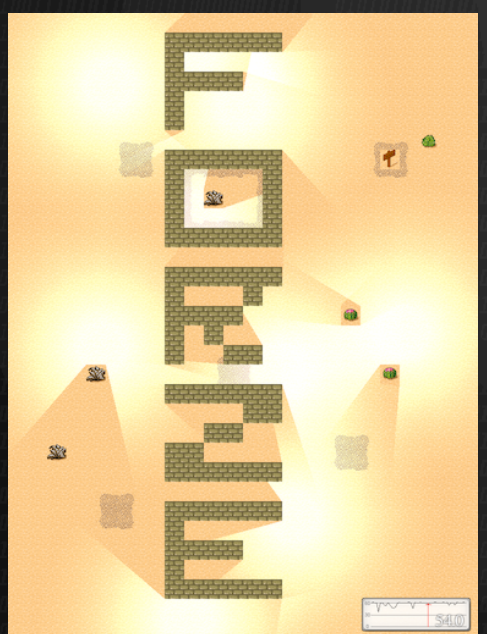
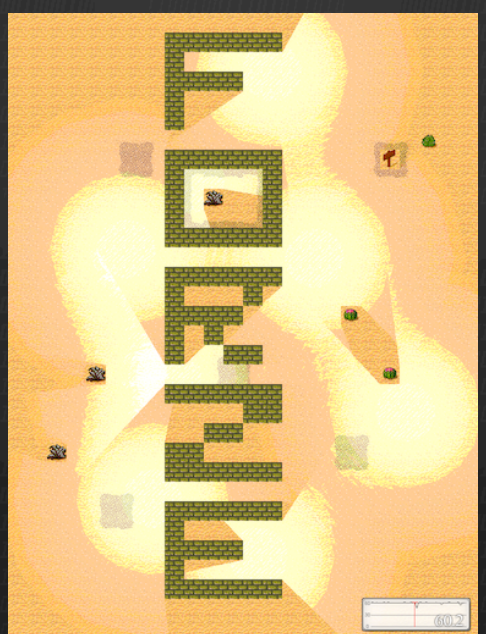

beta
TOKSESO

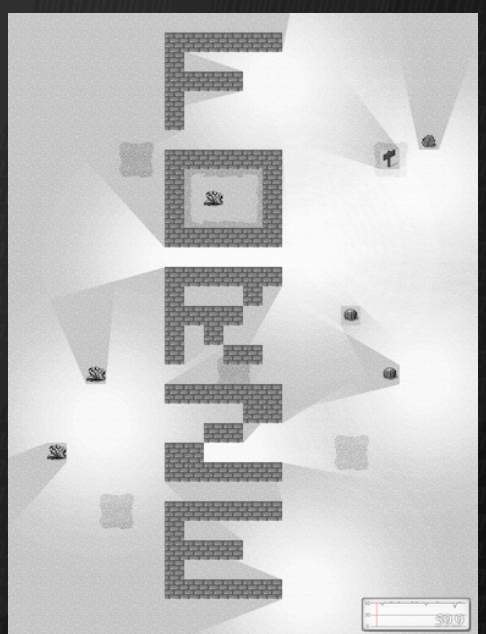
# WHAT'S FORZE2D?

***FORZE2D*** **is a 2D graphic engine based in *cocos2d* rethought to be 100% cross platform, fast, easy and optimized for the modern hardware.**

(TMXTiledMap + Lighting + Post-processing filters)



