

Measurements Documentation

version 1.0

for Unity

Documentation: [Online Documentation](#)

Support: vrchewal.com/unity/measurements

Asset Store: <http://u3d.as/16hd>

Jump To:

Introduction	How To
API	Troubleshooting
Support	

Introduction

In most games, it helps to know certain measurements for your game objects. For example, it may help to know the width of a gameobject in inches, the height of a collider or even the distance between two objects closest edges. Unfortunately, Unity does not offer much natively without writing some code.

Measurements is a simple tool that allows you to get real time information on an object's width, height and depth for its mesh, renderer and collider. You can also choose another gameobject and Measurements will tell you the distance between the two objects from their centers as well as their closest edges. All of this in any unit of measurement you like (*centimeters, meters, kilometers, inches, feet, miles, yards*).

How To

Just add the Measurements script to your gameobject.

You can choose the *Measurement Unit* for your results. The options are: centimeters, meters, kilometers, yards, inches, feet and miles. Your results will vary depending on your *Measurement Source*. The options are Renderer, Mesh and Collider. If you wish to know information about your model, choose either Renderer or Mesh. Collider is very useful when you want to know the distance between your collider and some other gameobject.

You may also select another gameobject to be the *Distance Object*. Measurements will calculate the distance between both object's centers and closest edges.

API

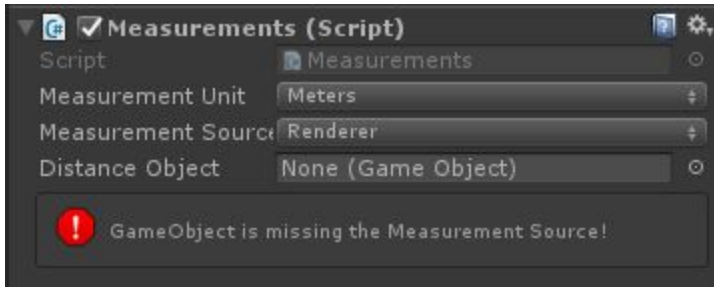
The following methods are open via the Measurements API.

setMeasurementUnit void	setMeasurementUnit(MeasurementUnits unit); Sets the unit of measurement for your results.
getMeasurementUnit MeasurementUnits	getMeasurementUnit(); Returns the unit of measurement for your results.
setMeasurementSource void	setMeasurementSource(MeasurementSources source); Sets the measurement source (Renderer, Mesh, Collider).
getMeasurementSource MeasurementSources	getMeasurementSource(); Returns the measurement source (Renderer, Mesh, Collider).
setDistanceObject void	setDistanceObject(GameObject obj); Sets the distance object for distance calculations.
getDistanceObject GameObject	getDistanceObject(); Returns the distance object used for distance calculations.
getDimensions Vector3	getDimensions(); Returns a Vector3 with x being width, y being height and z being depth. This returns the values converted to your Measurement Unit.
getDimensionsInMeters Vector3	getDimensionsInMeters(); Returns a Vector3 with x being width, y being height and z being depth. Returns the values in meters (Unity units).
getDistance Vector2	getDistance(); Returns a Vector2 with x being the center to center distance and y being the edge to edge distance. Returns the values converted to your Measurement Unit.
getDistanceInMeters	getDistanceInMeters(); Returns a Vector2 with x being the center to center distance and y being

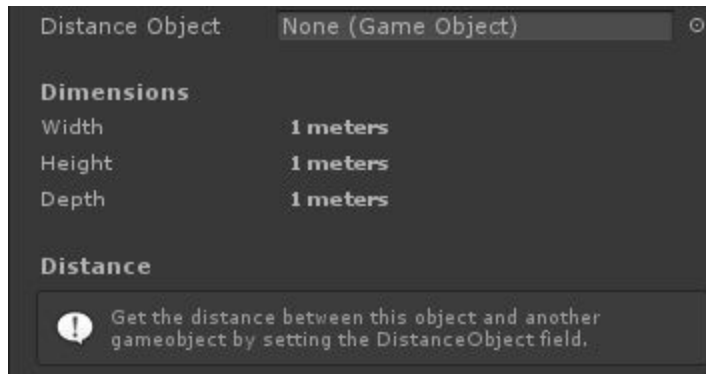
Vector2	the edge to edge distance. Returns the values in meters (Unity units).
getConversionValue float	getConversionValue(MeasurementUnits unit); Returns the conversion value from meters to your selected unit. Useful for managing your own calculations.

Troubleshooting

Oops, something went wrong. You have come to the right place.

1	<p>Problem You see a warning with the following:</p> <p>GameObject is missing the Measurement Source!</p> <p>Answer Choose a different Measurement Source or add the source that is missing. For example, if you choose Collider and the gameobject does not have a collider component, you will need to add the Collider before using that as a source.</p> 
2	<p>Problem You get 0 for a width, height and depth.</p> <p>Answer Your measurement source does not have a size. Choose a different source.</p>
3	<p>Problem You see a notification for Distance with the following:</p> <p>Get the distance between this object and another gameobject by setting the Distance Object field.</p> <p>Answer You don't have to do anything, but if you want to know the distance information for this object</p>

and another, drag another gameobject into the Distance Object field in the inspector.



Support

You can get your question answered by emailing support@vrchewal.com

Thank you for your purchase of Measurements!