Corona® SDK

Code less. Play more.

Corona SDK
AMBASSADOR

www.coronalabs.com

```
display.newImage( "sky.png" )
local ground = display.newImage("ground.png", 0, 400)
local crate = display.newImage("crate.png", 160, 50 );
crate.rotation = 30

local physics = require("physics")
physics.start()

physics.addBody( ground, "stati
physics.addBody( crate, { dens
```

Corona

The ultimate mobile platform

Write once

Develop 10x faster

Publish to top stores

Available on the App Store

Download for Android

Available on amazon appstore for Android

Available on nook

Monetize and distribute

inneractive.

InMobi

Developers Like You

Indies + Game Studios + Publishers + Agencies

150,000

Developers in the Corona Community

1,000,000,000+

app sessions

in 10 months

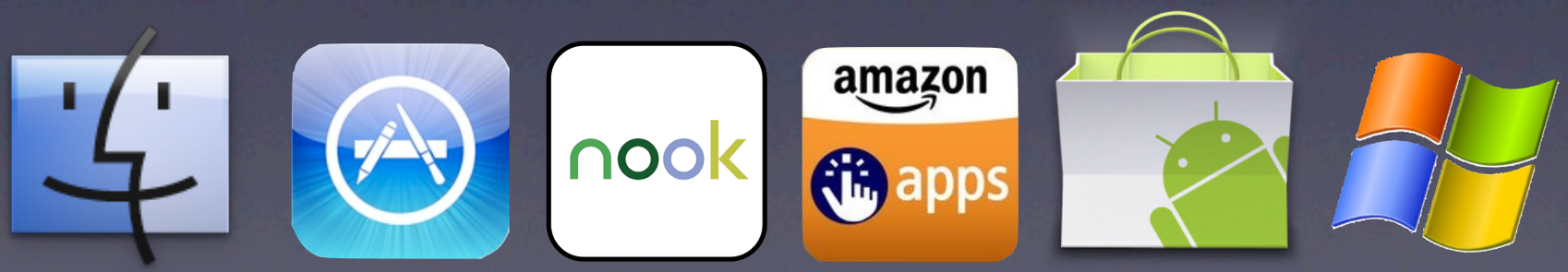
coronalabs.com

What is Corona?

SDK for native apps

develop 5-10x faster

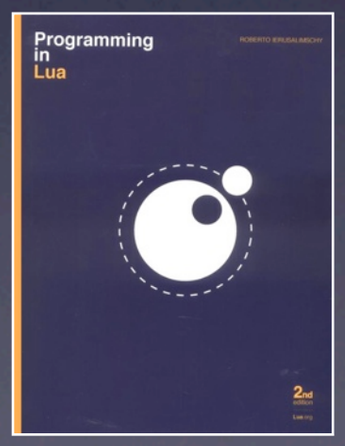
same code, multiple stores

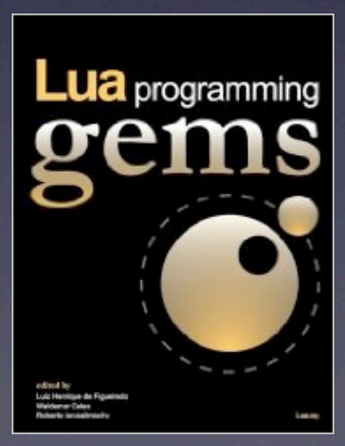


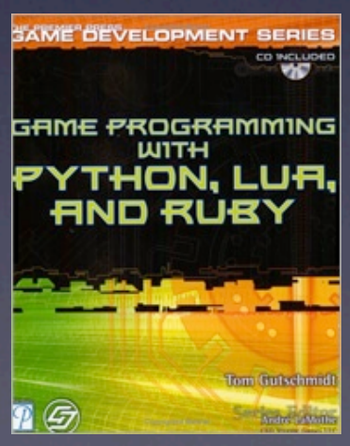
Lua: An Industry Standard

WORLD OF WARCRAFT®

Lr
ADOBE® PHOTOSHOP® LIGHTROOM® 2







Other Languages –> Lua

```
if (!carMoving) {
    // do something
} else if (noGas) {
    // something else
}
```

```
if (not carMoving) then
    -- do something
elseif (noGas) then
    -- something else
end
```

```
for (i=1; i<=10; i++) {
    print(i)
}
```

```
for i = 1,10 do
    print(i)
end
```

```
for (j=100; j>0; j--) {
    print(j)
}
```

```
for j = 100,1,-1 do
    print(j)
end
```

