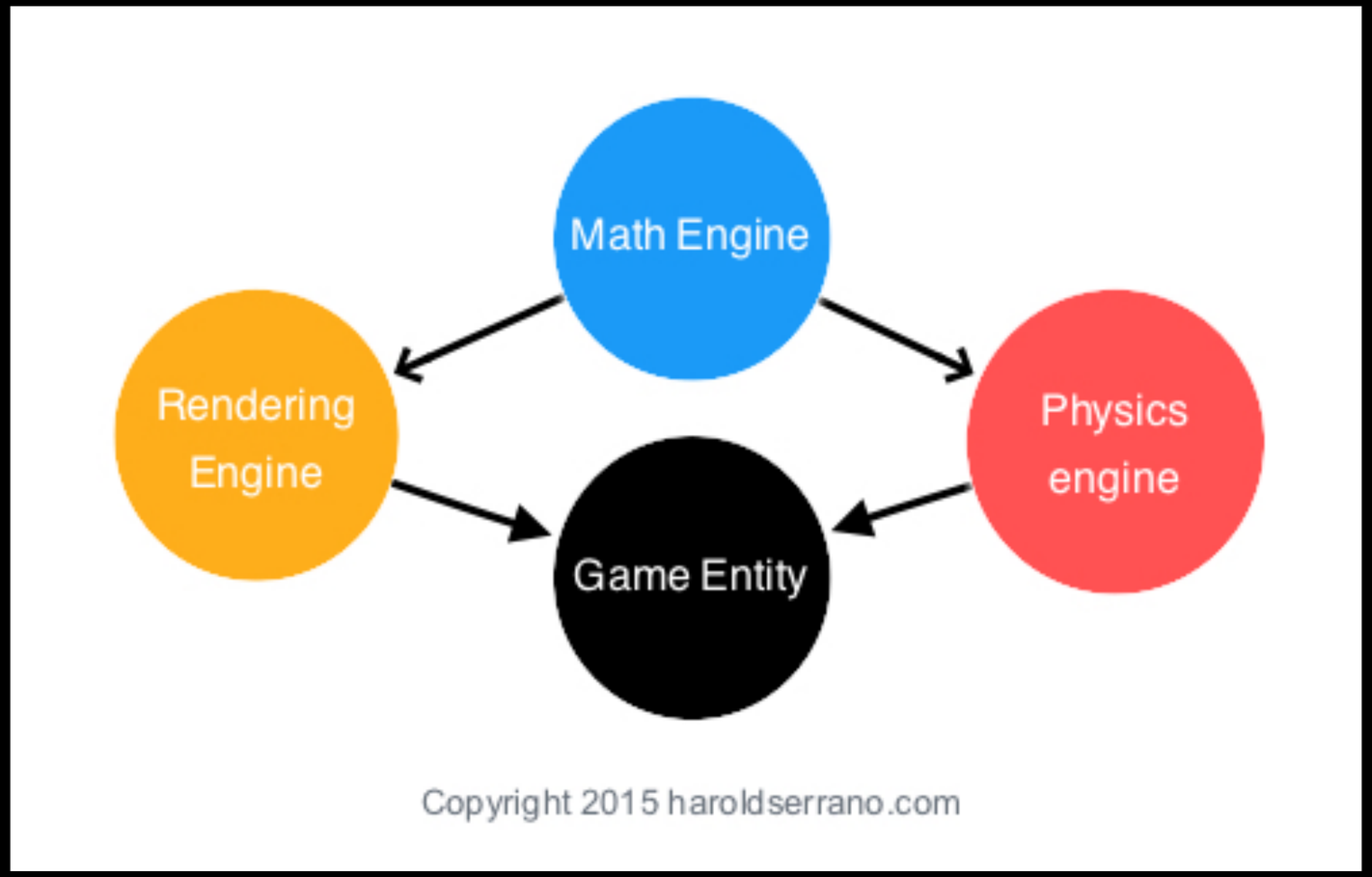
