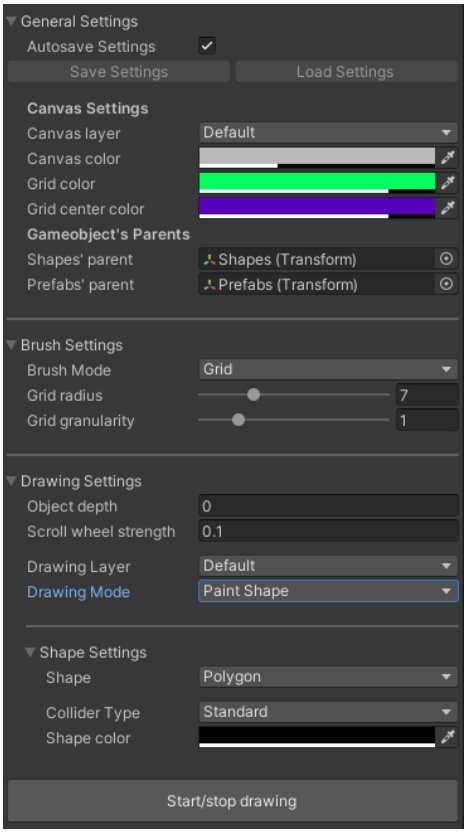
A World Prototyping Tool

By Raúl Martín ~ 11/2021

Introduction



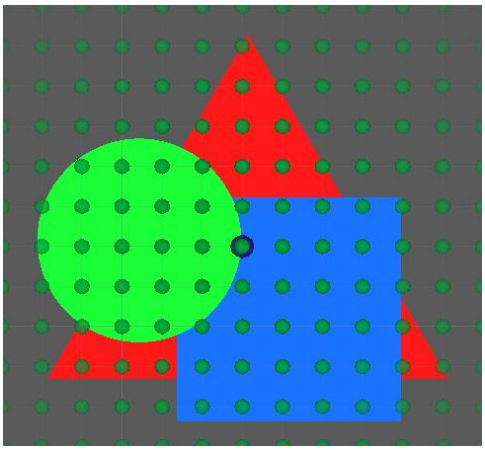
This is the offline documentation of the World Painter 2D tool. It explains in detail all the features and settings of the tool, as well as some examples and important considerations to take into account when using the tool.

If you prefer explanatory videos, refer to the package page in the Unity Asset Store.

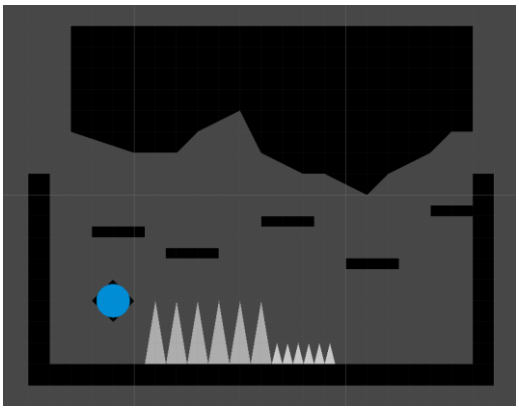
Table of Contents

1. Introduction to the Tool

1.1. Summary



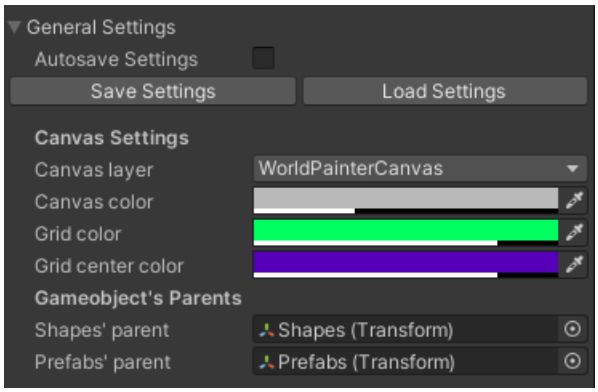
World Painter 2D is a tool designed to make Unity's 2D world prototyping much easier. This tool grants the user the ability to paint the scenery by drawing simple shapes with the mouse. Among these shapes, the user can draw rectangles, circles, or polygons. This list may be expanded in the future.



