

Time Game Tiled Guide

Programs:

- Tiled: <http://www.mapeditor.org/>
- Tiled2Unity: <http://www.seanba.com/tiled2unity>

Important Custom Properties

- unity:tag – Applies tag to **Tile Layer** or selected object in **Object Layer**. Tag must exist in the Unity project for this to be applied.
- unity:scale – Sets scale of Tiled Map when imported into Unity
- unity:isTrigger – Turns colliders into triggers when applies to a **Tile Layer** or an object in the **Object Layer**
- unity:sortingLayerName – When applied to a **Tile Layer** sets which sorting layer it should appear on

Tag Key:

- Background: Apply to selected **Tile Layer** to prevent bullets from passing through them. Needs to be applied to Walls

Note: Gates and Connectors should not be placed in the Tiled Map. Only apply the Tags for them to the map so our Unity Project can find those tags and create the Gate/Connector at the location.

Gate Tags:

Connector2LinkRight
Connector1LinkBotRight
ConnectorNoLinkLeft

- **Connector Naming System:**
 - **Connector:** Designates object as a connector
 - **# of Links {2,1,No}:** Tells how many directions 'wires' spread from the Connectors
 - **Link:** I don't this is useful anymore but that's how the names are set now. Goes after # of Links
 - **Keyboard Orientation {Bot, Left, Right}:** Comes immediately after **Link**. Sets the orientation of the keyboard. Bot means the keyboard is facing the bottom of the screen. Right means the keyboard is facing the right of the screen. Left means the keyboard is facing the left of the screen
 - **Wire Orientation {Bot, Left, Right, Top}:** Comes immediately after **Keyboard Orientation**. Only applies to 1 Link Connectors. Tells which direction wire should be facing relative to the screen.
- **Example:**



Connector1LinkLeftTop



Connector2LinkRight



ConnectorNoLinkLeft



Connector1LinkBotLeft

GateH
GateV

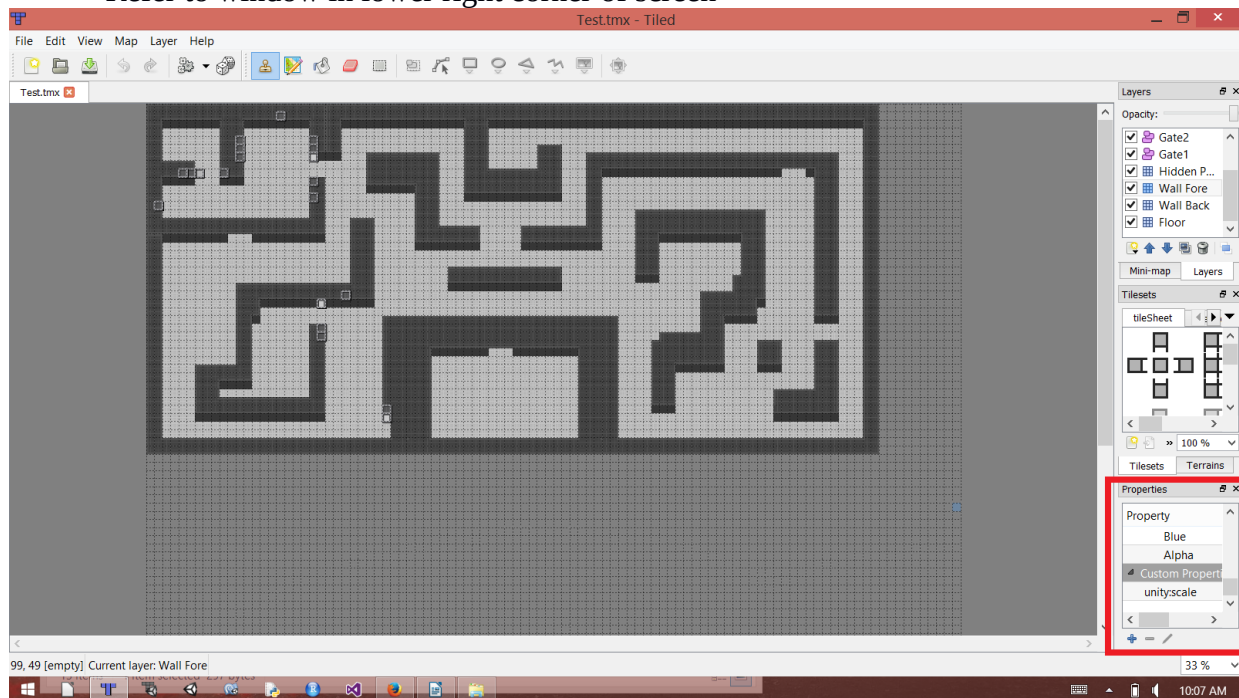
- **Gate Naming System:**
 - **Gate:** Designates object as a Gate
 - **Gate Orientation {H, V}:** Tells if the gate goes horizontally (H) or vertically (V)