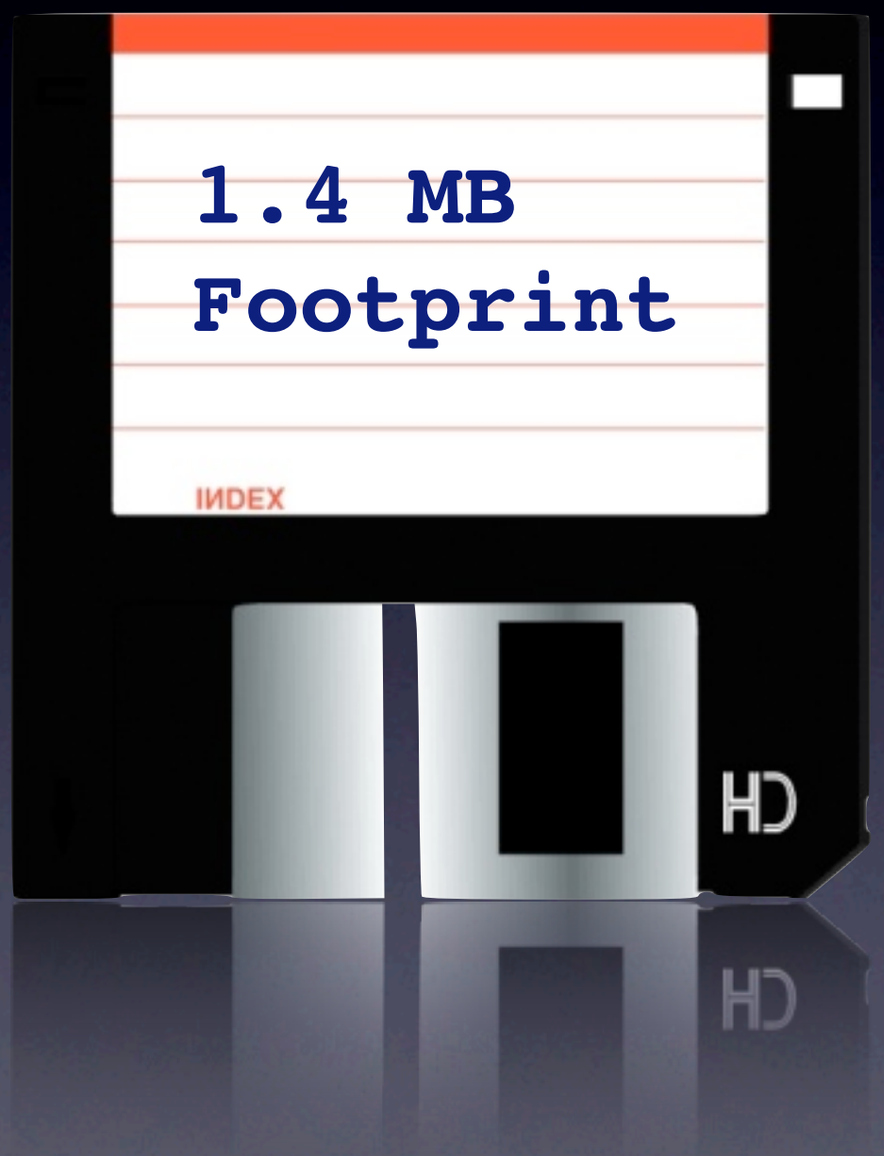
Lua Objects are Tables

```
myTable = { "a", "b", 100, "hello" }

otherTable = { x=5, y=7, name="Joe" }

newTable = {}

newTable[1] = "a"

newTable.x = 5

newTable.hasProperties = true

newTable["name"] = "Joe"

newTable["printJoe"] = function() print("Joe") end
```

Tons of Features/APIs

- Content scaling
- Multi-resolution images
- Simulator (instant refresh)
- Cloud services
- And more!

Small Code Size



Develop 5-10x faster

"Angry Birds"

36 hours

code+graphics+sound
(complete 2 level game)

"Fruit Ninja"

14 hours

code+graphics+sound
(gameplay only)

"Tiny Wings"

12 hours

code+graphics+sound
(gameplay only)

> "Developing directly in Xcode would have been at least 5x more code than Corona"
> – Unicorn Labs, Top 20 iPad eBook

Hard Problems Made Easy

(i.e., how Corona will teach you to love physics)