The tool also grants the ability to clone objects with the click of a button. This functionality can be helpful to, for example, designing a level main structure by cloning platforms, or clone a repetitive pattern in the level, such as grass objects or particles.

1.2. Features

The tool consists of an editor window with the following foldouts:



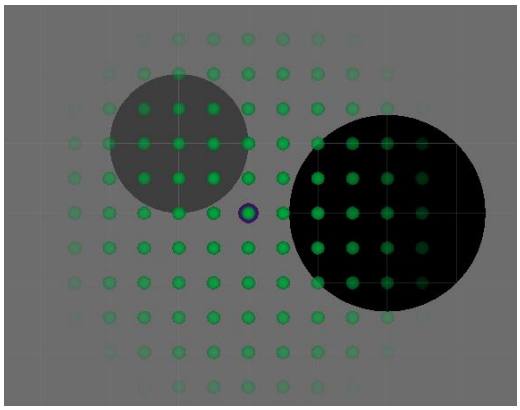
General Settings

This section contains all the settings that are thought to be changed only a few times. Among these we can find the following:

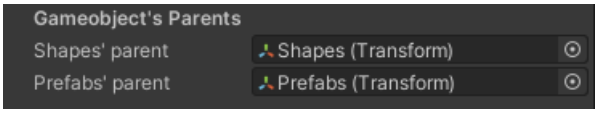
- **Saving:** These settings are related to storing and saving all the current tool settings. These settings can be saved automatically, or manually by using the buttons.
- **Canvas layer:** This setting is used by the tool to determine in which layer the canvas is located. It is recommended to create a unique layer for the usage of this tool. Nonetheless, it is not mandatory, but it can lead to some precision issues with the mouse.

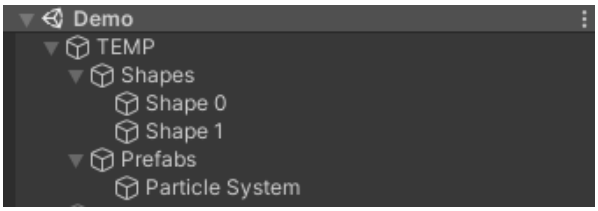
- **Canvas and grid color**: Used to change the color of the canvas and the grid. Alpha channel is ignored for the grid colors.

  It is recommended to have a *low alpha* value for the canvas color to achieve the following effect. Note the leftmost circle is behind the canvas and the rightmost one is in the foreground. This can be deduced by the shade of black each circle has.



- **Gameobject's parents:** These settings are used by the tool to store the new drawings and prefabs created while drawing as a child of the given gameobjects. Here you can see an example, in which the new drawn shapes are stored in an empty gameobject called Shapes and same goes with the cloned prefabs.



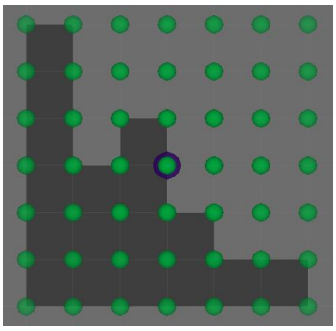


Brush Settings

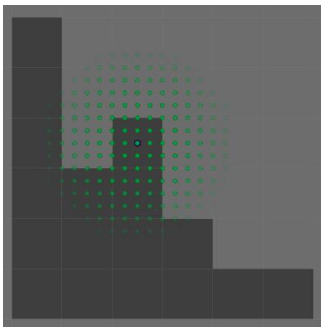
This simple section allows the user to change the drawing mode between Free and Grid. In Free mode the user can draw anywhere on the scene. In Grid mode, the shapes are restricted to the grid settings.

The *grid radius* defines the radius of the circumference around the mouse in which the grid will be drawn.

The *grid granularity* defines the step of the grid. In other words, how many units are between each dot on the screen.

