



# OptimizeIt!

## ActionScript Tips for iPhone Games

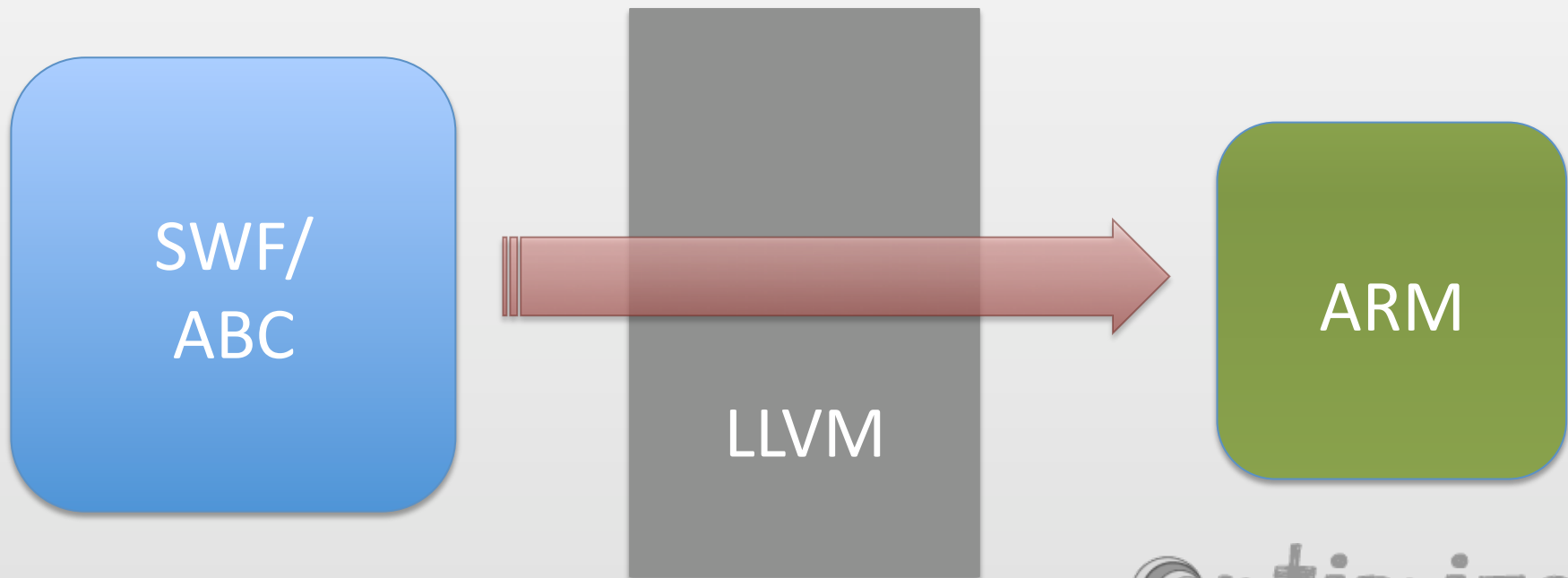
Renaun Erickson  
Platform Evangelist  
Adobe Systems  
2010.03.10

Twitter - renaun  
Blog renaun.com



## Packager for iPhone - PFI

- **LLVM**
  - **Capable of generating machine code for various targets, including x86 and ARM processors**



