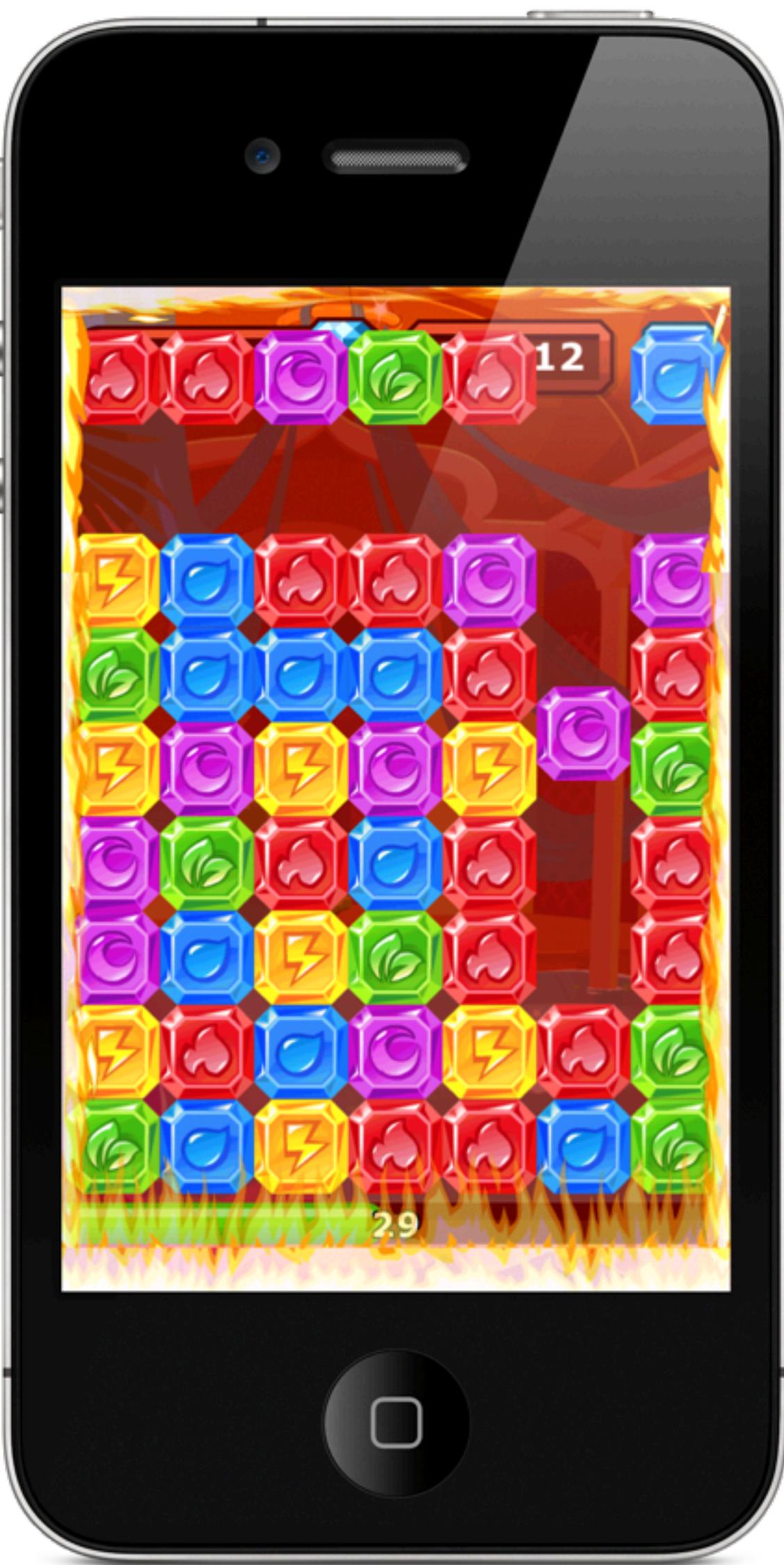




Ole Begemann





2D Game Framework

Ported from Python

By @ricardoquesada



BSD License

cocos2d-iphone.org



Started in 2008

1.0.1-stable

1.1-beta

2.0-alpha



1.x branch: OpenGL ES 1.1
(Mac: OpenGL 1.5)

2.x branch: OpenGL ES 2.0

cocos2d-android/javascript

cocos2d-x (C++)

Core Animation on Steroids



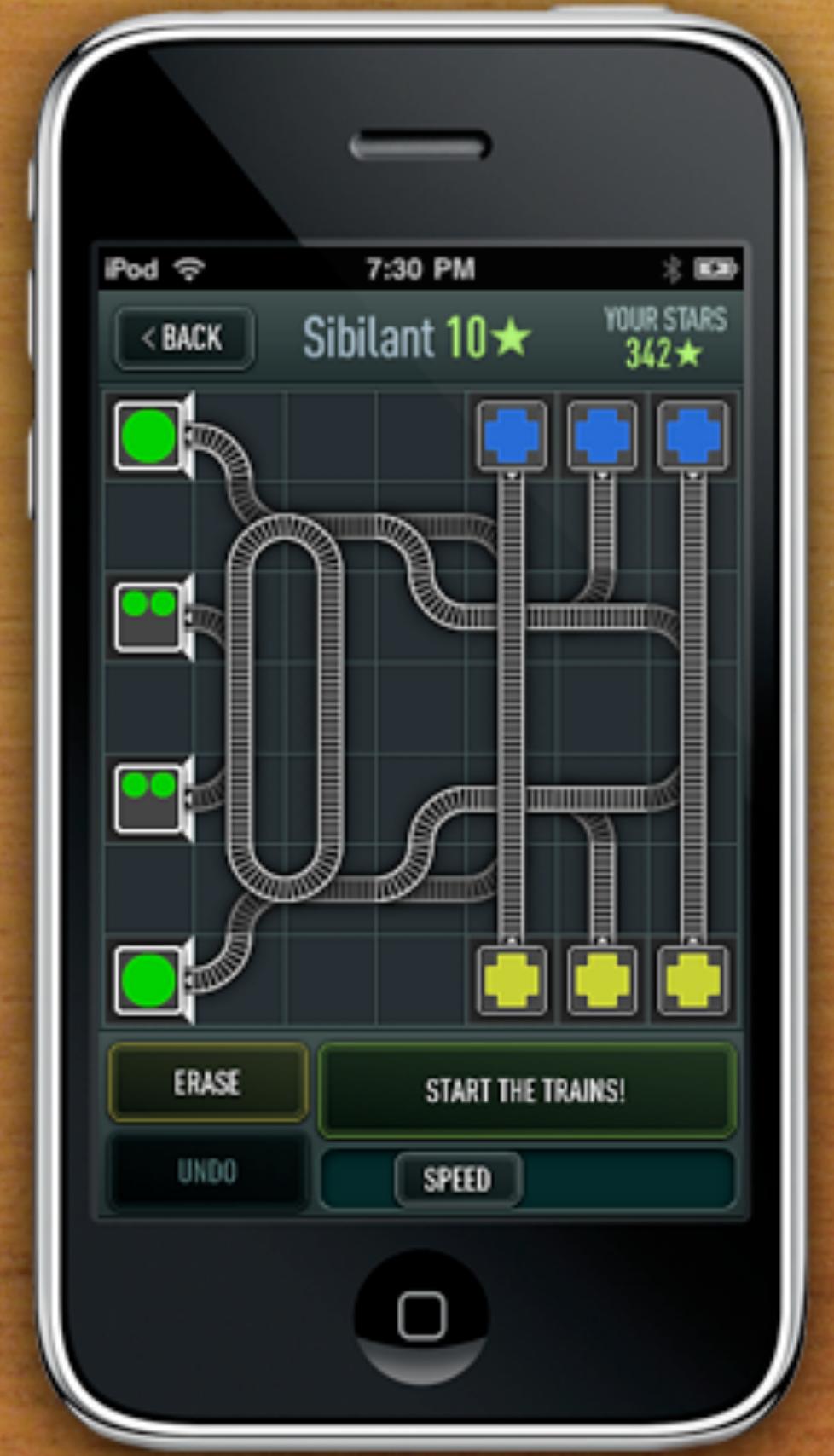
2D

(or 2.5D)