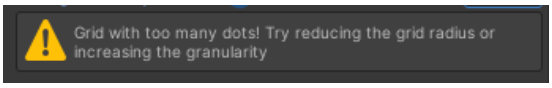


Shape drawn with grid granularity set to 1

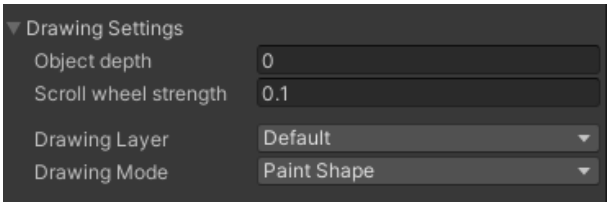


Grid granularity set to 0.25 over the previous shape

Note that drawing too many dots might reduce the performance of the editor, so the tool will not draw any dot if the number of dots is too high, and a warning will be displayed.



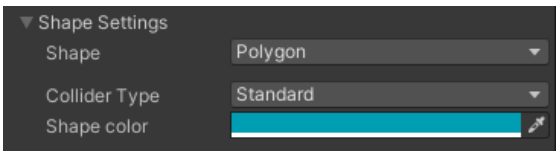
Drawing Settings



In this section the user can configure the more prone to change settings of the tool.

- **Object depth:** The z-axis value on which prefabs may be instantiated or shapes may be drawn. This can be useful in the 2D scenery creation because by using a perspective camera you can get a free parallax effect by having objects on the background or foreground. However, make sure to have an adequate layer and order in layer setting for the rendering to be done correctly.
- **Scroll wheel strength:** This setting changes by how many units the scroll wheel action modifies the object depth while drawing. More on this in section '2. How to Use'.
- **Drawing Layer:** In which layers the prefab or object is to be drawn.
- **Drawing Mode:** In this dropdown the user can choose between the two drawing modes the tool offers. Painting shapes and instantiating prefabs or copies of objects in the scene. Depending on the value of the drawing mode setting, one of the following two settings foldouts may appear:

❖ **Shape settings**

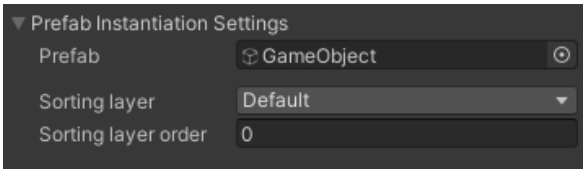


This foldout is used to configure the drawing shape settings:

- **Shape:** The shape can either be Polygon, Rectangle or Circle.
  - **Polygon:** This shape can have as many vertices as the user requires. The polygon must be convex to be rendered properly.
  - **Rectangle:** Define a rectangle by setting the two opposite vertices of a diagonal.
  - **Circle:** Define a circle by clicking on the screen to set its center and then the radius.
- **Collider Type:** It can be a standard collider or a trigger. NOTE: Triggers will display a color while being drawn, but as soon as the shape is finished, the color will disappear. This color can be brought back by selecting the shape on the hierarchy and enabling the Mesh Renderer component.

- **Shape Color:** The color the shape will render.

❖ **Prefab Instantiation Settings**

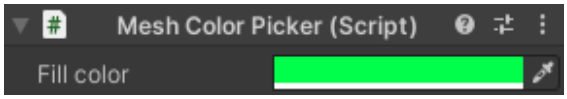


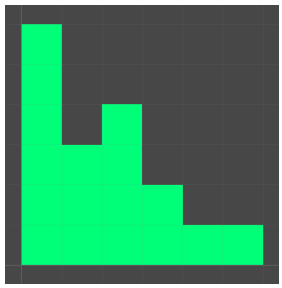
This foldout is used to configure the prefab instantiation settings:

- **Prefab:** Either a scene gameobject or a prefab. This is the object that will be cloned by the tool.
- **Sorting Layer:** The sorting layer in which the object will be instantiated.
- **Sorting Layer order:** The order value in the sorting layer.

