Yet another way to animate in Angular

Douglas Duteil
@douglasduteil

But first...

ng-europe

Patrick Aljord

(organizer)

Katiuska Gamero

(co-organizer)

Josh Moont

(collaborator)

Olivier Louvignes

(collaborator)

Thierry Lau

(collaborator)

Julien Bouquillon

(collaborator)

Cyril Balit

(collaborator)

Douglas Duteil

(collaborator)

Thanks

Yet another way to animate in Angular

Douglas Duteil
@douglasduteil

Me, Myself and I

2014 - 24 y/o

- Paris 8 University
- SFΞIR

Me, Myself and I

angular-ui/

- ui-codemiror
- ui-ace
- ui-layout

- ui-utils
- angular-ui-publisher

Me, Myself and I

First mission : web site

First mission : code

First mission : code

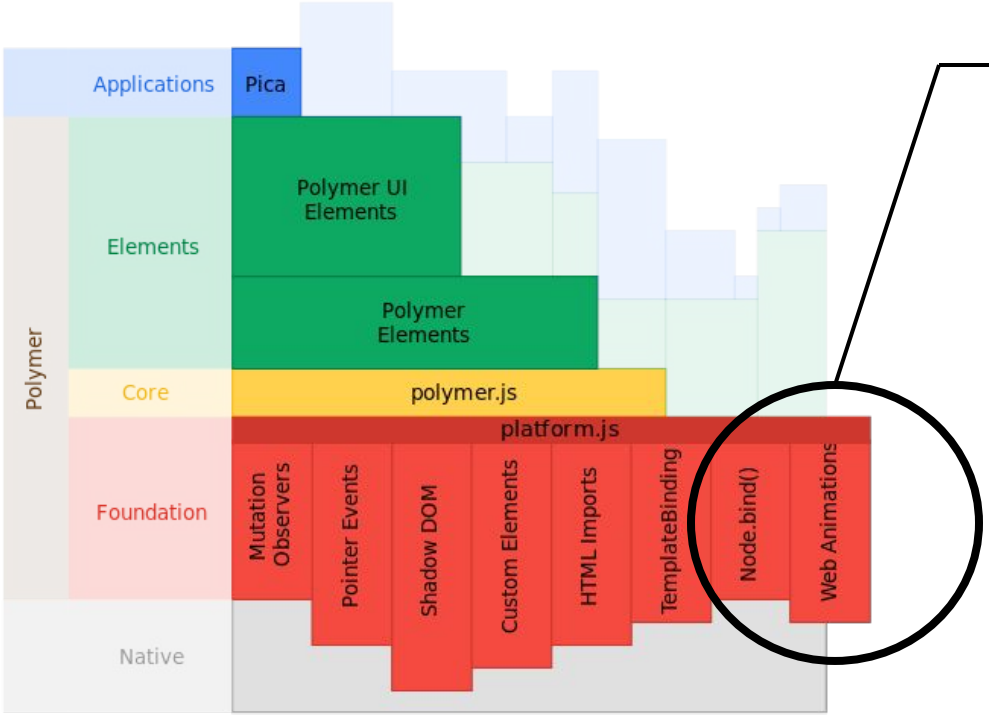
Back in November 2013

- Angular 1.1.5 but not using ngAnimate
- Manual “post-event” processing
- A lot of massive link directives
  - Purely animate elements (TweenMax)
  - Animate transitions to view (no ng-view or ui-view)

First mission : dude

Entering W3C Web Animations

Entering W3C Web Animations



Web Animation

Gihub repos :

- w3c/web-animations
- web-animations/web-animations-js
- web-animations/web-animations-next

W3C Web Animations

- Animation group (parallel, sequential)
- Animation player access

- Unify under the same API
  - CSS Transition / CSS Animation
  - SVG Animation / SMIL
  - requestAnimationFrame()

Web Animations

```css
@keyframes fadeOutDown {
  0% {
    opacity: 1;
  }

  100% {
    opacity: 0;
    transform: translate3d(0, 100%, 0);
  }
}

.foo {
  animation: fadeOutDown 2s;
}
```