**one screen
GAMES**

**SIDE
SCROLLERS**

**MAP-BASED
GAMES**



TRAINYARD



THE ELEMENTS



FISHING FRENZY

CCDirector

Singleton

Set up Cocos2D Environment

Manage Scene Graph

```
application:didFinishLaunchingWithOptions:
```

```
if (![CCDirector setDirectorType:kCCDirectorTypeDisplayLink]) {  
    [CCDirector setDirectorType:kCCDirectorTypeDefault];  
}
```

```
CCDirector *director = [CCDirector sharedDirector];
```

```
EAGLView *glView = [EAGLView viewWithFrame:[window bounds]  
    pixelFormat:kEAGLColorFormatRGB565 depthFormat:0];  
[director setOpenGLView:glView];  
[director enableRetinaDisplay:YES];  
[director setDeviceOrientation:kCCDeviceOrientationPortrait];  
[director setAnimationInterval:1.0/60]; // 60 fps  
[director setDisplayFPS:YES];
```

```
MainMenuScene *mainMenu = [MainMenuScene node];  
[[CCDirector sharedDirector] runWithScene:mainMenu];
```

CCScene

Represents the current screen/level

CCNode subclass

Should be subclassed to add your
custom code (or CCLayer)

CCNode

Equivalent to CALayer

Subclasses:

**CCLayer, CCScene, CCSprite,
CCLabelTTF, CCMenuItem, ...**

CCAction

For timed animations

Equivalent to CAAnimation

Subclasses:

CCMoveTo, CCMoveBy, CCFlipX,
CCScaleTo, CCActionEase, ...

MyScene -init:

```
CCLayer *textLayer = [CCLayer node];
CCLabelTTF *label = labelWithString:@"Hello World"
    fontName:@"MarkerFelt-Wide" fontSize:24.0f];
label.position = ccp(100.0f, 250.0f);
[textLayer addChild:label];

CCLayer *spriteLayer = [CCLayer node];
CCSprite *mainCharacter = [CCSprite spriteWithFile:@"mainCharacter.png"];
mainCharacter.position = ccp(240.0f, 160.0f);
[spriteLayer addChild:mainCharacter tag:MAIN_CHARACTER_TAG];

[self addChild:textLayer];
[self addChild:spriteLayer];
```

```
- (BOOL)ccTouchBegan:(UITouch *)touch withEvent:(UIEvent *)event
{
    CGPoint point = [touch locationInView:[touch view]];
    CGPoint touchPoint = [[CCDirector sharedDirector] convertToGL:point];

    CCSprite *mainCharacter = [self getChildByTag:MAIN_CHARACTER_TAG];

    CCMoveTo *moveToAction = [CCMoveTo actionWithDuration:0.5
        position:touchPoint];
    CCEaseInOut *easeAction = [CCEaseInOut actionWithAction:moveToAction];
    [mainCharacter runAction:easeAction];

    return YES;
}
```

Direct Access to Render Loop

On any node:

```
[self scheduleUpdate];
```

Then:

```
- (void)update:(ccTime)delta
{
    // This method is called every frame before rendering
    CGPoint newPosition = ccp(
        self.position.x + velocity.x * delta,
        self.position.y + velocity.y * delta
    );
    self.position = newPosition;
}
```

Texture Atlas/ Sprite Sheets

CCSpriteBatchNode

CCSpriteFrameCache



Key	Type	Value
▼frames	Diction...	(18 items)
▶ diamond_fireball1.png	Diction...	(5 items)
▶ diamond_fireball2.png	Diction...	(5 items)
▶ diamond_fireball3.png	Diction...	(5 items)
▶ diamond_fireball4.png	Diction...	(5 items)
▶ diamond_fireball5.png	Diction...	(5 items)
▶ diamond_fireball6.png	Diction...	(5 items)
▶ diamond_fireball7.png	Diction...	(5 items)
▶ diamond_fireball8.png	Diction...	(5 items)
▼gems_blue.png	Diction...	(5 items)
frame	String	{2,370},{90,90}}
offset	String	{0,0}
rotated	Boolean	NO
sourceColorRect	String	{0,0},{90,90}
sourceSize	String	{90,90}
▶ gems_blue_bw.png	Diction...	(5 items)
▶ gems_green.png	Diction...	(5 items)
▶ gems_green_bw.png	Diction...	(5 items)
▶ gems_purple.png	Diction...	(5 items)
▶ gems_purple_bw.png	Diction...	(5 items)
▶ gems_red.png	Diction...	(5 items)
▶ gems_red_bw.png	Diction...	(5 items)
▶ gems_yellow.png	Diction...	(5 items)
▶ gems_yellow_bw.png	Diction...	(5 items)
▶ metadata	Diction...	(3 items)

In scene init method:

```
CCSpriteBatchNode *spriteBatchNode = [CCSpriteBatchNode  
    batchNodeWithFile:@"myTextureAtlas.png" capacity:90];
```

```
spriteBatchNode.position = ccp(0,0);  
[self addChild:spriteBatchNode tag:SPRITE_BATCH_NODE];
```

```
[[CCSpriteFrameCache sharedSpriteFrameCache]  
    addSpriteFramesWithFile:@"myTextureAtlas.plist"];
```

Animations:

