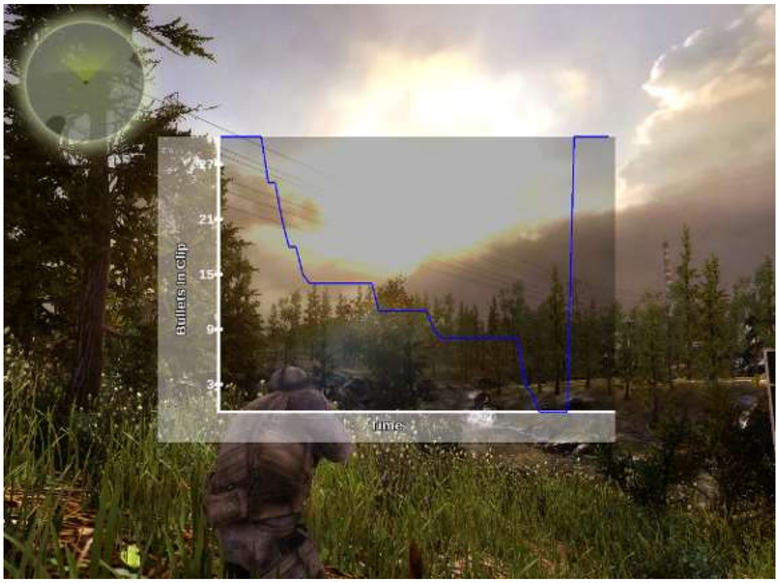
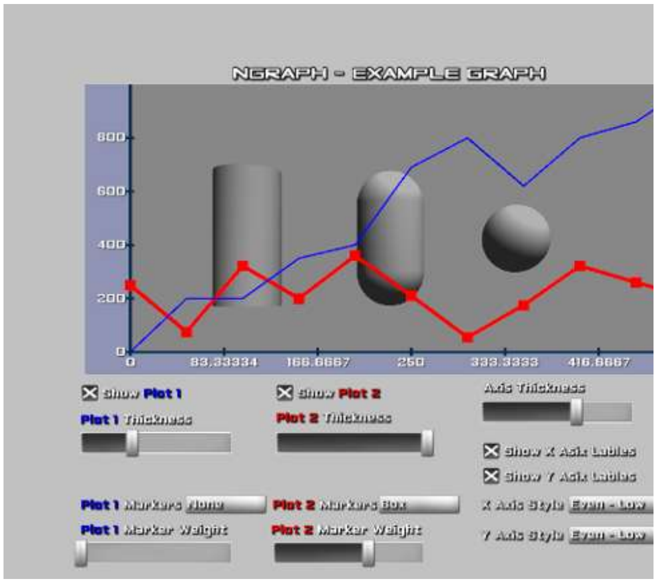
HOME UNITY ASSET STORE

GRAPH MASTER

DOCUMENTATION





☒ Show Plot 1

Plot 1 Thickness

☒ Show Plot 2

Plot 2 Thickness

Axis Thickness

☒ Show X Axis Lables

☒ Show Y Axis Lables

Plot 1 Markers None

Plot 1 Marker Weight

Plot 2 Markers Box

Plot 2 Marker Weight

X Axis Style Even - Low

Y Axis Style Even - Low

EXAMPLES

Interactive
Live Update
Equations

SUPPORT

Support is available via e-mail for those who have purchased Graph Master though the Unity Asset Store.
graphMasterSupport@niugnepsoftware.com

The example at the botom of this page is the best place to start. If you need something more raw, then the API documentation can be found here.

INTRODUCTION

Graph Master is a Unity run-time graphing tool available in the Unity Asset Store. Its purpose is to allow Unity developers to display dynamic graphs to their players. Graphs can be a great way to show the player information both during and after a game. With Graph Master you can now easily display information to the player with only a few lines of code.

Currently, Graph Master works with Daikon Forge or NGUI 3.0.4 (lower if you delete the Graph Master/Editor folder). This means that you must have Daikon Forge (available in the Unity Asset Store) or NGUI (available in the Unity Asset Store) imported into your project before you import Graph Master. There are plans to also support the new Untiy 4 native GUI system when it is released.

GETTING STARTED

Step 1: (Only for 3rd part GUIs) Prerequisites

If you are using a 3rd party GYU system, it must be installed. Install Daikon Forge, 2DTK, or NGUI from the Unity Asset Store, then install Graph Master from the Unity Asset Store. Once these packages are imported, you can create your first graph.

