

## Grid Editor Tool


Brandon Coffey

Welcome to the Grid Editor Tool!

This is a level designer tool for 2D games that makes creating and storing grid-tile based maps easy. The tool comprises of a Object Designer and a Level Designer.

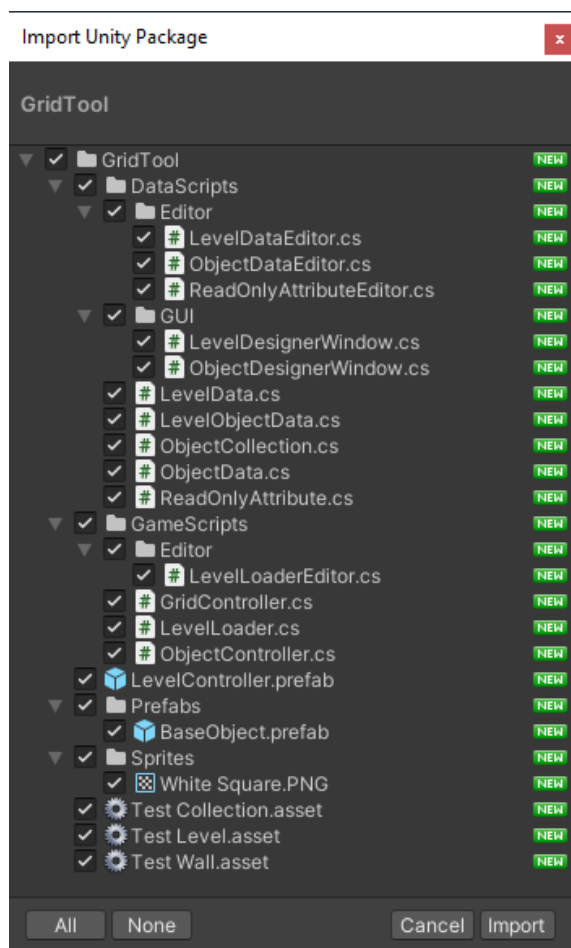
## Installation

First, you should have the below file downloaded.

 **GridTool.unitypackage**

This is a unity package, which is used to install various tools, prefabs, and scripts into Unity.

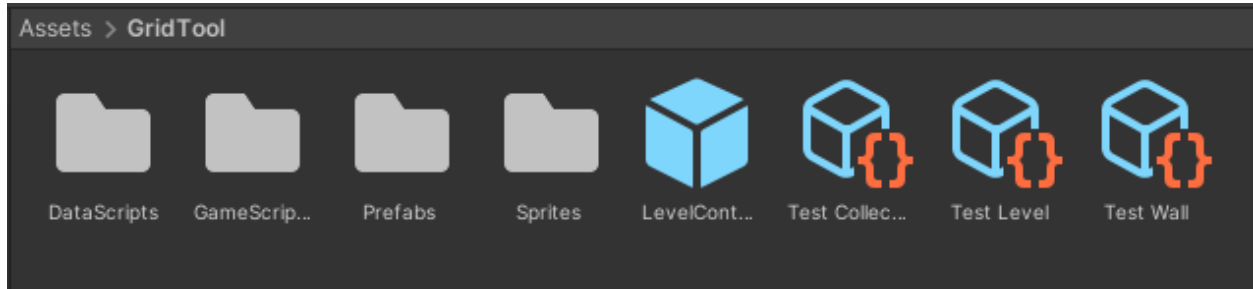
Once Unity is open (2020.3.0f1 or above) to a new or existing project, double click on the GridTool.unitypackage and it will open the following window in Unity.



Select import with everything checked. The Grid Tool is now installed!