# Packager

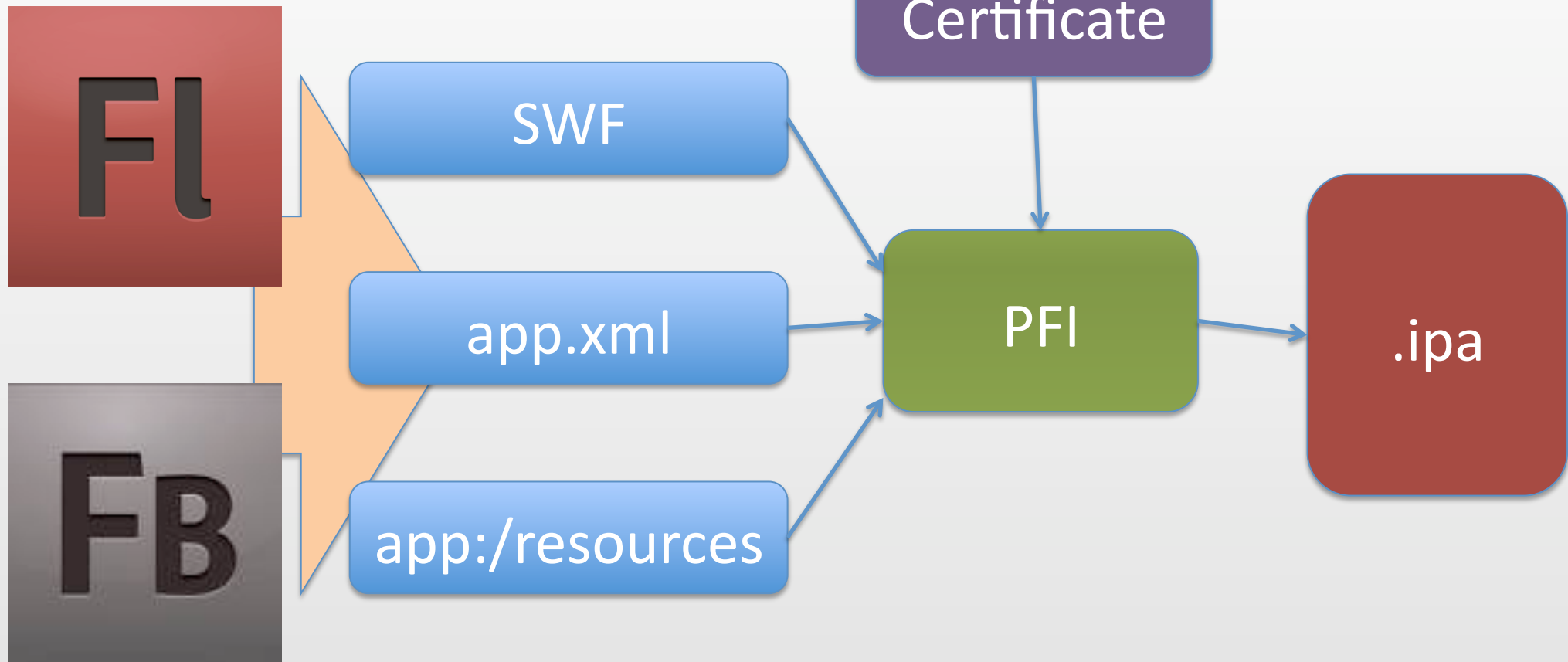


**Develop**

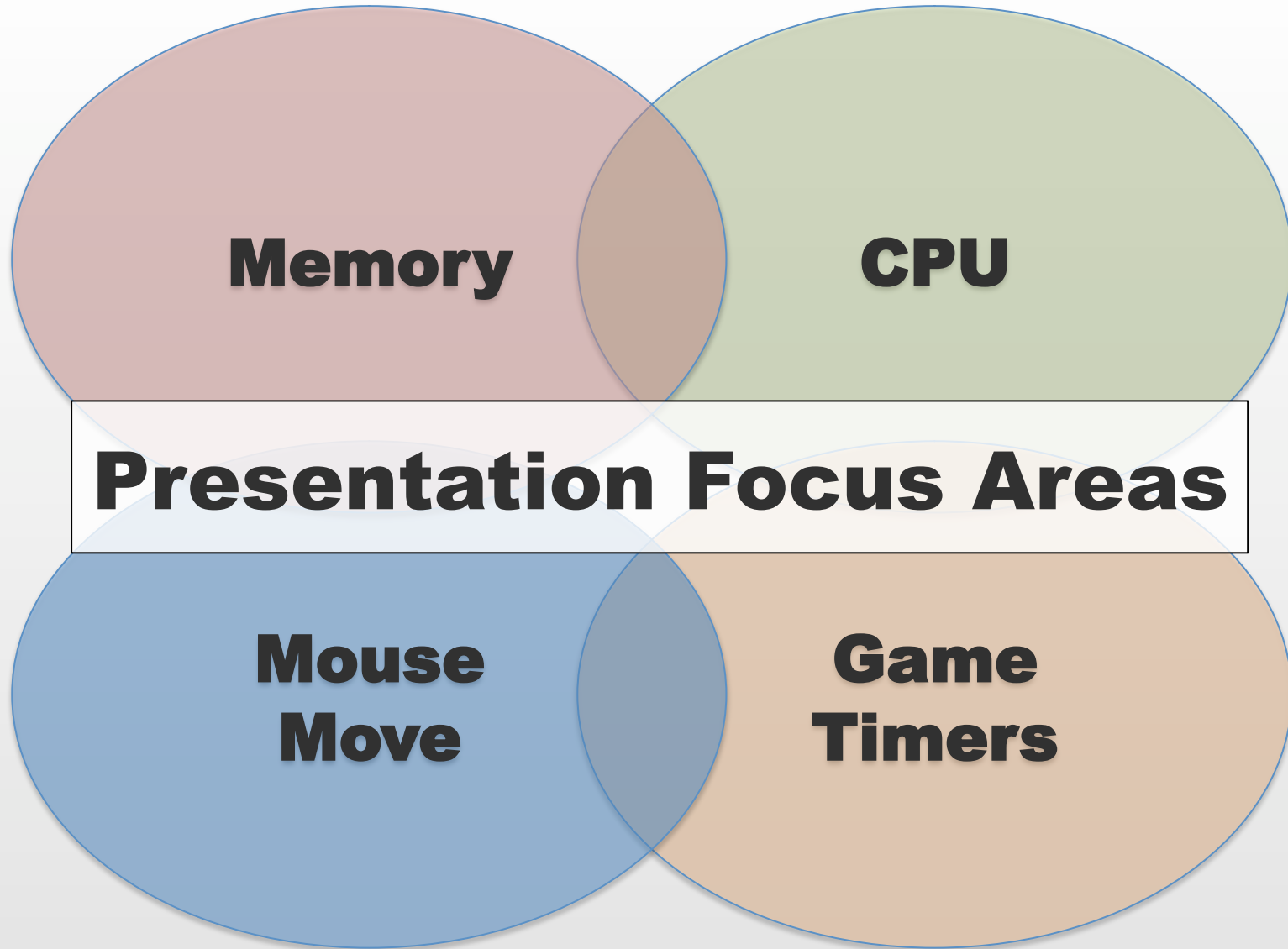
**Publish**

**Package**

**Deploy**



# Low Down



- **Devices**
  - **iPhone 3GS**
  - **iPod Touch 2<sup>nd</sup> Generation**
  - **iPod Touch 1<sup>st</sup> Generation**
- **Approach**
  - **Start with baseline and change one variable**
  - **Looking for trends and general areas**
- **Things to Remember**
  - **Real world more complex then baseline**
  - **Hardware limits: CPU, GPU, and RAM**

## GPU, Memory, and CPU

iPhone 3GS

PowerVR SGX

256 MB

600 MHz  
ARM Cortex-A8

iPod Touch 2<sup>nd</sup>

PowerVR MBX Lite

128 MB

533 MHz  
ARM11/ARM7

iPod Touch 1<sup>st</sup>

PowerVR MBX Lite

128 MB

400/412 MHz  
ARM11



MAssetEmbed

```
[Embed(source="assets/Spinner.png")]  
private var asset:Class;
```

- **Embed**
  - **Uses mxmlc**
  - **Exports as mx.core.BitmapAsset**
  - **A lot of class dependencies**

