Contents

1 frequent courses (Ablauf-Checklisten)

1.1 course of map editing

1. edit map

2. save map: Strg+S or top left File⟩Save, in your map folder that shouldn't be in the game folder

3. export map, top left File⟩Export as⟩file ending has to be .json or .tmj, level maps go into the game folder to assets/maps/, non-level maps go into the game folder to assets/maps/non-level-maps

4. if the game is running, stop it by closing the command line window (black thing that only has text)

5. start game by double clicking START.bat

1.2 course of creating a map

1. open Tiled

2. click on File⟩New⟩New Map

3. select:

   - Orientation: Orthogonal
   - Tile layer format: Base64 (uncompressed)
   - Tile render order: Right Down
   - Map size: fixed, with whatever sizes you want
   - Tile size: 32 px each

4. click ok

5. at the bottom right at Tilesets, click on New Tileset...

6. MAKE SURE THAT Embed in map IS SELECTED!!!!

7. select the spritesheet by clicking on Browse... next to the source text input and select in the game folder assets/sprites/spritesheet.png

8. click ok

9. save map by hitting ctrl+s or File⟩Save, select your map project files folder (it's recommended to have the map project files folder outside the space-runner folder, to make sure it doesn't get accidentally deleted on updates)

1.3   course of setting up the game

1. if you don't know how to do this, ask one who does

2. install git

3. install nvm

4. create new folder

5. get INITIAL_INSTALL.bat from me

6. put INITIAL_INSTALL.bat into the new folder

7. double click on INITIAL_INSTALL.bat, wait for it to complete

8. now you should have a folder space-runner in your new folder, this one includes the game

9. create a folder for your map project files (.tmx files) in your new folder (NOT the space-runner folder)

2   meta info

2.1   about this guide

This guide is separated into two parts. First general tutorials about Tiled. Then the documentation of the custom magic stuff implemented by dexter to make you able to create complex levels without having to touch program code.

2.2   some names

- collisions/colliding/... if the player collides with something, he/she can hit it. It's like an obstacle instead of a background wall. The player can't walk through it, would stand on it, etc.

3   general info about Tiled

3.1   custom properties?

Everything you draw/place/... in Tiled has some properties, like position, color, etc. Additionally you can set custom properties that have a type and value you choose.

Custom properties can have one of various data types. The most important are:

- bool: yes/no, true/false, checked box/unchecked box, etc.

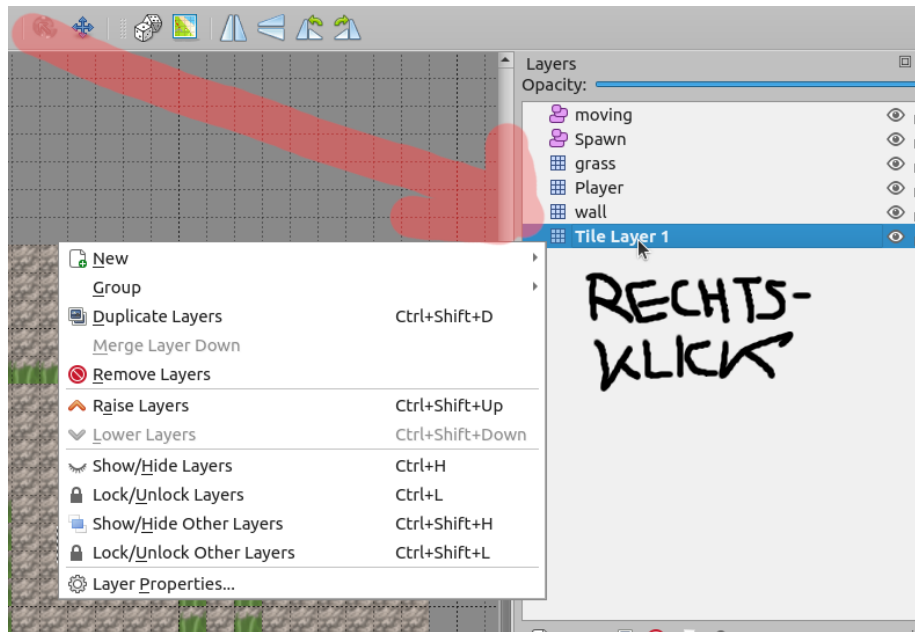- string: normal text

3.1.1 visual manual
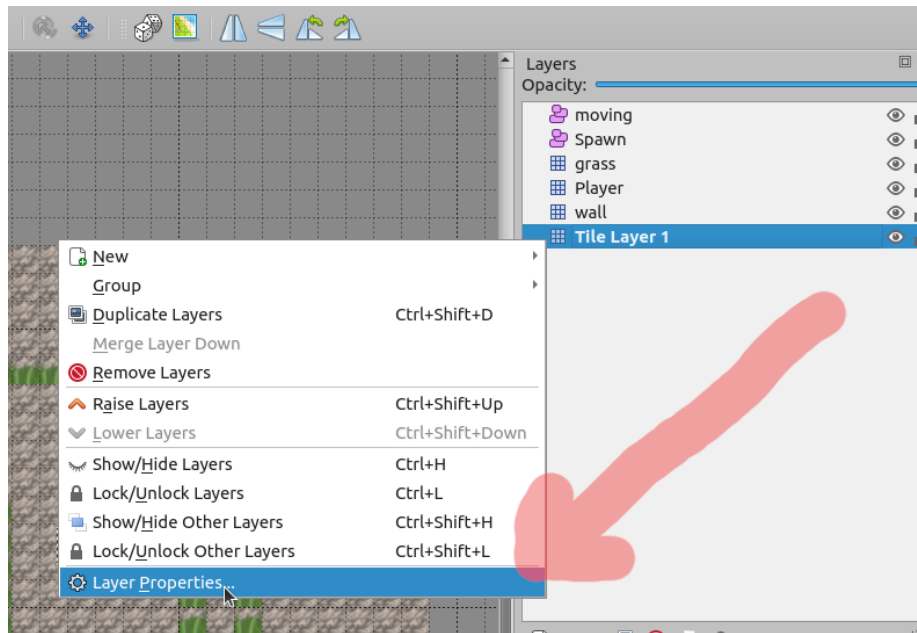

Figure 1: Right click on the layer.

