Easy Lighting: Studio

Documentation

System Functions:

These functions handle the entire lighting system. All of these functions must be called at least once for the system to work correctly.

ls_system_properties

-Sets the global properties for the entire lighting system.

-Arguments:

enabled - (bool) Toggles the lighting system on/off

mode - (const) Specifies lighting mode. It can be one of the following:

1. *ls_dm_dull*
2. *ls_dm_normal*
3. *ls_dm_vivid*

ambient - (color) Specifies the ambient color for the system. AKA, the color of the "darkness" when there is no light.

caps - (bool) Specifies whether or not to cap each shadow object with darkness (recommended).

-Returns: N/A

ls_system_draw

-Draws the system onto the screen. This should be called ONCE in the draw event.

-Arguments:

update - (bool) Toggles whether to upate the system and draw (true) or just draw it (false).

debug - (bool) If toggled, lights aren't blended and debug data is displayed.

-Returns: N/A

View Functions:

These functions handle where the system is actually drawn, and whether or not to draw it in a view. Either ls_view_attach or ls_noview_attach must be called at least once for anything to be drawn.

ls_view_attach

-Attaches a view to the system. Every view attached will have it's own personal render of the system.

-Arguments:

view - (real) The view number to attach.

-Returns: (bool)

true - If the view was successfully attached.

false - If it failed to attach the view (E.g., if the view was already added to the system).

ls_view_detach

-Removes a view from the system. The resources allocated for this view will be freed, and the view will no longer display lighting.

-Arguments:

view - (real) The view number to remove.

Returns: (bool)

true - If the view was successfully removed.

false - If it failed to remove the view (E.g., if the view was not currently added to the system).

ls_view_recalculate

-Wipes all the data for all views and recalculates everything. This is normally only needed if the size of a visible view is changed.

-Arguments: NONE

-Returns: (bool)

true - If the views are successfully cleaned and reset.

false - If it fails to reset the views (E.g., if there are no views currently in the system).

ls_noview_attach

-Tells the system it should be drawn over the entire room and over every view. This is normally only desired if views are not being used and the room is small.

-Arguments: NONE

-Returns: (bool)

true - If the new render is created.

false - If it fails to be created (E.g., if it already exists).

ls_noview_detach

-Tells the system to remove the global render, and only draw on the views.

-Arguments: NONE

-Returns: (bool)

true - If the system is successfully removed.

false - If the system is NOT successfully removed (E.g., it doesn't exist).

Light Functions:

These functions control and create the lights themselves. Note that the ls_light_set_* functions CANNOT be called on static lights! With the exception of ls_light_set_static.

ls_light_create

-Defines a new light in the system that will be drawn with the next call.

-Arguments:

x	*- (real) x position in the room*
y	*- (real) y position in the room*
radius	*- (real) The distance the light will shine from it's origin.*
spread	*- (real) Between 0 and 360. The number in degrees the light will shine around it's origin.*
rot	*- (real) The direction in degrees which it should face. 0 degrees being right.*
color	*- (color) The color of the light.*
precision	*- (real) (The number of points used to define the circle around the light. NOTE: With precision of 32, a spread of 45 has the same detail as a 45 degree chunk from a light with the spread of 360.*
static	*- (bool) Whether or not the light should be updated with the system. True means "no."*
shadows	*- (bool) Whether or not this light should cast shadows.*

-Returns: (real)

id	*- The id of the newly created light.*

ls_light_destroy

-Removes a specified light from the system.

-Arguments:

id	*- (real) The id of the light to be removed.*

-Returns: (bool)

true	*- The light was successfully removed.*
false	*- The light was not removed or not found.*

ls_light_set_pos

-Gives a new position to an existing light.