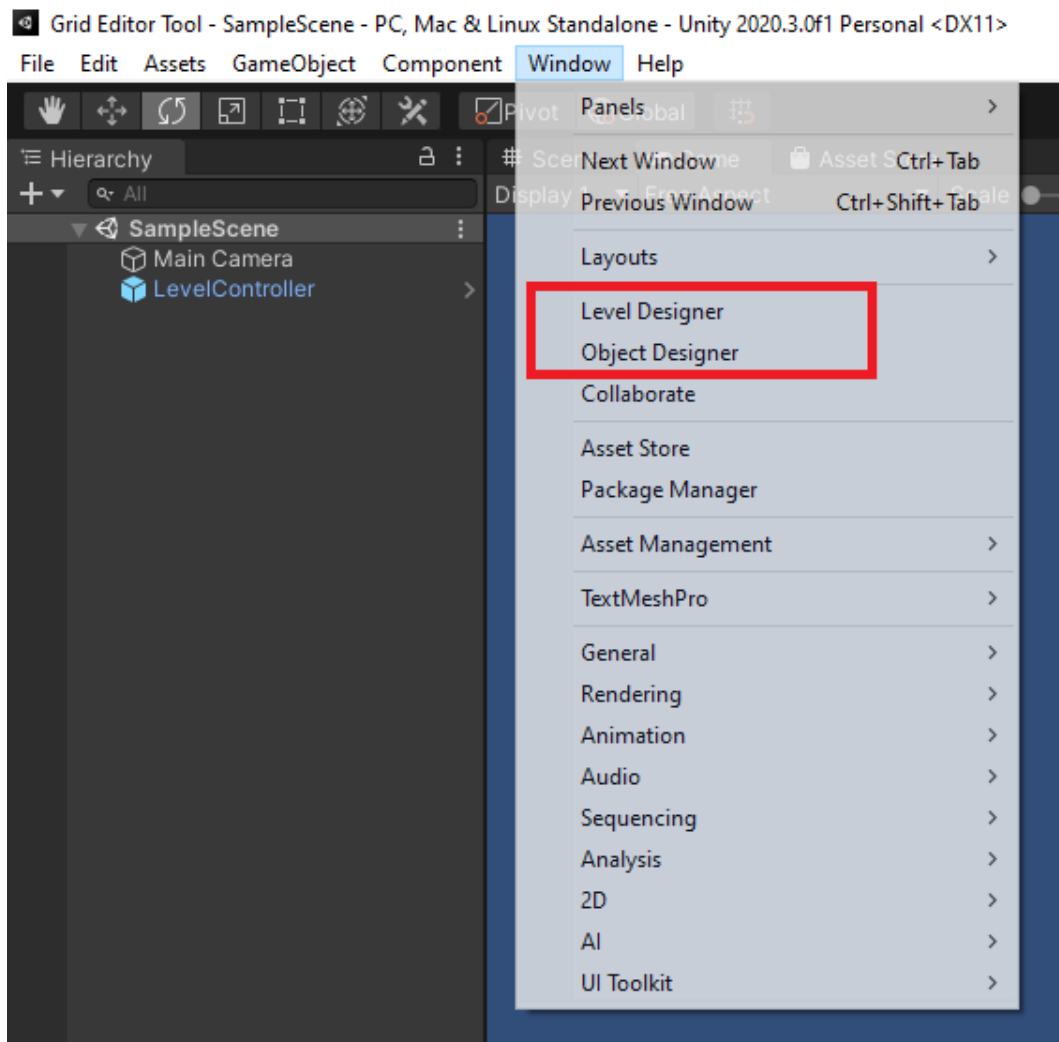
## Assets

After installing, there will be a new folder under assets called GridTool. The LevelController prefab and three scriptable objects on the right. These are the demo objects, feel free to test around with them. The Level Controller should be brought into the main scene where you will load levels into.