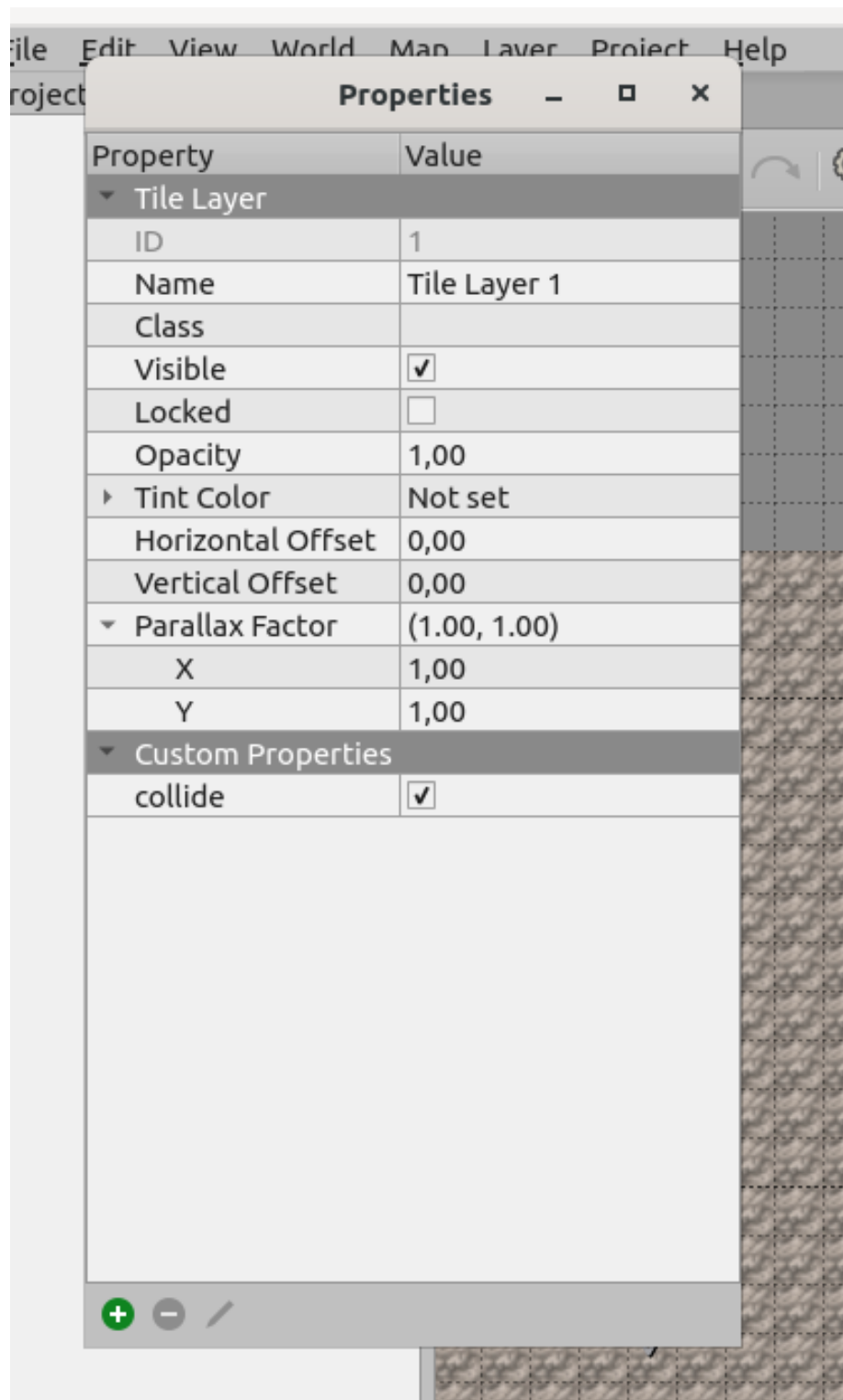Figure 2: Left click on "Layer Properties...".