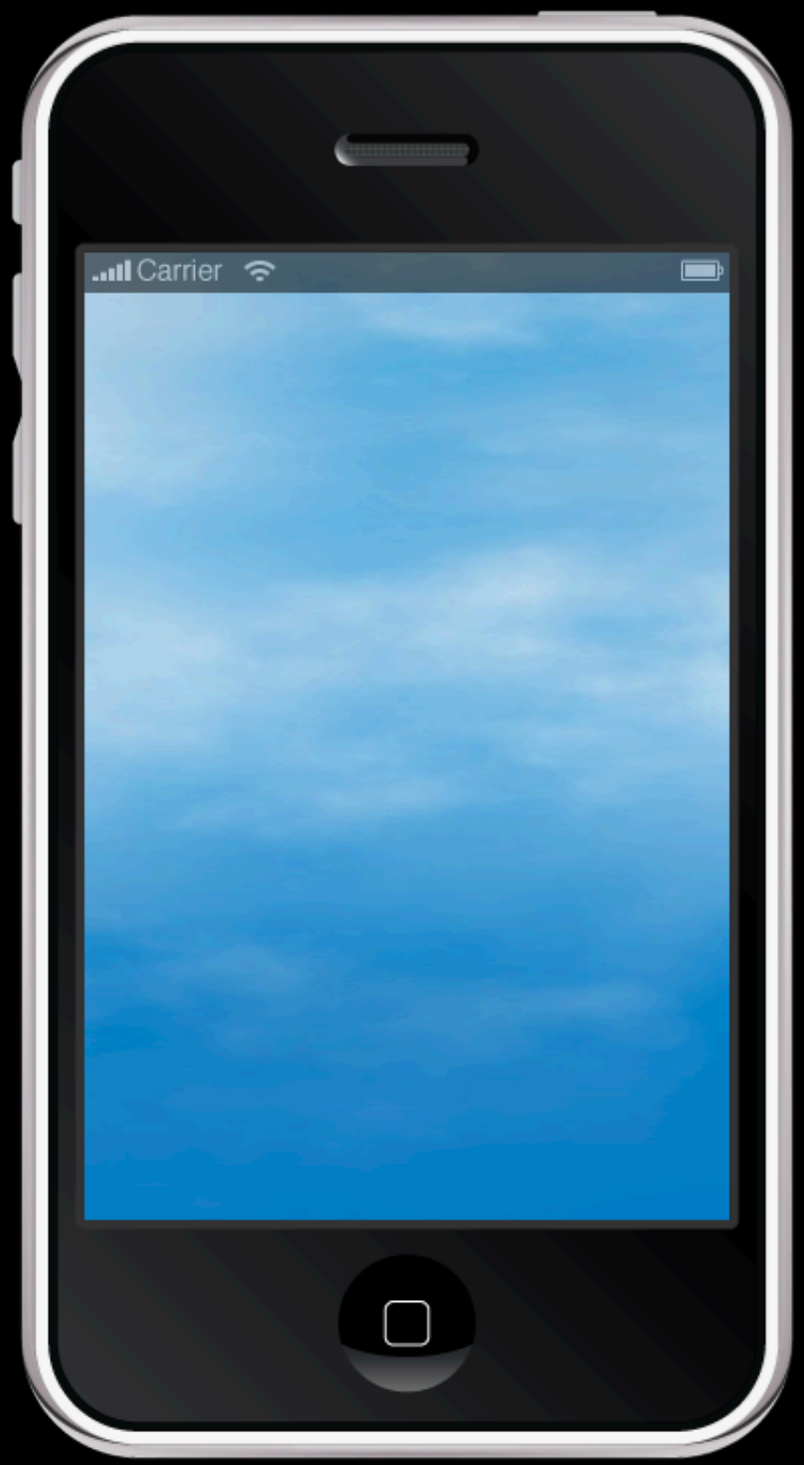
OpenGL in one line

```
Phone SDK.
// Display "myImage.png"

// ----------------------------------------------------------------------------
// OpenGLESTextureAppDelegate.m
// ----------------------------------------------------------------------------

#import "OpenGLESTextureAppDelegate.h"
#import "EAGLView.h"
#import "OpenGLESTextureViewController.h"

@implementation OpenGLESTextureAppDelegate

@synthesize window=_window;

@synthesize viewController=_viewController;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    self.window.rootViewController = self.viewController;
    return YES;
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    /*
     Restart any tasks that were paused (or not yet started) while the application was
inactive. If the application was previously in the background, optionally refresh the user
interface.
     */
    [self.viewController drawFrame];
}

- (void)dealloc
{
    [_window release];
    [_viewController release];
    [super dealloc];
}

@end


// ----------------------------------
// EAGLView.m
// ----------------------------------

#import <QuartzCore/QuartzCore.h>
#import "EAGLView.h"

@interface EAGLView (PrivateMethod
- (void)createFramebuffer;
- (void)deleteFramebuffer;
@end

@implementation EAGLView

@synthesize context;

// You must implement this method
+ (Class)layerClass
{
    return [CAEAGLLayer class];
}

//The EAGL view is stored in the nib file. When it's unarchived it's sent -initWithCoder:.
- (id)initWithCoder:(NSCoder*)coder
{
    self = [super initWithCoder:coder];
    if (self) {
        CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;

        eaglLayer.opaque = TRUE;
        eaglLayer.drawableProperties = [NSDictionary dictionaryWithObjectsAndKeys:
                                        [NSNumber numberWithBool:FALSE],
kEAGLDrawablePropertyRetainedBacking,
                                        kEAGLColorFormatRGBA8,
kEAGLDrawablePropertyColorFormat,
                                        nil];
    }

    return self;
}

- (void)dealloc
{
    [self deleteFramebuffer];
    [context release];

    [super dealloc];
}

- (void)setContext:(EAGLContext *)newContext
{
    if (context != newContext) {
        [self deleteFramebuffer];

        [context release];
        context = [newContext retain];

        [EAGLContext setCurrentContext:nil];
    }
}

- (void)createFramebuffer
{
    if (context && !defaultFramebuffer) {
        [EAGLContext setCurrentContext:context];

        // Create default framebuffer object.
        glGenFramebuffers(1, &defaultFramebuffer);
        glBindFramebuffer(GL_FRAMEBUFFER, defaultFramebuffer);

        // Create color render buffer and allocate backing store.
        glGenRenderbuffers(1, &colorRenderbuffer);
        glBindRenderbuffer(GL_RENDERBUFFER, colorRenderbuffer);
        [context renderbufferStorage:GL_RENDERBUFFER fromDrawable:(CAEAGLLayer
*)self.layer];
        glGetRenderbufferParameteriv(GL_RENDERBUFFER, GL_RENDERBUFFER_WIDTH,
&framebufferWidth);
        glGetRenderbufferParameteriv(GL_RENDERBUFFER, GL_RENDERBUFFER_HEIGHT,
&framebufferHeight);

        glFramebufferRenderbuffer(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0, GL_RENDERBUFFER,
colorRenderbuffer);

        if (glCheckFramebufferStatus(GL_FRAMEBUFFER) != GL_FRAMEBUFFER_COMPLETE)
            NSLog(@"Failed to make complete framebuffer object %x",
glCheckFramebufferStatus(GL_FRAMEBUFFER));
    }
}

- (void)deleteFramebuffer
{
    if (context) {
        [EAGLContext setCurrentContext:context];

        if (defaultFramebuffer) {
            glDeleteFramebuffers(1, &defaultFramebuffer);
            defaultFramebuffer = 0;
        }

        if (colorRenderbuffer) {
            glDeleteRenderbuffers(1, &colorRenderbuffer);
            colorRenderbuffer = 0;