FlashCS4Assets.fla / MAssetSWC



SpinnerBD

Export: SpinnerBD



Export SWC

- **SWC**
  - **Flash CS Pro**
  - **Export as flash.display.BitmapData**

## To Embed or To SWC



**Asset**



**SWF  
Asset/SWC**



**SWF  
Asset/Embed**





## DisplayObject Base Sizes

236 bytes

**Shape/Bitmap**



414 bytes

**Sprite**



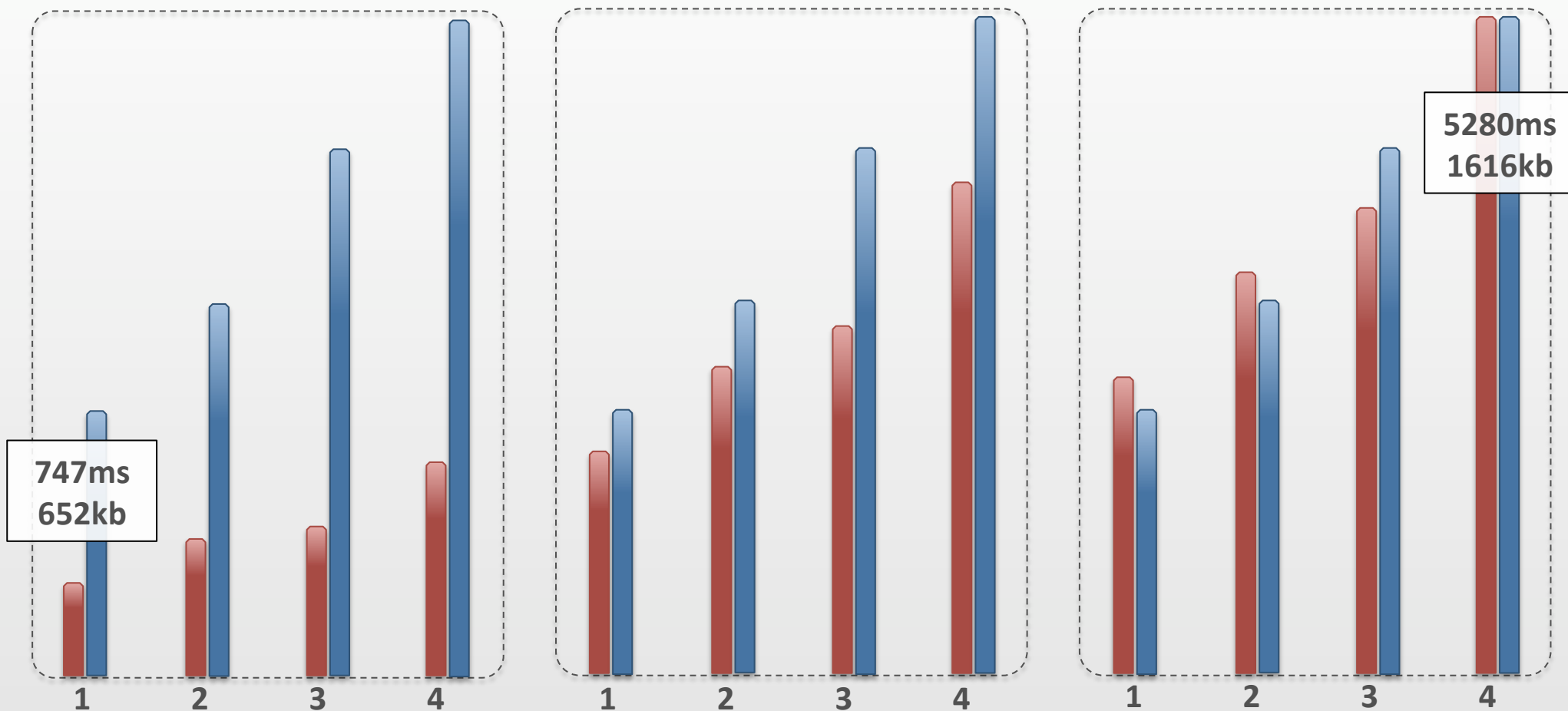
440 bytes

**MovieClip**



## Create/Add 2000 Objects

iPhone 3GS

iPod Touch 2<sup>nd</sup>iPod Touch 1<sup>st</sup>

1 - MTBitmapData

2 - MTShape

3 - MTSpriteBD

4 - MTSprite

- Time in milliseconds

- Size in kilobytes

# Memory



- **Packager for iPhone**
  - **GPU Acceleration requires caching**
  - **BitmapData does not center**
  - **Shape currently does not behave as expected when cached**

MTGSpriteBD

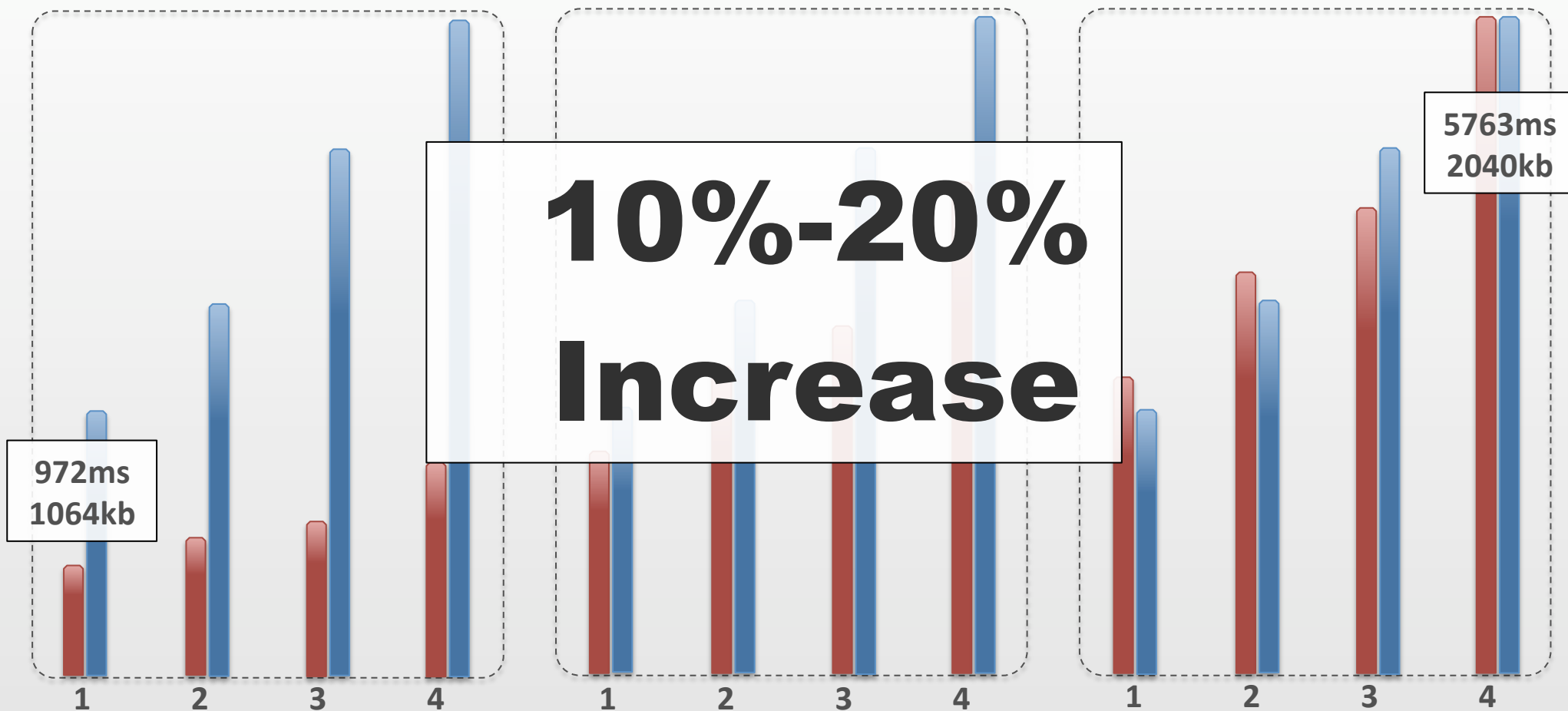
- **SpriteBD**

```
// SpinnerBD is 30x30 pixels
var bits:BitmapData = new SpinnerBD(0, 0);
var cacheMatrix:Matrix = new Matrix();
var matrix:Matrix = new Matrix(1, 0, 0, 1, -15, -15);
var obj:Sprite = new Sprite();
obj.graphics.beginBitmapFill(bits, matrix, false, true);
obj.graphics.drawRect(-15, -15, 30, 30);
obj.graphics.endFill();
obj.cacheAsBitmap = true;
obj.cacheAsBitmapMatrix = cacheMatrix;
addChild(obj);
```