Figure 3: Now you see the layer properties. Maybe you see it in some area in Tiled instead of a new window. It doesn't matter.
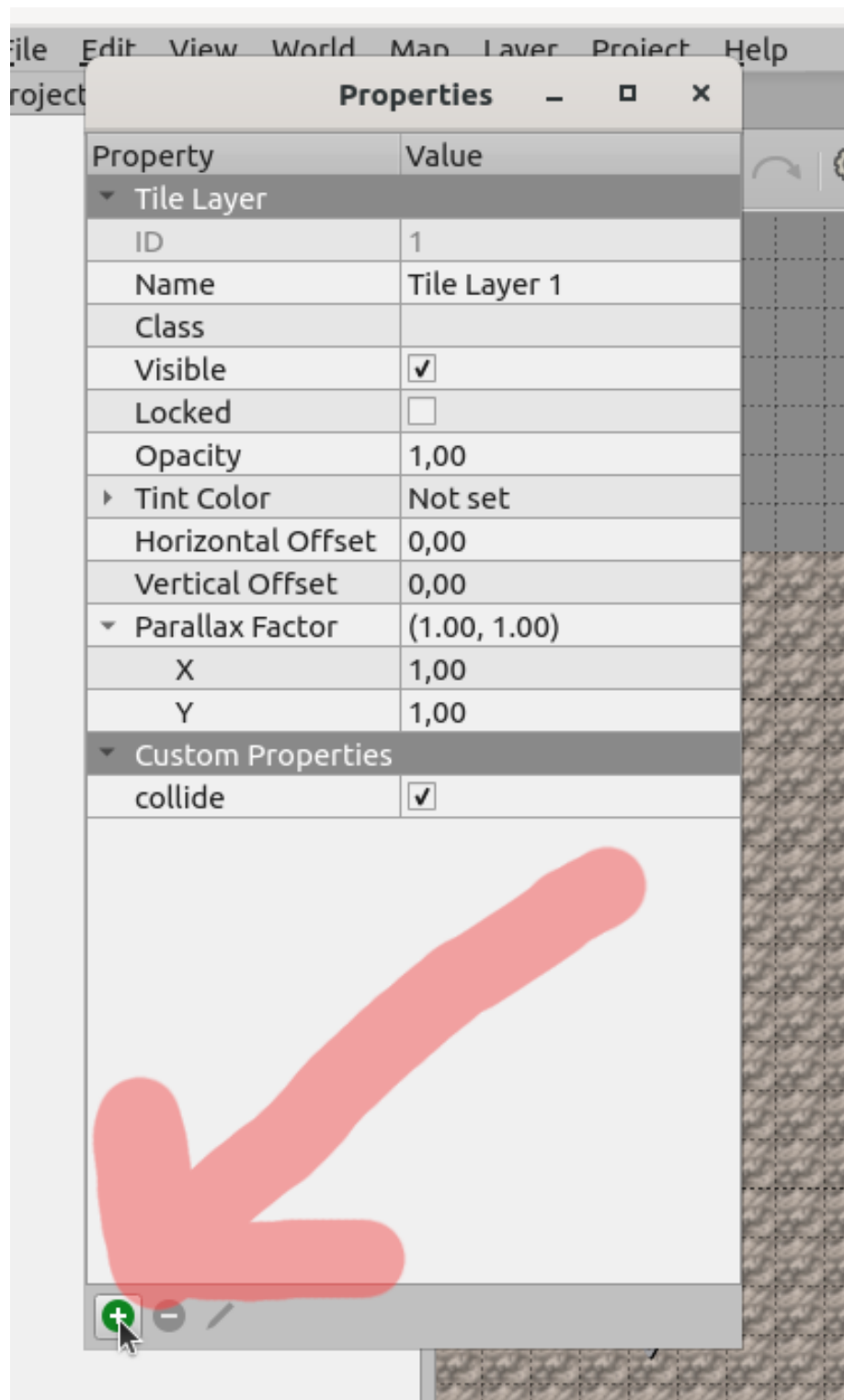
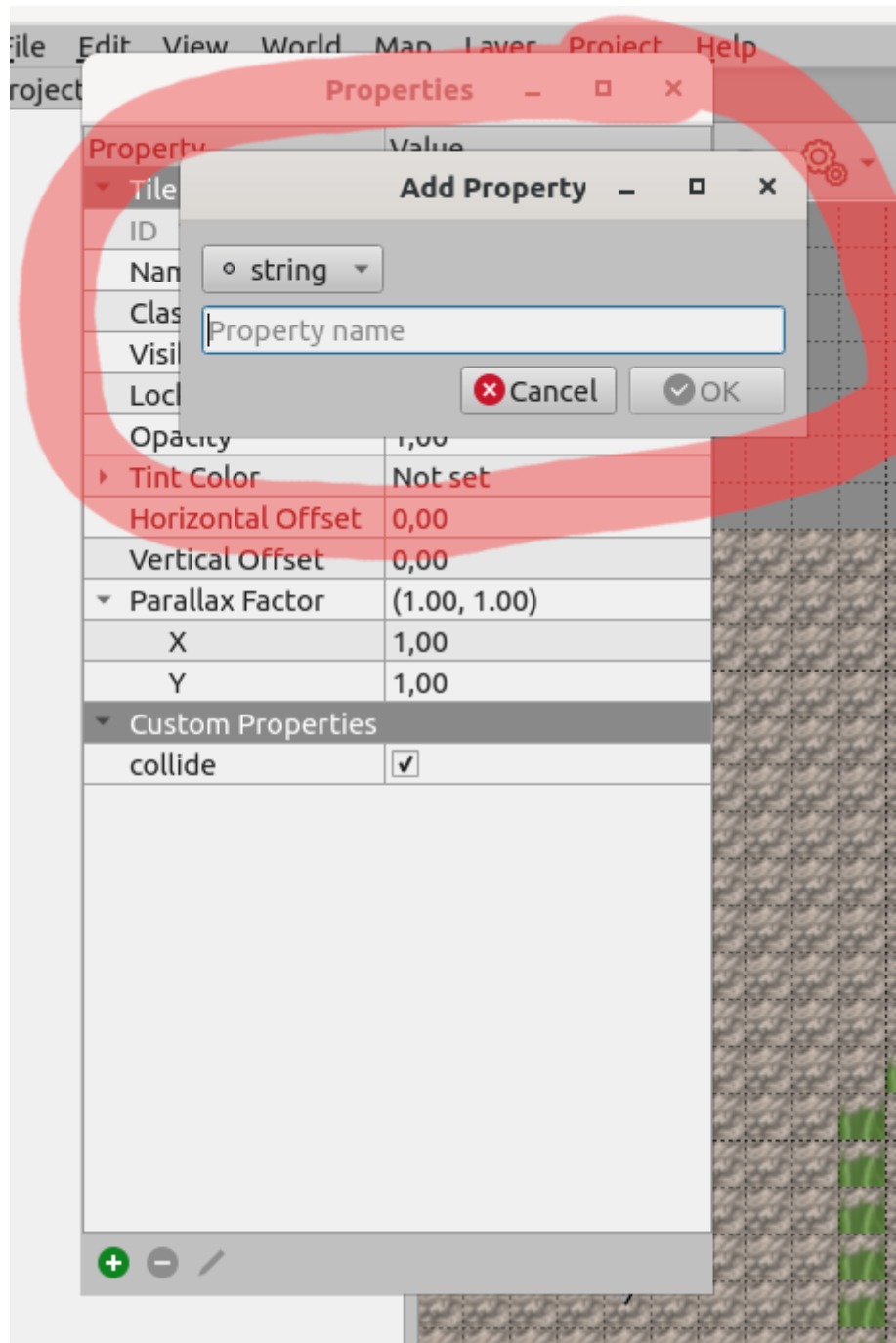Figure 4: To add a custom property, left click on the green + icon.