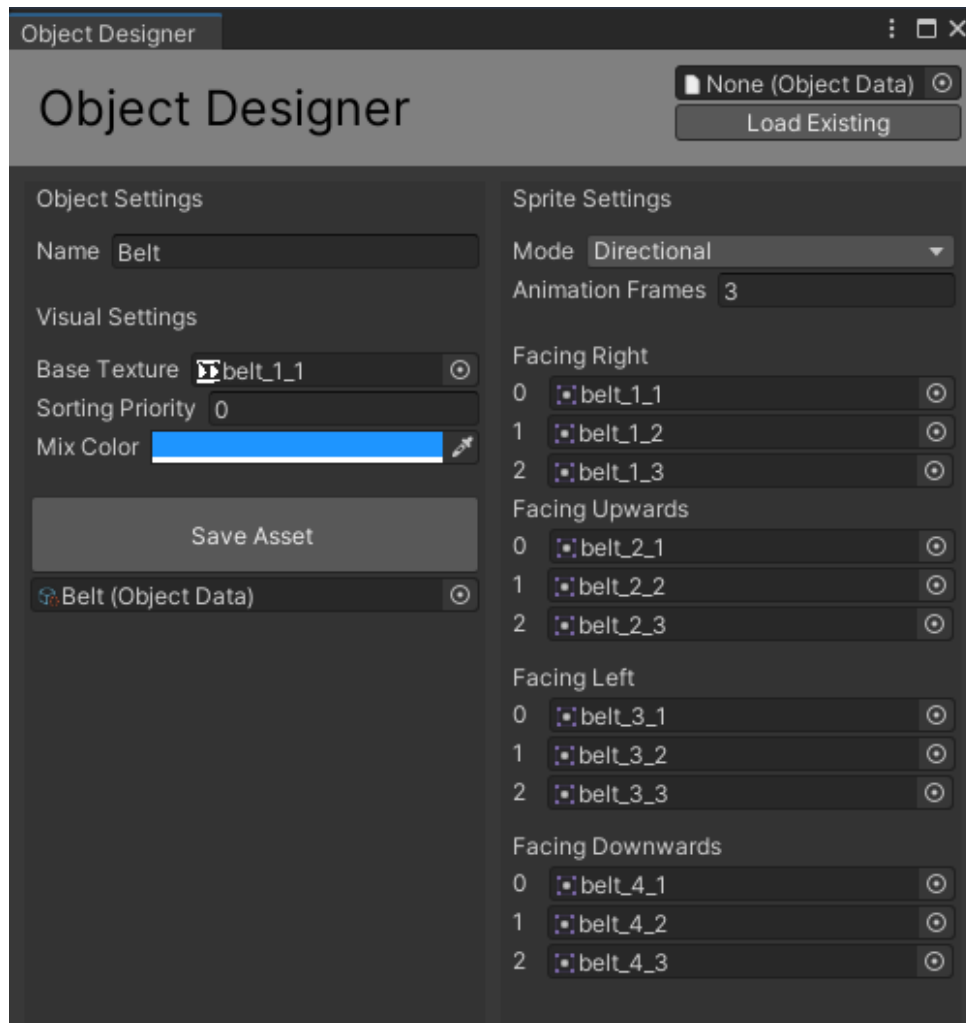
## Opening the Designer Windows

To open the Level Designer or the Object Designer, go to the top of Unity and select the Window dropdown. You should see Level Designer and Object Designer as shown below.



## Object Designer

The Object Designer is the main tool for creating 'objects', which are essentially a more advanced Sprite. They contain two sets of settings, as shown below.



On the top right, you will see an Object Data field and a Load Existing Button

- Drag any object data asset (or use the target icon) into this spot.
- Then, click Load Existing, and it will populate the Object Designer with the given Object Data asset and allow you to make changes and then save.

### Object Settings

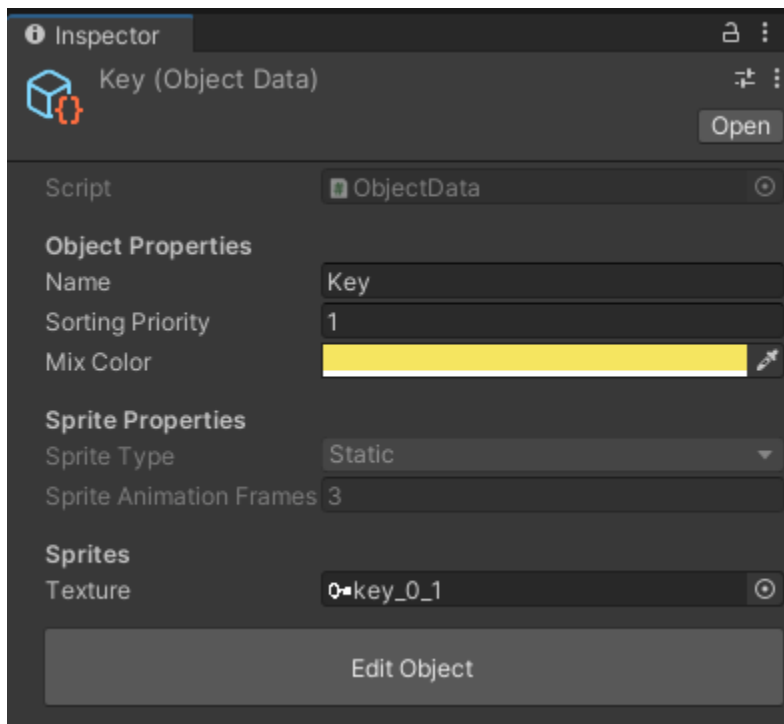
- Name. The name of the object. Displayed in the Level Data files
- Base Texture. Should be any of the sprites. Used to display texture for Level Designer.
- Sorting Priority. Sets the Sprite Renderer's sorting priority.
- Mix Color. If desired, a color can be multiplied on top of any given sprites.
- Create Asset (Shown if creating an asset instead of editing one).
- Save Asset (Saves the asset to the referenced object below).

