Tiled type properties

In Tiled, you can set a type and properties to objects in object layers. Case matters for porperty names and values defined in this document. Default values written here are used if the property is not explicitly defined.

Map properties

RedBox has some properties that will be read and interpreted when loading a tile map.

Name	Description	Default value
zInterval	Z interval between each layer. When layers are added to the state after being loaded. This will be the distance in Z between each layer for the render order. This will let you add bodies to the state between layers. If the Z interval is set to 0, the layers will still be rendered in the right order, but you will not be able to add bodies between the layers. If the value is negative, the order of the layers will actually be rendered in the reverse order.	100
zStart	Z of the first layer. The second layer would have a z of zStart + zInterval.	0

Layer properties

Only one property is read and interpreted by RedBox when reading a tile map.

Name	Description	Default value
z	Z coordinate of the layer. Overrides the Z calculated by the zStart and zInterval specified in the tile map's properties. Optional.	

Object types available

RedBox has some pre-defined types you can set that RedBox will read and initialize. To load the types in Tiled, open the “Preferences” window and select the “Object Types” tab. Then, you can either enter the types manually or import them from the “redbox_types.xml” file available with this documentation.

- Sprite
- InanimateSprite

Common properties

Here's the list of properties that RedBox will read for both "Sprite" and "InanimateSprite". Boolean values are represented with "0" for false and "1" for true.

If a given value is invalid (for example, if you put "bob" as the value for "velocity.x"), the default value will be used when loading the body.

Name	Description	Default value
velocity.x	Horizontal velocity of the object (float or integer value).	0
velocity.y	Vertical velocity of the object (float or integer value).	0
maximumVelocity.x	Maximum horizontal velocity of the object (float or integer value). Negative value for no maximum horizontal velocity.	-1
maximumVelocity.y	Maximum vertical velocity of the object (float or integer value). Negative value for no maximum horizontal velocity.	-1
acceleration.x	Horizontal acceleration in pixels/sec^2.	0
acceleration.y	Vertical acceleration in pixels/sec^2.	0
globalDrag	Deceleration applied when there is neither a horizontal nor a vertical acceleration until velocity reaches 0.	0
horizontalDrag	Horizontal deceleration applied when there is no horizontal acceleration until horizontal velocity reaches 0. So there could be a vertical acceleration while the horizontal drag is being applied.	0
verticalDrag	Vertical deceleration applied when there is no vertical acceleration until vertical velocity reaches 0. So there could be a horizontal acceleration while the vertical drag is being applied.	0
collidableSides	Tells which side can collide. A side that can't collide will let a collidable body pass through. The order of the list of booleans is "LEFT", "RIGHT", "TOP" and "BOTTOM". The value MUST be a string of 4 "0" and "1"'s.	1111
elasticity	Elasticity factor of the body, will determine how it should bounce in a collision. 0 means the body will not rebound at all when a collision is detected with another	0

	body. The body must be non-static for its elasticity to be applied. 1 would mean "perfect" rebound. So if you set an elasticity of 1 to an object falling to the ground, it would rebound at the same height infinitely. But, in reality, the elasticity calculation is not precise enough and it the height changes a bit each time.	
staticBody	Wether or not the body will react to collisions. A static body will collide, but it won't be moved or affected by the collisions.	0
offset.x	Horizontal offset added to the position when detecting collisions.	0
offset.y	Vertical offset added to the position when detecting collisions.	0
collidingBoxRatio.x	Width of the colliding box compared to the body's actual width. To make the colliding box have half the body's width, you'd put 0.5 here.	1
collidingBoxRatio.y	Height of the colliding box compared to the body's actual height. To make the colliding box have half the body's height, you'd put 0.5 here.	1
offsetRatio	Wether or not the offset is a proportion of the body's size. False makes it so the offset is interpreted in pixels.	0
textureKey	Key of the texture to use. Only used for rectangle objects and polygon objects. If not specified, the rectangle or polygon will be rendered as a shape which its color will be the same as the object's layer's.	
color	Color of the object. Must be in CSS rgb format. See http://www.w3.org/TR/css3-color/. Does not support HSL and HSLA.	rgb(255, 255, 255)

Specific properties

Sprite

The only difference a “Sprite” has with a “InanimateSprite” is the possibility to have animations. You can define animations for a sprite and the default animation. By default, a sprite doesn’t have any animations and only one frame.

For the frame property names “frame[i]”, replace “i” with the index of the frame. The index starts at 0. So if you define “nbFrames” with 3, it’ll expect definitions for “frame[0]”, “frame[1]” and “frame[3]”, else it’ll use default values. At least one frame must be defined.

For animation definitions, it uses a similar idea as the frame definitions. You define animation properties with “animation[name]” where you replace “name” with the animation’s name.

Name	Description	Default value
nbFrames	Number of frames the sprite has defined. Cannot be set to 0 (will be set to the default value if the number of frames is invalid).	1
frame[i].position.x	Horizontal position of the frame in the texture. Is only used for rectangle objects and polygon objects.	0
frame[i].position.y	Horizontal position of the frame in the texture. Is only used for rectangle objects and polygon objects.	0
frame[i].orientation	Orientation of the frame in the texture, used when the frame is rotated (for example, to save space in a texture atlas). The different possible values are "NORTH", "SOUTH", "EAST" and "WEST". The orientation represents the direction the top of the wanted frame image is pointing. For example, if the frame image is rotated 90 degrees clockwise, the orientation would be "EAST"	NORTH
animation[name].frames	The suit of frame indexes composing the animation. If an index is invalid, it will be interpreted as 0. The array is in the form "[0, 1, 4, 2, 2, 14, 5]" (surrounded by square brackets and frame indexes separated by comas, white spaces are all trimmed).	[0]
animation[name].timePerFrame	The time per frame (in seconds).	0.3
animation[name].nbLoops	Number of loops before the animation	-1

	stops/finishes. Negative value for infinite looping.	
defaultAnimation	Name of the default animation. If this property is not defined, no animation is started and the default frame is displayed.	
defaultFrame	Index of the frame to display by default if there is no default animation specified.	0

InanimateSprite

A “InanimateSprite” cannot be animated. It can only have one frame. Read the properties' description carefully, some of them can only be used for specific types of objects.

Name	Description	Default value
frame.position.x	Horizontal position of the frame in the texture. Only used for rectangle objects and polygon objects.	0
frame.position.y	Horizontal position of the frame in the texture. Only used for rectangle objects and polygon objects.	0
frame.orientation	Orientation of the frame in the texture, used when the frame is rotated (for example, to save space in a texture atlas). The different possible values are "NORTH", "SOUTH", "EAST" and "WEST". The orientation represents the direction the top of the wanted frame image is pointing. For example, if the frame image is rotated 90 degrees clockwise, the orientation would be "EAST".	NORTH