## Create/Add 2000 Objects with GPU caching

iPhone 3GS

iPod Touch 2<sup>nd</sup>iPod Touch 1<sup>st</sup>

1 - MTGBitmapData

2 - MTGShape

3 - MTGSpriteBD

4 - MTGSprite

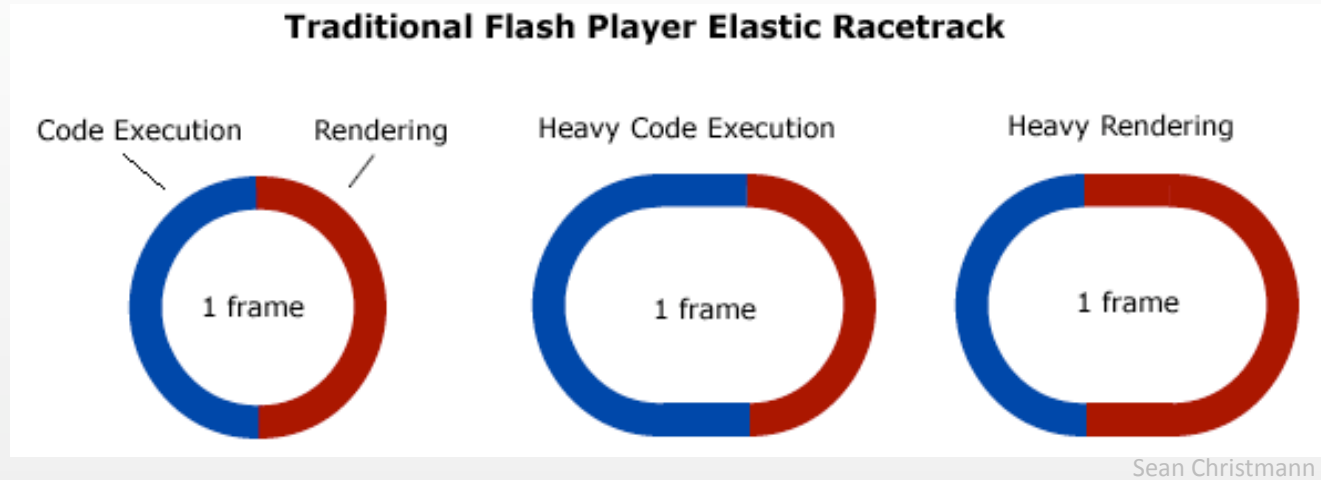
- Time in milliseconds

- Size in kilobytes

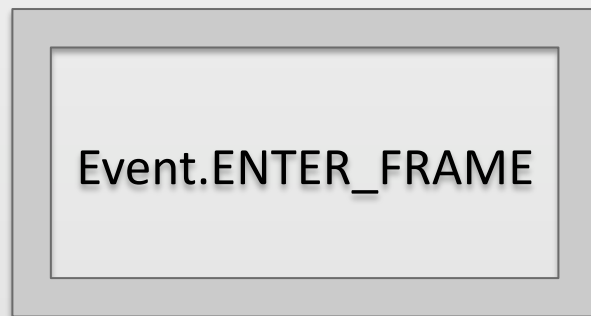
# Timing



## Elastic Racetrack



## Game Timers





# Game Timer

- **Event.ENTER\_FRAME**
  - **If code execution runs longer than a frame time slice it affectively slows down the animation**
- **Timer / updateAfterEvent()**
  - **Control logic and render**
  - **Assumes rendering will take longer than logic**



# Game Timer

## Optimized AS3 Game Timer Loop

<http://www.8bitrocket.com/>

**OptimizeIt.as**

©OptimizeIt!