## Sprite Settings

- Mode. Either Static or Directional. Sets the sprite to be directional, either up, down, left, or right. These will be displayed in game if the object is rotated.
- Animation Frames. Dynamically Sets the below sprites to have 1 or more frames. This will be actively used when the objects are brought into the scene.
- Sprites. Depending on the above settings, it will display a list of sprites (named) that should all be filled with various sprites. Drag and drop or use the target icon.

## Object Data

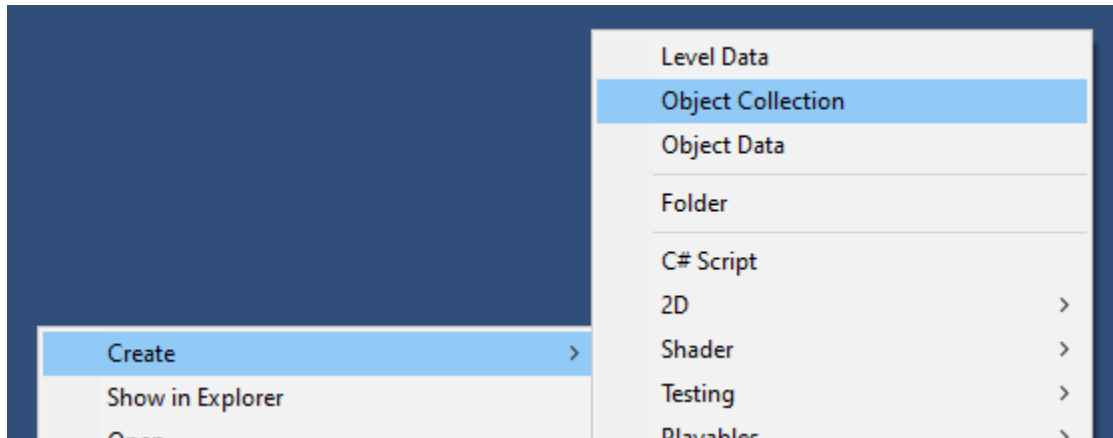
After creating or saving from the Object Designer, an Object Data will be saved into the assets folder (A popup will allow you to choose which folder to save to). This object contains all the information from the designer (some are locked).



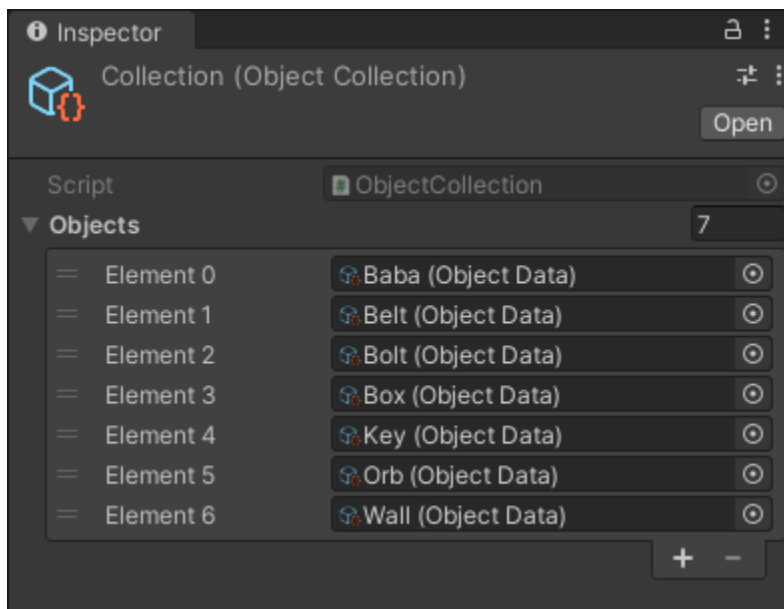
At the bottom of the Object Data, there is a large 'Edit Object' button that will open the Object Designer with the given object loaded. This allows for quick edits and updating data. Make sure to click save before closing.