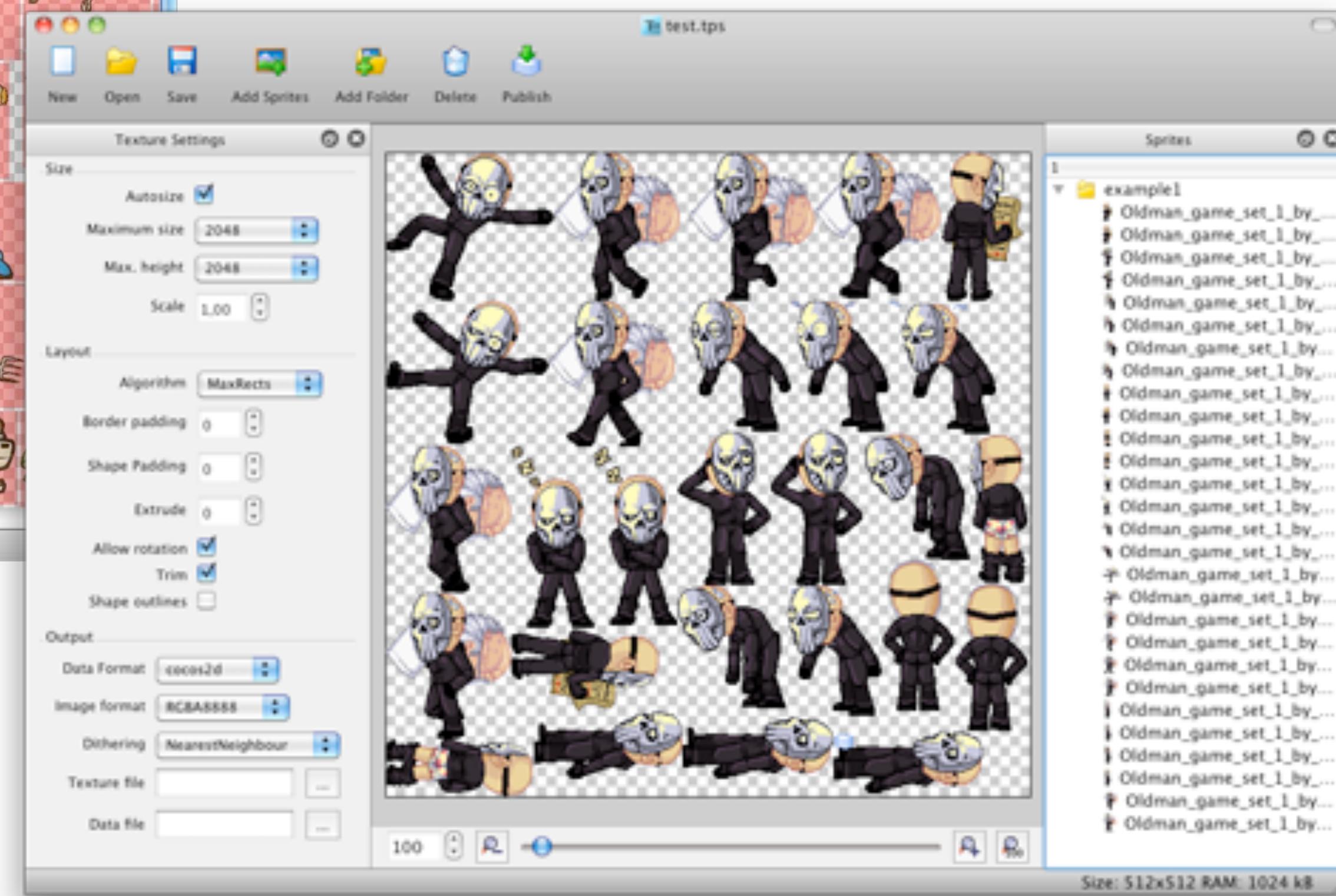
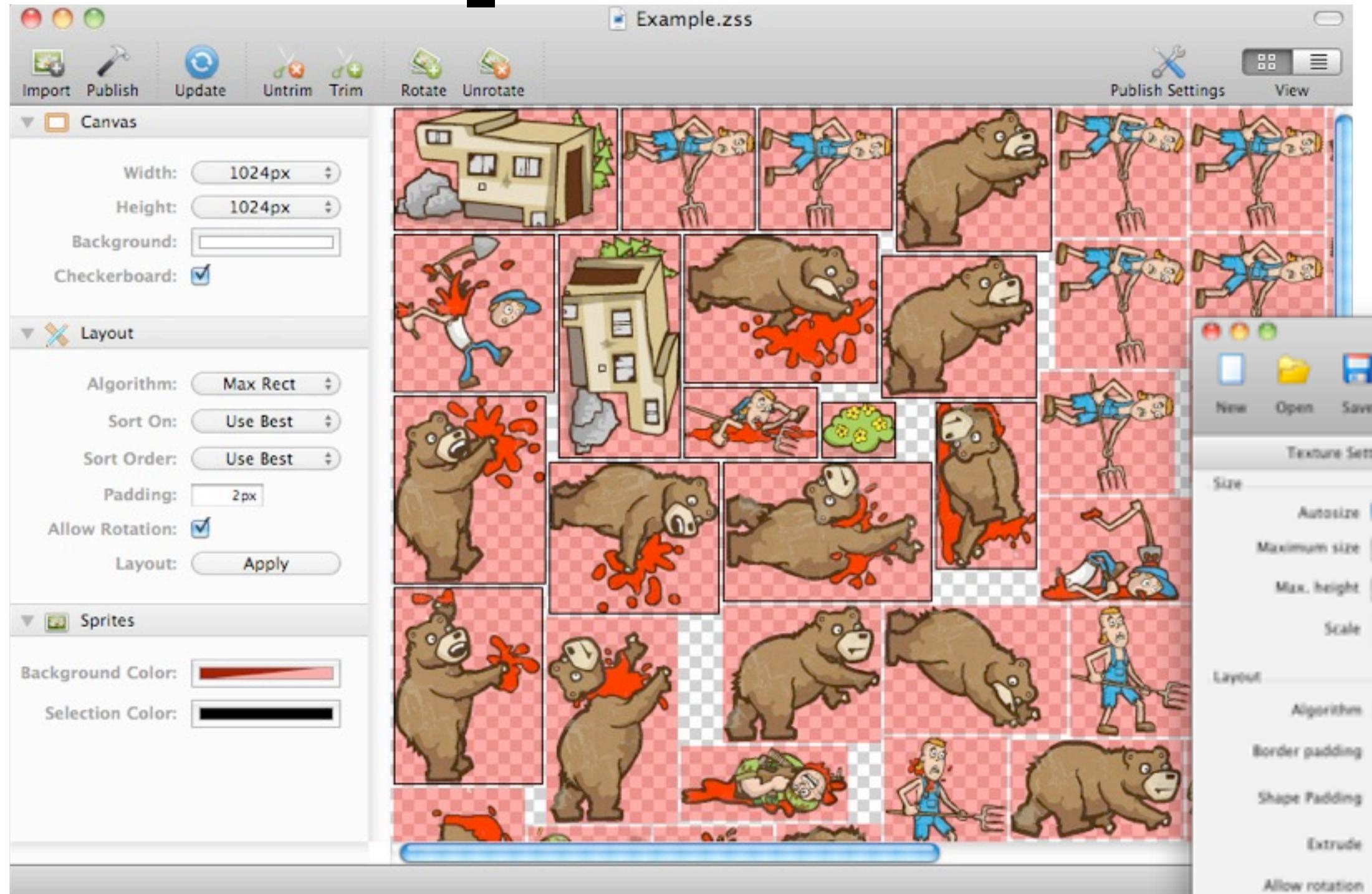
```
NSMutableArray *animationFrames = [NSMutableArray array];
for (int i = 1; i < 10; i++)
{
    NSString *file = [NSString stringWithFormat:@"animation_%d.png", i];
    CCSpriteFrame *frame = [frameCache spriteFrameByName:frame];
    [animationFrames addObject:frame];
}
CCAnimation *animation = [CCAnimation animationWithFrames:animationFrames
    delay:0.035];
[[CCAnimationCache sharedAnimationCache] addAnimation:animation
    name:@"myAnimation"];
```

...

```
CCAnimate *animateAction = [CCAnimate actionWithAnimation:animation
    restoreOriginalFrame:NO];
[mySprite runAction:animateAction];
```