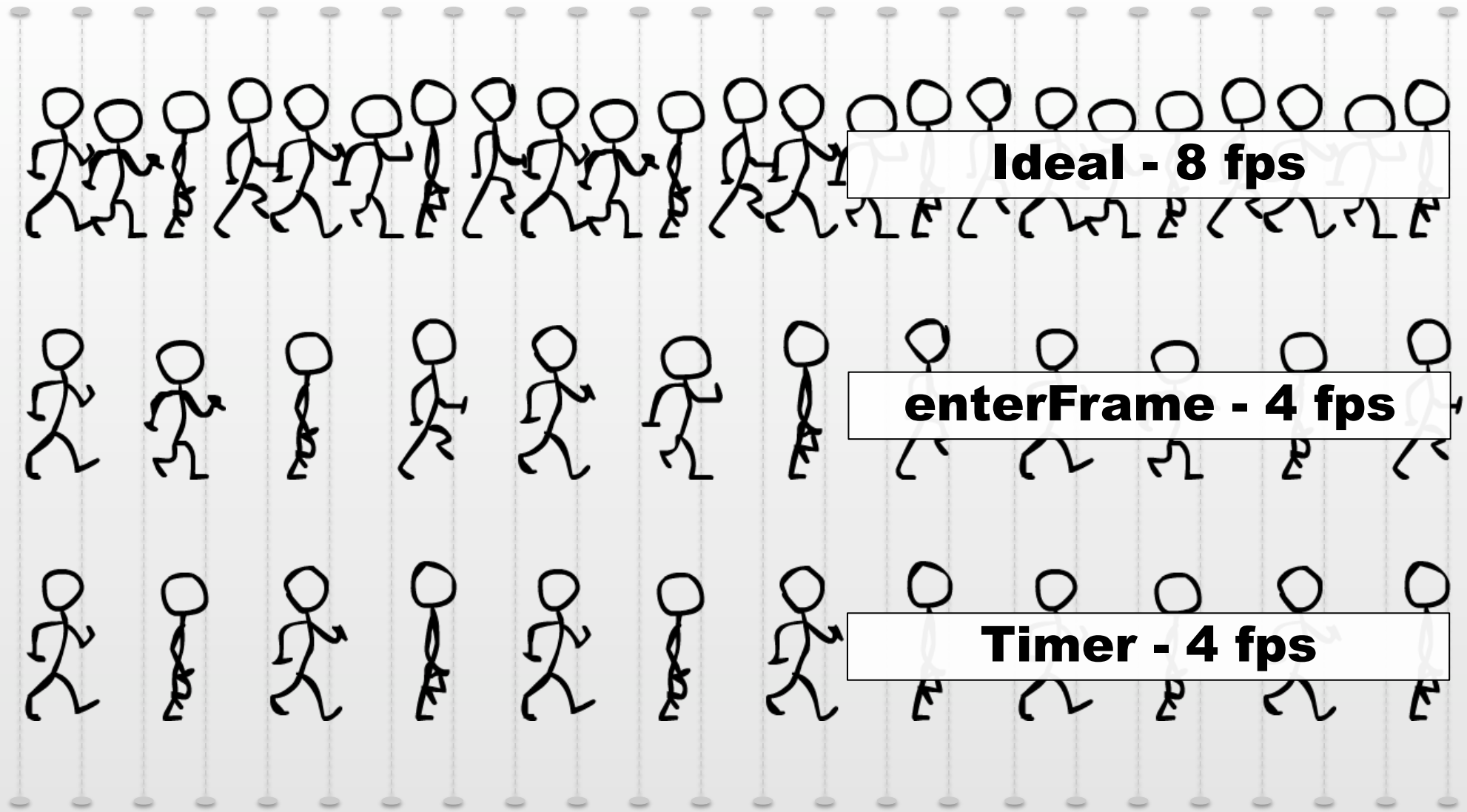
# Game Timer



0s

1s

2s





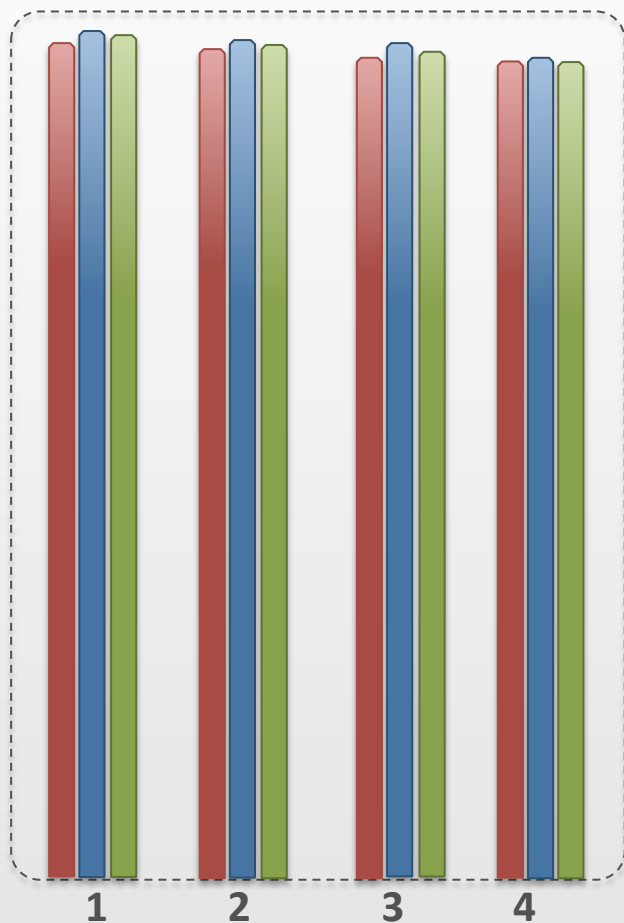
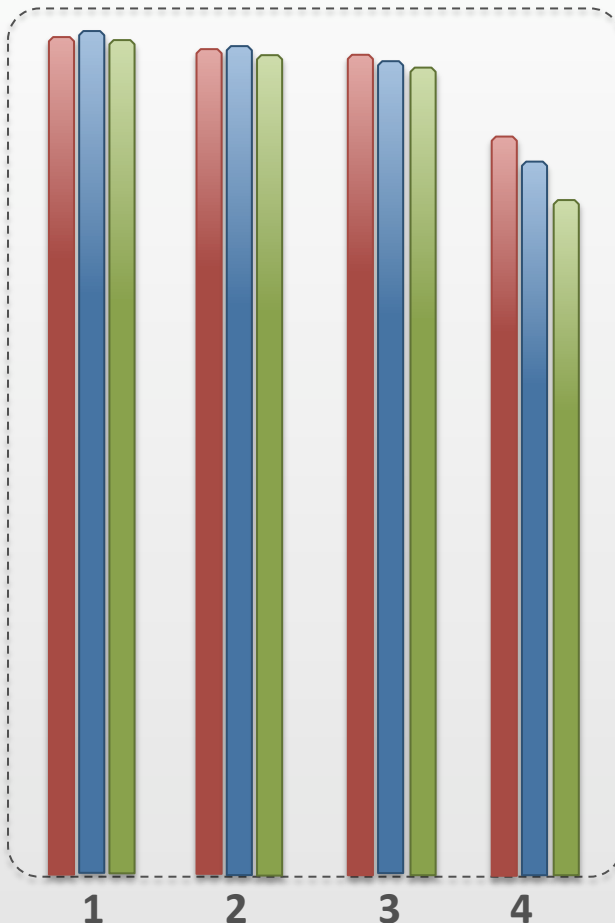
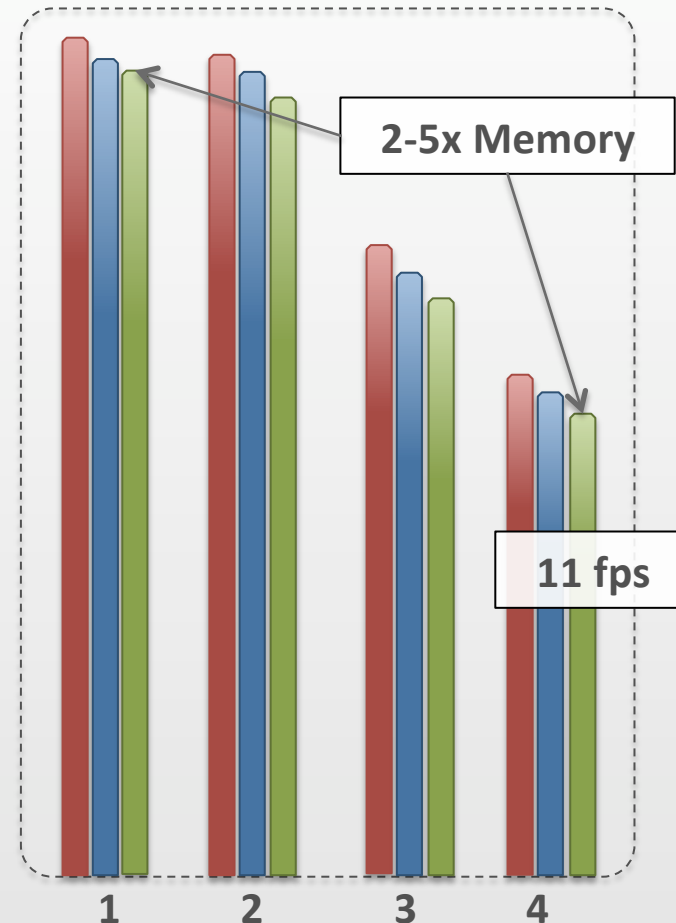
# Choose



- **Event.ENTER\_FRAME**
  - **Performs 5-10% better on slower devices**
  - **Possible less memory/data objects**
  - **Frames are always rendered, so everything slows down**
  - **Less memory comparing same test**
- **Timer / updateAfterEvent()**
  - **Animation consistent across different CPU speeds**
  - **Slowness has a drop frame affect**