## Object Collections

After creating a few objects, they will need to be gathered up into a collection before they can be used in the Level Designer. To create a new collection, right click in the project window (empty space in the assets folder) and go to Create > Object Collection. This is also available under Assets > Create > Object Collection at the top of the screen.



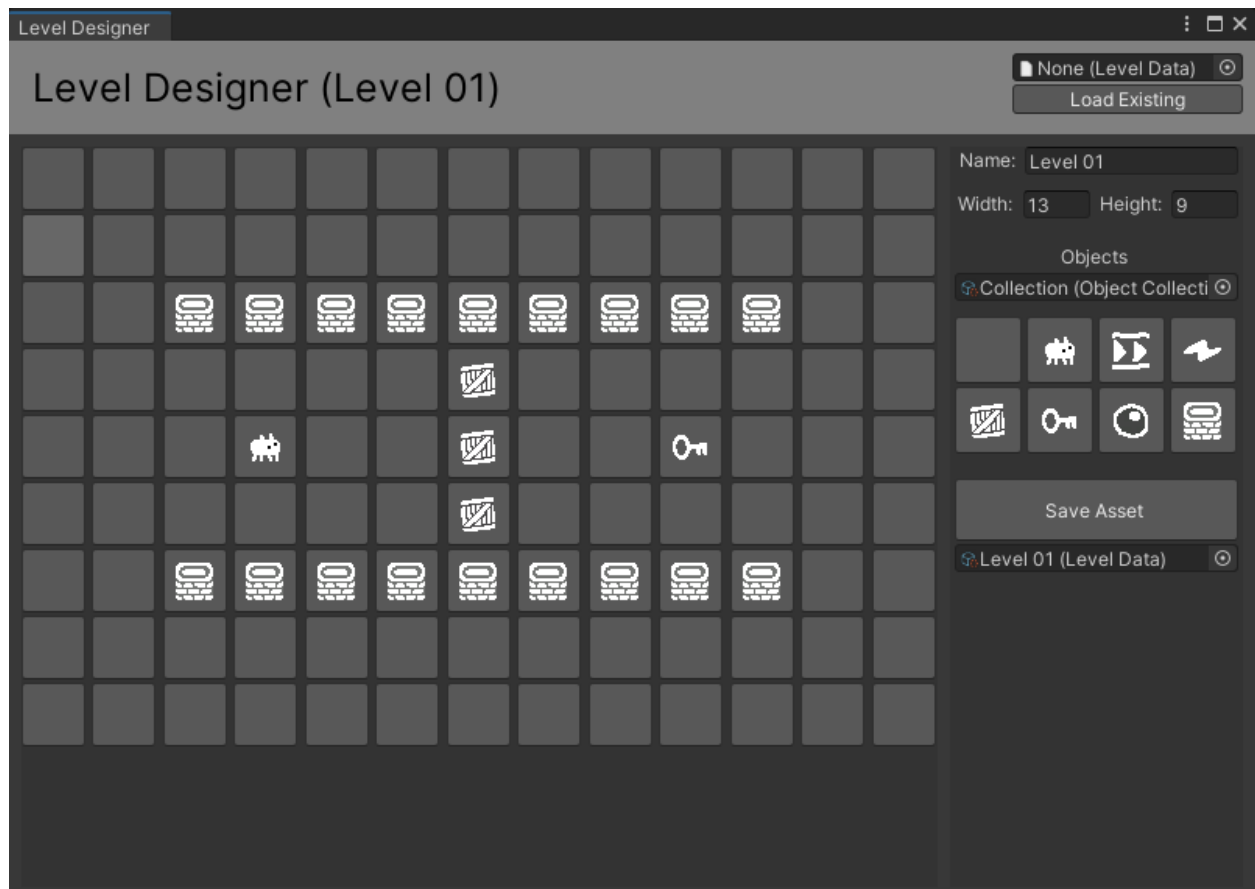
Once creating this object, you can simply drag all of the created objects into the Objects List on the Collection Object. A simple way to do this is to lock (top-right button) the inspector and select all assets, then dragging them over instead of one by one.



