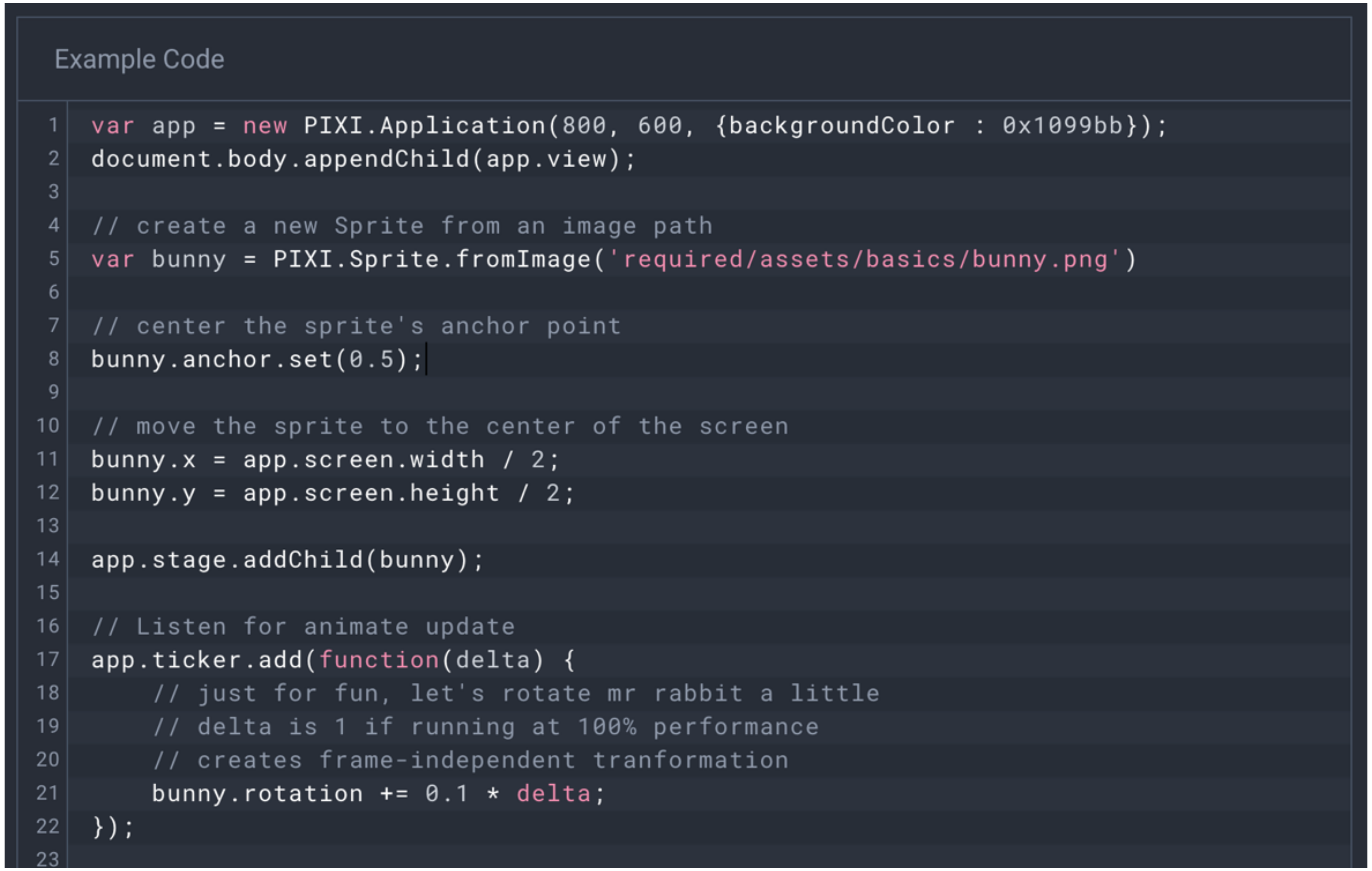
O2
OPEN ORIENTED
凹 凸 实 验 室

PixiJS-core源码学习

Simba

例子：　http://pixijs.io/examples/#/basics/basic.js


Example Code

```
var app = new PIXI.Application(800, 600, {backgroundColor : 0x1099bb});
document.body.appendChild(app.view);

// create a new Sprite from an image path
var bunny = PIXI.Sprite.fromImage('required/assets/basics/bunny.png')

// center the sprite's anchor point
bunny.anchor.set(0.5);

// move the sprite to the center of the screen
bunny.x = app.screen.width / 2;
bunny.y = app.screen.height / 2;

app.stage.addChild(bunny);

// Listen for animate update
app.ticker.add(function(delta) {
    // just for fun, let's rotate mr rabbit a little
    // delta is 1 if running at 100% performance
    // creates frame-independent tranformation
    bunny.rotation += 0.1 * delta;
});

```

在不使用框架的情况下如何实现?