1.3. Extra Features

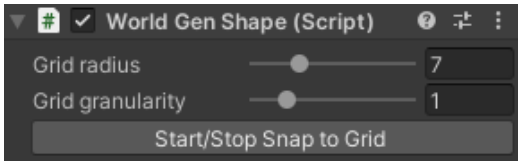
❖ **Color picker:** All the shapes created by the tool will include a component called *Mesh Color Picker*, which grants the user the ability to change the color of an already drawn shape, in chase they change their mind. The color will be changed from the editor.

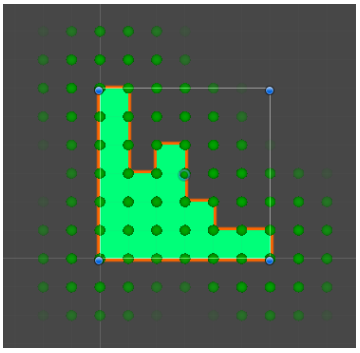




Here we can see how we changed the color of a shape used as an example in the section ‘1.2 Features ~ Brush Settings’.

❖ **Grid snapper:** Every shape drawn with the brush settings set to grid will contain a component called *World Gen Shape*, which grants the user the ability to move the shape in the grid, in case the user wants to relocate it.





A small grid will be drawn around every vertex of the shape, so make sure to have a configuration which draws a small amount of dots, as complex shapes can become really laggy.

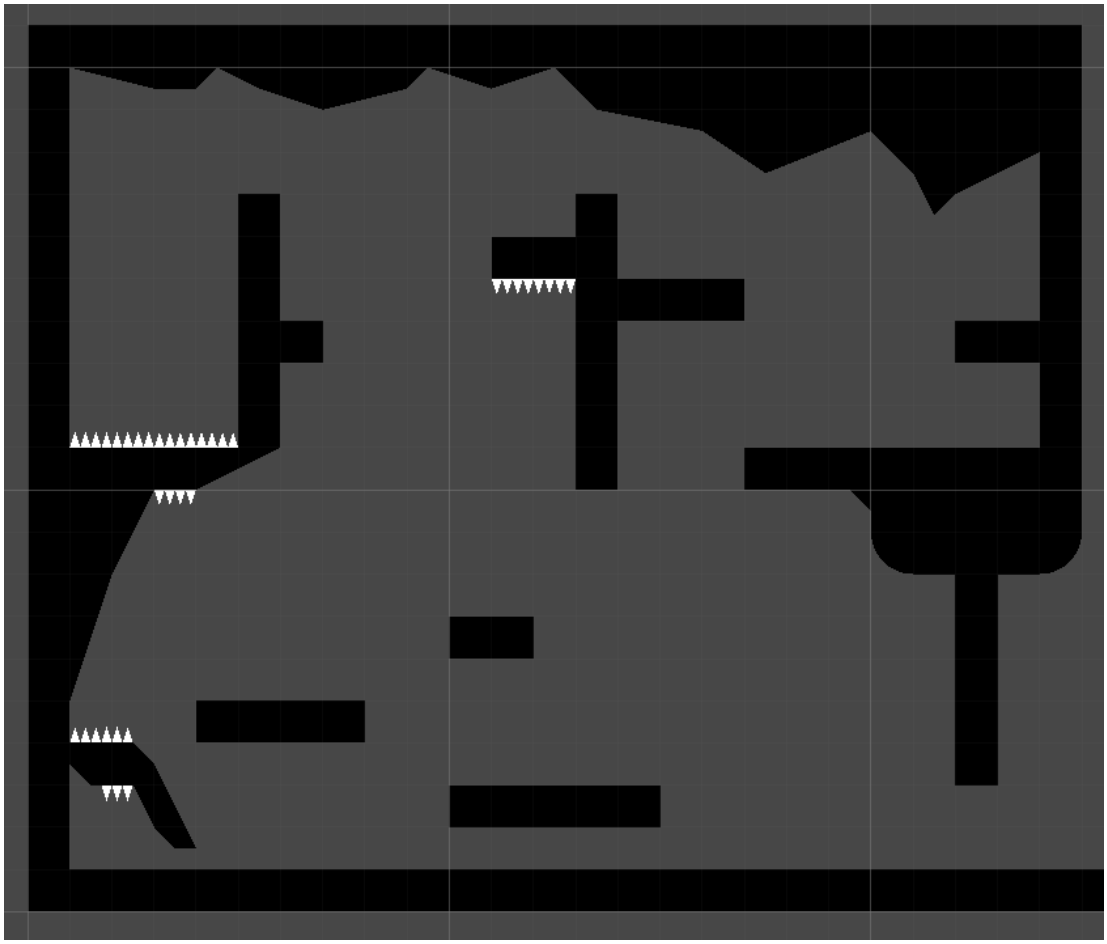
NOTE: The start/stop button enables or disables the snapping of every **selected** shape drawn by the tool. Make sure to stop snapping once you have finished relocating your shapes.