## Rotating Bitmaps at 20 fps

iPhone 3GS

iPod Touch 2<sup>nd</sup>iPod Touch 1<sup>st</sup>

1 – 50 Objects

2 – 100 Objects

3 – 150 Objects

4 – 200 Objects

- enterFrame CGSpriteBDF

- Timer CGSpriteBD

- Timer CGSprite

# CPU Tests

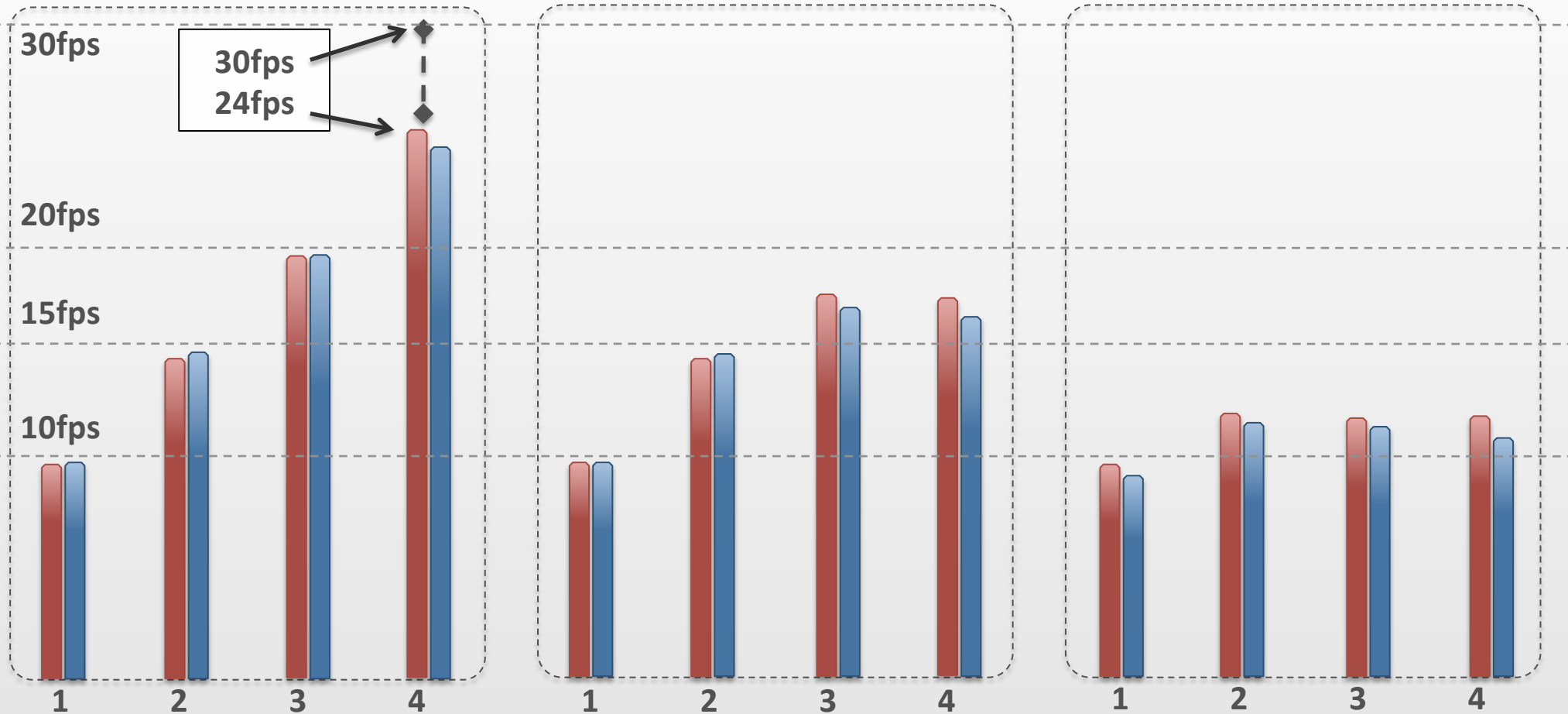


## Rotating 200 Objects at Various FPS

iPhone 3GS

iPod Touch 2<sup>nd</sup>

iPod Touch 1<sup>st</sup>



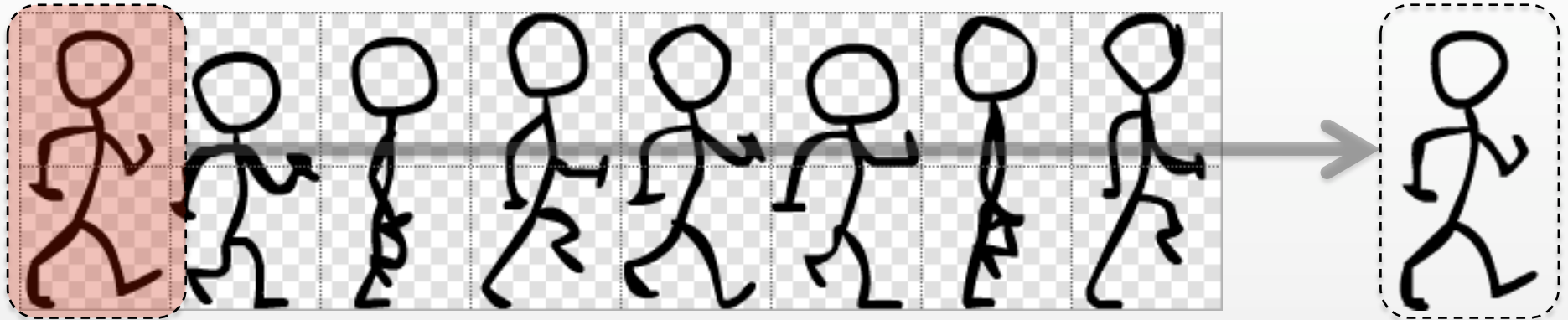
1 – 10 fps    2 – 15 fps    3 – 20 fps    4 – 30 fps

- enterFrame CGSpriteBDF

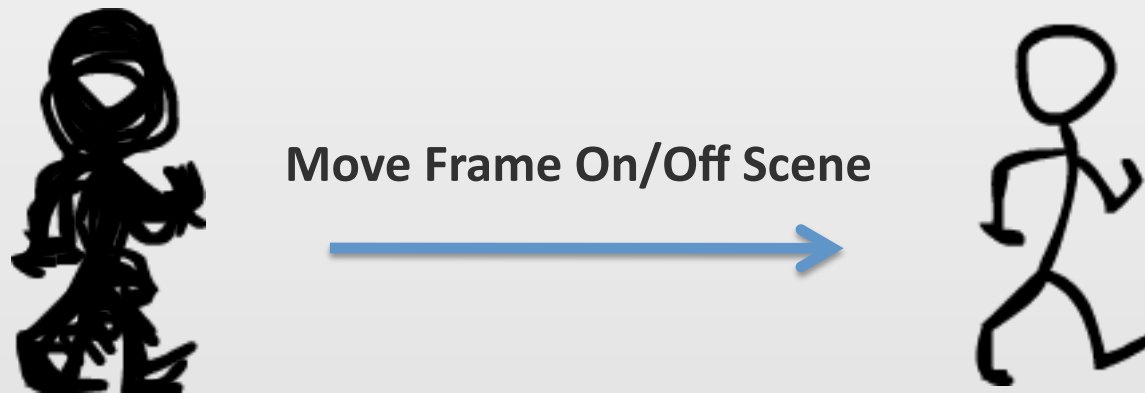
- Timer CGSpriteBD

OptimizeIt!

