

How to use Cocos2d to build a successful mobile game

Ashik Raj Manandhar

Senior Mobile Application Engineer
Pocket Gems

Agenda

- Cocos2d and Me
- Overview
- Walkthrough
- Limitations
- Extensions
- Alternatives
- Questions



Ashik Raj Manandhar

Ashik was the third engineer hired at Pocket Gems, a Sequoia-backed mobile gaming company.

Over the past year, Ashik played a lead role in building Pet Hotel, a fun casual game for the iPhone that debuted as the #1 Top Grossing App and had millions of downloads. Ashik and his team worked hard to release nearly weekly feature and content updates that kept Pet Hotel consistently in the Top 10 Top Grossing apps. Pet Hotel was the fourth Top Grossing App Worldwide of 2011.

Ashik graduated from UC Berkeley with a BS in Electrical Engineering and Computer Science with a focus in Robotics. He was the Berkeley EECS Department 2009 Warren Dere Design Award recipient for the Most Outstanding Engineering Design for his work on an autonomous self-driving scaled model robotic car. Ashik worked on computer vision and media streaming software for large government projects at a Silicon Valley startup in the defense industry. Prior to Pocket Gems, Ashik did research in land robotics at the University of Michigan.



Pocket Gems

- Founded 2009, backed by Sequoia Capital
- 14 iOS and Android titles
- Pioneer in mobile games
 - 1st farm game
 - 1st store game
 - 1st zoo game
 - 1st hotel game



Tap Zoo and Tap Pet Hotel

#1 and #4 Top Grossing iPhone Apps of 2011



Longevity



Tap Zoo – Released Sep. 2010
12 months straight in the top 10 grossing apps



Tap Pet Hotel – Released Apr. 2011
8 months straight in the top 10 grossing apps

Tap Pet Hotel



Cocos2d

Cocos2d

- OO wrapper around OpenGL
- Open Source
- Fast
- Easy



Cocos2d

- Large community of developers
- Used by over 3000 games on the App Store



Cocos2d

- OpenGL = lots of code

- Load images into memory
- Calculate rotations
- Create run loop to call rotations
- Call run loop

- Cocos2d

```
CCRotateBy *rotation = [CCRotateBy actionWithDuration:2 angle:360];  
CCRepeatForever *repeat = [CCRepeatForever actionWithAction:rotation];  
[gem runAction:repeat];
```