- **Undo/Redo:** The tool supports the undo/redo actions of unity when creating new instances of objects both in the prefab mode and the painting mode (only once the shape is considered as finished).
- **3D view:** The tool supports the scene view to be in 3D mode. It can be tricky to understand the depth of the scene but can be useful in some situations in which the user manipulates the depth of the objects, as stated in section 1.2.

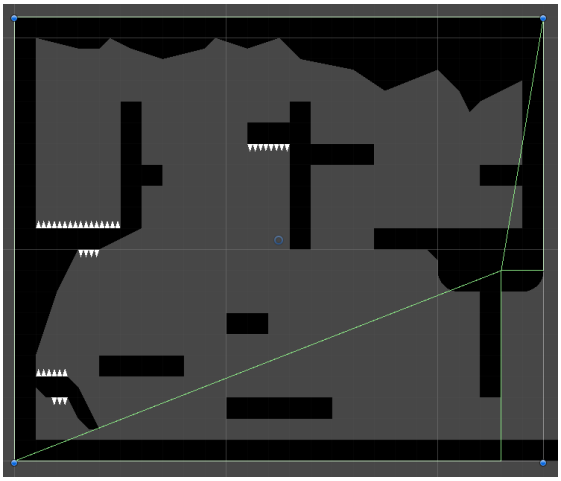
1.4. Usage examples

In this section we will see some usage examples of the tool.

Here we have an example room that has been designed in 5 minutes. It used all the shapes and demonstrates a few techniques that can be used when prototyping with this tool.



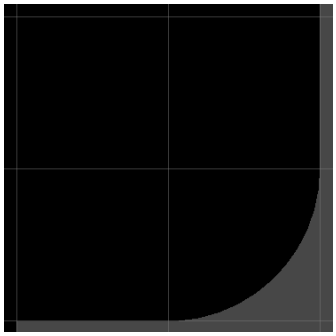
In the previous image we can see how the ceiling has an organic feeling of a cave-like scenery by using the polygon shape.

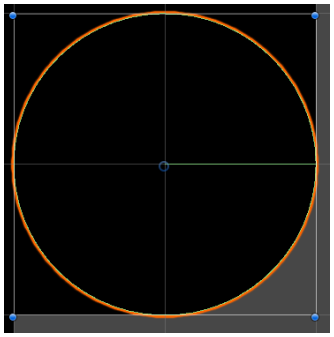


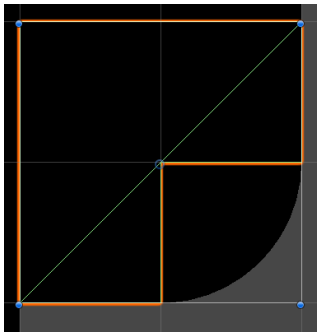
We have also defined some walls and platforms so the player can jump around the level. We can also define damage areas by creating a triangle and replicating it.

By using the trigger functionality, we can define a shape inside the area that plays music when the player is inside of it. It can also be used along with the Cinemachine package to define a camera confine area.