As a side note for this documentation, I am using sprite assets from the game Baba Is You. These are not included in the tool and I do not claim to own them, they are only for better visualizing the tool in action

## Level Designer

The Level Designer is a large window with a grid on the right. This will only appear this way after adding an Object Collection into the field on the middle right side.



Starting on the right side, there is a level name to identify this level. This is also repeated in the title of the Level Designer.

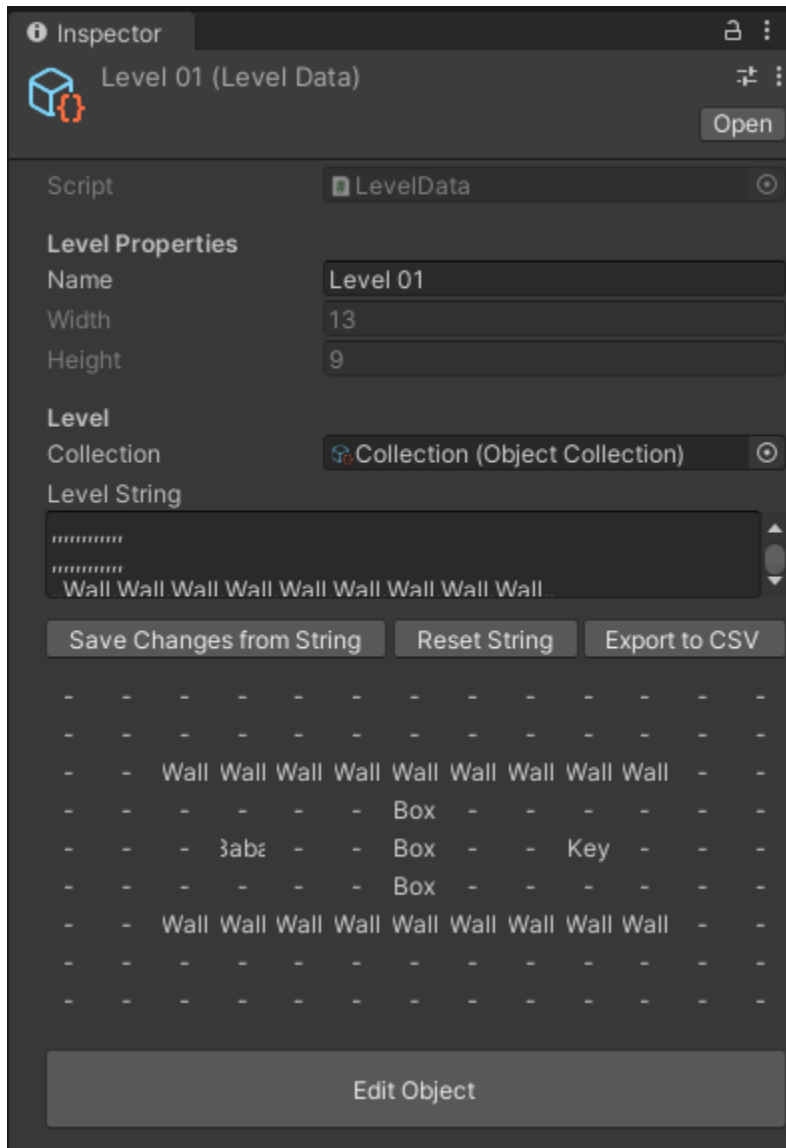
Below that, there are two integer fields for the width and height of the level grid. They will wait for you to finish typing a number and then dynamically update the grid size. Be careful, as this is a destructive action and cannot be undone.

Below that is the field for the Object Collection. Every object inside of the collection will be shown beneath this field as their own buttons. These buttons can be selected to allow the user to then click on buttons on the left side (the big grid) and 'draw' in that selected object into the scene.