Step 2: (Only for 3rd part GUIs) Extract The Correct GUI System

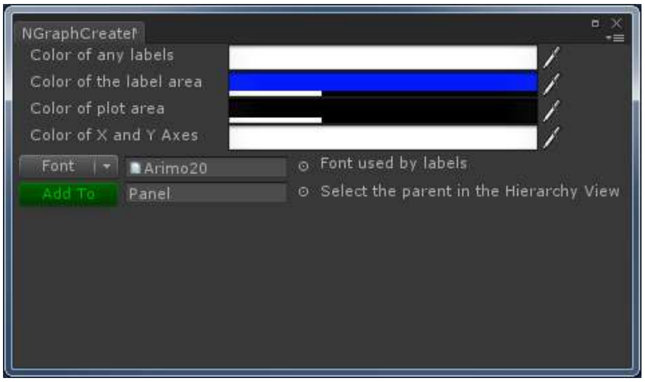
If you are using a 3rd party GUI system you will need to unzip an older version of the asset that is included. After unzipping the older version, delete the GraphMaster folder and keep the unzipped files. Inside the unzipped files, extract the Unity package named with the GUI system you own. If your project is using more than one of them, you can extract each one. **Do not extract packages that have not had thier corresponding assets imported.**

Step 3 (Native uGUI): Create a GUI - Most poeple start here

Only use this step if you are using Unity's native GUI.
This step may be skipped if you already have a GUI with a canvas in place.
Create a standard UI using the UI menu to get a Canvas in place. (Menu: GameObject → UI → Canvas)

Step 4: Create A Graph

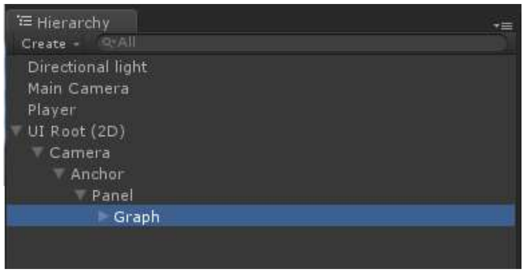
- Open the Graph Master "Create New Graph" Wizard. (Menu: Window → Graph Master → New Native Unity Graph)
- Fill out the form and click the green "Add To" button. (The "Add To" button will not be green until all the required items in the form are filled in - including the font.)



Color of any labels ⬜
Color of the label area 🟦
Color of plot area ⬛
Color of X and Y Axes ⬜
Font ▾ Arimo20 ○ Font used by labels
Add To Panel ○ Select the parent in the Hierarchy View

Step 5: Add Some Data

Create a custom script and add it to the newly created "Graph" game object. An example of a script that plots a data series can be found below:



```
Hierarchy
Create ▾  Q▾All
Directional light
Main Camera
Player
▼ UI Root (2D)
  ▼ Camera
    ▼ Anchor
      ▼ Panel
        ▶ Graph
```

InteractiveExampleGraphScript.cs

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

// This MonoBehavior expects to be attached to the sam
// that a Graph Master Graph is attached to.  This object
// you created it with the Graph Creation Wizard.
public class InteractiveExampleGraphScript : MonoBeh
 GraphMaster.UGuiGraph graph;
 GraphMaster.UGuiDataSeriesXy series1;


 void Awake() {
  // Capture the graph
  graph = gameObject.GetComponent<GraphMaster.U


  // Setup the graph - this can be done in the editor as v
  graph.setRanges(0, 500, 0, 1000);


  // Create the data we want to plot
  List<Vector2> data = new List<Vector2>();
   data.Add(new Vector2(0, 0));
   data.Add(new Vector2(50, 200));
   data.Add(new Vector2(100, 200));
   data.Add(new Vector2(150, 350));
   data.Add(new Vector2(200, 400));
   data.Add(new Vector2(250, 690));
   data.Add(new Vector2(300, 800));
   data.Add(new Vector2(350, 620));
   data.Add(new Vector2(400, 800));
   data.Add(new Vector2(450, 860));
   data.Add(new Vector2(500, 1000));


  // Add a X/Y Plot the the graph and capture the plot ar
   series1 = graph.addDataSeries<GraphMaster.UGuiDa


  // Apply our data to the plot.
   series1.Data = data;
 }
}
```