        }
    }
}

- (BOOL)presentFramebuffer
{
    BOOL success = FALSE;

    if (context) {
        [EAGLContext setCurrentContext:context];

        glBindRenderbuffer(GL_RENDERBUFFER, colorRenderbuffer);

        success = [context presentRenderbuffer:GL_RENDERBUFFER];
    }

    return success;
}

- (void)layoutSubviews
{
    // The framebuffer will be re-created at the beginning of the next setFramebuffer
method call.
    [self deleteFramebuffer];
}

@end


// ----------------------------------------------------------------------------
// OpenGLESTextureViewController.m
// ----------------------------------------------------------------------------

#import <QuartzCore/QuartzCore.h>
#import "OpenGLESTextureViewController.h"
#import "EAGLView.h"

@interface OpenGLESTextureViewController ()
@property (nonatomic, retain) EAGLContext *context;
@property (nonatomic, assign) CADisplayLink *displayLink;
- (void) loadTexture;
@end

@implementation OpenGLESTextureViewController

@synthesize animating, context, displayLink;

- (void)awakeFromNib
{
    EAGLContext *aContext = [[EAGLContext alloc] initWithAPI:kEAGLRenderingAPIOpenGLES1];

    if (!aContext)
        NSLog(@"Failed to create ES context");
    else if (![EAGLContext setCurrentContext:aContext])
        NSLog(@"Failed to set ES context current");

    self.context = aContext;
    [aContext release];

    [(EAGLView *)self.view setContext:context];
    [(EAGLView *)self.view setFramebuffer];

    [self loadTexture];

    self.displayLink = nil;
}

- (void) loadTexture
{
    glEnable(GL_TEXTURE_2D);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

    glGenTextures(1, &textureID);
    glBindTexture(GL_TEXTURE_2D, textureID);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_LINEAR);

    NSString *path = [[NSBundle mainBundle] pathForResource:@"myImage" ofType:@"png"];
    NSData *texData = [[NSData alloc] initWithContentsOfFile:path];
    UIImage *image = [[UIImage alloc] initWithData:texData];

    GLuint width = CGImageGetWidth(image.CGImage);
    GLuint height = CGImageGetHeight(image.CGImage);
    CGColorSpaceRef colorSpace = CGColorSpaceCreateDeviceRGB();
    void *imageData = malloc( height width 4 );
    CGContextRef image_context = CGBitmapContextCreate( imageData, width, height, 8, 4
width, colorSpace, kCGImageAlphaPremultipliedLast | kCGBitmapByteOrder32Big );
    CGColorSpaceRelease( colorSpace );
    CGContextClearRect( image_context, CGRectMake( 0, 0, width, height ) );
    CGContextTranslateCTM( image_context, 0, height - height );
    CGContextDrawImage( image_context, CGRectMake( 0, 0, width, height ), image.CGImage );

    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, width, height, 0, GL_RGBA, GL_UNSIGNED_BYTE,
imageData);
    [EAGLContext setCurrentContext:nil];

    [context release];

    [super dealloc];
}

- (void)viewDidUnload
{
    [super viewDidUnload];

    // Tear down context.
    if ([EAGLContext currentContext] == context)
        [EAGLContext setCurrentContext:nil];
    self.context = nil;
}

- (void)drawFrame
{
    [(EAGLView *)self.view setFramebuffer];

    // Replace the implementation of this method to do your own custom drawing.
    static const GLfloat squareVertices[] = {
        -0.5f, -0.33f,
        0.5f, -0.33f,
        -0.5f,  0.33f,
        0.5f,  0.33f,
    };

    static const GLfloat texCoords[] = {
        0.0, 1.0,
        1.0, 1.0,
        0.0, 0.0,
        1.0, 0.0
    };


    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glVertexPointer(2, GL_FLOAT, 0, squareVertices);
    glEnableClientState(GL_VERTEX_ARRAY);
    glTexCoordPointer(2, GL_FLOAT, 0, texCoords);
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);

    glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);

    [(EAGLView *)self.view presentFramebuffer];
}

@end
```