Then, there is the button for creating or saving the asset, depending on whether the user is editing or creating a new level. This works the same way as the Object Designer.

## Level Data

Once creating a level from the level designer, a 'Level Data' object will be created, as shown below.



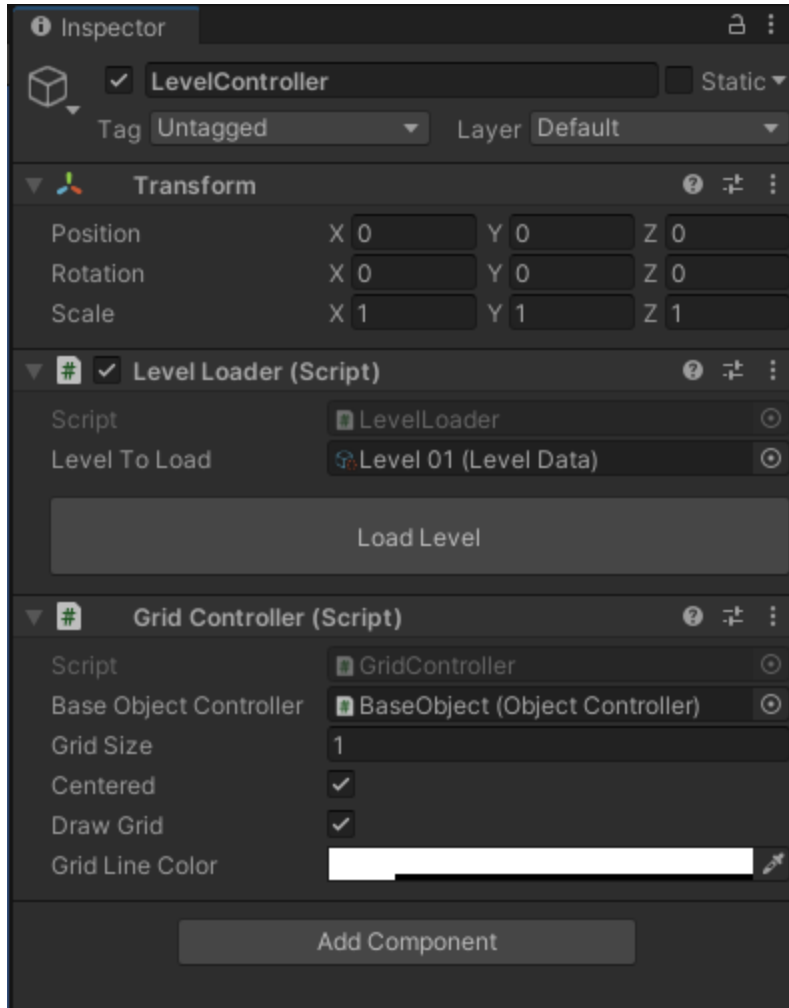
At the top, there are a few exposed properties (some locked) that show the name, width, and height of the level. Below that there is the collection of objects reference.

The Level String is a CSV equivalent of the Level, that can be exported (far right button) and copy-pasted back into the string box. Save Changes from String will override the below preview of the level with the CSV string values in the box. Reset String will rebuild the CSV string from the level.

The 'Edit Object' button will open up the Level Designer with the selected level loaded, for easy access and editing.

## Level Loader

Moving out of the data objects, the Level Loader is the main script for building a Level Data into an actual level shown on screen. This is done by selecting the Level to Load on the object and clicking Load Level at runtime.

