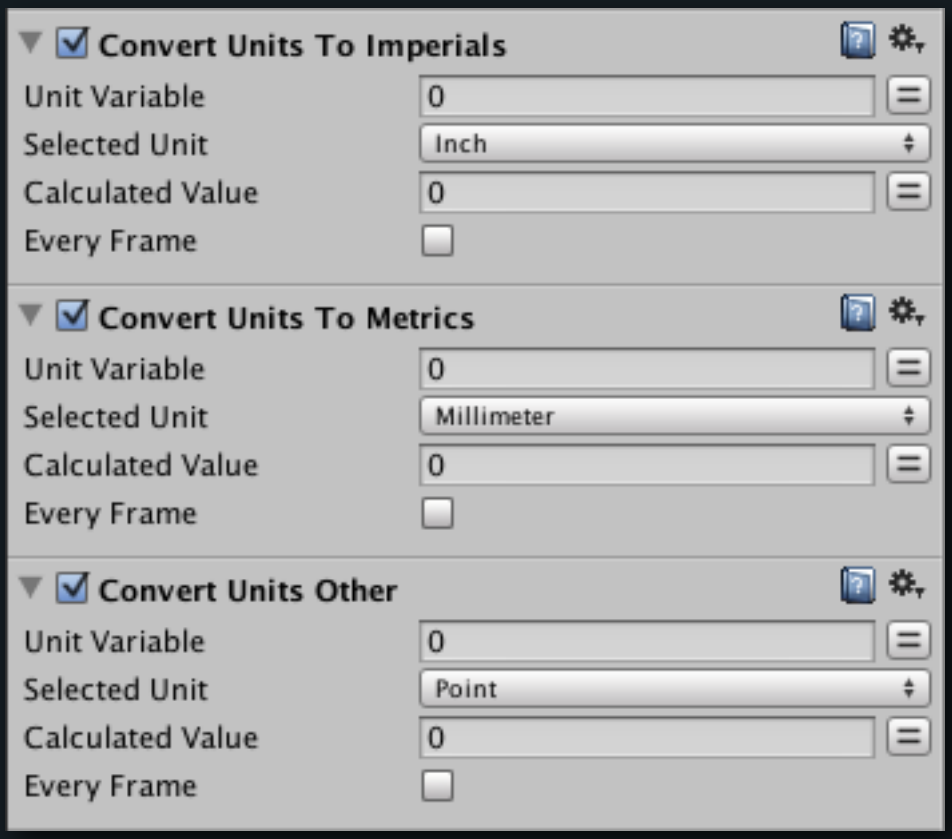
Current Value: The current value of the Bar.
Max Value: The maximum value of the bar.
Max Bar Width: The maximum value in floats, usually 1.0
Result: The result.
Every Frame: Repeats the action every frame.

Example:



If we had a health of 87, and a maximum health of 120, given that our healthbar has the maximum length of 1.0, our healthbar would have a current lenght of 0.725.

Unit Converters



Converts a float value to Metrical, Imperial and Other units.

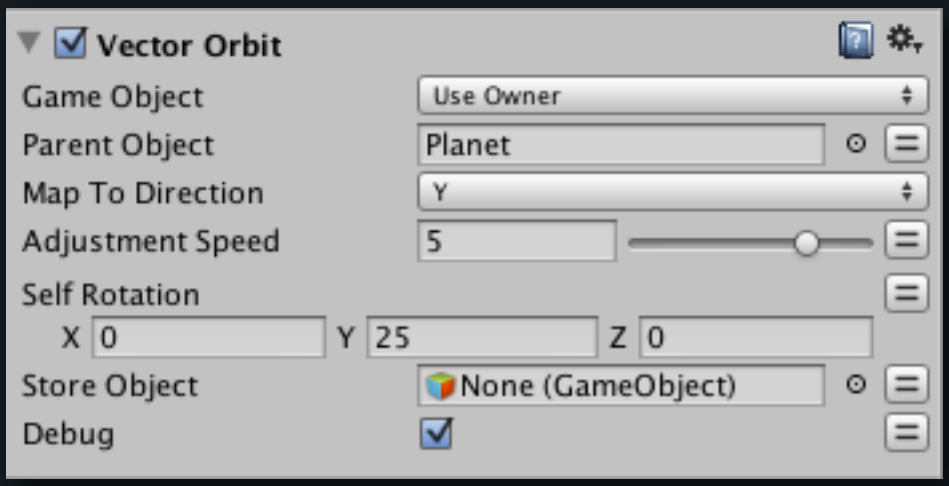
Unit Variable: The Value to be converted.
Selected Unit: The selected Unit to convert to.
Calculated Value: The resulted Value.
Every Frame: Repeats the action every frame.

Possible Units:

- Metrics:
  - Millimeter
  - Centimeter
  - Meter
  - Kilometer
- Imperials:
  - Inch
  - Foot
  - Yard
  - Mile
- Other:
  - Point
  - Span
  - Pace
  - Nautical Mile
  - Roman League

Important: In Unitys standard setting, 1 Unit in 3D Space equals 1 Meter.

Vector Orbit



Constantly orbits an object around another one, using a set direction and speed.

Game Object: Game Object to be used by the action.
Parent Object: Game Object to be orbited (will become parent).
Map To Direction: Direction of the orbit.
Adjustment Speed: Speed of orbit rotation.
Self Rotation: The current Game Objects own rotation around itself.
Debug: Shows direction and rotation informations in the scene view.

Tip:
Put the „Auto Rotate“ Action from this package onto your parent objects FSM to have the planet rotating also.

Place the Orbited and the oribiting objects next to each other with their pivots on the same Y position, like in the picture above, to prevent strange results.

Examples

You can find example scenes for all Actions in the Scenes folder, feel free to fiddle around with them.