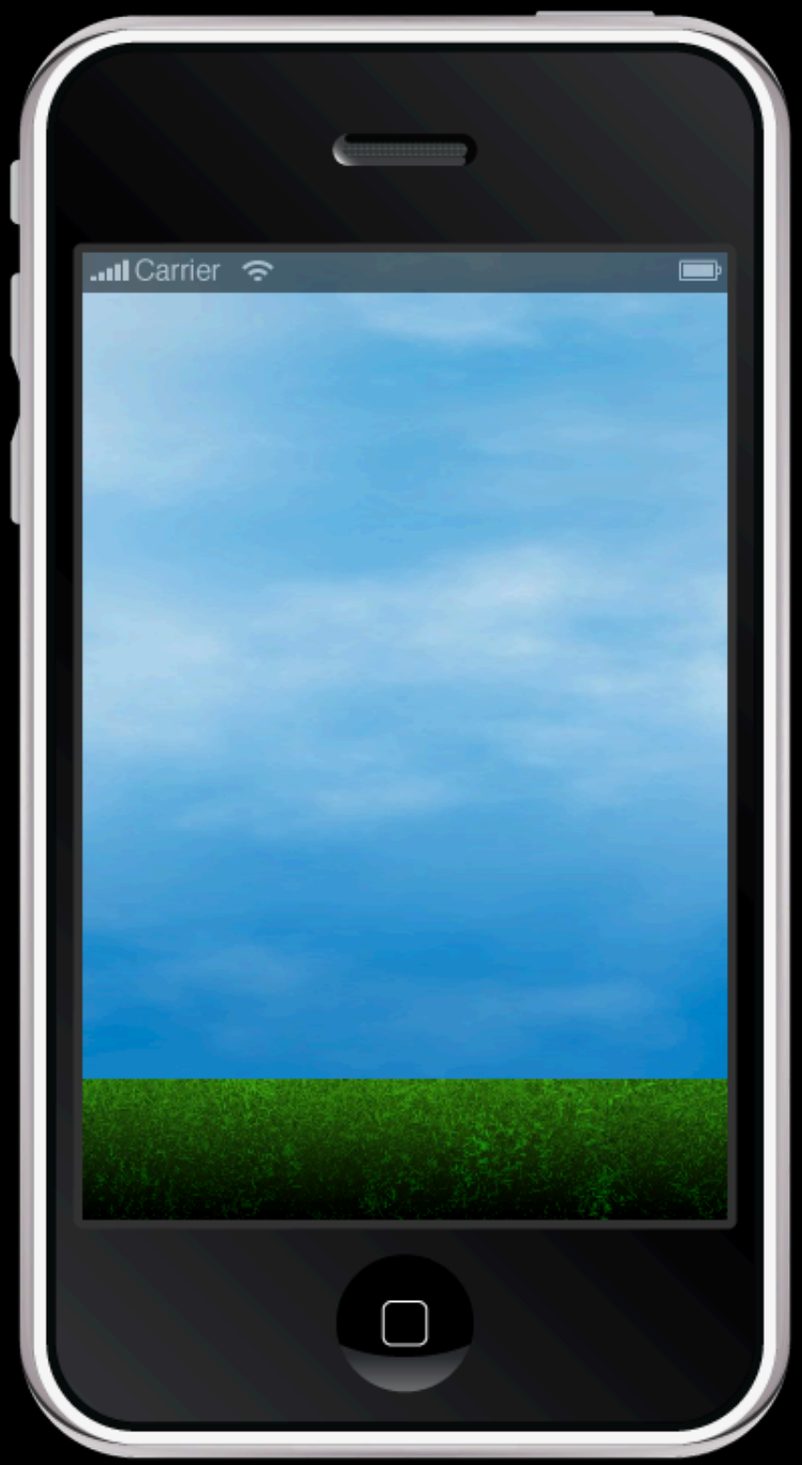
```
display.newImage("myImage.png")
```

coronalabs.com

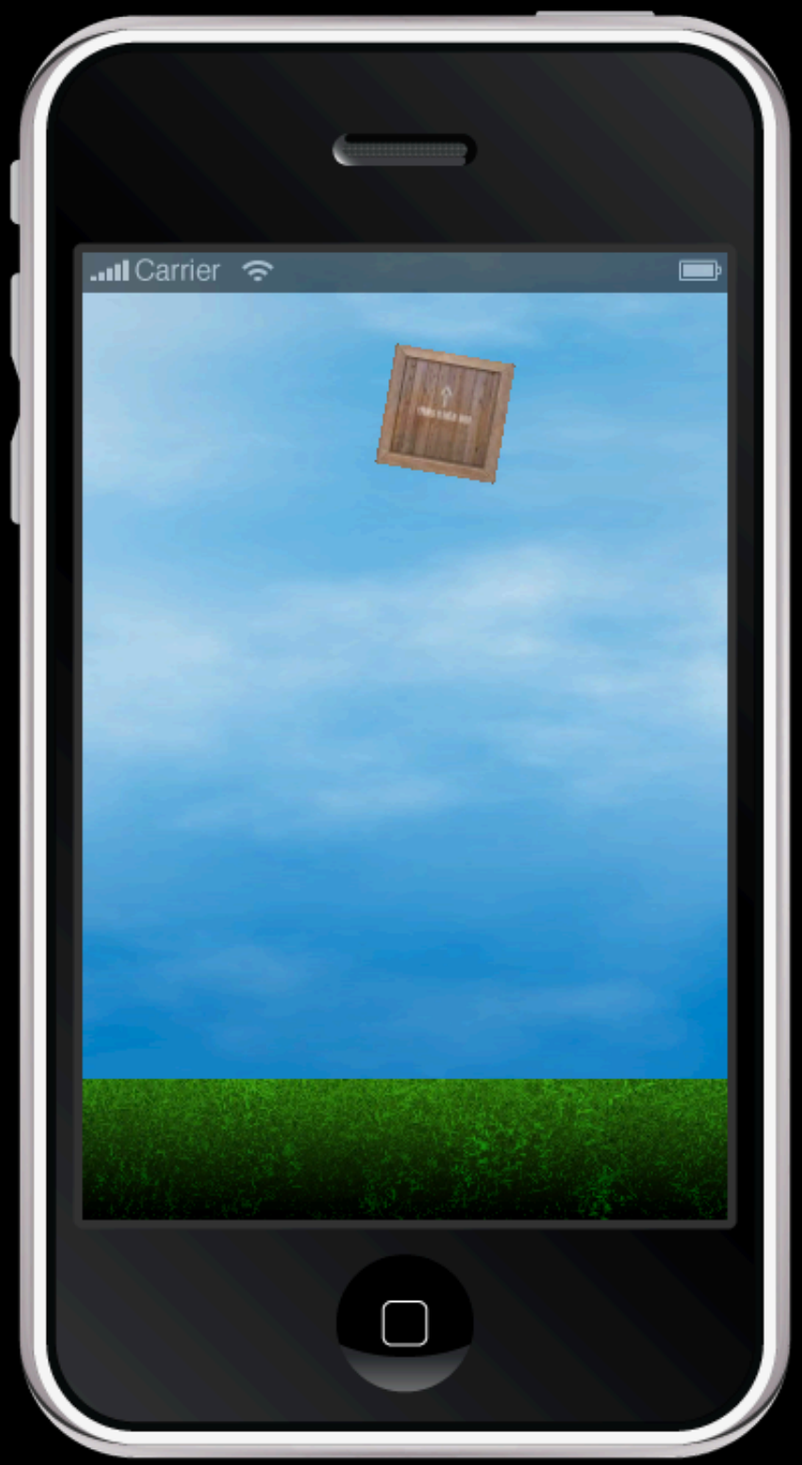
```
local sky = display.newImage( "clouds.png" )
```


Carrier

```
local sky = display.newImage( "clouds.png" )


local ground = display.newImage( "ground.png" )
ground.x = 160
ground.y = 445
```


Carrier

```lua
local sky = display.newImage( "clouds.png" )


local ground = display.newImage( "ground.png" )
ground.x = 160
ground.y = 445


local crate = display.newImage( "crate.png" )
crate.x = 180
crate.y = 80
crate.rotation = 10
```