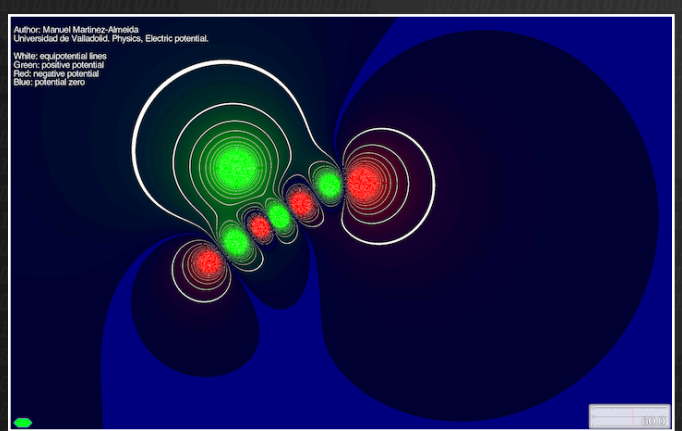




# WHY FORZE2D?

***FORZE, designed to be cross platform, to be fast, to be easy.***

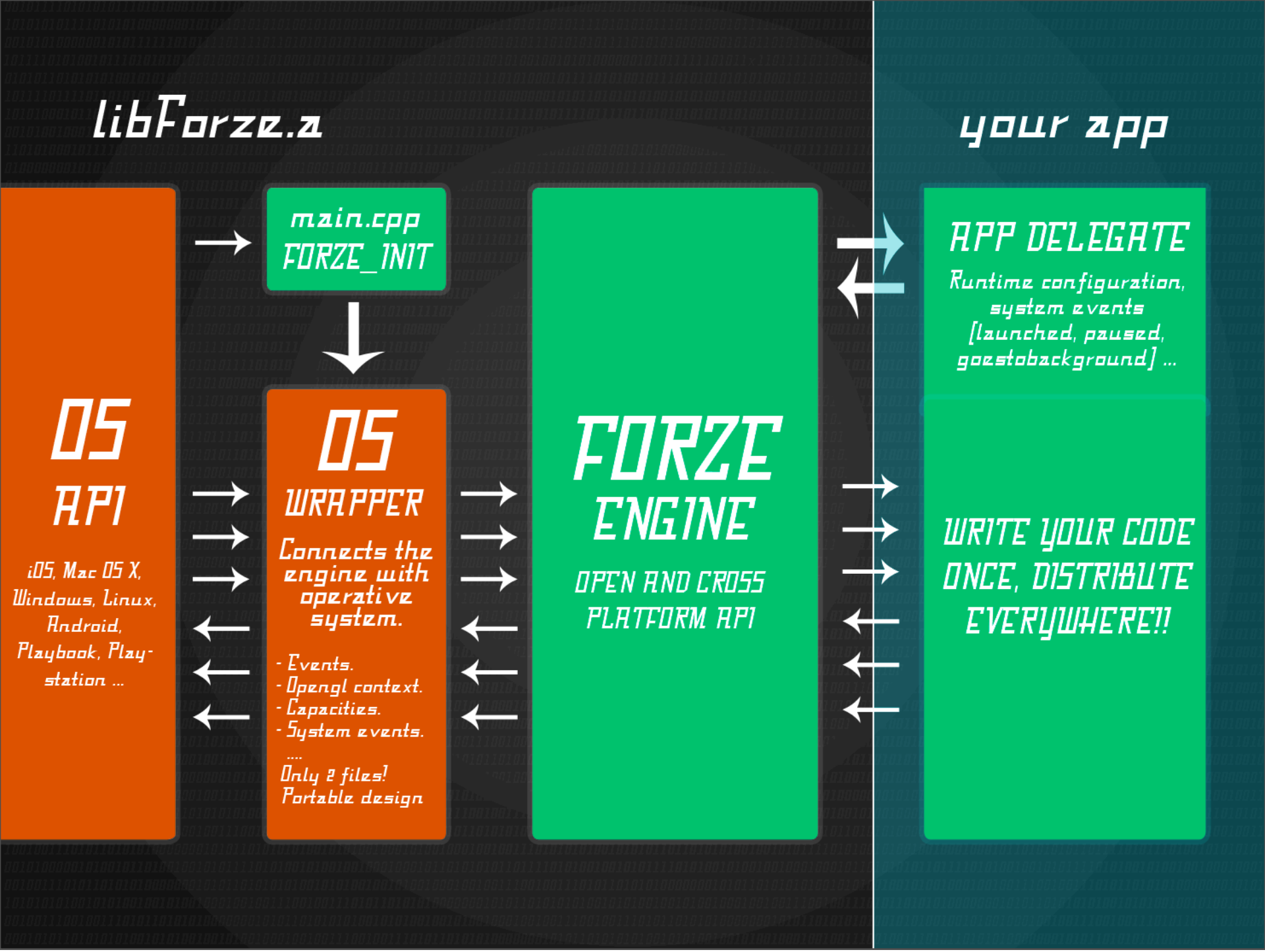
The main motivation was the creation a cross platform engine easy to port and maintain. From an intelligent point of view and following the cocos2d-iphone philosophy, FORZE is the only one cross platform version of cocos2d.