Building Blocks

- Sprites
- Labels
- Menus
- Sounds
- Actions
- Action Sequences



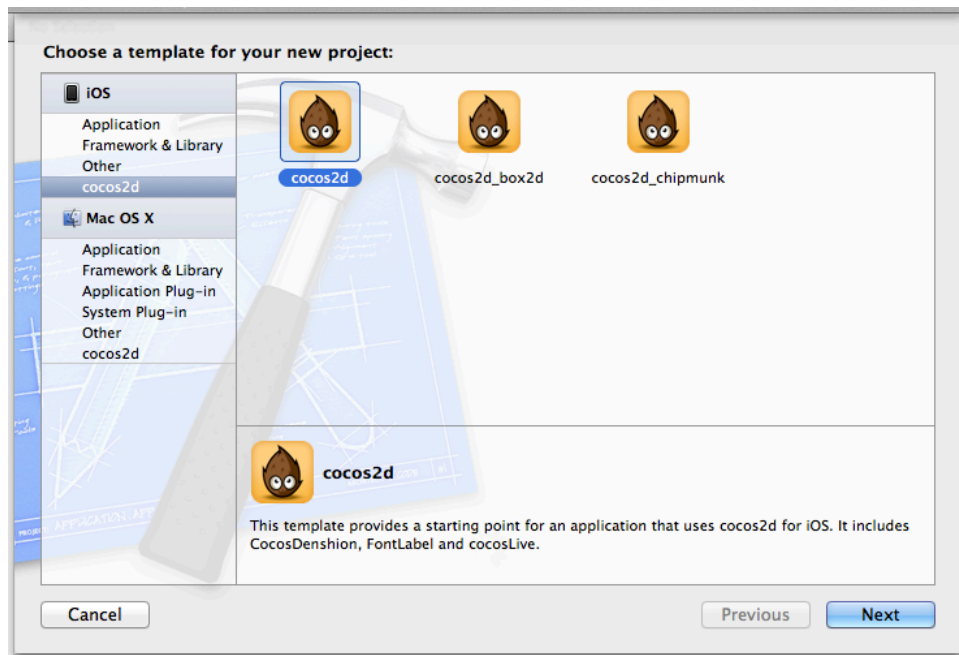
Pocket Full Of Gems

Simple game that uses D-Pad to move character and pick up gems



Walkthrough

Pocket Full Of Gems



Pocket Full Of Gems

- Add the character

// Load image and create character

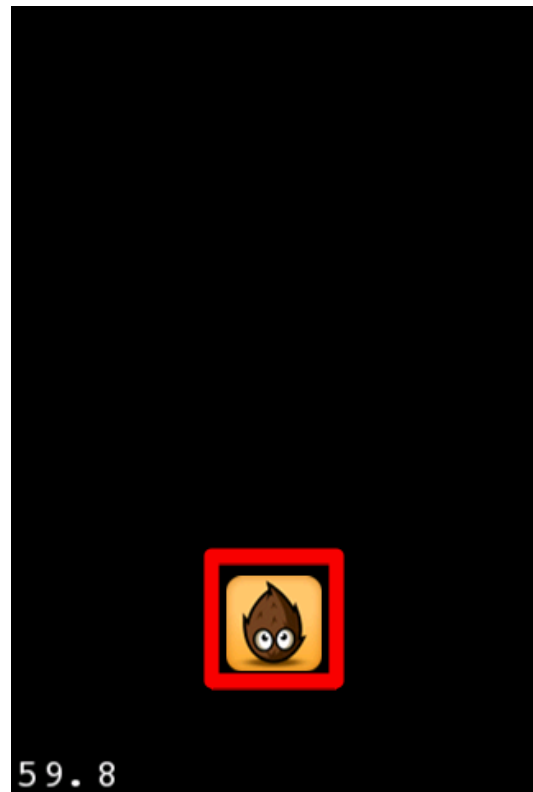
```
self.character = [CCSprite spriteWithFile:@"Icon-Small@2x.png"];
```

// Position the character

```
self.character.position = CGPointMake(size.width/2,  
                                     kBottomControls +  
                                     [self.character texture].contentSize.height/2);
```

// Place character on screen

```
[self addChild:self.character];
```



Pocket Full Of Gems

- Add the directional pad

```
// Load image and create left button sprites
```

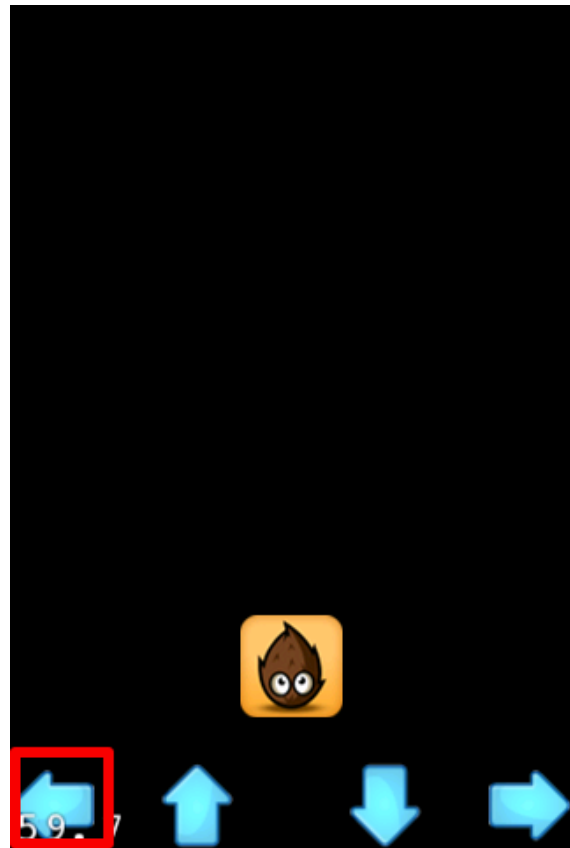
```
CCSprite *leftSprite = [CCSprite spriteWithFile:@"left.png"];  
CCSprite *leftSelectedSprite = [CCSprite spriteWithTexture:[leftSprite texture]];  
leftSelectedSprite.color = ccGRAY;
```

```
// Create left button menu item
```

```
CCMenuItemImage *leftButton = [CCMenuItemImage  
    itemFromNormalSprite:leftSprite  
    selectedSprite:leftSelectedSprite  
    target:self  
    selector:@selector(leftSelected)];  
leftButton.position = CGPointMake([leftSprite texture].contentSize.width/2,  
    [leftSprite texture].contentSize.height/2);
```

```
// Place on screen
```