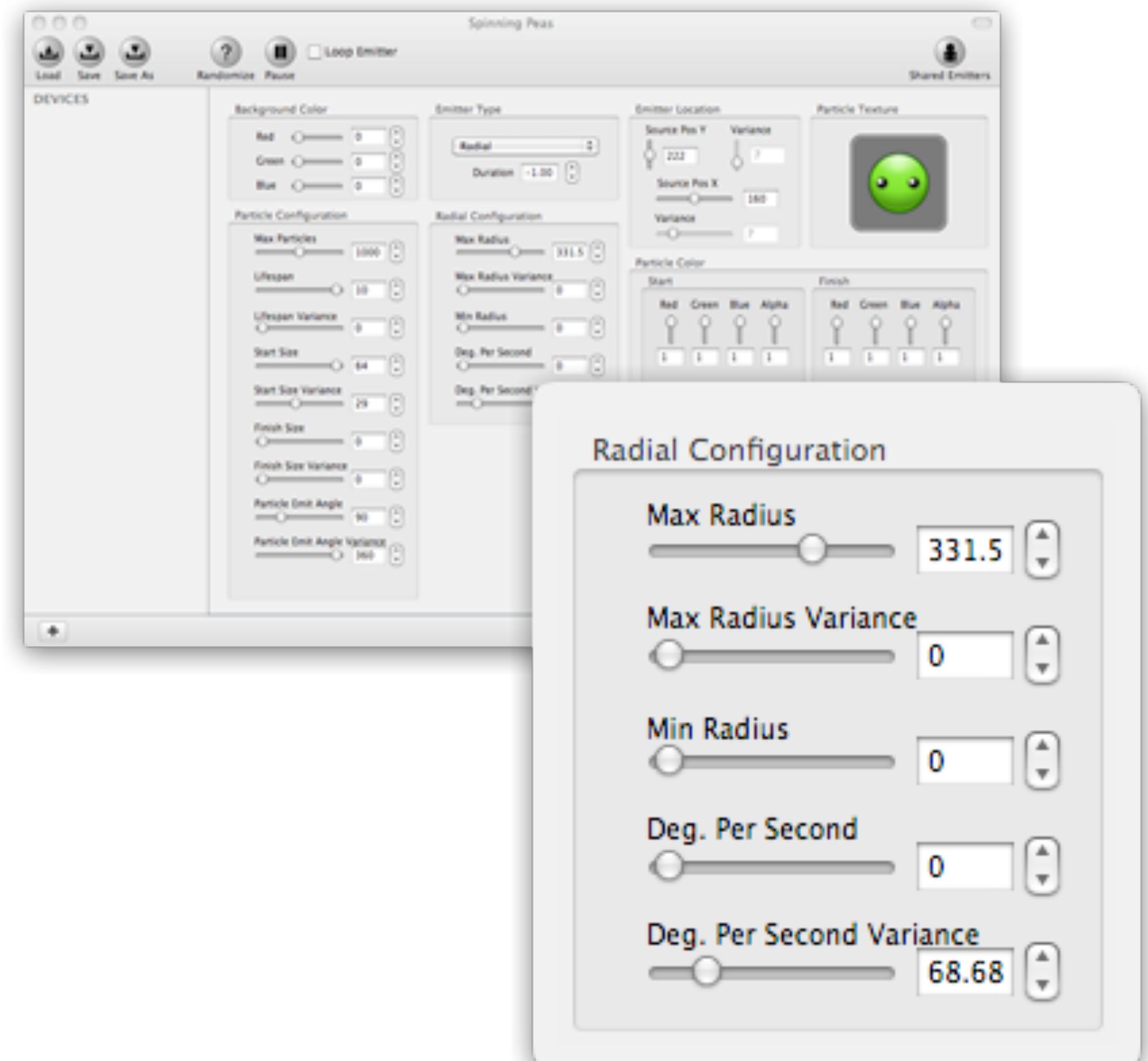
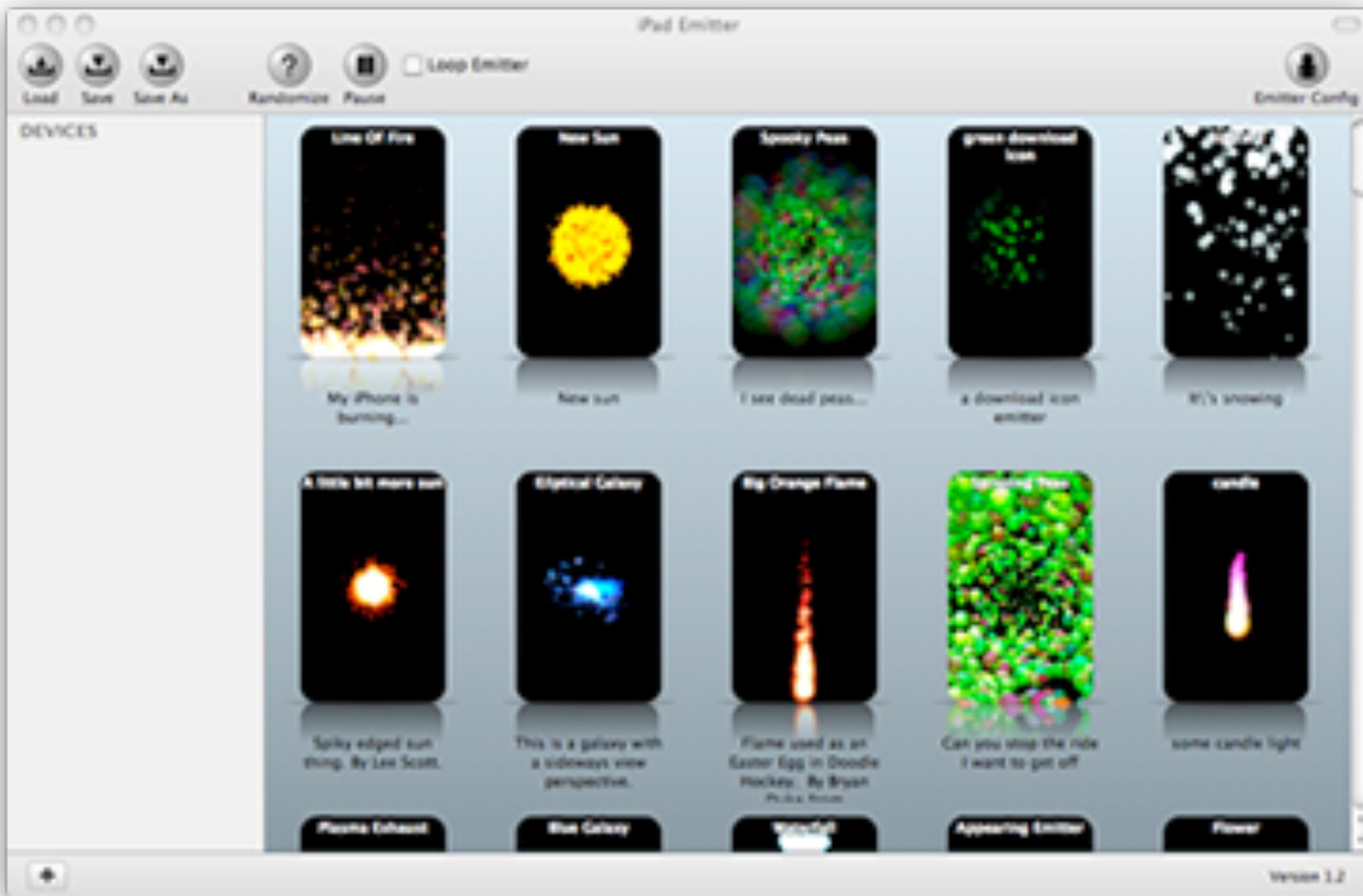
External Tools