# CROSS PLATFORM

**FORZE is not simply a engine that works in several platforms. It is designed to be cross platform.**

- All the OS dependencies are contained in a **OS wrapper** (just two files). e.g. iOS_support{.h, .cpp}
- Adding new devices is super easy and safe, **without breaking the rest of code**.
- The public engine API is the same for all platforms, from main.cpp to the event handling.
- **Autoresizing** ensures that your game will look the same.


libForze.a
your app
OS
API
iOS, Mac OS X, Windows, Linux, Android, Playbook, Play-station ...
main.cpp
FORZE_INIT
OS
WRAPPER
Connects the engine with operative system.
- Events.
- Opengl context.
- Capacities.
- System events.
....
Only 2 files!
Portable design
FORZE
ENGINE
OPEN AND CROSS PLATFORM API
APP DELEGATE
Runtime configuration, system events [launched, paused, goestobackground] ...
WRITE YOUR CODE ONCE, DISTRIBUTE EVERYWHERE!!

# AUTORESIZING

1. Designing your games using a arbitrary canvas size (e.g. 320x480) means than you don't have to work directly with pixels and screens anymore.
2. Choose the resizing mode.
   - Fit to screen: It scales your app without deforming it leaving black bars.
   - Fit to screen and fill: This technique is like to the previous one but the black bars are filled with "more content".
   - Fit to screen and fill (integer scale): Like the previous one, by in this case the scaling factor is always an integer value.
   - Expand to screen: It deforms your game if the screen's ratio is different, in order to fill the whole window.
   - No resizing.
3. FORZE ensures that your game will look the same in every device.
4. (Optional) Add x2, x3, x4 textures if you want your game to look perfect in an higher resolution.

# AUTORESIZING (USAGE EXAMPLE)

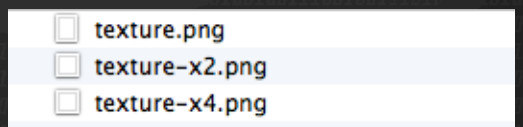
## Universal app: iPhone original, iPhone retina, iPad, iPad HD.

1. Set the smaller canvas size. In this case, iPhone original (320x480)

```
int main(int argc, char *argv[])
{
    // kFZSize_iPhone = fzSize(320x480)
    FORZE_INIT(new AppDelegate(), kFZSize_iPhone, argc, argv);
    return EXIT_SUCCESS;
}
```

2. Default resizing mode is “Fit to screen”
Nothing to do here.

3. Adding high resolution textures.
x2: iPhone retina and iPad original.
x4: iPad HD




FORZE will use the proper resources automatically.

# POWERFUL

**FORZE2D includes all the features of another popular 2D engines like cocos2d.**

- TMXTiledMaps.
- Actions ( MoveTo, ScaleBy… )
- Menus
- Scene management (transitions…)
- Particle system.
- Physics engines ( box2d, chipmunk… )
- Bitmap fonts ( Angel Code )
- Sprite atlas ( zwoptex, TexturePacker… )
- .png, .pvr, pvr.ccz, pvr.gz.
- Frame-by-frame animations.