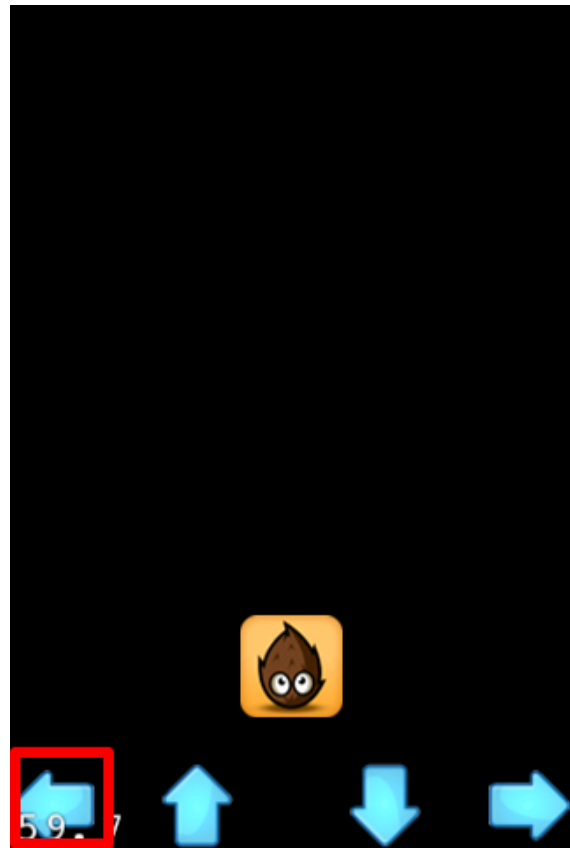
```
[menu addChild:leftButton];
```



Pocket Full Of Gems

- Update the position

```
- (void) leftSelected {  
    // Calculate new character position  
    int xPosition = self.character.position.x;  
    xPosition += [self.character texture].contentSize.width/2;  
  
    ... check bounds ...  
  
    // Update character position  
    self.character.position = CGPointMake(xPosition, self.character.position.y);  
  
    // Check to see if you picked up any gems  
    [self checkForCollisions];  
}
```



Pocket Full Of Gems

- Add the score

// Create the Score Label

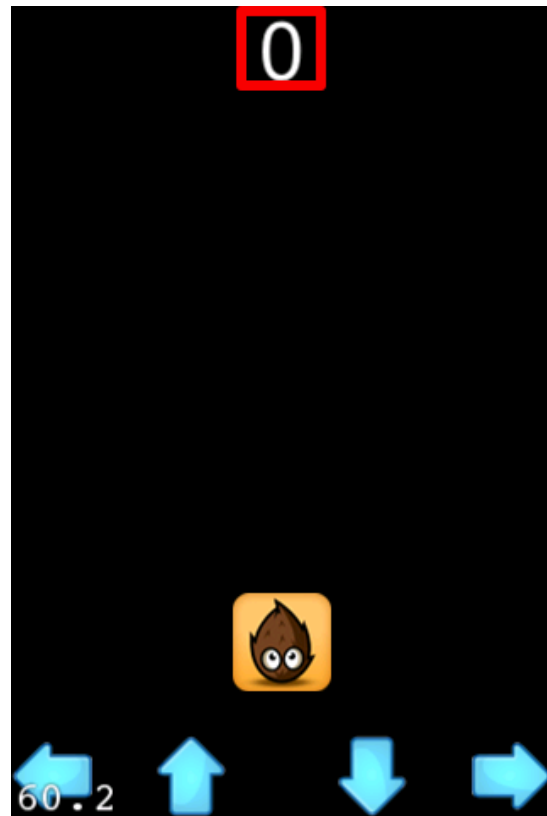
```
self.score = [CCLabelTTF labelWithString:@"0"  
              fontName:@"Arial"  
              fontSize:48];
```

