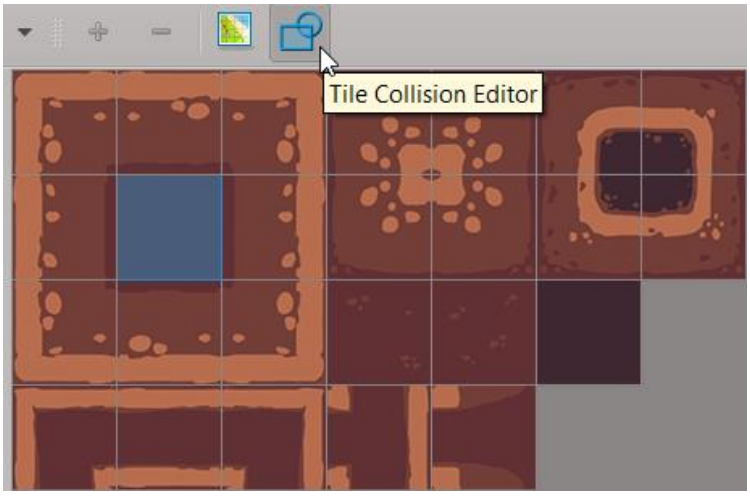
Modifying the imported level

Do not modify the imported scene directly. Instead, instantiate it or create a new scene inherited from it. If you do modify the source scene, all your changes will be lost next time you reimport.

How to edit your tileset's collision shapes

Starting with Tiled 1.0, tileset files are separate from maps. To edit the tileset's collisions, open it, click a tile to select it, and click on the "Tile Collision Editor" icon in the toolbar above the viewport.



Tile collision editor icon

Draw collision shapes in the newly opened tab. To snap to the grid, check the corresponding option in the View menu -> Snapping -> Snap to Grid.

Tip: you can copy and paste collision shapes between tiles. iIn the collision editor, press S to activate the "select object" tool, click the shape to select it, Ctrl C to copy and Ctrl V to paste it.

Warning: Godot only supports 1 collision shape per tile.

Object layers: supported types

Any shape you draw in Tiled is called an "object". The addon treats them as collision shapes by default, but it can also turn them into navigation paths or light occluders in Godot. To do that, select the shape, and in the properties editor, fill one of 2 keywords in the "Type" field:

1. `navigation`
2. `occluder`

Note this doesn't work with ellipses, as Godot doesn't support those shapes for navigation and light occluders. Also, if you use an object layer, you can't set its content as `navigation`.

Custom properties

Add custom properties in Tiled

Read the metadata in Godot

Post-import script: modify the imported scene

You can run any script with a post_import method. The plugin will pass it the imported scene, so you can append level elements built in Godot.

Add a Godot node to the level. Use it to load level elements that were designed in Godot:

```
extends Node

var my_node = load("res://my_scene.tscn")
func post_import(scene):
  var new_node = my_node.instance()
  # Set the node's owner to the current scene so it can be added to it from t
he editor
  new_node.set_owner(scene)
  scene.add_child(new_child)
  return scene
```