Figure 5: The property adding window opens. You can select the data type (in the image string is selected) and the property name.
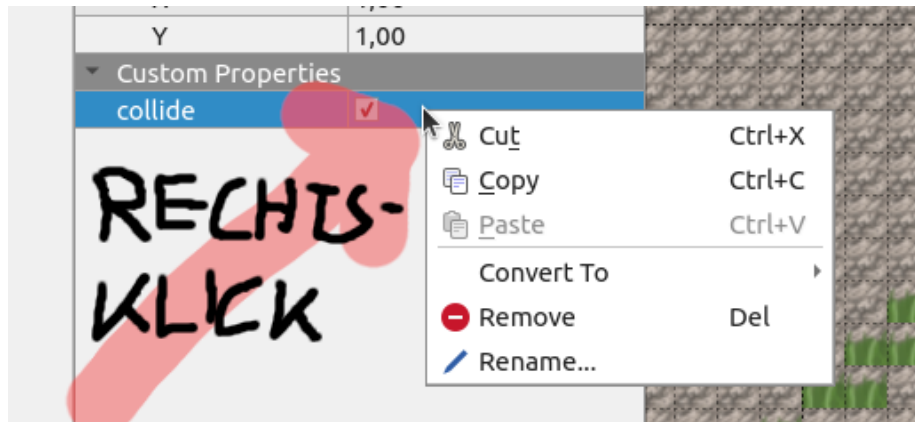
Figure 6: To edit a custom property, right click it.

3.2 layers?

3.2.1 tile layers

A tile layer consists of tiles. Tiles are the individual parts of the spritesheet. Tile layers are used for directly visible stuff. Here you're drawing.

3.2.2 object layers

An object layer consists of objects. Objects can be whatever you want and aren't directly visible. We use them for example for spawn points, platform movement paths, etc.

Objects can be of various types. Currently we only use points. As the name implies, they simply are a point in the map.

3.2.3 object layer lists

You can see the list of objects of an object layer and edit the individual name and properties of an object.
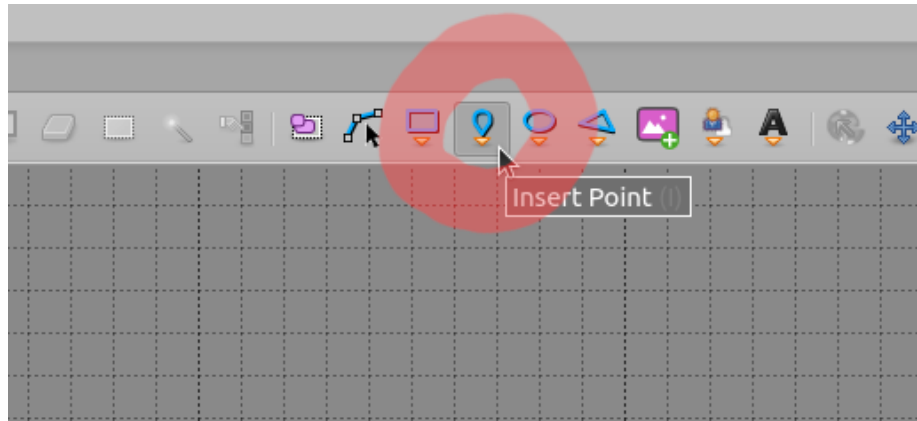
3.2.4 visual manual