// Position the score

```
self.score.position = CGPointMake(size.width/2,  
                                  size.height - [self.score texture].contentSize.height/2);
```

// Place it on screen

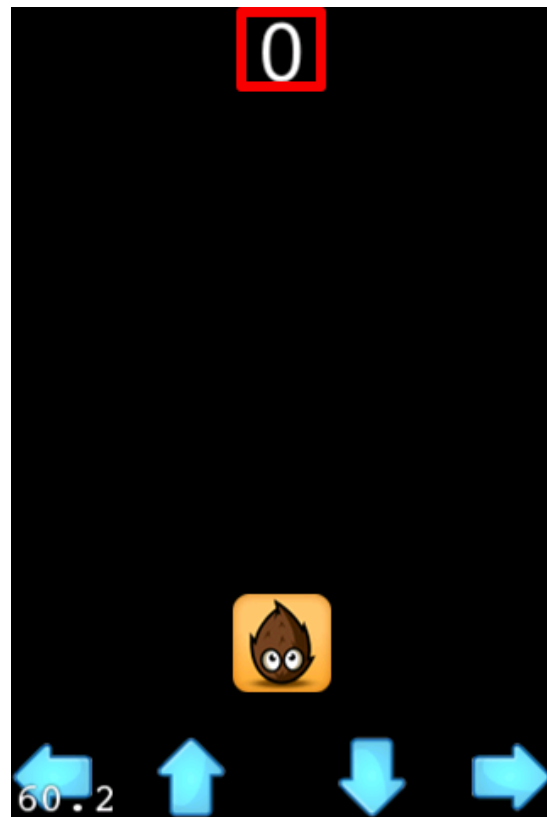
```
[self addChild:self.score];
```



Pocket Full Of Gems

- Update the score

```
- (void) updateScore {  
    // Update the score label  
    [self.score setString:[NSString stringWithFormat:@"%d",  
        self.points]];  
}
```



Pocket Full Of Gems

- Add the gems

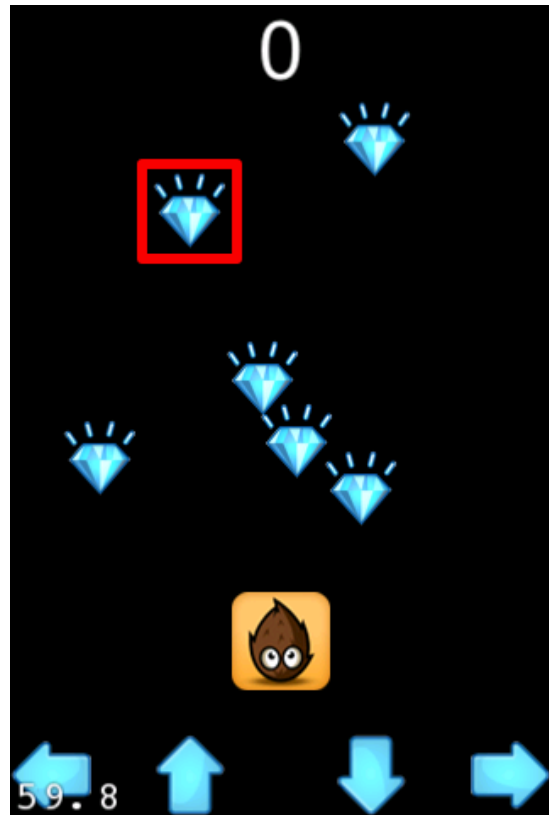
```
// Load the image and create a gem  
CCSprite *gem = [CCSprite spriteWithFile:@"gem.jpg"];
```