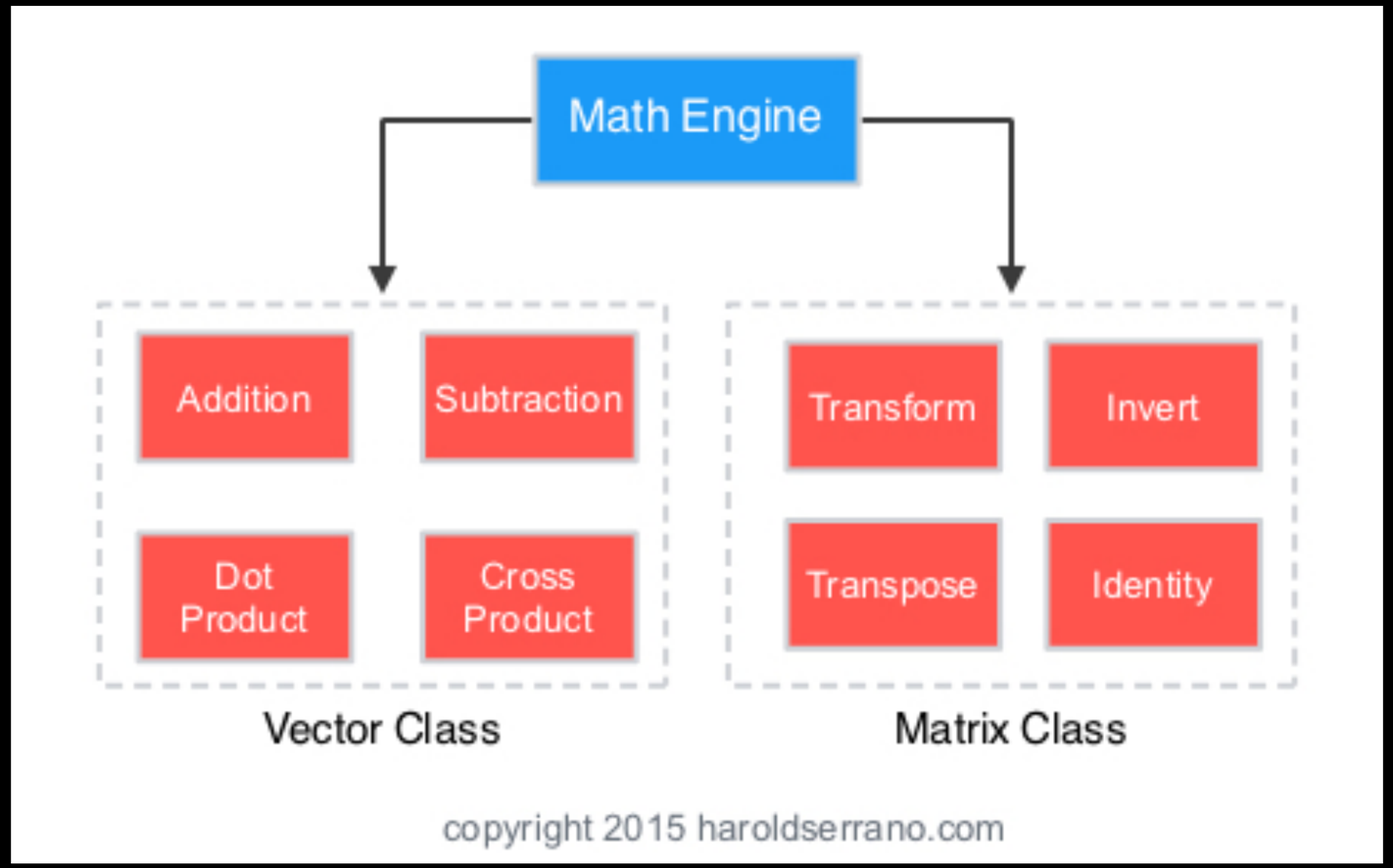
Javascript
Module