Figure 7: This is the button to select the point adding tool which can add point objects.
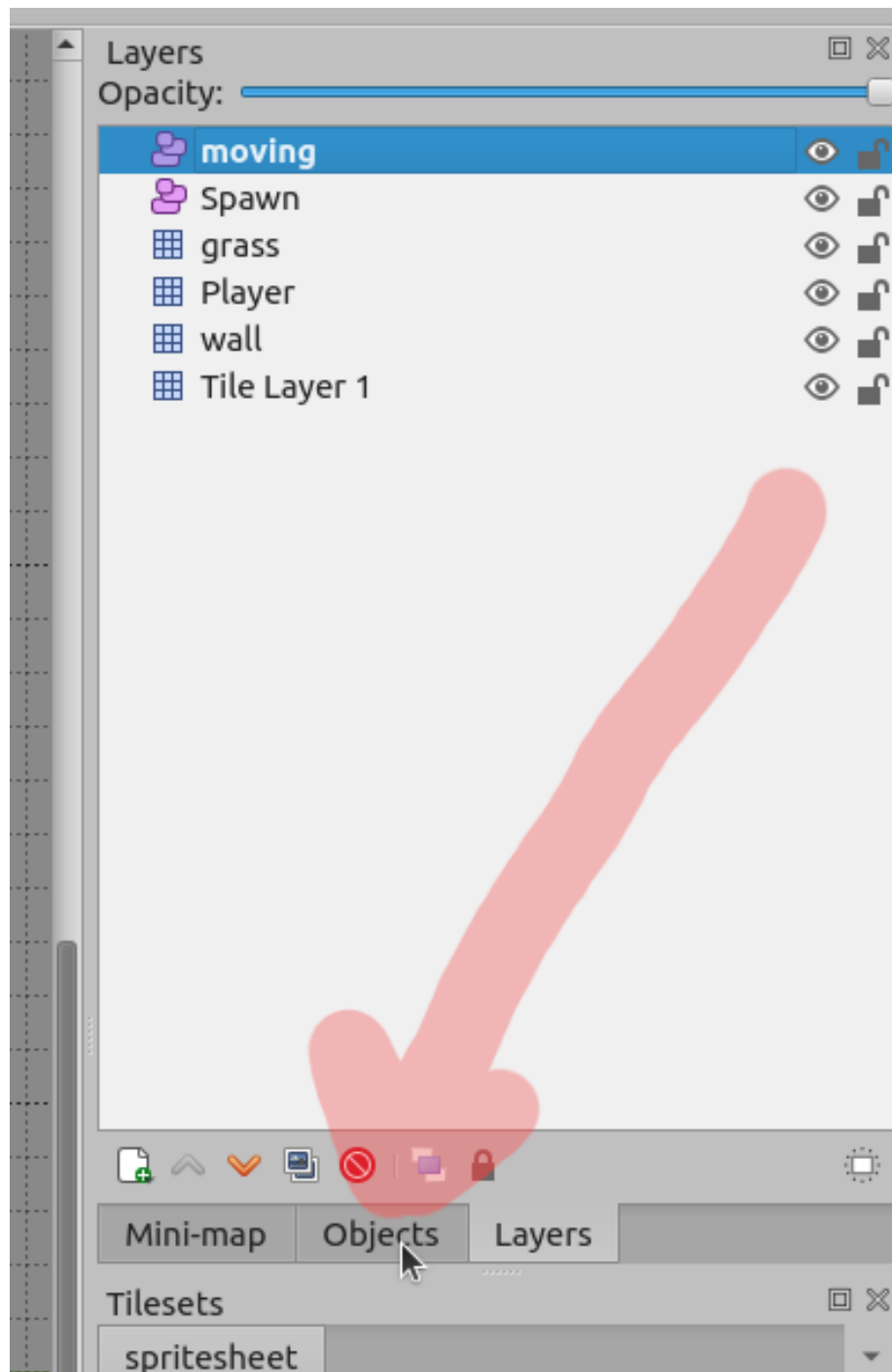
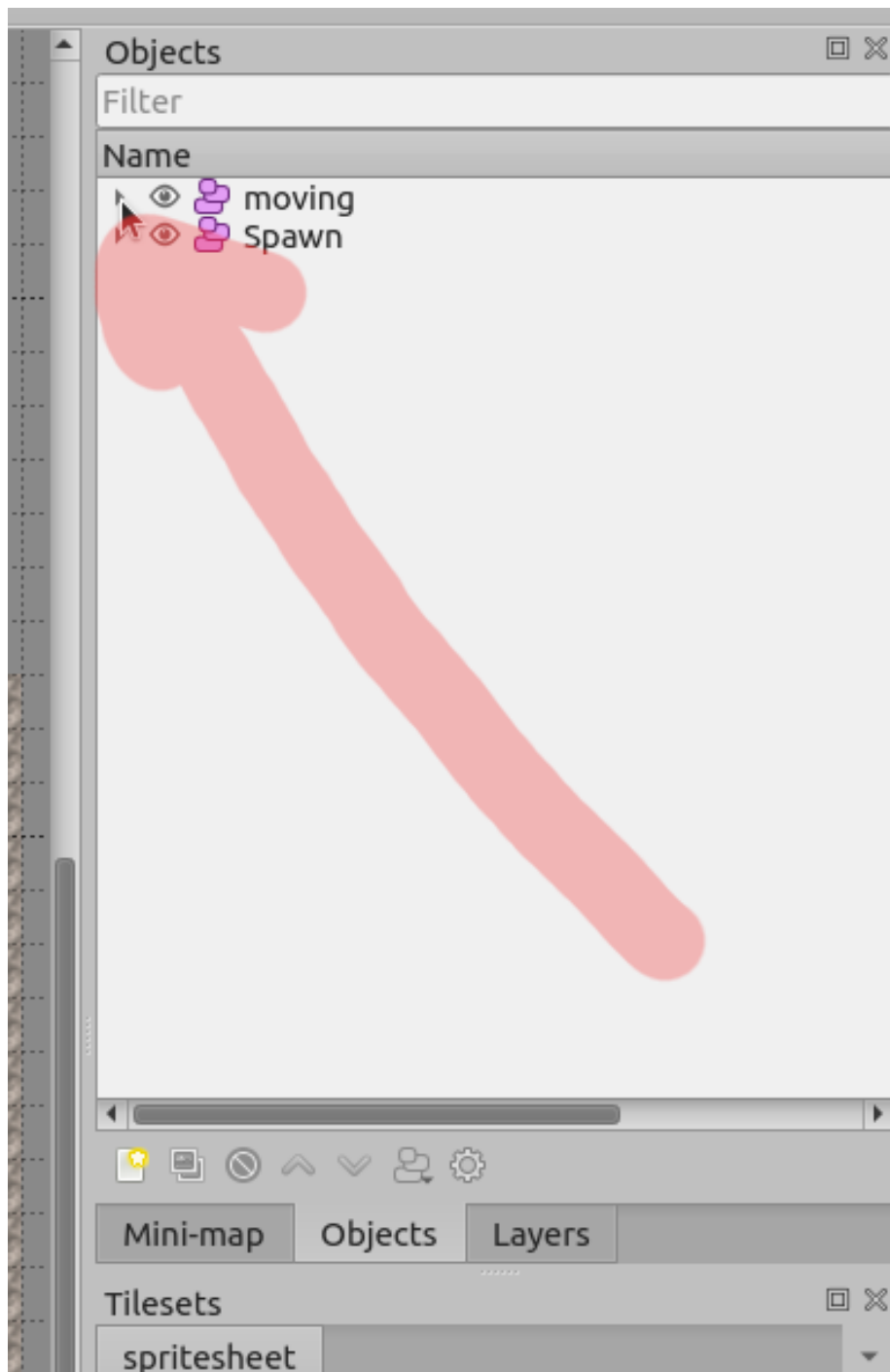Figure 8: To see the object lists, click on the Objects tab.