CSS

```js
foo.animate(
  // fade out down effect
  [
    { opacity : 1 },
    {
      opacity : 0,
      transform: 'translate3d(0, 100%, 0)'
    }
  ],
  // timing
  {
    duration: 2000 // ms
  }
);
```

JS

```
/**
 * @type {Animation}
 */
var anim = new Animation(
  // target
  foo,
  // fade out down effect
  [
    { opacity : 1 },
    {
      opacity : 0,
      transform: 'translate3d(0, 100%, 0)'
    }
  ],
  // timing
  {
    duration: 2000 // ms
  }
);
```

```
/**
 * Just run the animation on the document
 * timeline.
 */
foo.ownerDocument.timeline.play(anim);

////

/**
 * @type {AnimationPlayer}
 */
var player = document.timeline.play(anim);
```

Web Animations : player

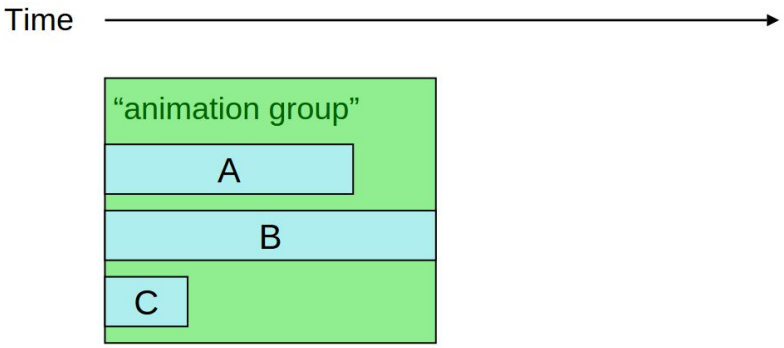
```css
.foo {
  animation-play-state: running;
  animation-play-state: paused;
}
```

CSS

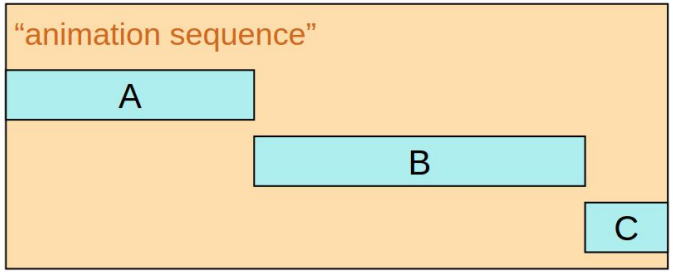
```js
player.play();
player.pause();

player.reverse();
player.finish();
player.cancel();

/**
 * @type {EventHandler}
 * @removed after 5 June 2014 in Editor's Draft
 */
player.onfinish;

// for seeking
player.currentTime;
```

JS

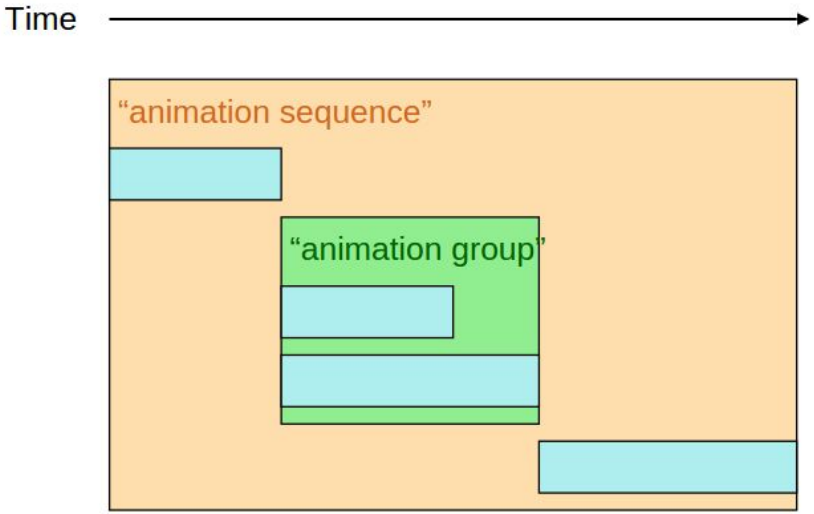
Web Animations : timing group



(a)



(b)