What is cocos?

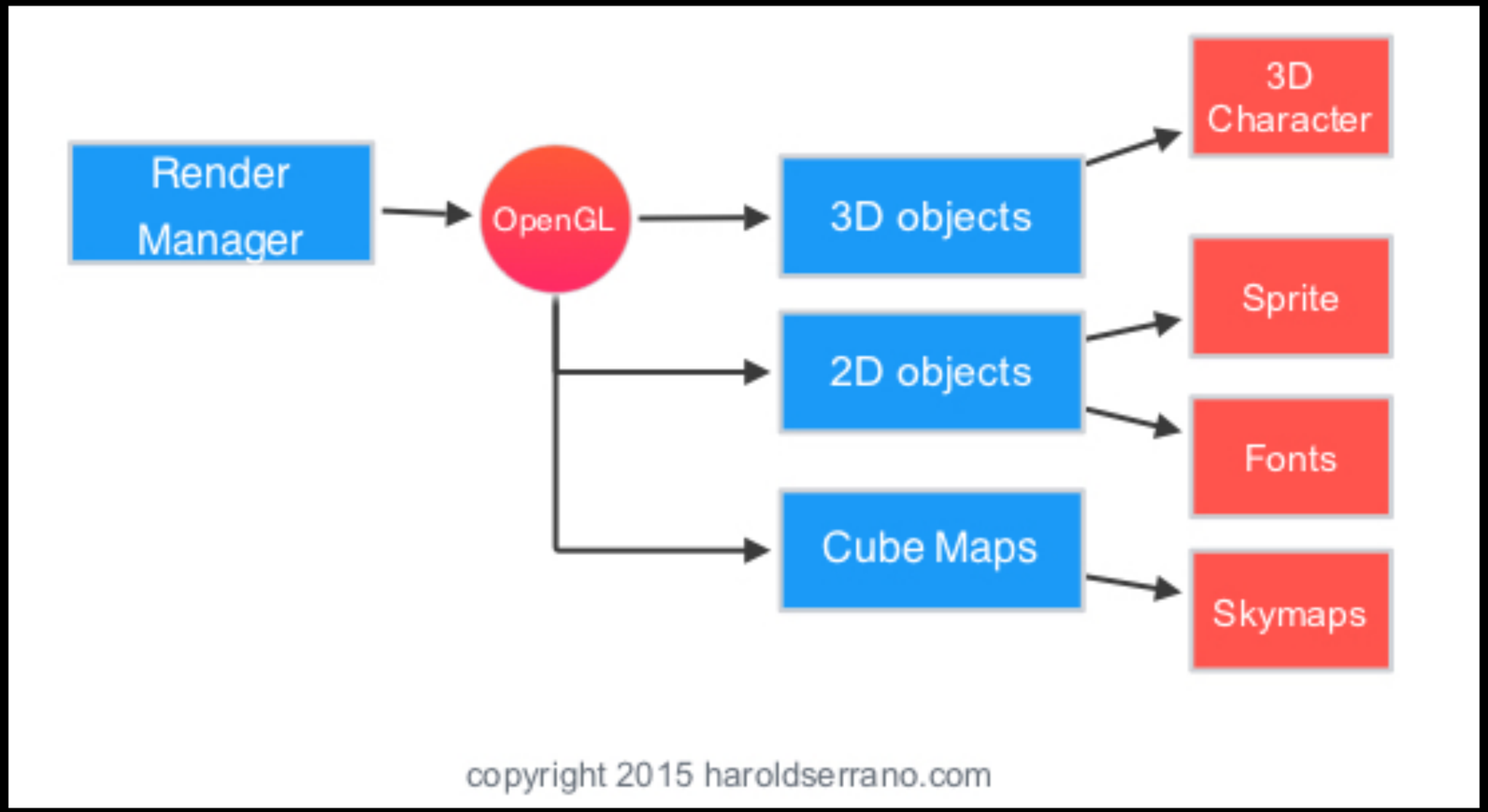
What inside a game engine?