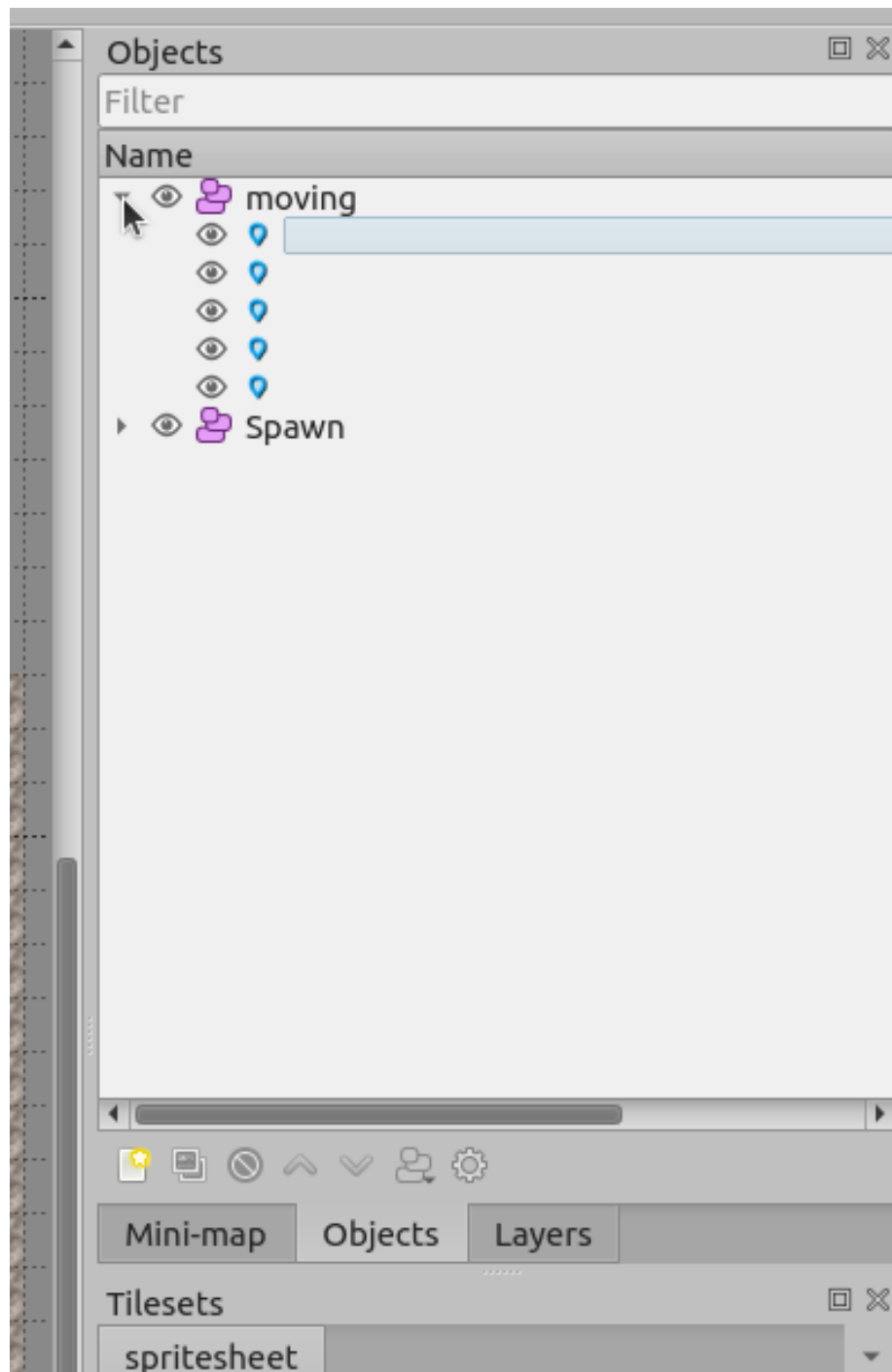Figure 9: Expand the list of the wanted layer.

