

ElySION Frameworks
ElySIONKronos Thorium Bridge

Library Name „elysion.kronos“

Dependencies: None other except for all the dependencies ElySIONKronos has

Classes

TAppLayer

Important: Call with **Application**, TAppLayer is already initialized

Description: Handles general functions to control the application and window

Function return value	Function name	Description
void	showCursor()	Displays the standard black-and-white cursor on the screen
void	hideCursor()	Hides the cursor (Default)
void	takeScreenshot(string filename)	Takes a screenshot of the window Warning: Do not use! Results in an access violation at the moment
bool	setVideoMode(int width, int height, int colorbits, bool fullscreen)	Sets the window according to the parameter values if possible
void	toggleFullscreen()	Toggles between fullscreen and window mode: If the window is currently in window mode, it will then be in fullscreen after the function call and vice versa
void	quit()	Quits the application
property string	caption	Read-write property accesses window caption
property int	width	Read-only property to get the window's current width Set value through setVideoMode
property int	height	Read-only property to get the window's current height Set value through setVideoMode
property int	bits	Read-only property to get the window's current bit depth Set value through setVideoMode
property bool	fullscreen	Read-only property to get the window's current fullscreen Set value through setVideoMode

elPoint

Description: Provides a simple 2D point (integer value)

Function return value	Function name	Description
Constructor	create	Creates the class with zero values
Constructor	new(int x, int y)	Creates the class with specified x and y value
property int	x	Read-write property to access x value
property int	y	Read-write property to access y value

elfPoint

Description: Provides a simple 2D point (float value)

Function return value	Function name	Description
-----------------------	---------------	-------------

Constructor	create	Creates the class with zero values
Constructor	new(float x, float y)	Creates the class with specified x and y value
property float	x	Read-write property to access x value
property float	y	Read-write property to access y value

elVector

Description: Provides a simple 3D vector

Function return value	Function name	Description
Constructor	create	Creates the class with zero values
Constructor	new(int x, int y, int z)	Creates the class with specified x, y and z value
property int	x	Read-write property to access x value
property int	y	Read-write property to access y value
property int	z	Read-write property to access z value

elColor

Description: Provides a simple RGB color class

Note: Will introduce Alpha value in near future

Function return value	Function name	Description
Constructor	create	Creates the class with zero values
Constructor	new(byte r, byte g, byte b)	Creates the class with specified r, g and b value
property byte	r	Read-write property to access r value
property byte	g	Read-write property to access g value
property byte	b	Read-write property to access b value

elRect

Description: Provides a rectangle

Function return value	Function name	Description
Constructor	create	Creates the class with zero values
Constructor	new(int x, int y, int w, int h)	Creates the class with specified x, y,w and h value
property int	x	Read-write property to access x value
property int	y	Read-write property to access y value
property int	w	Read-write property to access w value
property int	h	Read-write property to access h value

elSprite

Description: Loads and displays a sprite

Function return value	Function name	Description
Constructor	create	Constructs sprite class
Constructor	new	Constructs sprite class
void	loadFromFile(string filename)	Loads sprite from filename

void	loadFromFileCI(string filename, int x, int y, int w, int h)	Loads sprite from filename and clips the image See also: clipImage Warning: Function name will change if Thorium will support overloaded functions
void	clipImage(int x, int y, int w, int h)	Clips image according to the parameters
void	setTransparencyColor(byte r, byte g, byte b)	Sets the transparency of the sprite according to the parameters Warning: Function name will change if Thorium will support overloaded functions
void	setTransparencyPoint(int x, int y)	Sets the transparency of the sprite according to the parameters Warning: Function name will change if Thorium will support overloaded functions
void	move(int deltax, int deltax)	Moves the sprite deltax and deltax pixels across the screen Warning: Does not always work.
void	draw()	Draws sprite on the screen
property elColor	color	(Read-write) Sets the color of the sprite
property elPoint	offset	(Read-write) Sets the offset position of the sprite
property elPoint	offsetRot	(Read-write) Sets the offset rotation of the sprite
property elVector	position	(Read-write) Sets the position of the sprite
property elfVector	scale	(Read-write) Scales the sprite
property float	rotation	(Read-write) Rotates the sprite
property byte	alpha	(Read-write) Sets alpha value
property boolean	visible	(Read-write) Sets visibilty of the sprite If visible is set to false, the sprite will not be drawn when draw() is called
property int	width	(Read-only) Returns the width of the sprite Only works if LoadFromFile(...) has been called
property int	height	(Read-only) Returns the height of the sprite Only works if LoadFromFile(...) has been called
property boolean	onMouseOver	(Read-only) Returns true if the the mouse is over the sprite
property boolean	onClick	(Read-only) Returns true if user clicked on sprite

elTimer

Description: Timer class

Function return value	Function name	Description
Constructor	create	Constructs timer class
Constructor	new	Constructs timer class
void	start()	Starts timer
void	stop()	Stops timer
void	pause()	Pauses timer
void	unPause()	Resumes timer after being paused
int	getTicks()	Returns timer ticks
property int	interval	(Read-write) Accesses interval of the timer (in milliseconds)
property boolean	active	Returns true if timer is not stopped
property boolean	onEvent	Returns true after every interval