Zwoptex

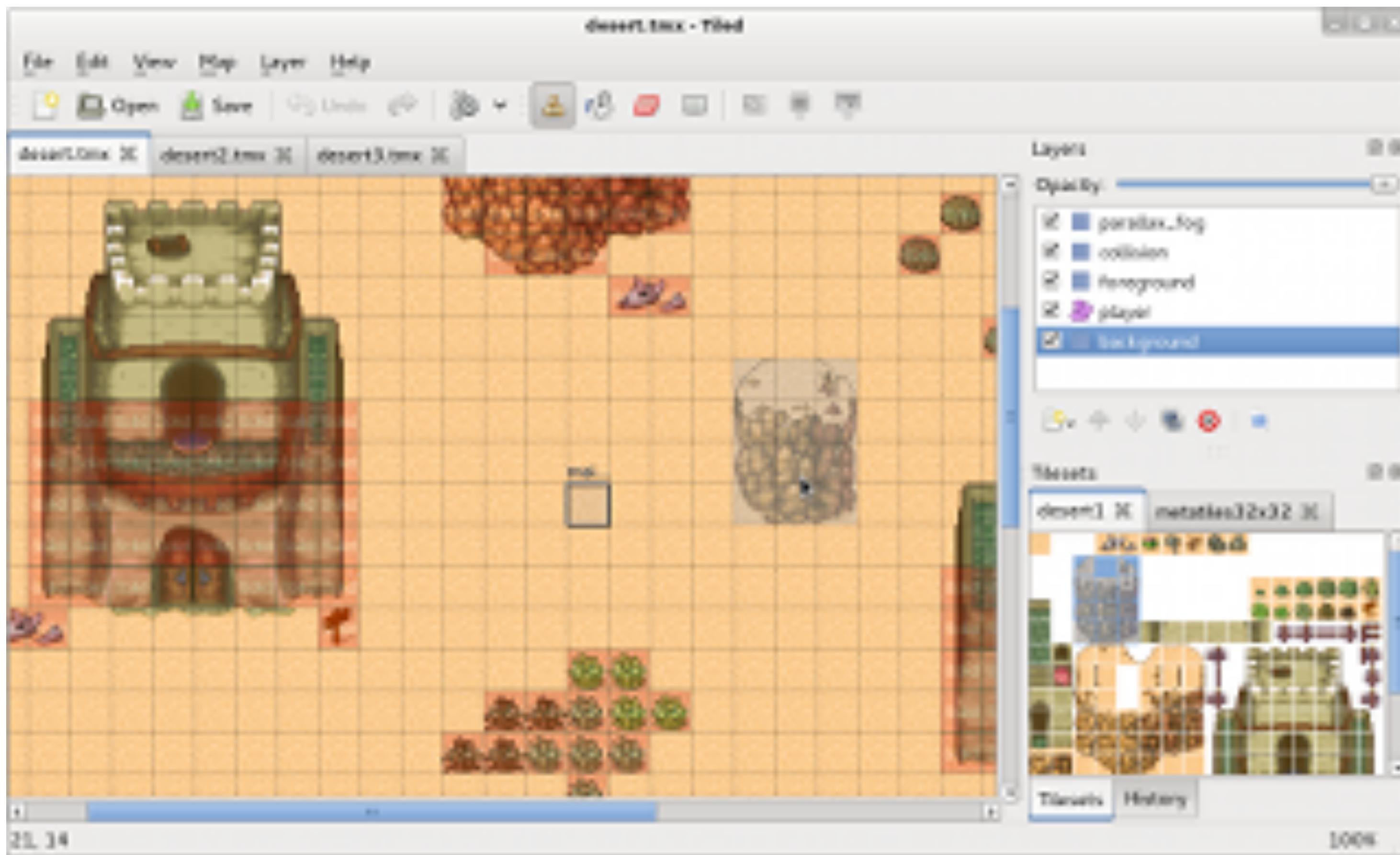


Texture Packer

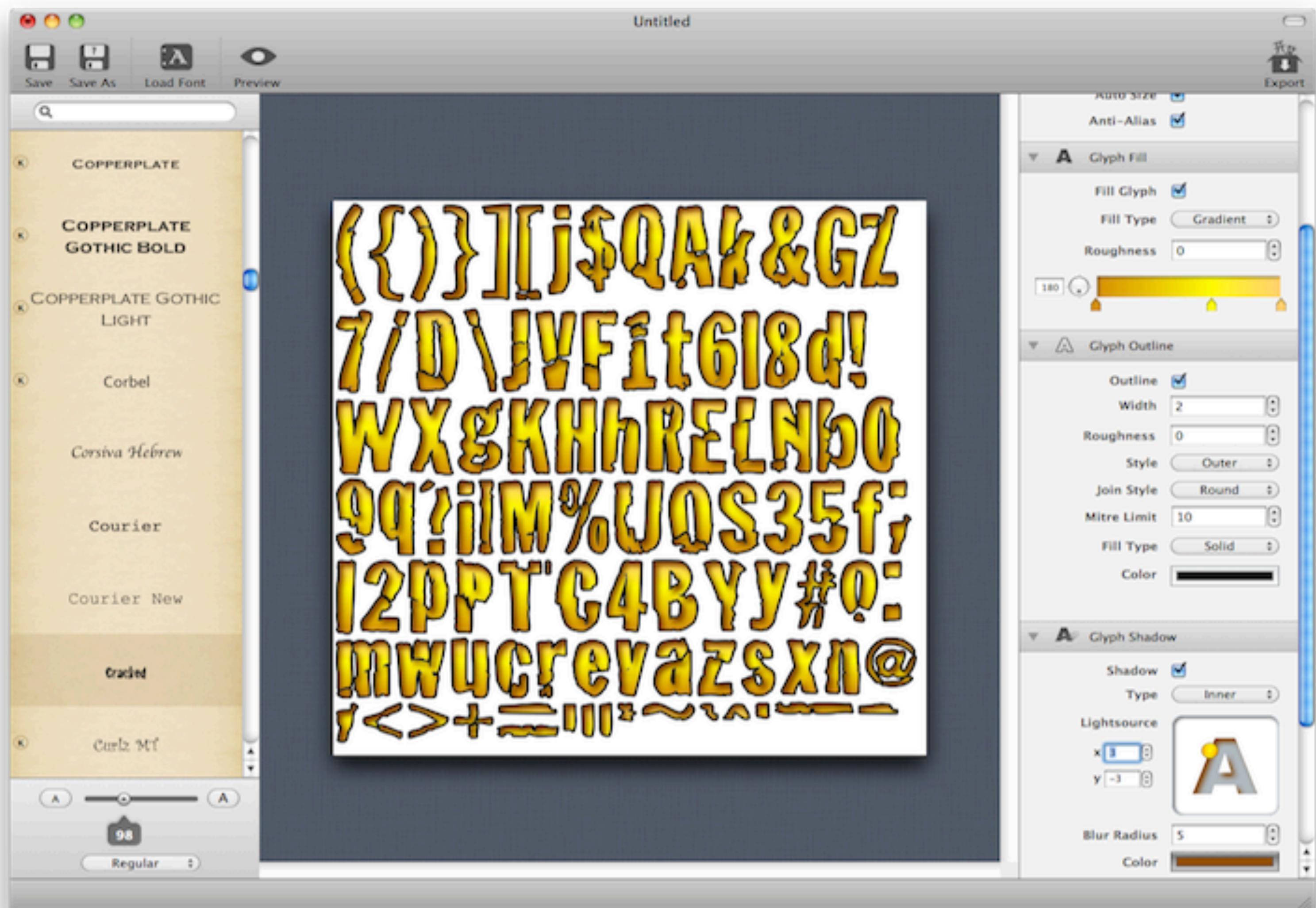
Particle Designer



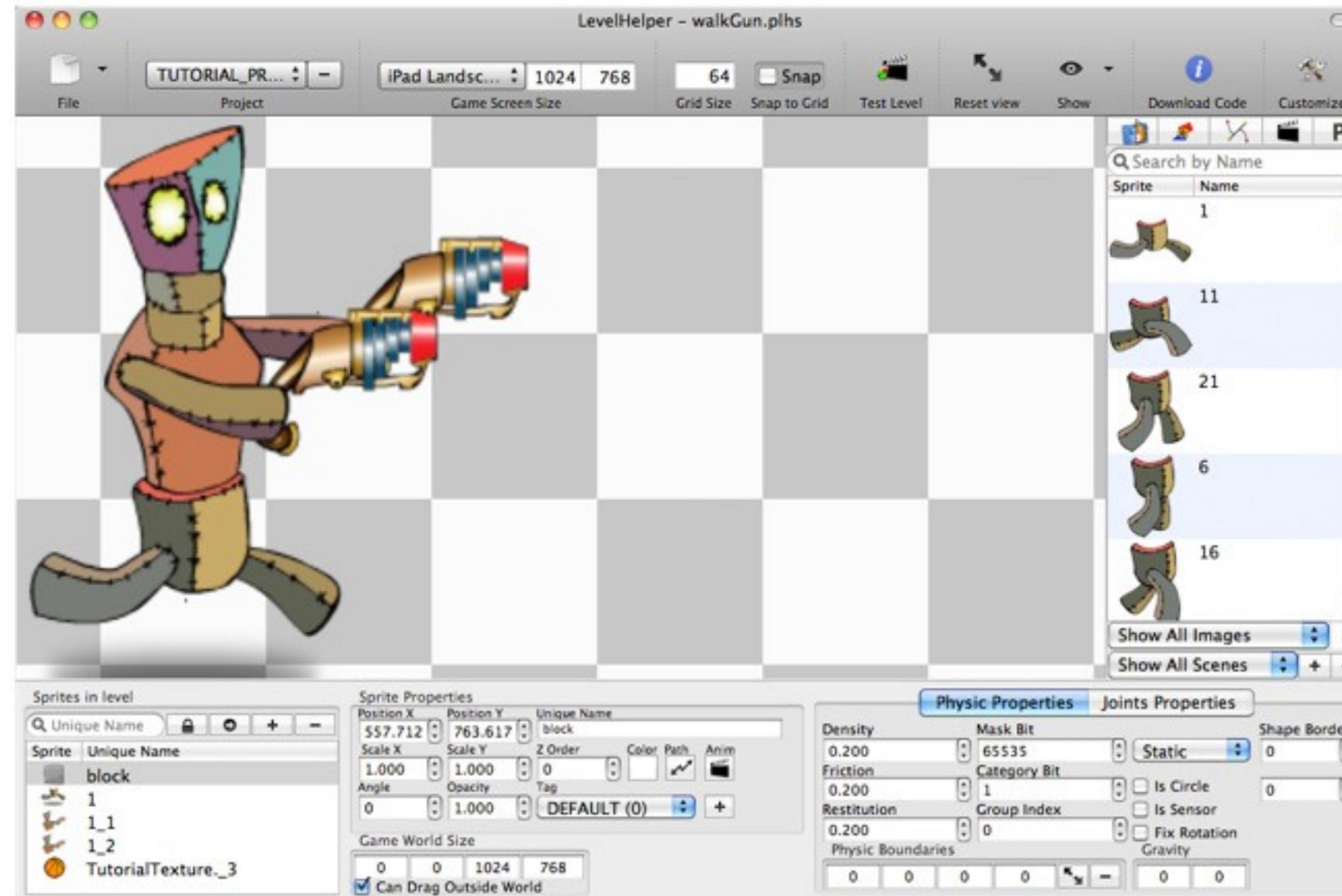