画布	图片元素	update
canvas	drawImage	RAF

目录

display

$$\begin{tikzcd}
\text{Container} & \text{DisplayObject} \arrow[l, "\text{extend}"'] & \text{EventEmitter} \arrow[l, "\text{extend}"']
\end{tikzcd}$$

stage

ticker

Ticker

帧控制器

renderers

$$\begin{tikzcd}
\text{CanvasRenderer} & & \\
& {} & \text{SystemRenderer} \arrow[l, "\text{extend}"'] \\
\text{WebGLRenderer} & &
\end{tikzcd}$$

renderer

目录

graphics

$$\begin{tikzcd}\text{Graphics} & \text{Container} \arrow[l, "\text{extend}"']\end{tikzcd}$$

绘制各种原始形状

sprites

$$\begin{tikzcd}\text{Sprites} & \text{Container} \arrow[l, "\text{extend}"']\end{tikzcd}$$

绘制各种纹理

目录

math

Point
ObservablePoint
Matrix
GroupD8

Circle
Ellipse
Polygon
Rectangle
RoundedRectangle

算法/几何形状

textures

BaseTexture	The source can be - image url, image element, canvas element.
VideoBaseTexture	A texture of a [playing] Video

纹理

目录

Application.js

入口

autoDetectRenderer.js

```
if (!forceCanvas && utils.isWebGLSupported())
    {
        return new WebGLRenderer(options, arg1, arg2);
    }

return new CanvasRenderer(options, arg1, arg2);
```

const.js　　settings.js

Ticker

TickerListener

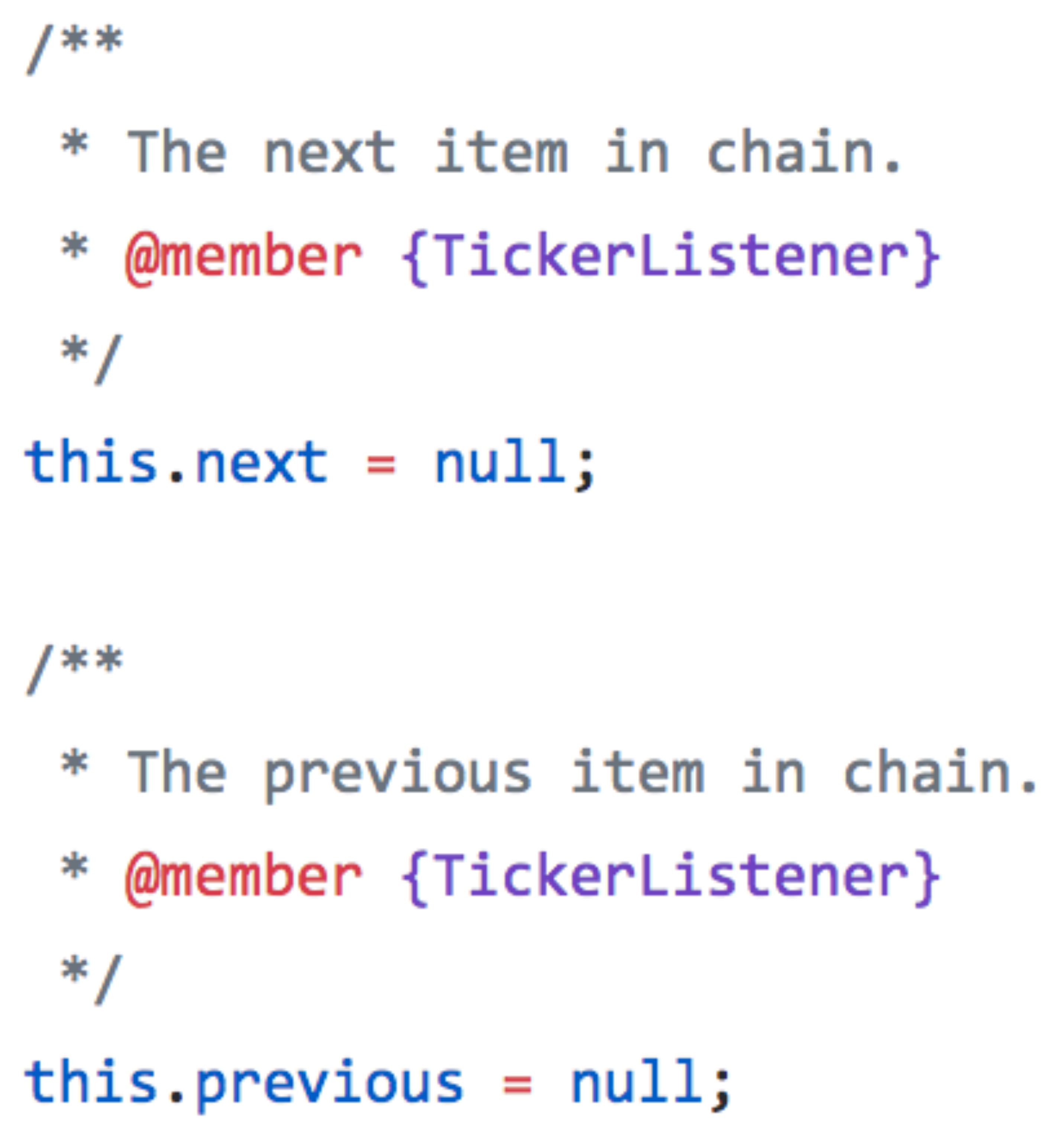
constructor(fn, context = null, priority = 0, once = false)