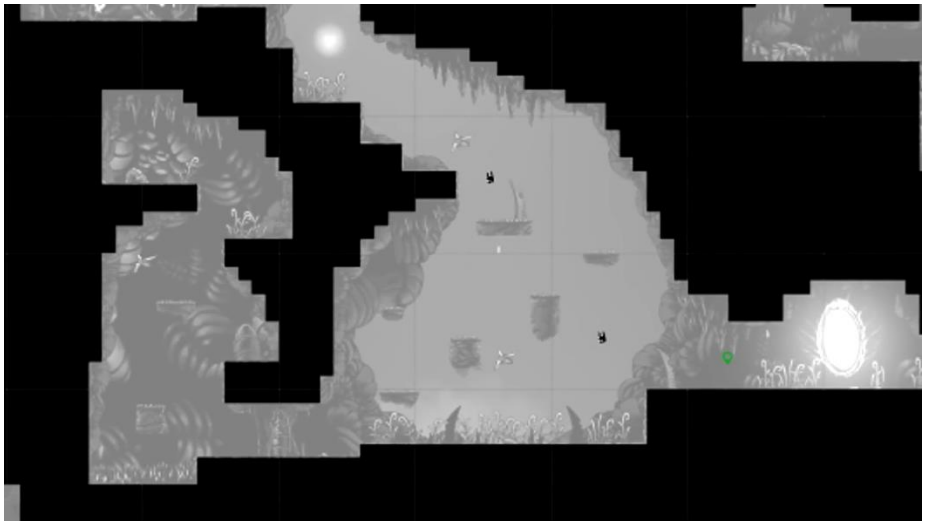
If we prefer a more smooth look on our scenery we can combine circles with rectangles or polygons.







Here we have another example of usage. PLEASE NOTE that this screenshot is from the video game Hollow Knight and belongs to Team Cherry.



This tool was NOT used by Team Cherry, but I got the inspiration from this screenshot to develop the tool. It represents a good example on how the tool may be really handy for level design.

2. How to Use

Configuring the Tool

Once you have installed the package via the package manager, open the window. It is located in Window > 2D > World Painter 2D.

Before starting using the tool, make sure to set a Canvas Layer and a Drawing Layer. You can also set the gameobjects parents to store the newly created shapes and prefabs. More information in section 1.2.

General Controls

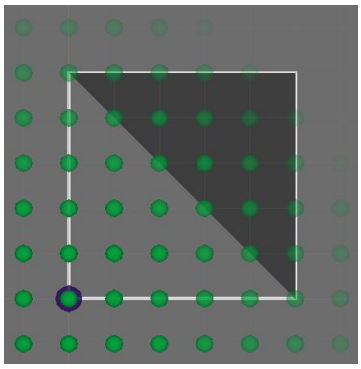
- **Ctrl + Mouse Scroll Wheel:** This grants the user the ability to modify the object's depth on the go while drawing, without the need to modify the *Object Depth* value by hand. The Scroll Whell Strength setting changes how many units the *Object Depth* setting is added or subtracted for each step of the wheel scrolled.
- **Left Click:** It behaves differently depending on the drawing mode.
    - **Painting shapes:** Adds a vertex to the current shape. If it's the last vertex needed, it finishes the shape.
    - **Instantiating prefabs:** Creates a new instance of the prefab stored in the *Prefab* field.
- **Right Click:** This action only takes place when painting shapes. If the shape is not finished, it cancels the creation of the shape. A shape is considered not finished when it has not enough vertices defined. For Rectangles and Circles, when the user has only clicked once. For Polygons, when the user has only clicked twice.

Drawing Simple Shapes

Once you selected *Paint Shape* in the drawing mode, choose the shape and color you want to draw.

- **Rectangles:** To draw a rectangle, click on the scene to set the first vertex. Click once more to set the opposite vertex of the rectangle and finish the shape.
- **Circle:** To draw a circle, click on the scene to set the center. Click once more to set the radius of the circle and finish the shape.
- **Polygon:** To draw a polygon, click at least 3 times on the screen to define 3 vertices. Once the shape is finished, right click to stop drawing the polygon and set it as finished.

Make sure that when you are drawing the polygon it is convex (make sure there are no edges overlapping). Failing to do so might result in a weird rendering of the mesh.