How-To

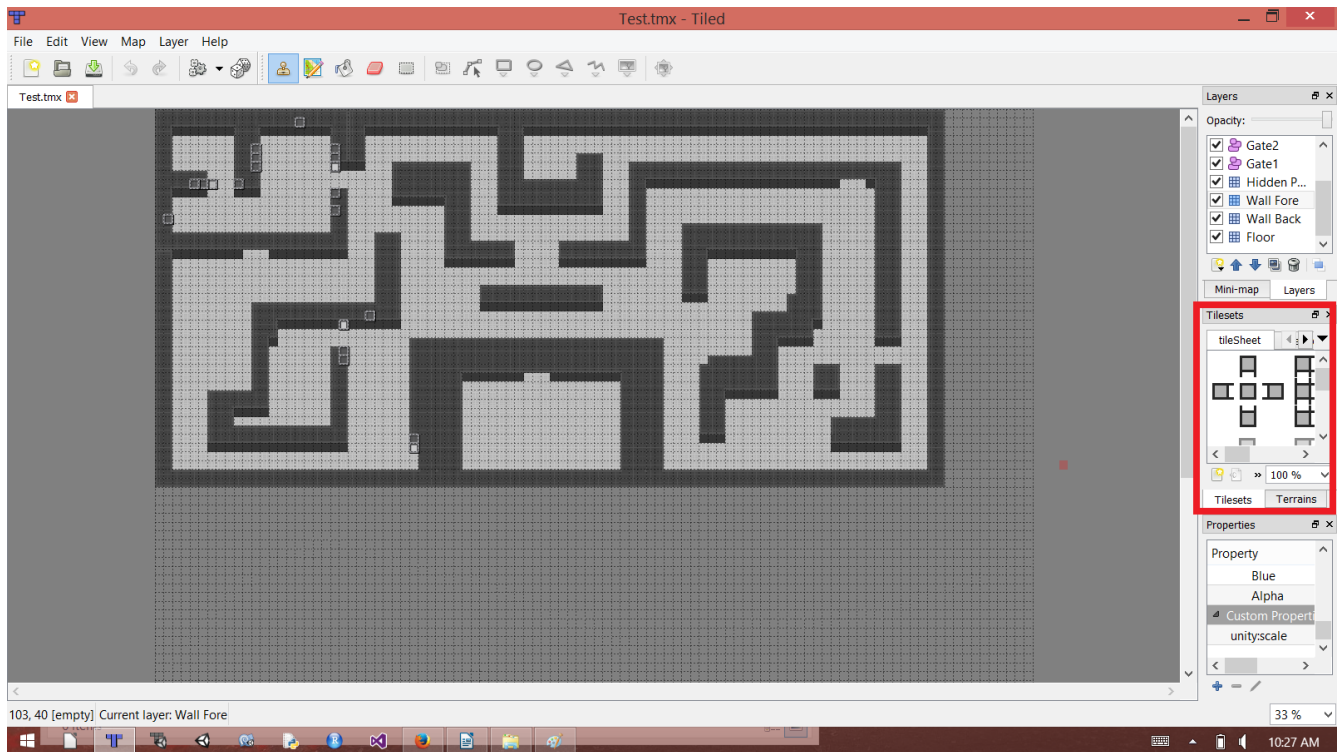
Tiled

First Steps:

- Enable Snap To Grid: Go to View → Snap to Grid
- Set the Scale:
 - Go to Map → Map Properties
 - Refer to window in lower right corner of screen

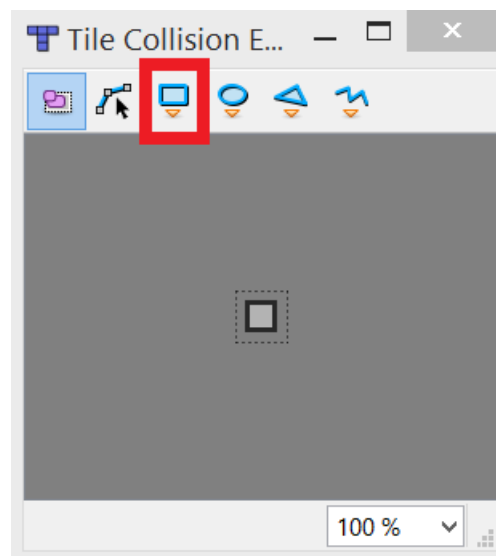


- Press the + sign.
- Name the Custom Property **unity:scale**
- Input **0.01** in the input box right of **unity:scale**
- Import the tilesheet
 - Drag tileSheet.png into the Tilesets window. Do this twice if you want to make Secret Passages later as you'll need a copy of tiles that look the same but do not have collision (Explained further in document)
 - Press Ok. The tileset should be automatically split
 - If not, makes sure Type: Based on Tileset Image
 - Width: 32 px
 - Height: 32 px