# POWERFUL

**But in addition, FORZE2D includes completely new features.**

- 2D lighting system
- Rendering-on-demand
- Advanced grid effects ( like Infinity Field ) (working progress)
- Post-processing filters ( grayscale filter, toon filter… )
- Modular animations ( Spriter ) (working progress)
- Recursive opacity.
- OpenGL, OpenGL ES 1.0, OpenGL ES 2.0
- Powerful diagnostic logs.
- Unified event handling.

# EASY

## Easy = less development time = more money for you

FORZE is designed to be intuitive. If you have used cocos2d or Corona SDK before, learning FORZE will be a breeze.

```
Sprite *sprite = new Sprite("myimage.png");
sprite->setColor(fzWHITE * 0.5f); // gray!!
sprite->setPosition(getContentSize()/2); // center
sprite->setRotation(45); // rotate

// move to point (0, 0) in one second
sprite->runAction(new MoveTo(1, FZPointZero));

// attach sprite to the layer
addChild(sprite);
```

```
// Post-processing effect
setFilter(new FilterToon());
```

# EXAMPLE CODE (hello world)

helloworld.cpp

**Initialize FORZE.**

```
// APPLICATION'S ENTRY POINT
int main(int argc, char *argv[])
{
    FORZE_INIT(new AppDelegate(), kFZSize_Auto, argc, argv);
    return EXIT_SUCCESS;
}
```

**AppDelegate:** Receives the system events.

- applicationLaunched (required)
- applicationPaused, resumed… (optional)
- Runtime configurations (optional)

```
// APPLICATION DELEGATE
class AppDelegate : public ApplicationProtocol {
public:
    AppDelegate() {}

    void applicationLaunched(void *options)
    {
        // INITIALIZE HELLOWORLD SCENE
        HelloWorld *scene = new HelloWorld();

        // PUSH SCENE
        Director::Instance().pushScene(scene);
    }
};
```

**HelloWorld:** Custom subclass of Scene. Draws a label with text “Hello world!” with font “helvetica.fnt” at the center of the screen.

```
// CUSTOM SUBCLASS OF FORZE::SCENE
class HelloWorld : public Scene {
public:
    HelloWorld()
    {
        // Label with text and font filename.
        Label *label = new Label("Hello world!", "helvetica.fnt");

        // Place label at the middle of the layer
        label->setPosition(getContentSize()/2);

        // Attach label to the layer
        addChild(label);
    }
};
```

# EXAMPLE CODE (event handling)

**setTrackedEvents()** sets the event kinds that you want to track, in this case: accelerometer and touch.

**bool event(Event)** receives the events.

This code creates sprites under your fingers (multitouch) and removes them when you stop touching the screen.

In addition the accelerometer data is used as gravity in box2d (for example).

```
class MyLayer : public Layer {
public:
    MyLayer()
    {
        setTrackedEvents(kFZEventType_Tap | kFZEventType_Accelerometer);
    }

    bool event(Event &event)
    {
        if(event.isType(kFZEventType_Accelerometer)) {
            setGravity(event.getVector());

        }else{
            // EVENT IS TYPE TAP
            switch (event.getState()) {
                case kFZEventState_Began:
                {
                    Sprite *sprite = new Sprite("finger.png");
                    sprite->setPosition(event.getPoint());
                    addChild(sprite);

                    event.setUserData(sprite);
                    return true; // I am interested in this event
                }
                case kFZEventState_Updated:
                {
                    Sprite *sprite = (Sprite*) event.getUserData();
                    sprite->setPosition(event.getPoint());
                    break;
                }
                case kFZEventState_Ended:
                case kFZEventState_Cancelled:
                {
                    Sprite *sprite = (Sprite*) event.getUserData();
                    removeChild(sprite);
                    break;
                }
                default: break;
            }
        }
        return false;
    }

    void setGravity(fzPoint3 vector) {
        // BOX2D STUFF
    }
};
```