Tiled Map Editor



Glyph Designer



Level Helper



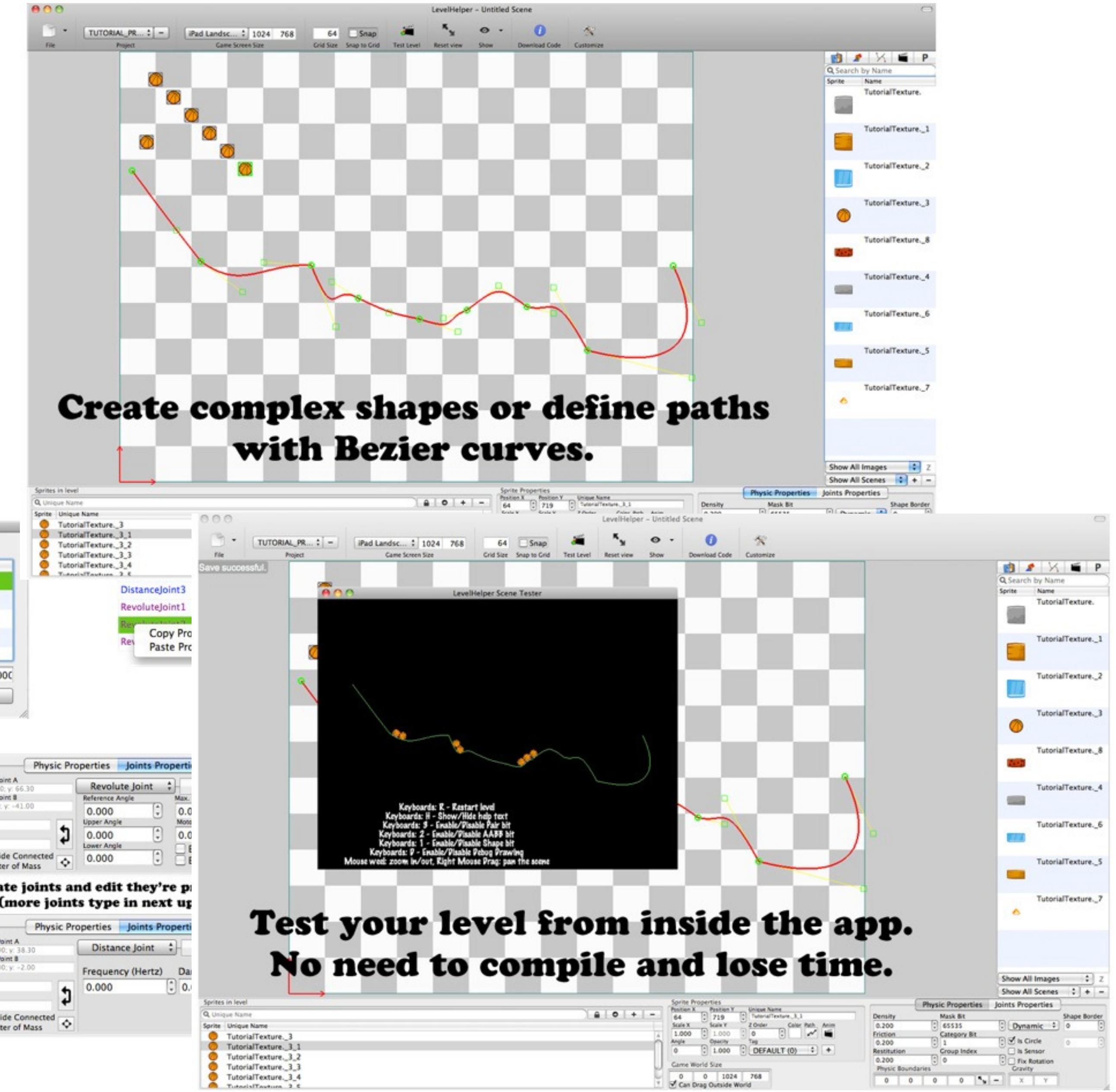
www.levelhelper.org

Edit physic properties and create complex interactions.