Figure 10: Now you see every object that's in the layer. In the image the layer 'moving' has five points.
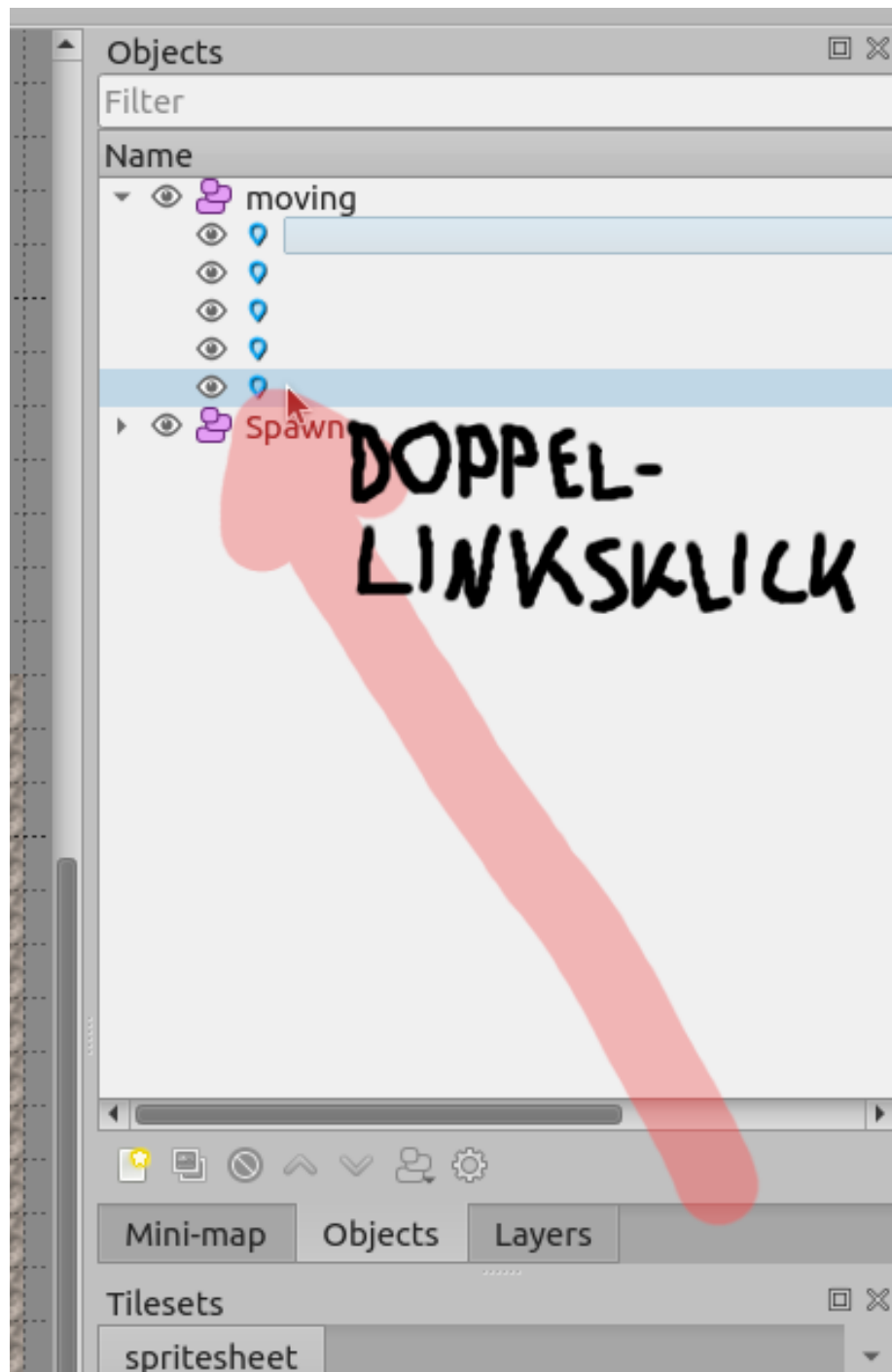
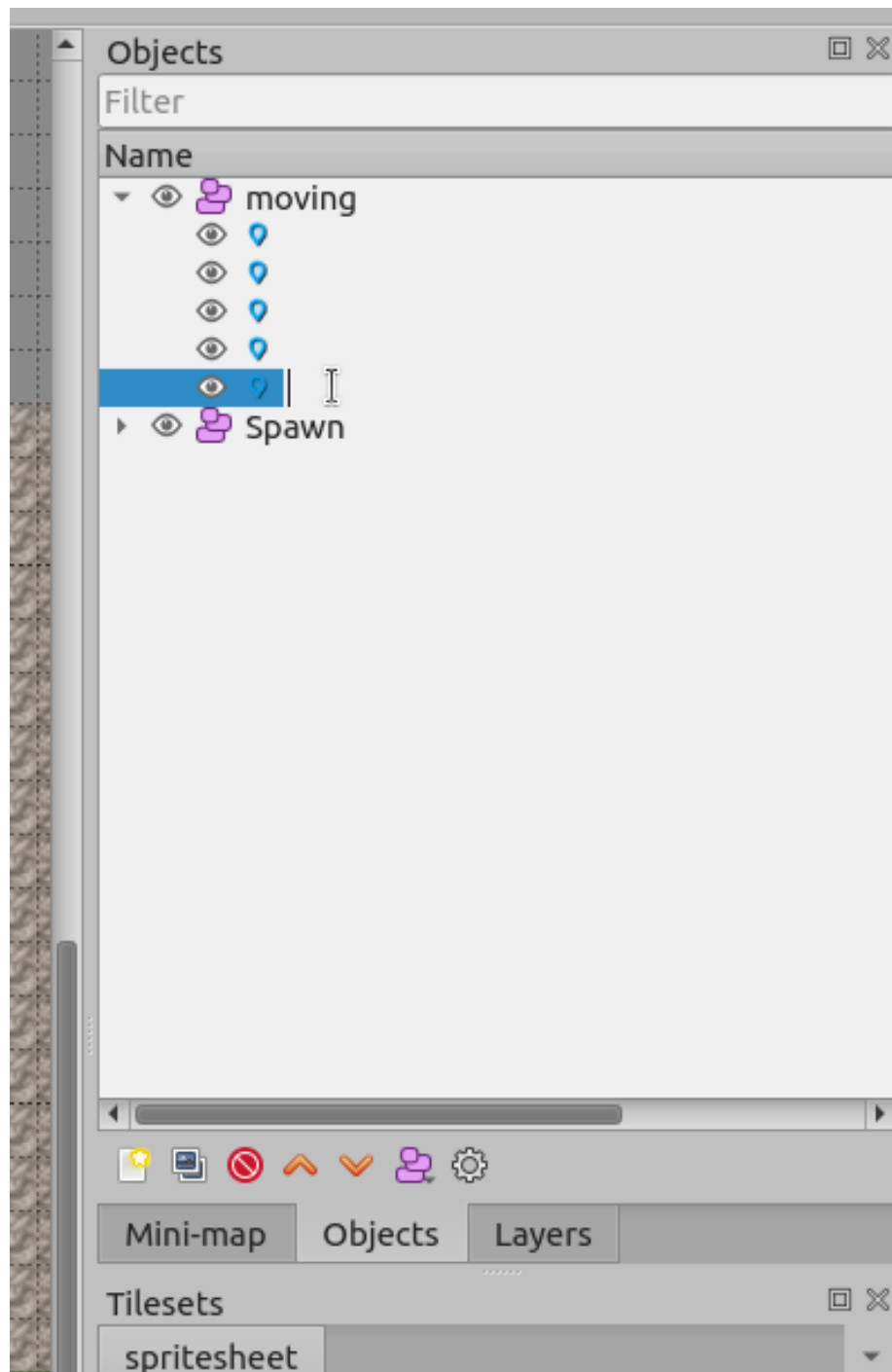Figure 11: To edit the name of an object, double left click it.