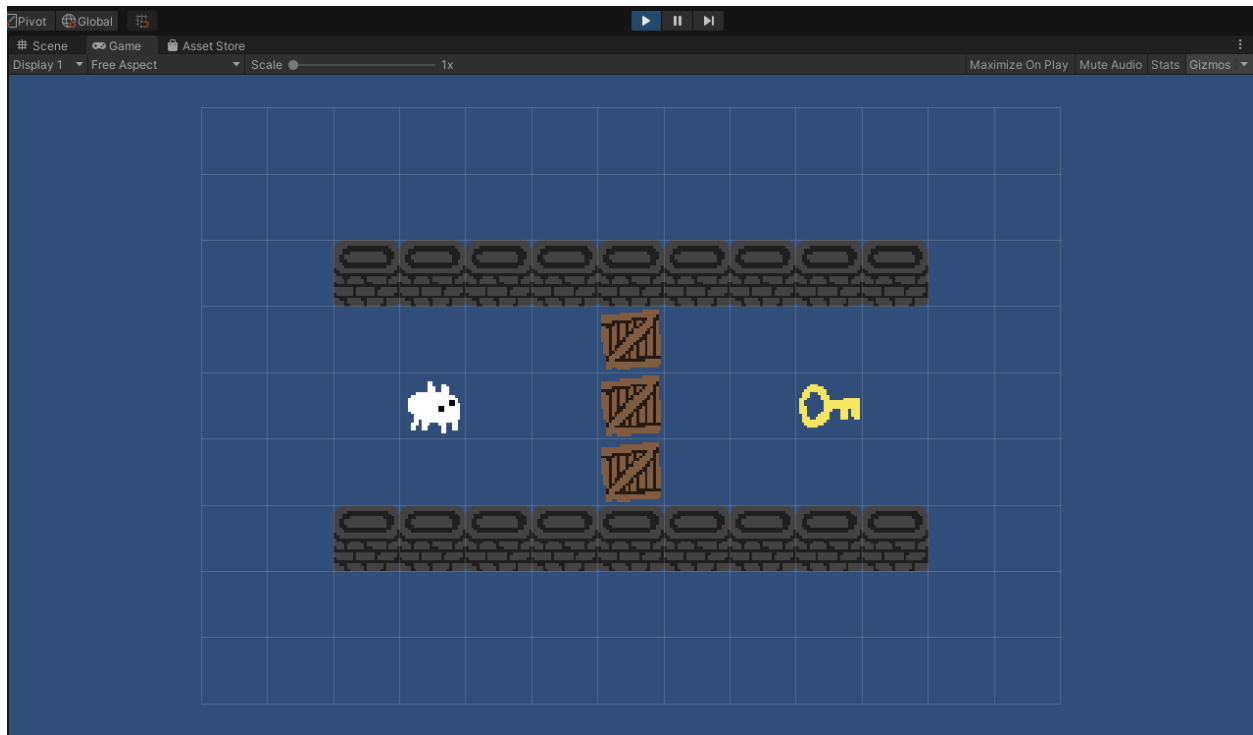


The Level Loader references a Grid Controller to create the grid. Feel free to specify a grid size and color. The Centered checkbox chooses whether to draw the grid centered at the transforms position or not. This can be disabled by the Draw Grid checkbox.

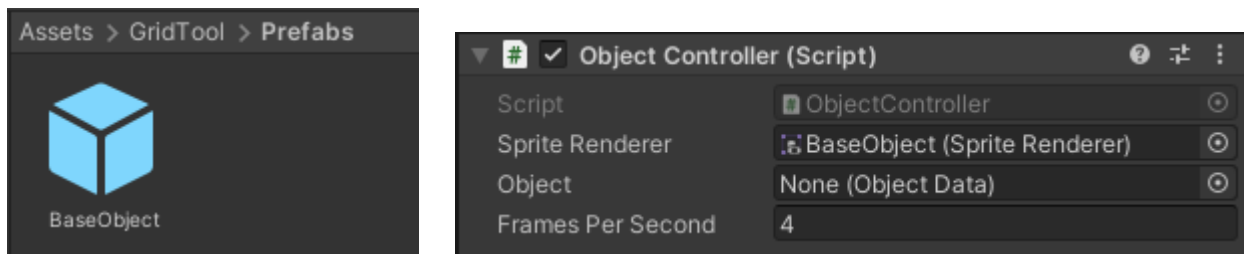


## Unity Preview

After setting up the level loader or just using the given LevelController prefab, a grid and the level objects will be built into scene. These will have animations if the animation frame is set to more than one and will multiply the mix color onto the sprite. All the below sprites are from a game called Baba Is You (I do not own rights to this game and the sprites are not included in the tool), they are just for tutorial purposes. The sprites in the game are all drawn as white sprites, and I have chosen the colors to be multiplied onto them at runtime, as shown below.



If you would like to edit the default animation speed (frames per second), it is stored on the BaseObject located in the prefabs folder. This is referenced by the Grid Controller.



Thank you and happy level designing!