

Jared Callupe Lopez
GAM200
Team: Luminosity
Game: KnightLight

Tiled Design Guide

Rules

Tilemap: When creating levels, we would like to keep our design consistent. Make sure you follow the level design rules explained further in this document.

Tilemap Settings: When creating a map in Tiled make sure to have the following settings:

Map

- Orientation: Orthogonal
- Tile layer format: CSV
- Tiled render order: Right Down

Map size

- Fixed: Up to the level scale. The custom engine does not support infinite-sized tilemaps.

Tile size

- Width: 64px
- Height: 64px

Tileset: The Tileset_64x64.tsx file contains an upscaled version of Kenney's 1-bit platformer tileset: monochrome_tilemap_packed64x64.png (monochrome_tilemap_packed.png originally). All 400 tiles have a 'name' string property with their own names used to be exported to the custom engine.

Tileset Settings: If creating a new tileset make sure you have the following settings:

Tileset

- Name: [Tileset name]_[Tileset resolution].tsx
- Type: Based on Tileset Image

Image

- Source: The monochrome_tilemap_packed64x64.png file location. In the Unity project, the file is located in Assets > SuperTiled2Unity > Import. In the team repo, the file is located in trunk > KnightLight > Engine > Assets > Textures.
- Use transparent color: unchecked
- Tile width: 64px
- Tile height: 64px
- Margin: 0px
- Spacing: 0px

One Single Tileset: The custom engine can only support one single tileset currently. It might be possible to add custom tiles to the current tileset, but this hasn't been attempted so far.

Export Settings: In Edit > Preferences > General make sure to check the 'Embed tilesets' option for proper export to the custom engine.

Unity Scale: In the Unity inspector, make sure to set the 'Pixel Per Unit' of the tilemap to 64. Additionally, when placed on a scene, set its X and Y scale to 0.5.

Level Design Rules

Terrain Tiles: When creating the map layout or terrain, make sure to use the ‘brick’ tiles (as they’re named in the tileset) located on the right side of the tileset. These go in the ‘MainLayer’ tile layer.

No Blank Tiles: Do not cover up empty tiles inside terrain with blank or completely black tiles. The game’s background is already black and it only congests the custom engine’s hierarchy.

Decorations: In the ‘Decorations’ tile layer, decorative, uninteractable, or background objects can be placed. This layer’s opacity is set to 0.5 which makes the objects look darker in a dark environment.

Cables: (Unity only) In the ‘CableN’ tile layer, use the CableTileset_64x64 tileset to draw cables that connect the lamp with their respective door. Make sure the player can follow it without losing track of it and they wrap around terrain tiles, and go through them only if needed.

Colliders: In the ‘Colliders’ object layer, use the Insert Rectangle tool (R key) while holding Ctrl to create the environment collision. Holding Ctrl makes your cursor snap to the grid, but just in case you accidentally let it go, check the rectangle’s width and height are multiples of 64.

Jump-Throughs: In the ‘JumpThrough’ object layer, make rectangles that go from one edge of the jump-through platform to the other. Make sure these colliders have a height of 0. Sometimes, the Unity build will require the colliders to have some height. If this is the case, set the height of the collider to 24 in Tiled.

Hazards: In the ‘Hazards’ object layer, make rectangles that cover up the whole sprite in the tile of the hazard. In the case of spikes (spikes_02 specifically), make sure their height is set to 36 and, since the tilemap is rendered from top to bottom, add 28 to their Y position.

Exit: In the ‘Exit’ object layer, make a square that covers up the exit door(s). The exit door tiles go in ‘Main Layer’, while the entrance goes in the ‘Decorations’ tile layer to signify not being able to interact with it (in the future you might be able to interact with them and go to previous rooms).

Lamps: In the ‘Lamps’ object layer, make 1 tile x 1 tile colliders that cover up the lamp tiles position.

Doors: In the ‘Doors’ object layer, make colliders that cover up where the doors go.

Color Coding

All colliders in Tiled will use the following color-coding:

Hazards: Red

Standard Colliders: Green

Jump Through Colliders: Blue

Doors: Cyan

Lamps: Yellow

Exit: Magenta

More details regarding the specific decimal code (R,G,B) can be seen in the layer specifications section of this guide.

Layer Specifications

The following items contain information for all layers in our tilemaps in the following format:

- **[Layer Name]:** (Layer description).
 - Layer Properties**
 - [Layer Property] = [Layer Property Value]
 - Custom Properties**
 - {Custom Property Type} [Custom Property Name] = [Custom Property Value]

Object Layers:

- **Doors:** Layer for all doors that block the player's path.
 - Layer Properties**
 - Color = rgb(0,255,255)
 - Custom Properties**
 - {string} name = doors
- **Lamps:** Layer for all lamps (door-triggering or healing) the player can interact with.
 - Layer Properties**
 - Color = rgb(255,255,0)
 - Custom Properties**
 - {string} name = lamps
- **Exit:** Layer for the exit of the room.
 - Layer Properties**
 - Color = rgb(255,0,255)
 - Custom Properties**
 - {string} name = exit

- **Hazards:** Layer for all hazards the player can receive damage from.

Layer Properties

- Color = rgb(255,0,0)

Custom Properties

- {string} name = hazards
- {string} unity:Tag = Hazards

- **Colliders:** Layer for all terrain colliders.

Layer Properties

- Color = rgb(0,255,0)

Custom Properties

- {string} unity:Tag = Environment
- {string} unity:layer = Environment

- **JumpThrough:** Layer for all platforms the player can jump through.

Layer Properties

- Color = rgb(0,0,255)

Custom Properties

- {string} unity:Tag = JumpThrough

Tile Layers:

- **MainLayer:** Layer for all foreground tiles and base appearance of the level.

Custom Properties

- {int} unity:SortingOrder = 4

- **Decorations:** Layer for all decorative, uninteractable, or background tiles.

Layer Properties

- Opacity = 0.50

Custom Properties

- {string} name = decorations

- **CableN:** Layer for each cable that connects a lamp and a door. 'N' is the number of the cable.

Layer Properties

- Opacity = 0.30

Custom Properties

- {int} unity:SortingOrder = 110

Useful Links

If all this information gets too overwhelming, here are some helpful links that can aid you understanding this document, or our level design process in general:

Importing SuperTiled2Unity To Your Unity Project:

https://www.youtube.com/watch?v=eyI_7GmIxMY

Tiled Documentation: <https://doc.mapeditor.org/en/stable/>

SuperTiled2Unity Documentation:

<https://readthedocs.org/projects/supertiled2unity/downloads/pdf/latest/>

For more specific questions or suggestions that can improve the legibility of this document, please contact the level designer.