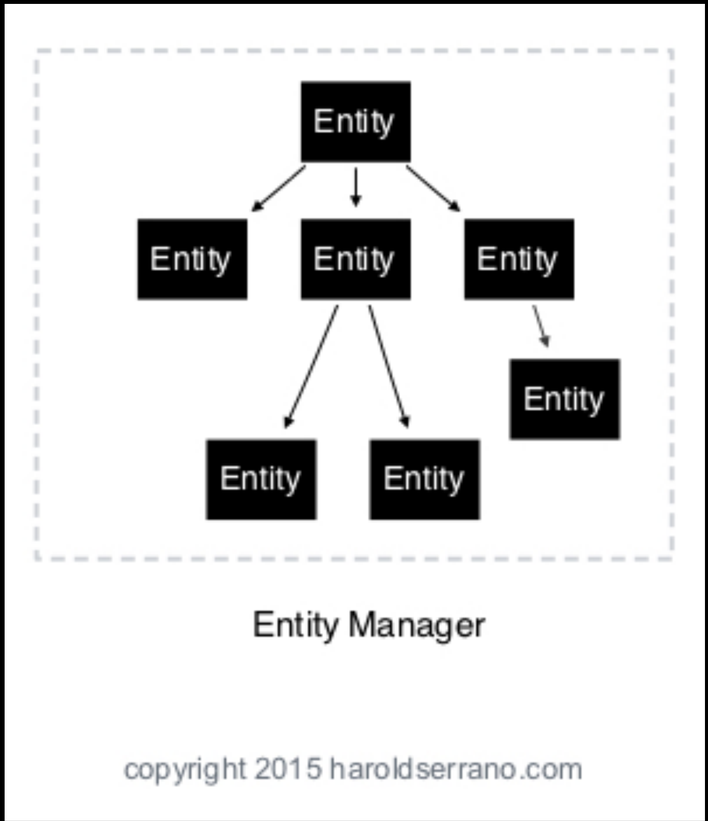
Math Engine

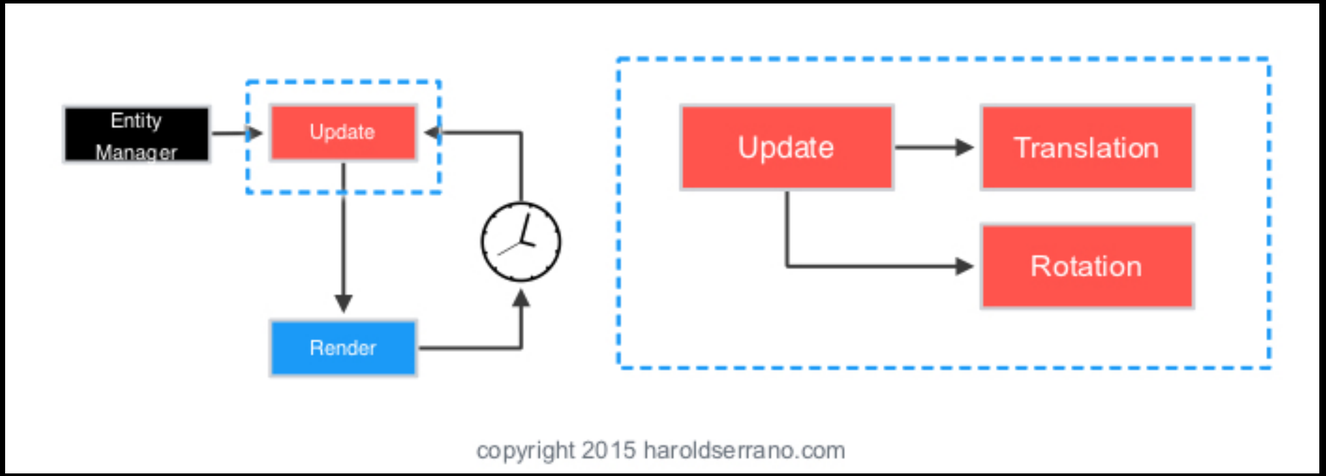

Math Engine
Addition
Subtraction
Dot Product
Cross Product
Transform
Invert
Transpose
Identity
Vector Class
Matrix Class
copyright 2015 haroldserrano.com

