Groups and Scenes

Groups

Groups are a collection of objects which are kept together; the most obvious example is a scene. Groups have their own new() function which operates identically but adds the object to the group, and their destructor removes all objects in that group.

Groups are implemented using tags; an “illegal” tag, group$<constant> is used in conjunction with core methods to add new or existing objects. An immediate consequence is that when the object is disposed, it is ‘removed’ from the group because it does not have the tag anymore (as tags are removed on dispose).

Groups are class system.group and are in system\group.lua

<instance> :new(className<string>,data<table>)

This overrides the new method (which would normally just create an instance) and creates an instance by calling the superclass new and adding it to the group (by tagging it with the ‘illegal’ tag), otherwise it is functionally identical.

<instance map>, <count> :getMembers()

getMembers is simply a *__getAllTaggedMembers* call with the illegal tag. It should be always remembered (as with other queries) that whilst all objects exist as part of the call, some may be disposed as a consequence of disposing others, so one should always check for isAlive()

Scenes

Scenes are a subclass of a group. Effectively they are a group of objects representing a current state in the game - both display objects and model objects.

Each Scene has an associated container with it which is used to clip the display to the boundary. This is passed into the constructor in the parameter container when the new() method is used.

Scenes are constructed and destructed on demand. Constructors and destructors operate as normal, either creating or deleting the scene and its component objects. Execution of the actual code can be done here; in the transition phase activity is stopped automatically by having the raw Update only mode set. It should be noted that if you use Corona’s transitions and timers they will continue as normal.

void setAlpha(alpha)

Sets the alpha for the current scene

void resetContainer()

Resets the container to its default status and brings it to the front

First version 26 Nov 2014
Redesigned scenes, 4th Jan 2015