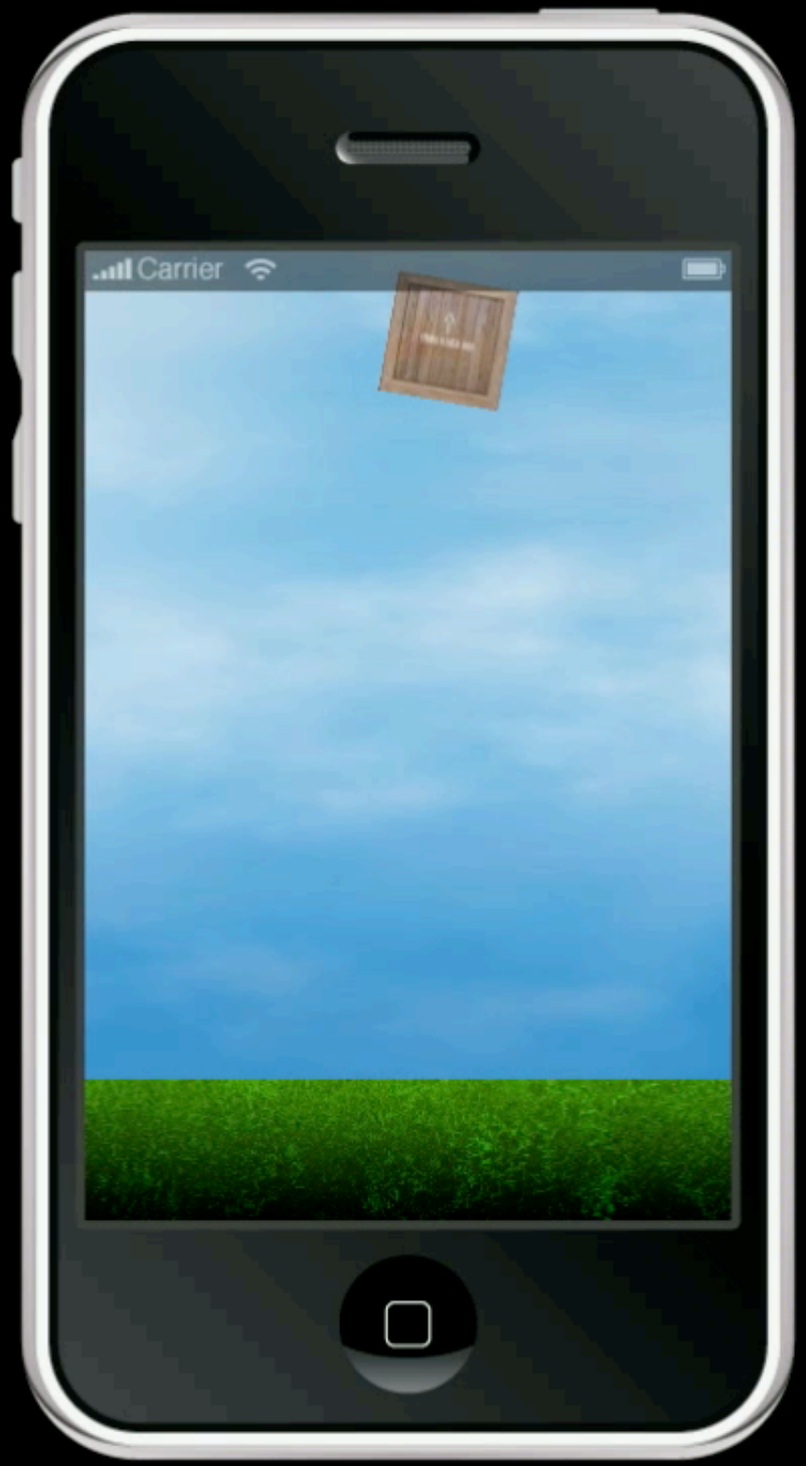

Carrier

```
local physics = require( "physics" )
physics.start()

local sky = display.newImage( "clouds.png" )

local ground = display.newImage( "ground.png" )
ground.x = 160
ground.y = 445
```

```
local crate = display.newImage( "crate.png" )
crate.x = 180
crate.y = 80
crate.rotation = 10
```

```
local physics = require( "physics" )
physics.start()

local sky = display.newImage( "clouds.png" )

local ground = display.newImage( "ground.png" )
ground.x = 160
ground.y = 445

physics.addBody( ground, { friction=0.5 } )
   ground.bodyType = "static"

local crate = display.newImage( "crate.png" )
crate.x = 180
crate.y = 80
crate.rotation = 10
```