Web Animations : SVG

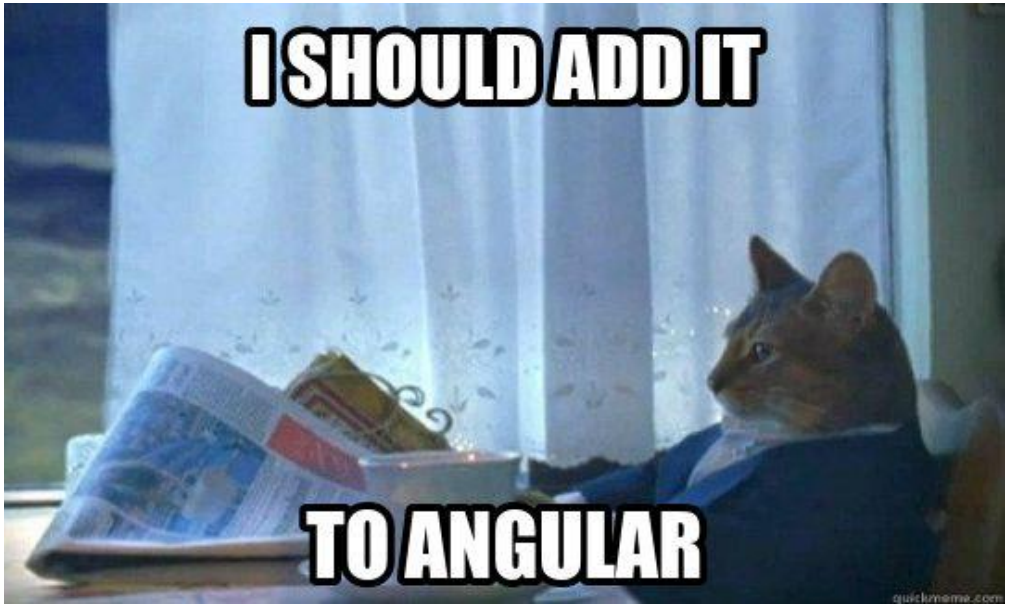
```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <defs>
    <path id=path d="M 100,100 a 75,75 0 1,0 150,0 a 75,75 0 1,0 -150,0"/>
  </defs>
</svg>
<script>
  var animFunc = new MotionPathEffect(document.querySelector('#path').pathSegList);
  var animation = new Animation(targetElement, animFunc, 2000);
</script>
```

Web Animations : custom

```
element.animate(
  // EffectCallback
  function sample(timeFraction, currentTarget, animation){
    // ...
  },
  10000
);
```


AWESOME

Web Animations And Angular



WaAaAaaAaaa...

WaAaAaaAaaaAaaaaaAaaaaaaaaA...

web animation and angular

WaAaaaaAaaaAaaaaaAaaa...

WaAaAaaAaaaaAaaaaaaAaaaaaaaaaA…

Ideas:

- Skeletal animation (separate skin and bones)
- Functional classes

WaAaAaaAaaaaAaaaaaaAaaaaaaaaaA...

```
var HACK_ANGULAR_MODULE = angular.module;
angular.module = function fakeModule() {
  return angular.extend(HACK_ANGULAR_MODULE.apply(this, arguments), {

    // Add a "waAnimation" function to add new animations ;)
    waAnimation: function () {
      this._invokeQueue.push(["waAnimationProvider", "register", arguments]);
      return this;
    }

  });
};

angular.module('waAnimate', [])
  // Use 'waAnimate' provider to register animations
  .provider(waAnimation.name, waAnimation)
```