## Blitting / copyPixels()



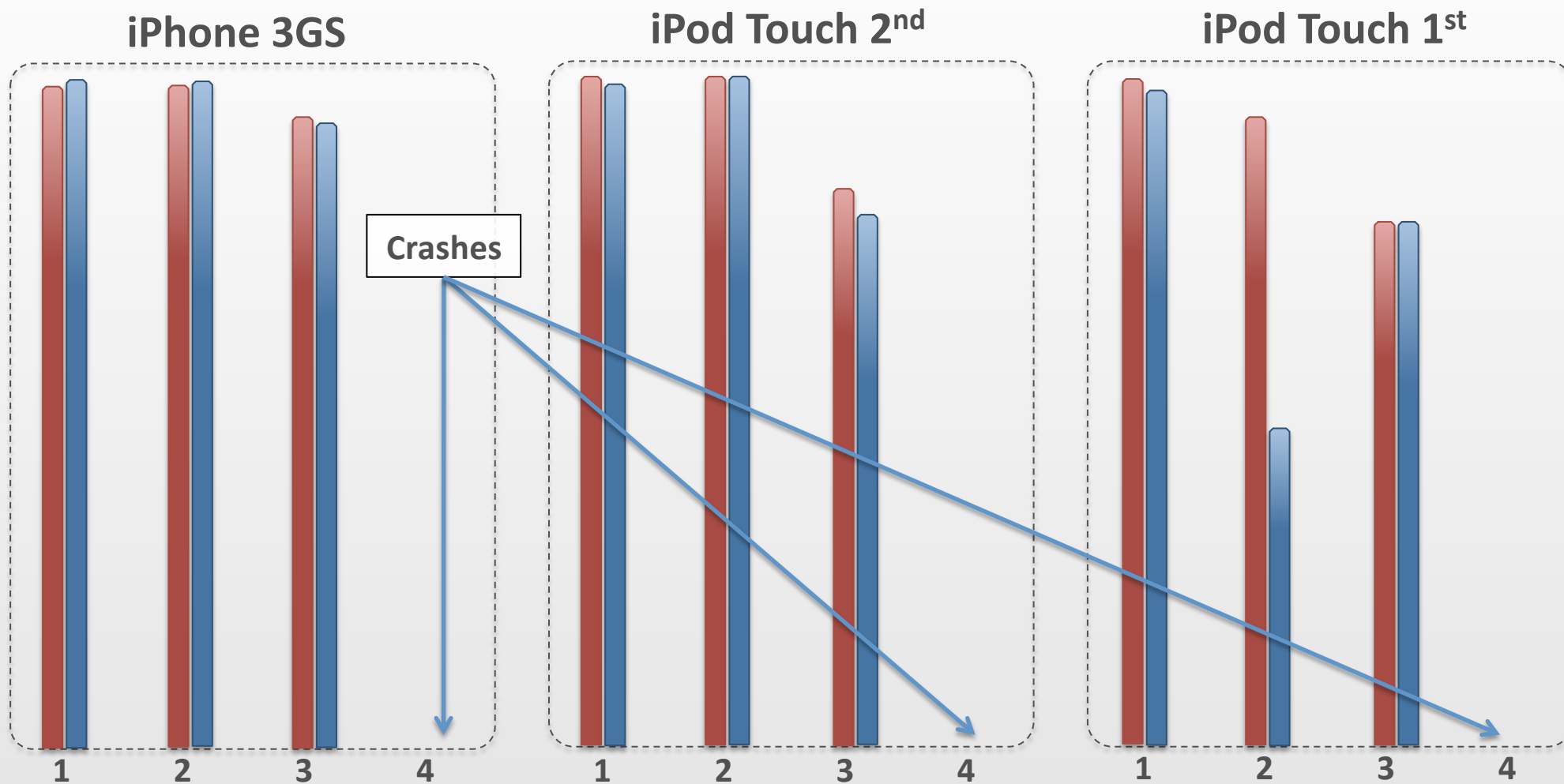
## SpriteDB Cached Frames



# CPU Tests



## Animation of 20 and 40 Objects



- enterFrame

- Timer

1 – CWalkingBlit(F) 20

2 – CWalkingBlit(F) 40

3 – CGWalkingBD(F) 20

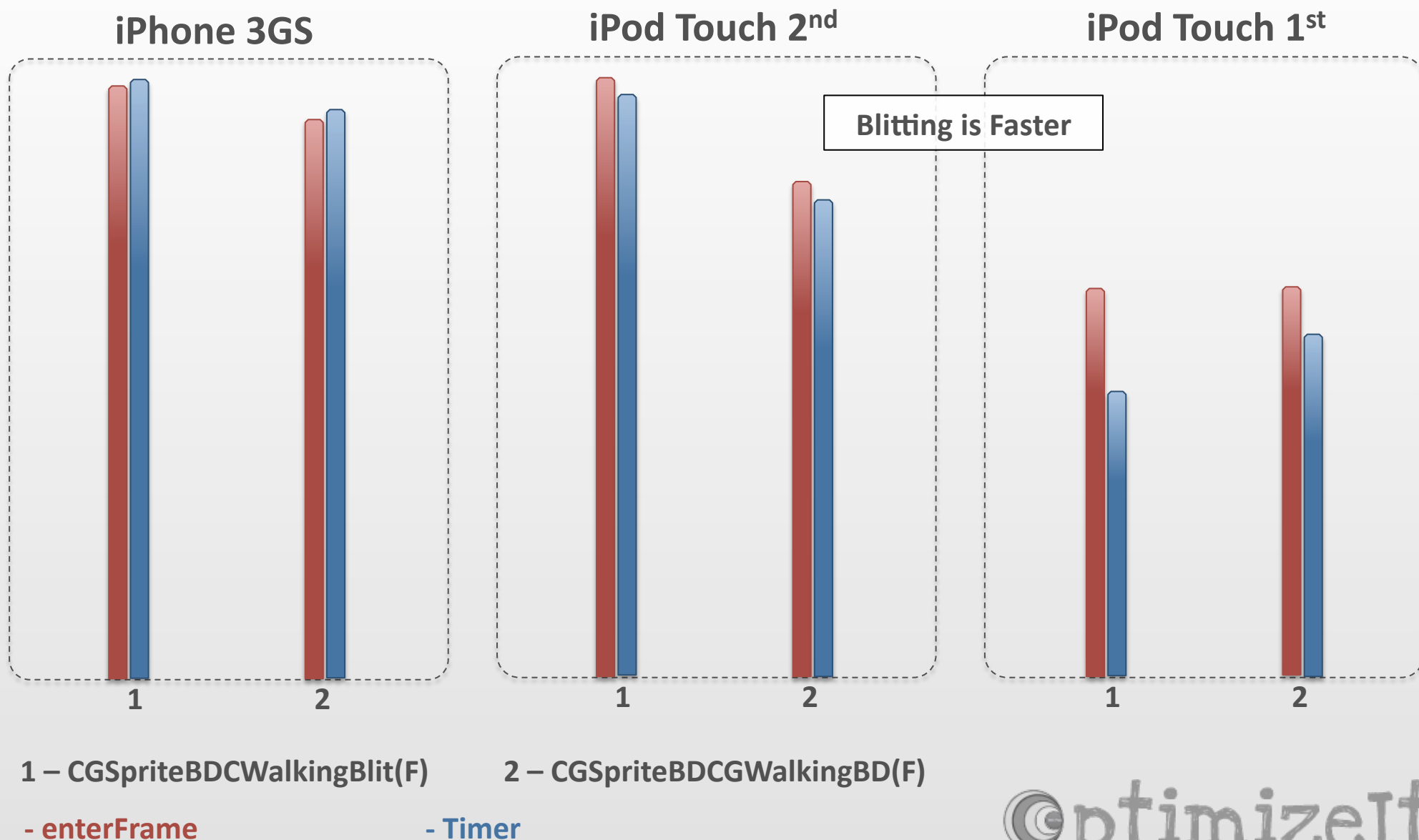
4 – CGWalkingBD(F) 40

# Choose



- **Pixel GPU Limits**
  - **Rotating Case**
    - **$200 * 30 * 30 = 180k$**
  - **Animation**
    - **$40 * 73 * 141 * 8 = 3.23 \text{ mil}$**