Rendering Engine - Render Objects

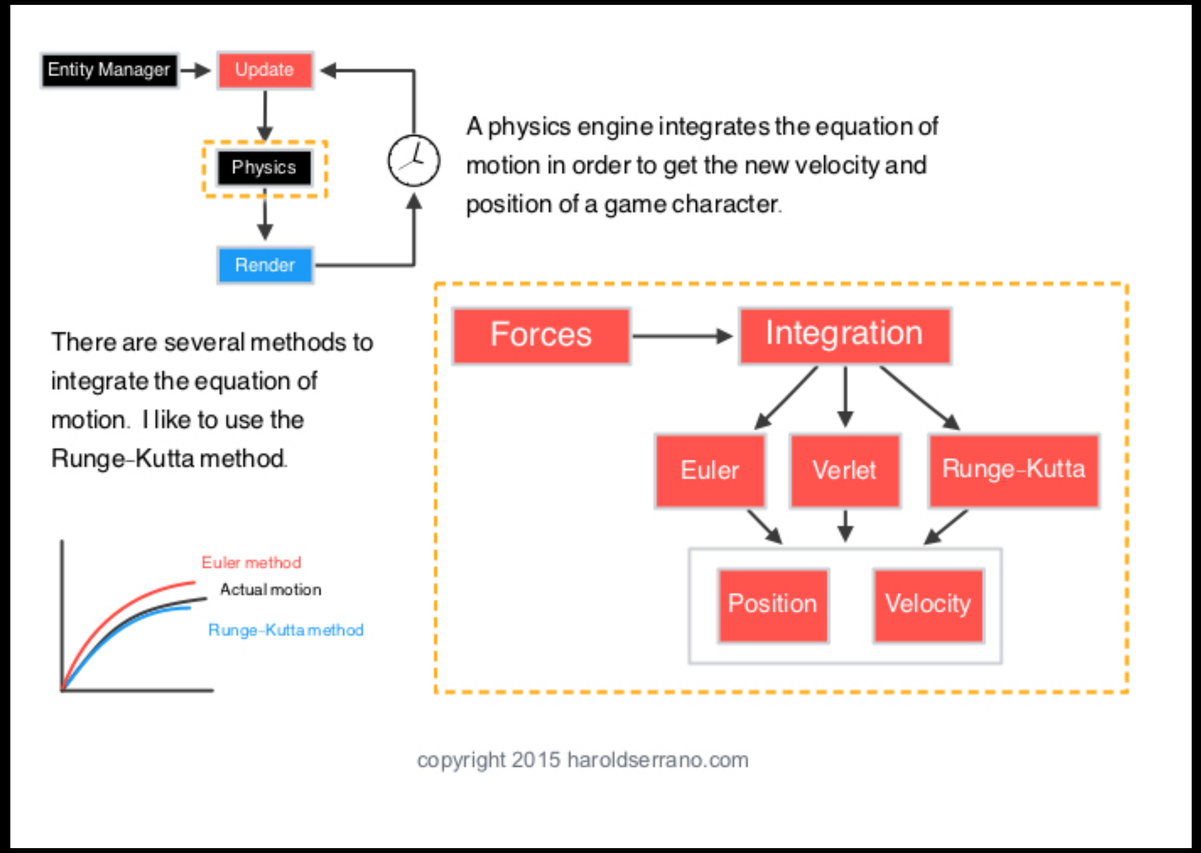


Scene Graph - Event Loop





Physics Engine



Now do you want to write
all that things

in one single file

like that

cocos2d-js.js

cc-Kproject > build > web-mobile > cocos2d-js.js > ...

```
89781       _global.CC_DEV = false;
89782       _global.CC_DEBUG = true;
89783       _global.CC_JSB = false;
89784       _global.CC_WECHATGAMESUB = false;
89785       _global.CC_WECHATGAME = false;
89786       _global.CC_QQPLAY = false;
89787       _global.CC_RUNTIME = false;
89788       _global.CC_SUPPORT_JIT = true;
89789       false;
89790       var engineVersion = "2.1.3";
89791       _global["CocosEngine"] = cc.ENGINE_VERSION = engineVersion;
89792     }), {} ]
89793   }, {}, [ 411 ]);
```

What is module?

- Modules are just clusters of code, but it should
  - highly self-contained with distinct functionality
  - can be shuffled, removed, or added as necessary
  - aims to lessen the dependencies on parts of the codebase as much as possible

What is module look like?