# FASTER

**FORZE's API is similar to other alternatives, but internally they are very different. FORZE2D is designed to be fast.**

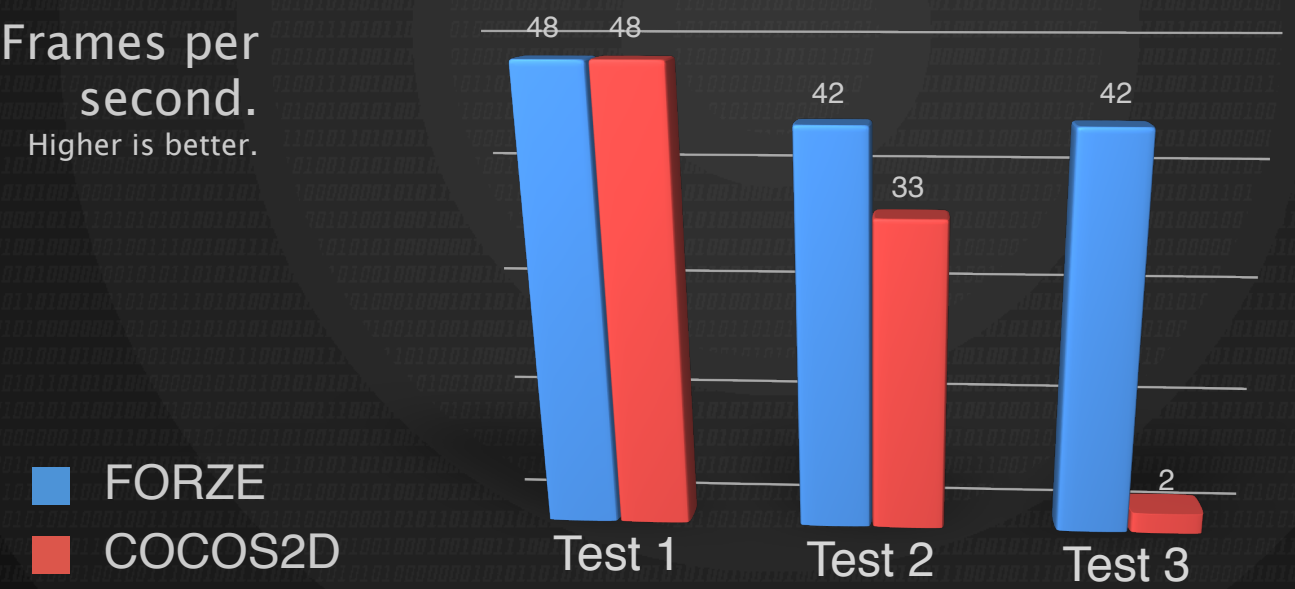
- Innovative caching techniques.
- Better algorithms: much of the most common tasks are O(1).
- Memory management: less memory per node, less fragmentation, faster.
- Multithreading.
- Balanced charge between CPU and GPU.
- Custom assembly ARM NEON and Intel SSE code.
- Efficient code.
- C++

# PERFORMANCE TEST

**FORZE2D 0.0.92 alpha vs. cocos2d 2.0 rc0a. Release mode, same build settings. iPad 1.**

- Test 1: 7000 static sprites
- Test 2: 7000 sprites + 7000 actions (rotating forever)
- Test 3: 7000 sprites + each frame (adding 100 sprites (at back) + removing 100 (at front) )

Frames per second.
Higher is better.



# OPEN SOURCE

## FORZE2D is using the liberal MIT license!

# Thanks for your time.

Contact

Manuel Martínez-Almeida
manu@forzefield.com
Skype: manucorporat
@forzefield

Visit us at

forzefield.com