## 50 Rotating / 20 Animated

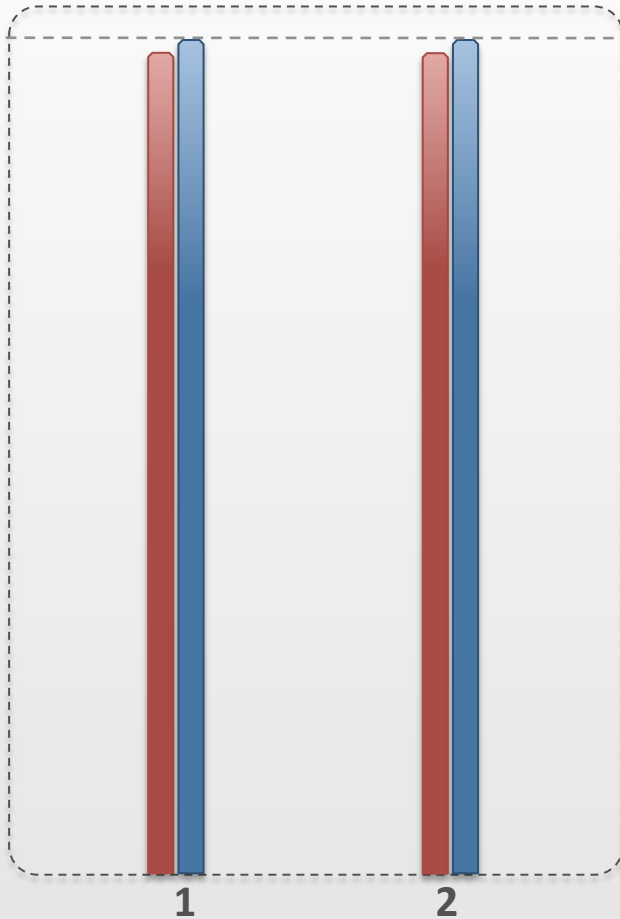


# Mouse Move

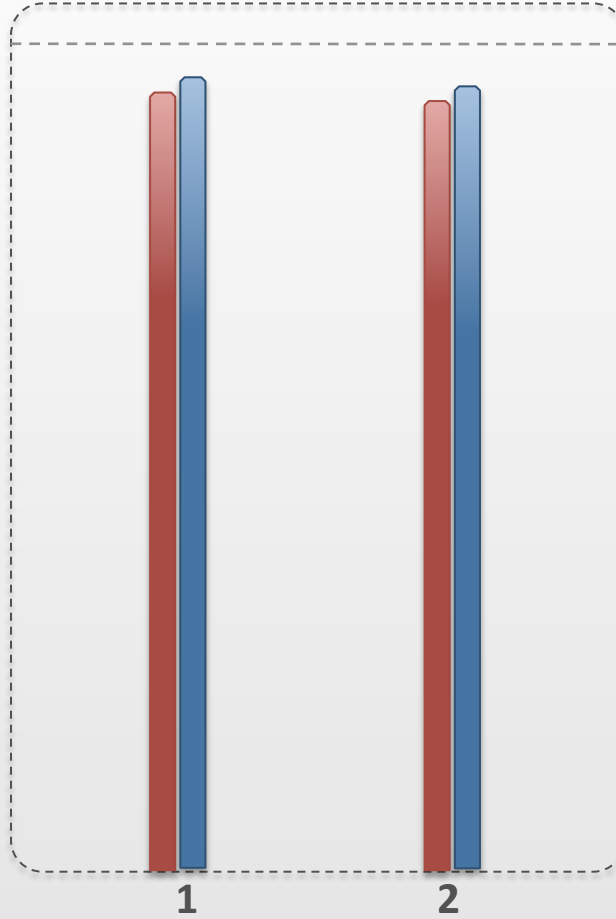


## Rotating 150 Objects While Mouse Moves

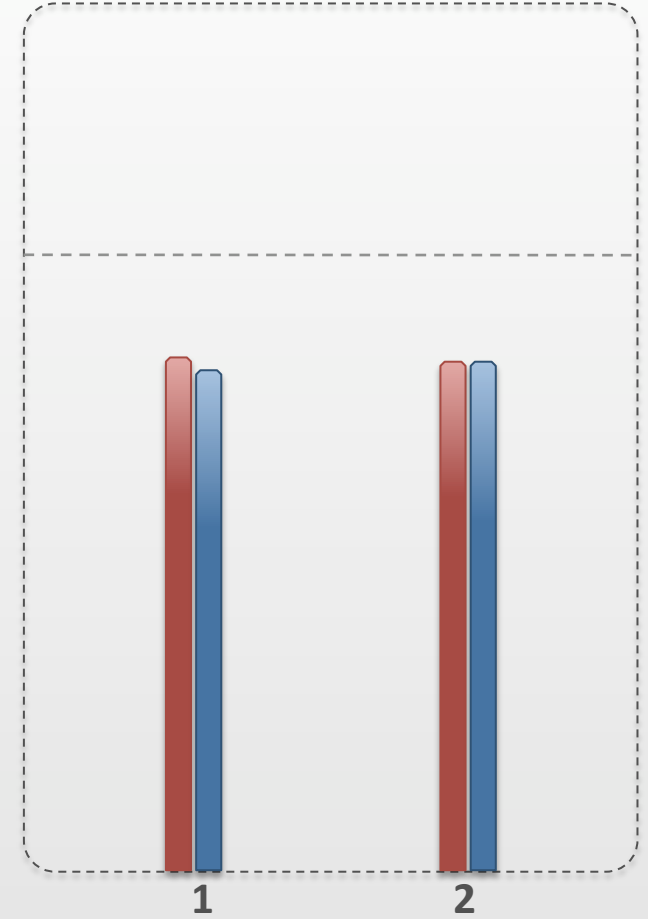
iPhone 3GS



iPod Touch 2<sup>nd</sup>



iPod Touch 1<sup>st</sup>



1 – CMouseEvent(F)

2 – CMouseStage(F)

- enterFrame

- Timer





# Tests For the Future

Object Pooling

Math Class Inlining

Loop/Conditional Performance Tests

## Resources

<http://www.bytearray.org/?p=1363>

<http://gskinner.com/talks/quick/#100>

<http://www.mikechambers.com/blog/2010/02/23/fitc-amsterdam-slides-creating-high-performance-iphone-applications-with-actionscript-3/>