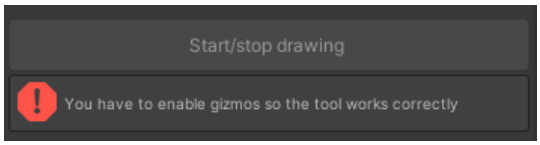
Remember also to NOT close the loop i.e., set the first and last vertices in the same position. This leads to one triangle not rendering, as it can be seen in this screenshot.

Instantiating Objects

To start instantiating objects, simply set drawing mode to *Instantiate Prefab* in the Drawing Settings Foldout and select a gameobject from the scene or prefab from the project folder into the Prefab slot. Then click the Start/Stop drawing button and enjoy cloning.

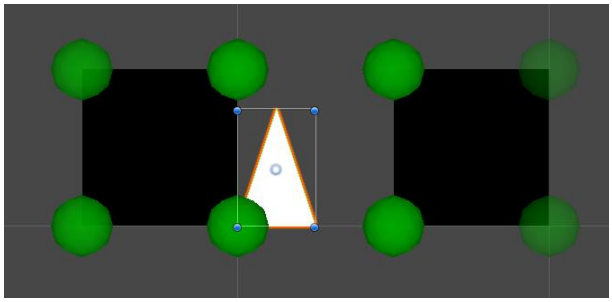
3. Fundamental Considerations

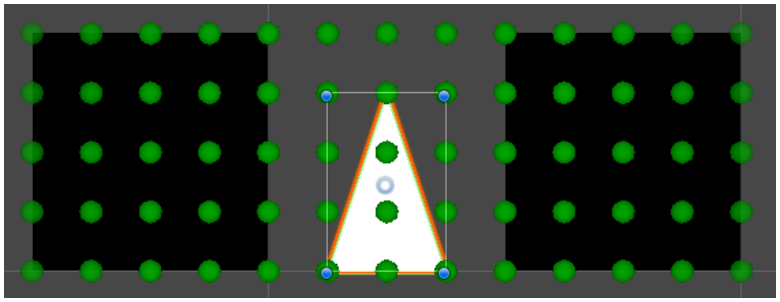
There are several considerations to take into account when using the tool. Not following them may result in an improper operation of the tool.



- **Gizmos:** The tool *requires* gizmos to be enabled in the scene window in order to work correctly.
- **Scene Window:** The tool is thought to be used with a single scene window open. If you use more than one window some punctual issues might appear. Consider closing one scene view if the issue is annoying.
- **3D Canvas:** When using the tool with the scene view in 3D mode, make sure to draw in the canvas from the correct side. Z-Axis must be towards $+\infty$.

- **Small shapes snapping:** If you defined a shape with a grid with small granularity and you are trying to snap it to a bigger granularity, it might not snap correctly. Here we can see an example. We created this spike with a granularity of 0.125, but when we try snapping it to a grid of 0.5 only one vertex will be in position. If you want to snap it centered, try increasing the granularity snapping.





- **Grid disappearing:** There is a known issue with unity where sometimes graphics disappear or become really small. This is an issue with the zoom level. To reset it select any object from the hierarchy and press F to focus it. This should fix the issue.
- **Prefab instantiation:** The tool is programmed to remember which prefab to clone with the save and load functionality. This allows closing the unity editor and when it is opened again, it will let you clone the last prefab you selected. To ensure it has a reference to such object, the tool clones the object into the Resources folder of the Package. If you find any issue related with saving or loading taking too long it is because it is copying the prefab. Consider not using heavy prefabs with this tool' s instantiation mode. Also note that some objects might not be fully cloned or might be missing some components.

If you find any bug, closing the editor window and opening it again might fix it. Any problem please notify me. My contact information is in the Unity Asset Store page of the package.

Other smaller details about the tool:

- **WorldPainter2D Tool gameobject:** The tool uses a gameobject to execute its logic and display the grid. This gameobject appears always the editor window is open. To remove it make sure to close the tool. It will disappear automatically.
- **Canvas object:** While drawing, a new canvas will be created by the tool, but it will be hidden in the hierarchy. It will be automatically deleted when not drawing or the editor window is closed.