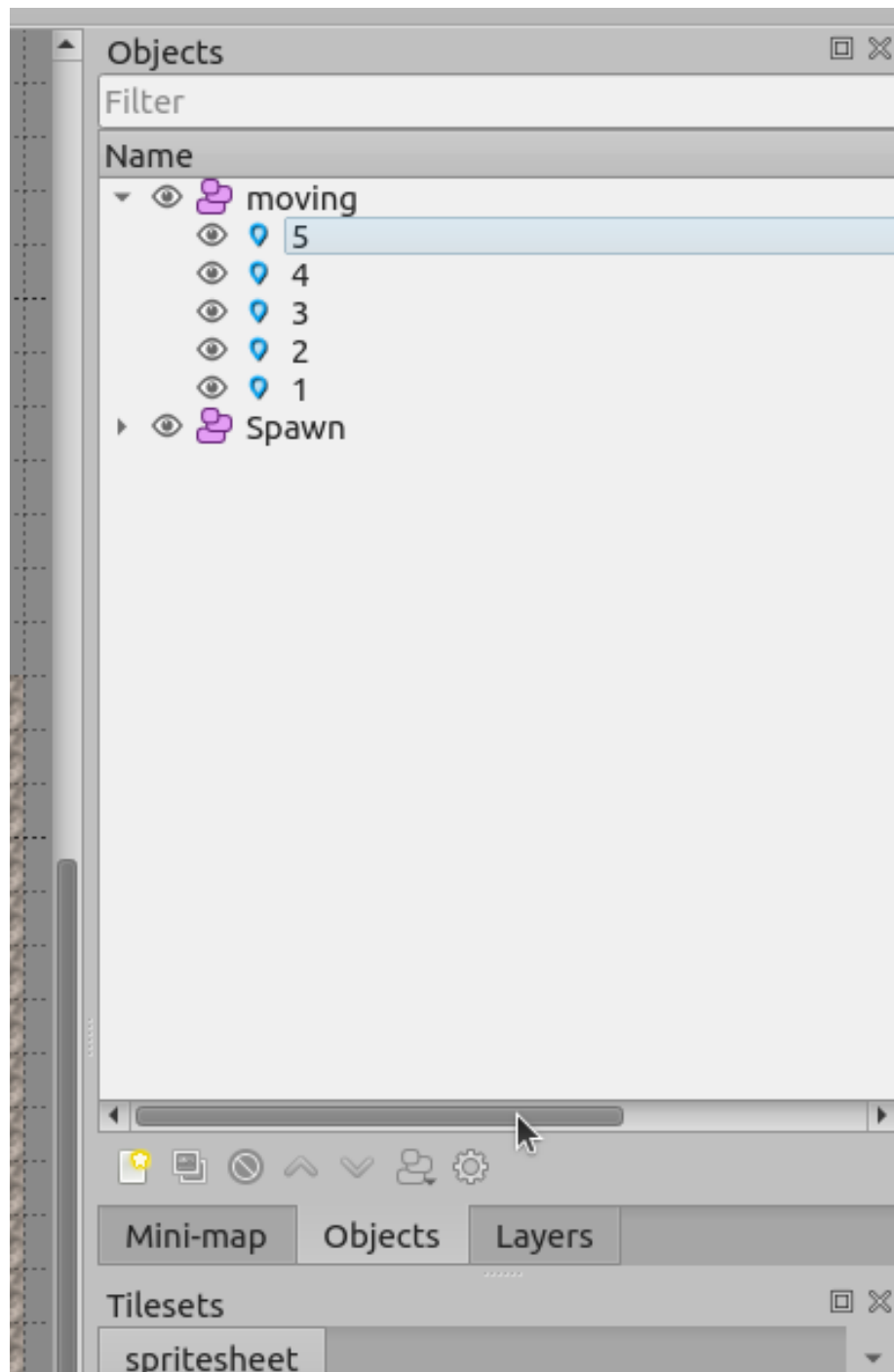Figure 12: Then type whatever you want.

Figure 13: In the image the five points were numbered 1 to 5.

4 Tiled special custom properties

Until now all special custom properties are used on layers and NOT an individual tiles. For the collide property this had serious impact on file sizes and using this style everywhere makes the usage easier.

Update: Now, all layers collide with the player by default. Instead you can use the background property to make the player fall/walk through layers.

Custom properties and special layer names doen't care about upper/lower case letters. E. g. you could write teLEPoRtTomaP instead of e. g. teleportToMap or spawn instead of Spawn.

4.1 background

- Name: background
- Type: bool
- Meant for: Tile layers

All tiles of the tile layer will NOT collide with the player. They're like a background or foreground.

4.2 kill

- Name: kill
- Type: bool
- Meant for: any layer

Everything in this layer will kill the player on contact. Tiles, enemies, etc.

4.3 teleportToPlace

- Name: teleportToPlace
- Type: string, name of an object layer, has to be exactly the same name, upper/lower case matters
- Meant for: tile layer

When the player hits anything in this layer, he/she gets teleported to the first object of the map with the name set in the teleport_to_name property.

4.4   finish

- Name: finish

- Type: bool

- Meant for: any layer

On contact the level is finished. The contents of these layers are what the player tries to reach.

4.5   unlocktask

- Name: unlocktask

- Type: bool

- Meant for: tile layers

On contact the player unlocks a task. Each level finish and unlocktask layer unlocks a different task. Therefore the total amount of levels and unlocktask layers (across all levels!) should be exactly the amount of tasks that are supposed to be distributed via the game (probably 28).

4.6   checkpoint*

- Name: checkpoint

- Type: bool

- Meant for: tile layers

On collision the colliding tile is set as the current checkpoint of the player. If the player dies, he or she gets teleported back to his/her current checkpoint.

4.7 visual examples

Figure 14: This layer has the collide property set to true. The player will collide with every tile of it.

Figure 15: This layer has the kill property set to true. The player will die when hitting something of this layer. This includes tiles and NPCs that got spawned because of the layer. (until now these two things are the only killing ones)

Figure 16: This layer has the teleport_to property set to "secret_chamber_map".
If the player hits anything in this layer, he/she will get teleported to the map
that's saved in the file named "secret_chamber_map".

5 Tiled special layers

5.1 Spawn

- Name: Spawn

- Type: Object layer

Contains the players spawn. The first object of the layer is selected and used
as the spawn position.

5.2 Player

- Name: Player

- Type: Tilelayer

Contains nothing. The layer at which the player will be displayed. Layers
above will be displayed above the player and layers underneath will be displayed
behind the player.

5.3 moving*

- Name: moving*write whatever you want here, only the "moving" prefix is
  important*

- Type: Object layer

Contains two or more points. The points mark the path which the moving
platform will follow. The object layer itself has the following parameters (still
work in progress):

- platform look

- platform size

- platform speed

In the Tiled Objects view, the points have to be numbered. The movement path just loops through the points going from lowest to biggest. E.g. with the points named 1, 2, 3 and 4, the platform will first go to 1, then 2, then 3, then 4, then 1, then 2, then 3, then 4, then 1 and so on.

5.4   npc*

- Name: npc*write whatever you want here, only the "npc" prefix is important*

- Type: object layer

Contains points. Each npc is by default the dude type. If you want another type, you have to set the custom property 'type' which determines the type of NPC. Todo: work in progress Points with an invalid type (including no type) will be ignored. A point only determines the spawn location of an NPC. After spawning they behave on their own. If an npc dies, it respawns at its spawn location.

To set enemies, i.e. NPCs that kill on contact, add the custom 'kill' property to the layer.