The following document lists the entity-components with their respective systems and usage.

- Grayed out components are used internally and not to be modified by the client-developer.
- Property names in triangle brackets (e.g <property>) are placeholders.
- Property names in quotation marks (e.g. “set/d*”) are pattern matched.
- The T column indicates wether the property can be set in the Tiled Level Editor. In theory, it is possible to set every attribute in the Editor, but it is advised to restrain setting some attributes.

Component	T	Type	Description
entityid		integer	unique entity identifier
name	x	string	textual name or representation of entity
player	x	boolean	indicator for player entity
mapdata		boolean	indicator for tiled map level data, set by level reader
mapentity		boolean	indicator for generic map entity (tile, object, image)
tilelayer		boolean	indicator for tile layer tile type, set by level reader
imagelayer		boolean	indicator for image layer type, set by level reader
objectgroup		boolean	indicator for object layer tile type, set by level reader
cameratrack	x	boolean	entity is tracked by the camera
point	-	point	entity represents point, tile coords
polygon	-	polygon	entity represents polygon data, tile coords
ellipse	-	boolean	entity represents ellipse, implies rectangle
rectangle	-	rectangle	entity represents rectangle, tile coords
hitbox	x	{w,h} (float)	width, height of bump collision box (can be different from entity width, height)
collision	x	boolean	adds hitbox {w = width, h = height}]to entity when added to collision system
bumptype[1]	x	string / boolean	bumptype when other collides with entity, (“touch", "cross", "slide" or "bounce", default is “touch”), alternative set boolean to false if all collisions should be ignored
bumptag	x	integer [1,64]	tag (group) this entity belongs to for the purpose of collision handling (supports 64 groups)

---

[1] In the context of collision handling, entity describes the colliding object and other describes the object the entity collides with.

bumpmask[2][3]	x	integer (64 bit)	only collide with with entity bumptag **j** if bit **j** in other bumpmask is set (**j** in [1,64])
pos	x	vector	2d “top-down” position, tile coords
immobile	x	boolean	indicator that entity should not be moved
moved		boolean	indicator that entity has moved since last update
follow	x	table	entity follows a given object, if object, translated to follow.target, value is a reference to an object with a pos component or x,y attributes ,
follow.target	x	object[4]	target to follow, either a component with an object reference or a table with pos component
follow.speed	x	float	the speed in m/ms with which the entity follows the target, default 0.1
follow.pause	x	boolean	pause following, default false
follow.mindist	x	float	minimum distance to target, do not come closer than this, default 0
follow.tolerance	x	float	tolerance for distance to target, default 10
patrol	x	object	(special case of follow) entity follows a given path, value is a reference to a polygon/polyline object
patrol.speed	x	float	the speed in m/ms with which the entity follows the path, default 0.1
patrol.pause	x	boolean	patrol is paused, default false
patrol.reverses	x	boolean	wether the entity follows the path back once it is completed
patrol.repeatcount	x	int	number of times the patrol path should be repeated after the initial run, default -1 (always repeat)
patrol.iteration	-	int	current repeat iteration of the followed path
width	x	float [0, inf]	physical width of an entity in tile coords
height	x	float [0, inf]	physical height of an entity in tile coords
rotation	x	float	rotation of entity clockwise in degrees
mass	x	float [0, inf]	mass of object [kg]
lindamp	x	float [0,1]	factor modelling friction/drag
acc		vector	current acceleration [m/ms^2]
vel		vector	current velocity [m/ms]

[2] Only applies if *entity.bumpmask* and *other.bumptag* exists. Otherwise these properties are ignored when resolving collision.
[3] To convert a binary number to an integer in Tiled you can use the Tiled Console (bottom-left icon) and type in “parseInt(“binary number”, 2), e.g. parseInt(“0011”,2) outputs the number 3.
[4] Needs to be either an object with a reference (Tiled) or an Entity with a “*pos*” component.

force[5]		array of vector	accum. forces [kg*m/ms^2], depending on ForceMode (Force, Impulse, Acceleration, VelocityChange)
sprite		sprite	sprite, render component
state		StateMachine	manage current entity state
event.<subject>		table	events listen to messages with the header <subject> (e.g. collision) and execute actions upon those events
event.<subject>.repeats	x	boolean	if set to false event subject is consumed, after the first accepted message is processed, i.e. ignore following messages
event.<subject>.ignore	x	boolean	event does not listen to any more messages for the subject (useful for dynamic blocking of events)
event.<subject>."set/d*"[6]	x	object	object target reference for setting values, defaults to message body if not given or 0
event.<subject>."set/d*".<property>	x	any[7]	sets the property of the given target to the given value, accepts dot notation for <property>

Event/Message

Subjects

The following events and messages are defined.

- "collision" - body = entity (colliding entity)
- "property" - body = {name, value}

Example

[5] Manipulate force using *Entity:addForce(fx, fy, mode)*. Parameter mode can be one of the following Force, Impulse, Acceleration or VelocityChange, defaults to Impulse.
[6] Matches against the literal characters "set" followed by zero-or more digits. Valid names are set, set1, set2, etc.
[7] Value can be any type, strings are treated special: "nil" is treated as deleting that property, "self:<property>" references an own property and "body:<property>" references a property in the message body.