A screenshot of the LevelHelper interface showing a physics properties panel for a Cocos2d-X With Box2d joint. It displays anchor points, body A and B, and collision settings. The text "Edit physic properties and create complex interactions." is overlaid on this panel.

Support for all top engines
(more to come)

A screenshot of the LevelHelper interface showing a physics properties panel for a WiEngine joint. It displays anchor points, body A and B, and collision settings. The text "Support for all top engines (more to come)" is overlaid on this panel.



Physics Engines

Not strictly part of Cocos2D

Box2D (C++) or Chipmunk (C)

Set up physics model and update in
render loop

Box2D Init:

```
b2Vec gravity = b2Vec(0.0f, -10.0f);  
bool allowBodiesToSleep = true;  
world = new b2World(gravity, allowBodiesToSleep);
```

Define static container body:

```
b2BodyDef containerBodyDef;  
b2Body *containerBody = world->CreateBody(&containerBodyDef);  
b2PolygonShape screenBoxShape;  
int density = 0;  
screenBoxShape.SetAsEdge(lowerLeftCorner, lowerRightCorner);  
containerBody->CreateFixture(&screenBoxShape, density);  
...
```

Add a dynamic box to the world:

```
b2BodyDef bodyDef;  
bodyDef.type = b2_dynamicBody;  
bodyDef.position = [self toMeters:mySprite.position];  
bodyDef.userData = mySprite; // connection to Cocos2D  
b2Body *body = world->CreateBody(&bodyDef);
```

```
b2PolygonShape dynamicBox;  
float sizeInMeters = [self toMeters:SPRITE_SIZE_IN_POINTS];  
dynamicBox.SetAsBox(sizeInMeters * 0.5f, sizeInMeters * 0.5f);
```

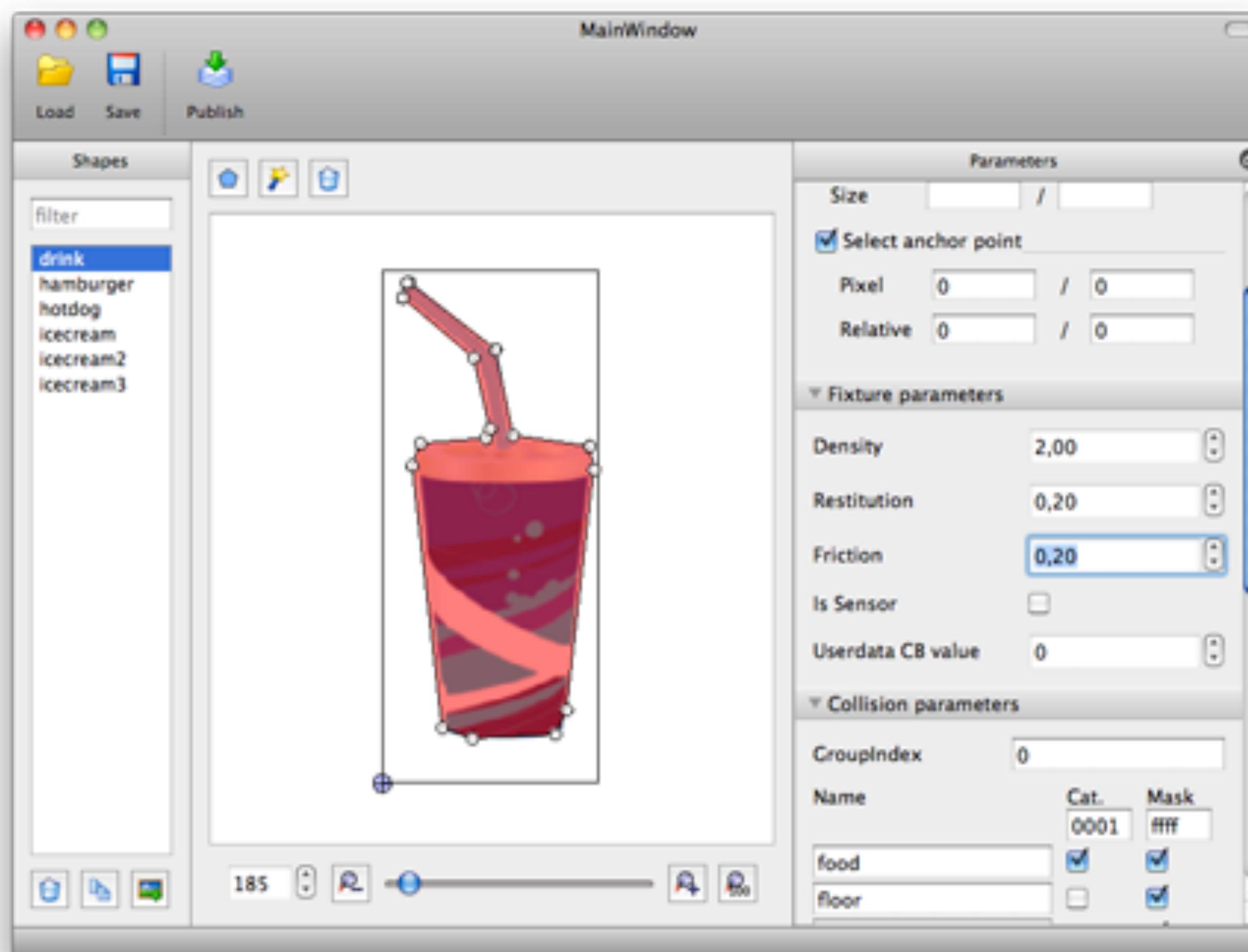
```
b2FixtureDef fixtureDef;  
fixtureDef.shape = &dynamicBox;  
fixtureDef.density = 0.3f;  
fixtureDef.friction = 0.5f;  
fixtureDef.restitution = 0.6f;  
body->CreateFixture(&fixtureDef);
```

Updating the world:

```
- (void)update:(ccTime)delta
{
    float timeStep = 0.03f;
    int32 velocityIterations = 8;
    int32 positionIterations = 1;
    world->Step(timeStep, velocityIterations, positionIterations);

    for (b2Body *body = world->GetBodyList(); body != nil; body->GetNext())
    {
        CCSprite *sprite = (CCSprite *)body->GetUserData();
        sprite.position = [self toPoints:body->GetPosition()];
        sprite.rotation = CC_RADIANS_TO_DEGREES(body->GetAngle()) * -1;
    }
}
```

