

Cocos2d-x Programmer's Guide

C++ and Lua

Table Of Contents

1. About Cocos2d-x
 - A very brief history:
 - Launched in July 2010, cocos2d-x is the best open-source game engine available.
 - Versioning information (i.e C++, JS, Lua, not related to Cocos2d-iphone, etc)
 - Cocos2d-x comes in a variety of flavors:
 - * c++ [c++](#)
 - * Lua [Lua](#)
 - Prerequisites:
 - Operating systems and tools supported:
 - * For Android development.
 - * For iOS and OSX development.
 - * For Linux development.
 - * For Windows and Windows Phone development.
 - Why choose Cocos2d-x vs other game engines?
 - Where to get help:
 - [Official Forums](#)
2. Basic Concepts and Essentials
3. Sprites
 - What are Sprites?
 - Creating Sprites:
 - Creating a Textured Sprite.
 - Creating an Untextured Sprite.
 - Creating a Sprite From a SpriteFrame.
 - Creating a Sprite From SpriteCache.
 - Creating a Sprite From a Rect.
 - Sprite Manipulation:
 - Resizing.
 - Anchor Point and Position.
 - SpriteSheets.
4. Actions

- What are Basic Actions:
 - Animate.
 - Fade In/Out.
 - Move.
 - Rotate.
 - Scale.
 - Tint.
 - Tweening and Easing.
- What are Sequences:
 - Spawn.
 - Reverse.
- Running Actions and Sequences.

5. Building and Transitioning Scenes

- What is a Scene?
- Creating a Scene:
 - Creating a Node Tree.
 - Node properties to its descendants.
- Coordinate Systems:
 - Converting between coordinate systems.
- Transitioning between Scenes.

6. UI

- Labels:
 - BMFont.
 - TTF.
 - Label Atlas.
 - SystemFont.
- Label examples.
- Menu/Menu Items:
 - What makes up a menu?
 - Menu Items and adding to a Menu.
 - Examples:
 - * Create Menu with 1 item.
 - * Create Menu from array of items.
 - * Lambdas.
- Buttons.
- Scroll.
- Layout.

7. Other Node Types

- TMX.
- Particle.
- Parallax.

8. Event Dispatcher

- What is the EventDispatch mechanism?
 - Responds to user events.
 - The basics:
 - * Event listeners encapsulate your event processing code.
 - * Event dispatcher notifies listeners of user events.
 - * Event objects contain information about the events.
- 5 types of event listeners:
 - **EventListenerTouch** - responds to touch events
 - * describe what to override.
 - **EventListenerKeyboard** - responds to keyboard events
 - * describe what to override.
 - **EventListenerAcceleration** - responds to accelerometer events
 - * describe what to override.
 - **EventListenerMouse** - responds to mouse events
 - * describe what to override.
 - **EventListenerCustom** - responds to custom events
 - * describe what to override.
- Registering event with the dispatcher.
- Removing events from the dispatcher.

9. 3D

- 3D Sprite.
- 3D Actions.
- 3D Animations.
- Placeholders for: Lights, Shadows, Cameras.

10. Lua

- call custom c++ from Lua.
- bindings to c++.
- subclassing.
- Placeholders for: memory management, Debug a Lua Game.

11. Services

- Other SDK's.

- Plugin-X.
- Placeholders - IAP, FB.

12. Physics

- What options and why integrated physics engine?
- Physics concepts.
- Physics world, bodies.
- Collision.
- Examples.

13. Audio

14. Advanced Topics

- Best Practice - optimization, memory, performance, profiling.
- Sound.
- SQLite.
- Subclass Cocos2d-x classes.
- Data structures (i.e Vector).
- Custom OpenGL and Shaders (what to cover here? CustomCommand?).
- c++11 usage.
- Rendering pipeline (notes about this in the wiki).
- Networking - curl, http.
- ccConfig.h (include and the defines that you can enable for debugging memory leaks, drawing, etc.).
- File system access.
- Resolution independence.