GameModule

Inherits From: NSObject
Declared In: Solitaire/GameModule.h

Class Description

GameModule is an abstract superclass providing a generic framework for **.solitaire** game modules. It contains a number of methods which can be overridden to provide game-specific behavior.

The methods **startGame:**, **restartGame:**, and **checkForWin:** should always be overridden.

Symbolic Constants

SYNOPSIS

IBOutlet

Instance Variables

IBOutlet NSView* **inspector**;
IBOutlet NSWindow* **gameWindow**;
IBOutlet GamePref* **prefs**;
IBOutlet NSWindow* **rulesWindow**;
NSBundle* **bundle**;

NSString* **gameName**;
NSString* **frameName**;
CardSize **cardSize**;
BOOL **ignoreSizePref**;
CardBack **cardBack**;
NSColor* **desktopColor**;

inspector	Our inspector view
gameWindow	Active game window
prefs	Game's preference object
rulesWindow	Window containing rules
bundle	Our bundle for this module
gameName	The name of our game
frameName	Frame save string for gameWindow.
cardSize	Current size of the cards
ignoreSizePref	TRUE if only one size of cards is
cardBack	Pattern on card backs
desktopColor	Color of desktop

Method Types

Initialization/deallocation	- initFromBundle:withName:
Our inspector	- inspector: - inspectorWillBeRemoved - inspectorInstalled

Showing rules	- showRules:
Starting/stopping play	- startGame: - restartGame: - endGame:
Win/lose actions	- win - lose - checkForWin
Setup	- windowShouldClose: - windowDidMove: - nibPathForCardSize:realSize: - loadGameWindow:ofSize: - commonGameSetup

Instance Methods

checkForWin

- (void)**checkForWin**

Determine if the game has been won. Always override (unless your game is impossible to win). If you determine that the game has been won, then call [**self** win].

commonGameSetup

- (void)**commonGameSetup**

HELPER METHOD. DO NOT OVERRIDE. We load the correct game nib after rescanning our generic preferences.

endGame:

- (void)**endGame:**(id)*sender*

Sent when the game engine is about to switch to a different game. Default implementation hides the game window and rules window.

initFromBundle:withName:

- **initFromBundle:**(NSBundle*)*aBundle* **withName:**(NSString*)*name*

Designated initializer for the GameModule class. If this method is overridden in a subclass, always pass message back to this class via **super**.

inspector:

- (NSView*)**inspector:**(id)*sender*

Returns the module's inspector view.

inspectorInstalled

- (void)**inspectorInstalled**

Sent immediately after this module's inspector view has been installed in the Game Selection Panel. Override as needed. The default implementation does nothing.

inspectorWillBeRemoved

- (void)**inspectorWillBeRemoved**

Sent just before the module's inspector is removed; i.e. before switching to a new game. The default implementation does nothing.

loadGameWindow:ofSize:

- (void)**loadGameWindow:**(NSString*)*path* **ofSize:**(CardSize)*size*

Load the specified NIB file containing the game window.

lose

- (void)**lose**

Called when the game has been lost. Override to create new "lost" behaviour. Most games will not bother to detect losing situations.

nibPathForCardSize:realSize:

- (NSString*)**nibPathForCardSize:**(CardSize)*cardSize*
  realSize:(CardSize *)*realSize*

Attempt to find the NIB containing the indicated CardSize. If not found, try to find *any* game NIB. Returns the actual size found in '*realSize*'.

restartGame:

- (void)**restartGame:**(id)*sender*

Sent when user selects "Restart Game" option. The game should start over, without shuffling the deck. If overridden, always send the message back to the GameModule class via **super**.

showRules:

- (void)**showRules:**(id)*sender*

Displays the rules panel.

startGame:

- (void)**startGame:**(id)*sender*

Start a new game; i.e. shuffle the deck and deal. If overridden, always send the message back to the GameModule class via **super**.

win

- (void)**win**

Called when the game has been won. By default, invokes the generic "win" routine. Override to create custom "game won" behaviour.

windowDidMove:

- (void)**windowDidMove:**(NSNotification *)*notification*

Save the game window frame in the defaults database when the window has moved. (I thought this could be done automatically for you).

windowShouldClose:

- (BOOL)**windowShouldClose:**(id)*sender*

If the closing window is the game window we set our outlet to **nil** and set it's delegate to **nil** too.