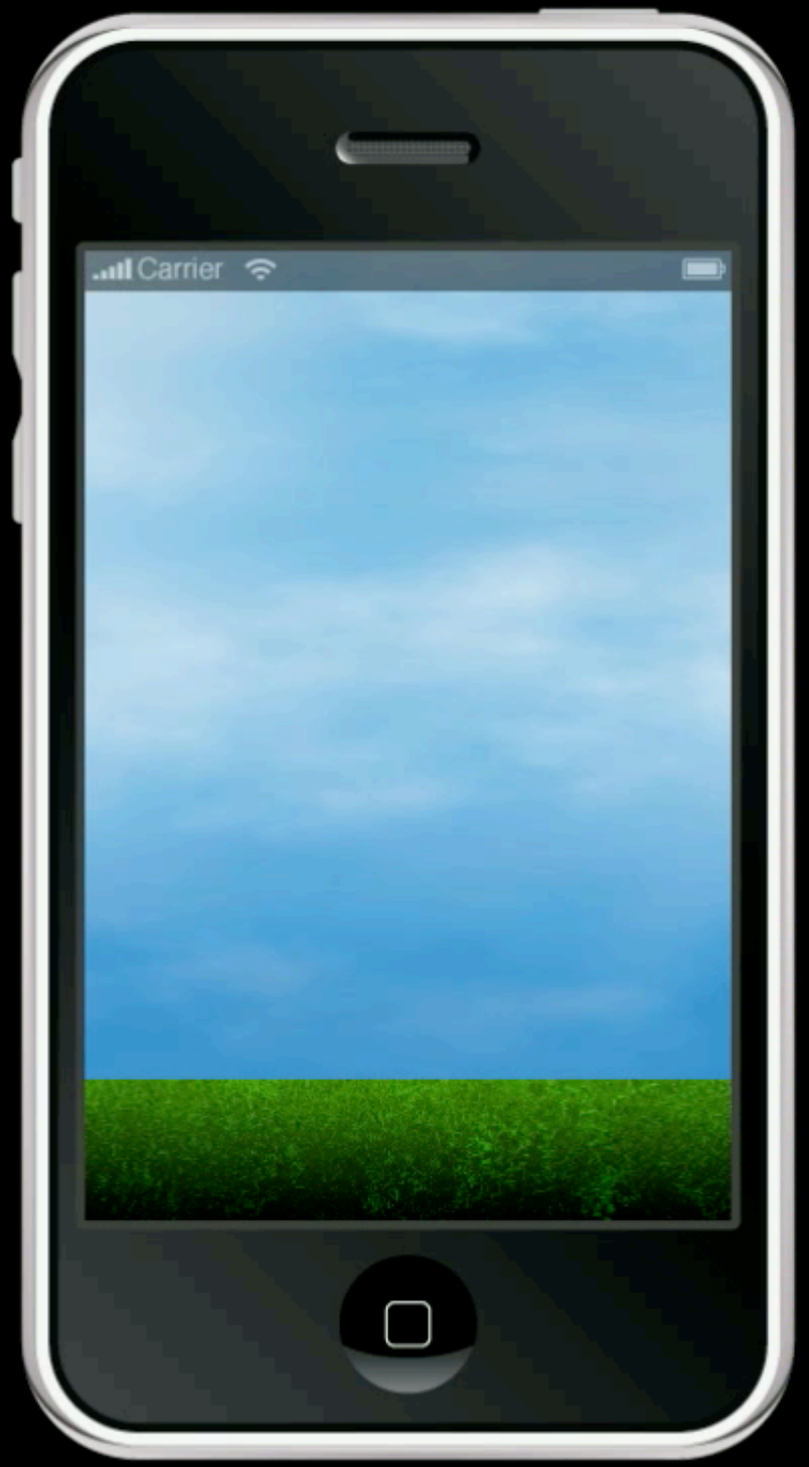
```
local physics = require( "physics" )
physics.start()

local sky = display.newImage( "clouds.png" )

local ground = display.newImage( "ground.png" )
ground.x = 160
ground.y = 445

physics.addBody( ground, { friction=0.5 } )
   ground.bodyType = "static"

local crate = display.newImage( "crate.png" )
crate.x = 180
crate.y = 80
crate.rotation = 10

physics.addBody( crate, { density=2.0,
     friction=0.5, bounce=0.3 } )
```


Carrier

What if we want lots of crates?

```
local crate = display.newImage( "crate.png" )
crate.x = 180
crate.y = -100
crate.rotation = 10
physics.addBody( crate, { density=2.0,
      friction=0.5, bounce=0.3 } )
```

```lua
local function spawnCrate()
  local crate = display.newImage( "crate.png" )
  crate.x = math.random( 320 )
  crate.y = -100
  crate.rotation = 10
  physics.addBody( crate, { density=2.0,
              friction=0.5, bounce=0.3 } )
end

timer.performWithDelay( 500, spawnCrate, 50 )
```


Carrier

What if gravity was **up** rather than down?

```
physics.setGravity( 0, 9.8 )
```

```
physics.setGravity( 0, -9.8 )
```


Carrier

Take a Look at What's Possible

"Corona is a 'godsend'. My game runs at 1 fps (frame/sec) in Flash but runs in full 30 fps in Corona."
– Jeff Fulton, Flash Developer