... find a random position ...

```
// Find the position  
gem.position = CGPointMake(xPosition, yPosition);
```

```
// Rotate the gem  
... Create rotation loop ...
```

```
// Add it on screen  
[self addChild:gem];
```



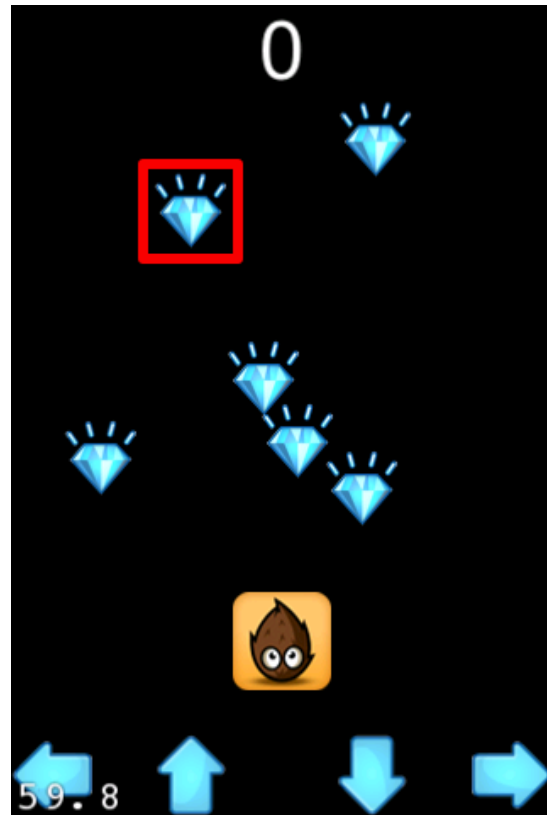
Pocket Full Of Gems

- Collision detection

```
// If the character and the gem overlap
if (ccpDistance(self.character.position - gem.position) < minDistance) {
    // Remove the gem off screen
    [self removeChild:gem cleanup:YES];

    // Add points
    self.points++;
}

// Update the score on screen
[self updateScore];
```



Pocket Full Of Gems



Walkthrough

Additional Features

Scenes

A low-angle, upward-looking photograph of the Atlas statue in a courtyard. The statue, a muscular man holding a globe, stands on a stone pedestal. It is surrounded by tall, light-colored stone buildings with many windows. In the background, a very tall skyscraper reaches towards a blue sky with some clouds. The perspective creates a sense of height and scale.

Atlasing

Debugging





Texture Management



Isometric Support

High Quality Games



Walkthrough



Limitations

Limitations

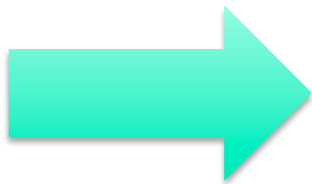
- Touch handling
- No support for gestures
- No support for scroll lists
- Performance can be sluggish when you add 10,000x things on screen

Culling



Limitations

Mipmap



Limitations

Extensions

- Subclassing the basic classes to create novel features
- Overriding the `draw` and `update` methods
- Compositing
- Improved Atlasing

Extensions

- Physics Engines
 - Box2D
 - Chipmunk
- Other open source extensions

When is Cocos2d the Wrong Choice?

- Real time 3D
- Complex/intricate menus

Alternatives

	Pros	Cons
OpenGL	+ Great performance	- Low-level - Hard to iterate fast
Unity	+ Cross Platform + 3D	- Proprietary
Corona	+ Wrapper around OpenGL + Cross Platform	- Proprietary

Questions?



Ashik Raj Manandhar

Senior Mobile Application Engineer
Pocket Gems

@AshikRaj

ashik.raj@pocketgems.com