1. Single responsibility
2. Independency
3. Encapsulation
4. Small

Why?

1. Easy to read
2. Easy to debug
3. Easy to maintain
4. Reusability
5. Avoid namespace pollution

How to make module in javascript?

Once upon time

```
<script src="./lib/singleton.js"></script>
<script src="./lib/howler.js"></script>
<script src="./lib/pixi.js"></script>
<script src="./lib/pixi-layer.js"></script>
<script src="./lib/pixi-spine.js"></script>
<script src=“./lib/pixi-stats.js"></script>
<script src="./lib/memfs.js"></script>
<script src="./lib/gsap.js"></script>
```

- Problem
  - Lack of dependency
  - Pollution of global namespace.
  - Loading issue

Module Pattern Javascript

```
var Card = (function(){
    var _display = document.createElement("div");
    var _cover = document.createElement("div");
    var _image = document.createElement("img");
    _display.appendChild(_image);
    _display.appendChild(_cover);
    return {
        open: function(){
            _cover.style.display = "none";
            _image.style.display = "block";
        },
        close: function(){
            _cover.style.display = "block";
            _image.style.display = "none";
        }
    }
})();
```

Javascript Constructor

```
var Card = (function(){
    var _display = document.createElement("div");
    var _cover = document.createElement("div");
    var _image = document.createElement("img");
    _display.appendChild(_image);
    _display.appendChild(_cover);
    return {
        open: function(){
            _cover.style.display = "none";
            _image.style.display = "block";
        },
        close: function(){
            _cover.style.display = "block";
            _image.style.display = "none";
        }
    }
})();
```

```
// constructor
function Card(index) {
    this.element = document.createElement("div");
    this._cover = document.createElement("div");
    this._image = document.createElement("img");
    this.element.appendChild(this._image);
    this.element.appendChild(this._cover);
    this.index = index;
}
```

Javascript prototype

```
function Card() { };
console.log(Card.constructor);
// Function() { [native code] }
console.log(Card.prototype);
// {constructor: f}
Card.prototype.constructor === Card;
const card = new Card();
console.log(card.__proto__);
// {constructor: f}
card.__proto__ === Card.prototype;
const card2 = new Card();
card.__proto__ === card2.__proto__;
```

```
Card.prototype._active = true;
Card.prototype.active = {
    get() {
        return this._active;
    },
    set(value) {
        this._active = value;
        let displayStyle = value ? "block" : "none";
        element.style.display = displayStyle;
    }
}
Card.prototype.show = function (value) {
    this.active = value;
}
```

Javascript prototype

```
Card.prototype._active = true;
Card.prototype.active = {
    get() {
        return this._active;
    },
    set(value) {
        this._active = value;
        element.style.display = value ? "block" : "none";
    }
}
Card.prototype.show = function (value) {
    this.active = value;
}
var card = new Card();
```

Javascript class

```
// constructor
function Card(index) {
    this.element = document.createElement("div");
    this._cover = document.createElement("div");
    this._image = document.createElement("img");
    this.element.appendChild(this._image);
    this.element.appendChild(this._cover);
    this.index = index;
}
```

```
class Card {
    constructor(index) {
        this.index = index;
        this.element = document.createElement("div");
        this._cover = document.createElement("div");
        this._image = document.createElement("img");
        this.element.appendChild(this._image);
        this.element.appendChild(this._cover);
    }
}
```

Javascript class

```
Card.prototype._active = true;
Card.prototype.active = {
    get() {
        return this._active;
    },
    set(value) {
        this._active = value;
        let display = value ? "block" : "none";
        element.style.display = display;
    }
}
Card.SIZE = 200;
Card.prototype.show = function (value) {
    this.active = value;
}
```

```
class Card {
    constructor(index) {
        // ...
        this._active = true;
        // static property
        Card.SIZE = 200;
    }
    get active() { return this._active; }
    set active(value) {
        this._active = value;
        this.element.display = value?"block":"none";
    }
    show(value){
        this.active = value;
    }
}
```

Make our game engine

Rendering

Engine

1. Node
2. Sprite
3. Label

- Position: x, y
- width, height
- scaleX, scaleY
- active