fn:　update时执行

context:　fn的上下文

priority:　优先级

once:　是否只执行一次



```
/**
 * The next item in chain.
 * @member {TickerListener}
 */
this.next = null;

/**
 * The previous item in chain.
 * @member {TickerListener}
 */
this.previous = null;
```

TickerListener链

```
Ticker._head = new TickerListener(null, null, Infinity);

Ticker.add =  this._addListener(new TickerListener(fn, context, priority));

Ticker.add(this.render, this, UPDATE_PRIORITY.LOW);
```

TickerListener链的更新

update()

```
while (listener)
{
    listener = listener.emit(this.deltaTime);
}
```

this._tick()

```
this._tick = (time) =>
{
   this._requestId = null;

   if (this.started)
   {
       // Invoke listeners now
       this.update(time);
       // Listener side effects may have modified ticker state.
       if (this.started && this._requestId === null && this._head.next)
       {
           this._requestId = requestAnimationFrame(this._tick);
       }
   }
};
```

deltaTime

elapsedMS = 1 / settings.TARGET_FPMS 16.66ms

deltaTime = elapsedMS * settings.TARGET_FPMS

Container

Text Sprite Graphics

$$\begin{tikzcd}\text{Container} & \text{DisplayObject} \arrow[l, "\text{extend}"']\end{tikzcd}$$

addChild()

removeChild()

…

```
for (let i = 0, j = this.children.length; i < j; ++i)
 {
     this.children[i].renderCanvas(renderer);
 }
```

DisplayObject

alpha visible filterArea transform _filters _bounds

Transform

Transform ← TransformBase

position

scale

skew(rotation)

pivot(Point)

二维矩阵变换

```
updateTransform(parentTransform)
{
    const pt = parentTransform.worldTransform;
    const wt = this.worldTransform;
    const lt = this.localTransform;

    // concat the parent matrix with the objects transform.
    wt.a = (lt.a * pt.a) + (lt.b * pt.c);
    wt.b = (lt.a * pt.b) + (lt.b * pt.d);
    wt.c = (lt.c * pt.a) + (lt.d * pt.c);
    wt.d = (lt.c * pt.b) + (lt.d * pt.d);
    wt.tx = (lt.tx * pt.a) + (lt.ty * pt.c) + pt.tx;
    wt.ty = (lt.tx * pt.b) + (lt.ty * pt.d) + pt.ty;

    this._worldID ++;
}
```

THANKS

FOR YOUR WATCHING

OPEN ORIENTED

凹 凸 实 验 室