WaAa : Skeletal animation


```
{
  "bones" :
[{"parent":-1,"name":"Bone","pos":[1.62171,1.32006,1.77362e-07],"rotq":[0,0,0,1]},{"parent":-1,"name":"Bone.001","pos":[-1.52522,
1.32006,1.77362e-07],"rotq":[0,0,0,1]}],
  "skinIndices" : [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0],
  "skinWeights" : [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0],
  "animation" :
{"name":"ArmatureAction","fps":24,"length":4.08333,"hierarchy":[{"parent":-1,"keys":[{"time":0,"pos":[1.62171,1.32006,1.77362e-07
],"rot":[0,0,-0,1],"scl":[1,1,1]},{"time":0.0416667,"pos":[1.62171,1.32006,1.77362e-07],"rot":[0,0,-0,1]},{"time":2,"pos":[1.6217
1,4.9867,1.77362e-07],"rot":[0,0,-0,1]},{"time":4.08333,"pos":[1.62171,1.30368,1.77362e-07],"rot":[0,0,-0,1],"scl":[1,1,1]}]},{"p
arent":0,"keys":[{"time":0,"pos":[-1.52522,5.00929,1.77362e-07],"rot":[0,0,-0,1],"scl":[1,1,1]},{"time":2,"pos":[-1.52522,1.23815
,1.77362e-07],"rot":[0,0,-0,1]},{"time":4.08333,"pos":[-1.52522,4.81948,1.77362e-07],"rot":[0,0,-0,1],"scl":[1,1,1]}]}]}
}
```

WaAa : Functional classes

```
<div
    class="square"
    ng-repeat="s in squares"
    ng-click="doIt=!doIt; center=$index"
    ng-class=" {
      'fooAnimation({
        pos    : $index,
        length : squares.length,
        center : center
      })' : doIt
    }">
</div>
```

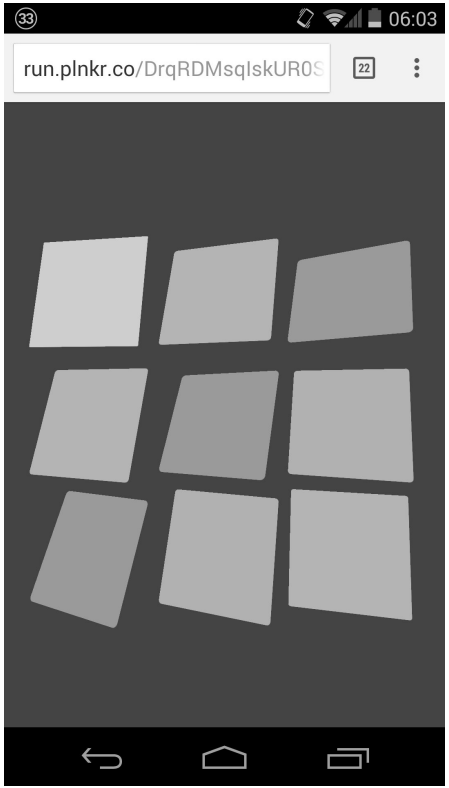
So…

So…

Work in progress

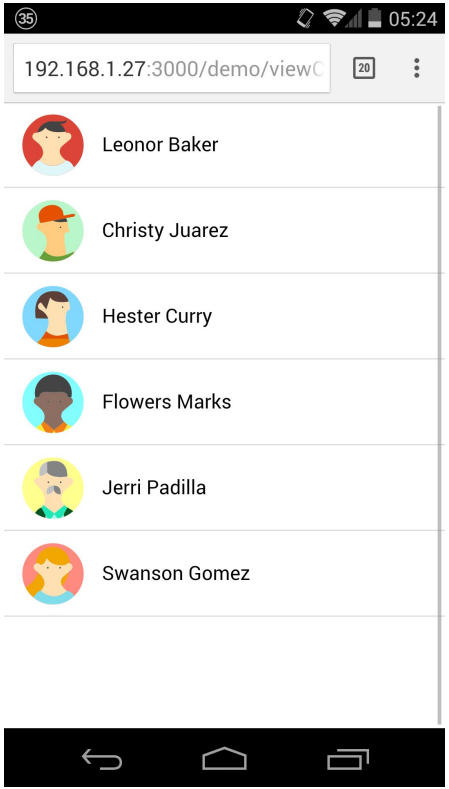
- Hard to deal with scope life
  - pre-analysing
  - post digest execution
- Player access gap
  - [hack] of the functional class

Demos



Square wave

Demos



Material Design List Transition

Inspired by http://codepen.io/nroviw/details/kCazJ/

Demos

Image defragmentation

Inspired by http://codepen.io/natewiley/pen/pFABJ

Merci

Thanks

Douglas Duteil
@douglasduteil