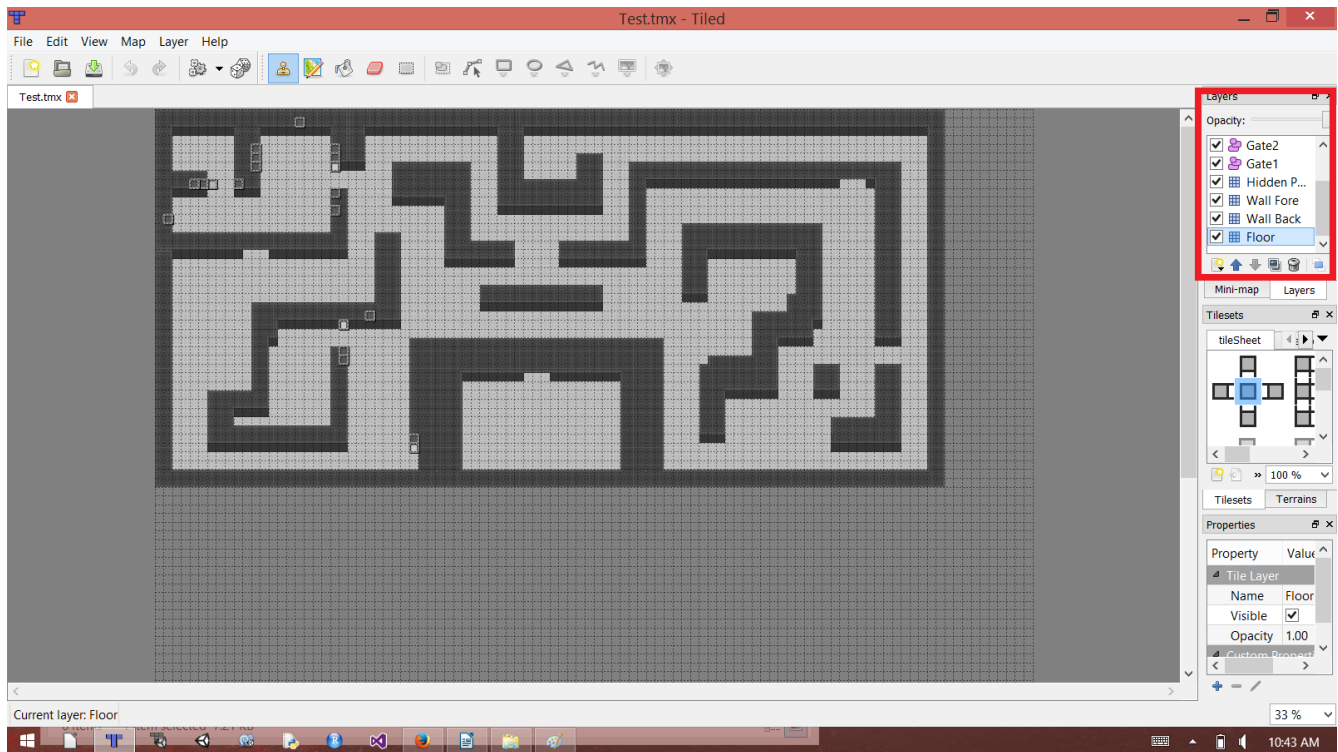
Setting up Collision for ALL copies of selected tile:

- Select a Tile which you want to apply collision on from the Tileset (indicated in the above image).
- View → Collision Editor



- Select the Rectangle Tool indicated by the red box. Select the tile with the Rectangle Tool to set the collision around the tile. Close the window once done. If you made a mistake and want to delete the box collider, select the pink button and click on the collider box you want to delete.

Setting Up Basic Layers:

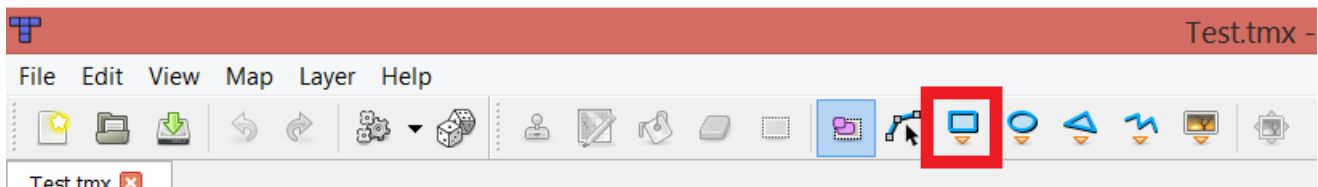


- There are 3 basic layers we need for the project: **Floor, Background Walls, Foreground Walls**. The actual names are optional. Additional **Tiled Layers** for organization purposes or otherwise are optional.
- **Floor:** Anything the player can walk over. This should include the blank wall tiles as the player can walk over them and they should not interfere with the bullet collision. Does not need to be tagged.
- **Background Walls:** Should contain MOST of the wall tiles not ALL. Whatever should be in the foreground should not be in this layer. Must be **tagged** in **Unity** as **Background** and given collision in order to prevent player and bullets moving through the walls.
- **Foreground Walls:** These are the sections of the wall that the player can go 'behind'. Must have no collision on these tiles and be given the Custom Property **unity:sortingLayerName Foreground**. These should always be the 'Top' Wall tiles forming the wall that points to the top of the screen.
- **Creating a Layer**
 - Right Click window shown in the red box and Add Object/Tile/Image Layer
 - For the Walls/Floors, use the Tile Layer
- **Tagging 'Background Wall' Layer**
 - Do not apply from **Tiled**. The Tag doesn't apply to the correct features apparently. Must tag the **Background Walls** within **Unity** by selecting the child object of the layer called **Collision**. Set the child **Collision** tag as **Background**
- **Setting Sorting Order for 'ForegroundWall' Layer**
 - Select the **Foreground Wall** Layer in the Layers Window
 - Create a new Custom Property

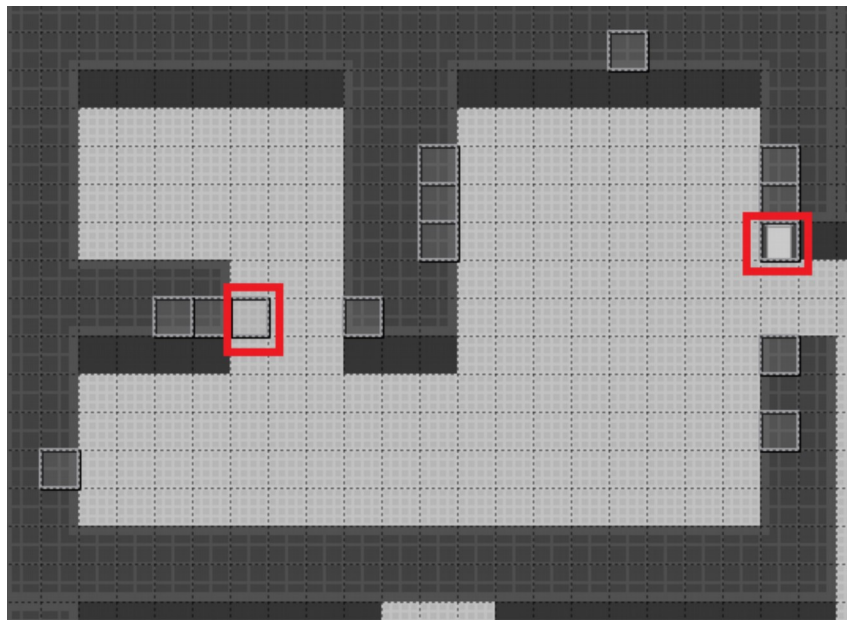
- Name the Custom Property **unity:sortingLayerName**
- In the **unity:sortingLayerName** property, input **Foreground**

Making Gates:

- Each **Gate** with its set of connectors needs its own Object Layer. Just name each Object Layer as Gate1, Gate2, Gate3.... Gate[n].
- Each Object Layer must have an object with the **unity:tag** of either **GateV** or **GateH** to indicate which gate should go over the object in the object layer. Only one object should be labeled unless you want multiple gates to open for the same set of **Connectors**
- Each Object Layer for a **Gate** must have at least one **Connector** object tagged that will open the Gate
- **Creating Connector and Gate Objects**
 - Create an object layer and name it Gate[number]
 - Select the Object Layer



- Select the Rectangle Tool indicated above
- Create 1 x 1 squares on map for where you want object to be placed (within the specified object layer)



- **Gates** should not be placed directly on collidable walls but beside them as indicated above
 - The Red Box to the left is how you place a Horizontal Gate
 - The Red Box to the right is how you place a Vertical Gate
 - **MAKE SURE** to add and apply the Custom Property **unity:isTrigger** and input **true**.

By default, all objects created in the Object Layer have colliders that must be turned into triggers if we don't want the player to collide with them in the actual map

- To apply individual tags to each “box” or “object” create in the Layer, select the pink tool and click on the desired box (or SHIFT click and select multiple desired boxes to apply properties to all of them) and give it the Custom Property **unity:tag**. From there you may tag it as a **GateH, GateV, ConnectorNoLinkBot, etc...**

Changes to Document:

- **February 26:** Background tagging information updated
- **March 3:** Changed background information and scaling info