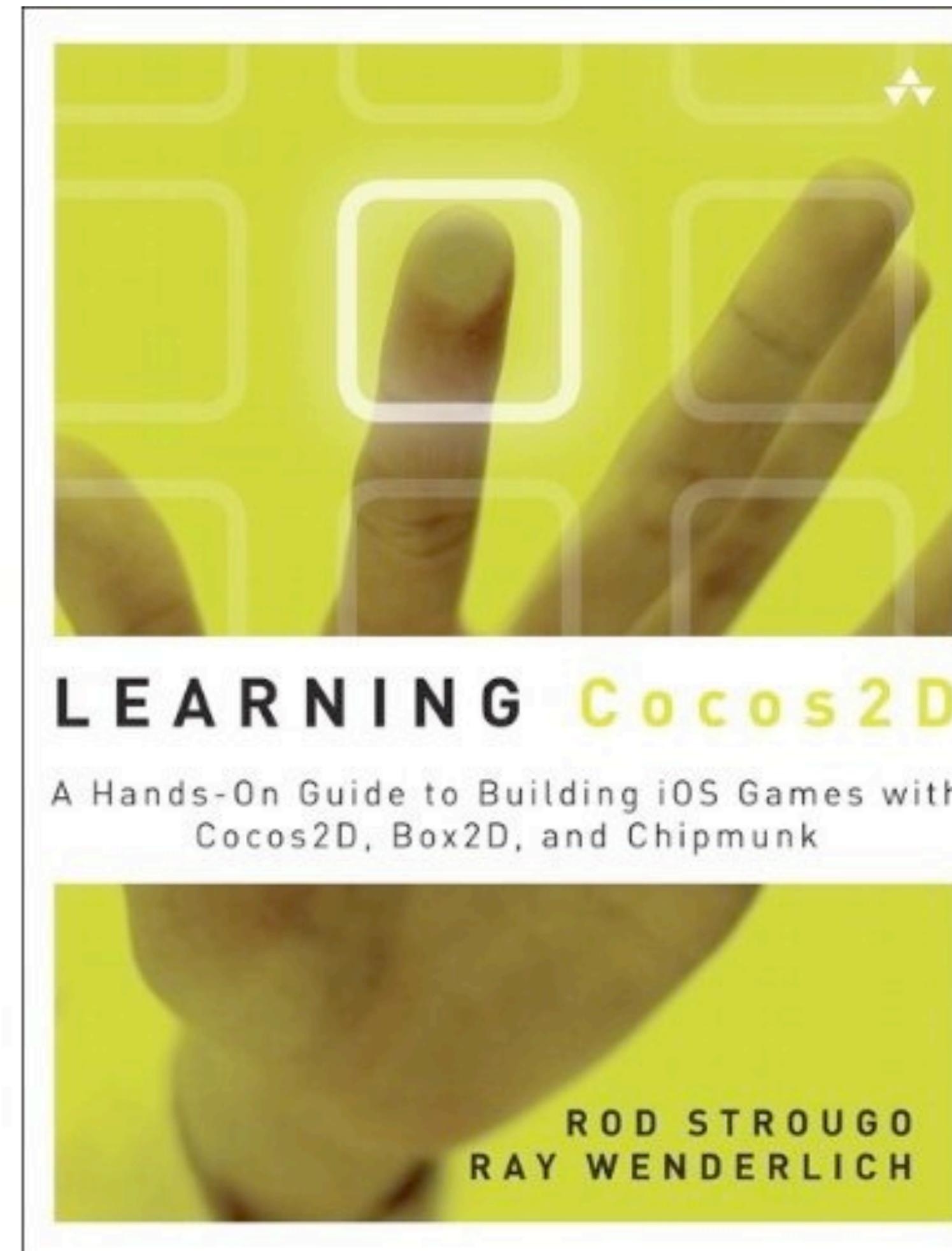
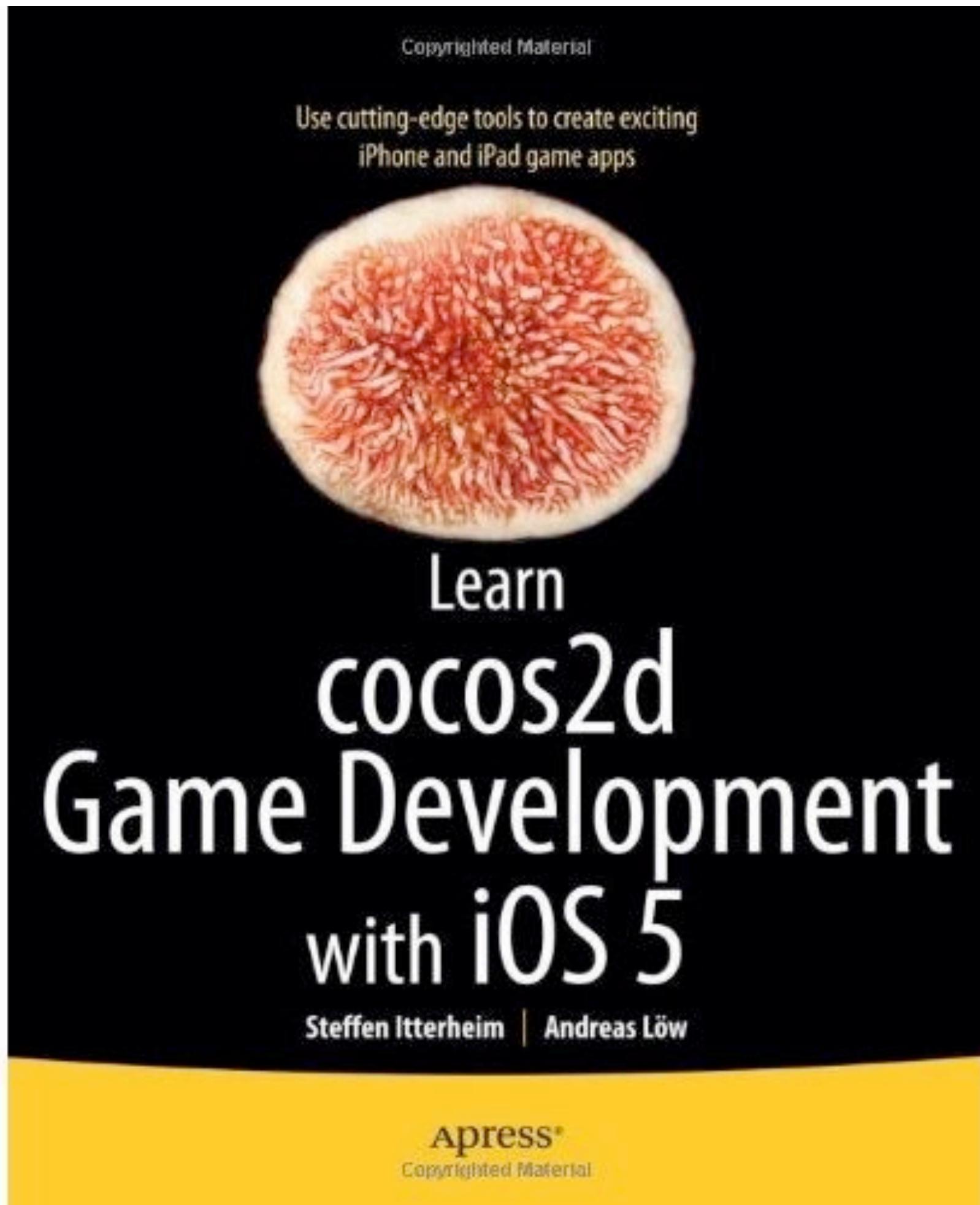
Physics Editor



cocos2d-iphone.org

learn-cocos2d.com

raywenderlich.com



Ole Begemann

oleb.net

@olebegemann