-Arguments:

id	*- (real) The id of the light.*
x	*- (real) The new x position in the room*
y	*- (real) The new y position in the room*

-Returns: N/A

ls_light_set_rot

-Rotates a light to the specified angle.
-Arguments:
id - (real) The id of the light.
rot - (real) The new angle of the light in degrees.
-Returns: N/A

ls_light_set_static

-Toggles whether or not a light is static.
-Arguments:
id - (real) The id of the light.
static - (bool) Whether or not it should be static
-Returns: N/A

ls_light_set_spread

-Defines a new spread for a given light.
-Arguments:
id - (real) The id of the light.
spread - (real) Between 0 and 360, the new spread.
-Returns: N/A

ls_light_set_color

-Defines a new color for a given light.
-Arguments:
id - (real) The id of the light.
color - (color) The new color to be assigned.
-Returns: N/A

ls_light_set_shadows

-Toggles whether this light casts shadows or not.
-Arguments:
id - (real) The id of the light.
shadows - (bool) Whether or not it should cast shadows.
-Returns: N/A

ls_light_get_x

-Finds the x position of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (real)
x position - The current x position of the light.

ls_light_get_y

-Finds the y position of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (real)
y position - The current y position of the light.

ls_light_get_rot

-Finds the rotation of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (real)
rotation - The current rotation of the light.

ls_light_get_static

-Finds the state of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (bool)
true - The light is static
false - The light is not static

ls_light_get_spread

-Finds the spread of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (real)
spread - The current spread of the light.

ls_light_get_color

-Finds the rotation of a given light.
-Arguments:
id - (real) The id of the light.
-Returns: (color)
color - The current color of the light.

ls_light_get_shadows

-Finds whether it is setup to cast shadows or not.
-Arguments:
id - (real) The id of the light.
-Returns: (bool)
bool - Whether or not shadows are toggled.

ls_light_manual_update

-Manually updates a given light and it's shadows. Generally only needed when updating a static light.
-Arguments:
id - (real) The id of the light.
-Returns: N/A

Shadow Functions:

These functions define and update the shapes and positions of shadows in the system. When defining points, they should be added in a clockwise or counter-clockwise manner. Concave shapes can occasionally cause problems, so try to keep them convex whenever possible.

ls_shadow_create

-Creates a new shadow shape object.
-Arguments: NONE
-Returns: (real)
id - The id of the new object.

ls_shadow_destroy

-Removes a specified shadow object from the system.
-Arguments:
id - (real) The id of the shadow shape.
-Returns: N/A

ls_shadow_addpoint_polar

-Adds a new point to a shadow shape using polar coordinates.
-Arguments:
id - (real) The id of the shadow shape to add to.
angle - (real) The angle from the relative origin
radius - (real) The distance from the relative origin.
-Returns: (real)
pos - The specific position of the point in the shadow object.

ls_shadow_addpoint_cartesian

-Adds a new point to a shadow shape using cartesian coordinates.
-Arguments:
id - (real) The id of the shadow shape to add to.
x - (real) Relative x position of the point.
y - (real) Relative y position of the point.
-Returns: (real)
pos - The specific position of the point in the shadow object.

ls_shadow_setpoint_polar

-Modifies the position of an existing point in a shadow shape in polar.
-Arguments:
id - (real) The id of the shadow shape.
pos - (real) The position of the point to modify.
angle - (real) The new relative angle of the point.
radius - (real) The new relative distance from the origin.
-Returns: N/A

ls_shadow_setpoint_cartesian

-Modifies the position of an existing point in a shadow shape in cartesian.
-Arguments:
id - (real) The id of the shadow shape.
pos - (real) The position of the point to modify.
x - (real) The new relative x position of the point.
y - (real) The new relative y position of the point.
-Returns: N/A

ls_shadow_attach_pos

-Attaches a defined shadow shape to a position in the room. Multiple attachments can be made per shape.
-Arguments:
id - (real) id of the shadow shape to use.
x - (real) x position in the room.
y - (real) y position in the room.
culling - (bool) Toggles culling for the shape at this position.
-Returns: (real)
id - id of the list holding the position.

ls_shadow_detach_pos

-Removes a specified position from the system so it is no longer drawn in the room.
-Arguments:
id - (real) id of the position to remove.
-Returns: (bool)
true - The position was successfully removed.
false - The system failed to remove the position.

ls_shadow_modify_pos

-Changes the location of a specified position.
-Arguments:
id - (real) id of the position to modify.
x - (real) The new x position in the room.
y - (real) The new y position in the room.
-Returns: (bool)
true - The position was successfully modified.
false - The position was not modified.

ls_shadow_clearpoints

-Removes all the points from a given shadow shape.
-Arguments:
id - (real) id of the shadow shape to clear.
-Returns: (bool)
true - The points were removed from the shape.
false - The system failed to remove the points.

Constants:

These are added system constants that can be used in certain functions.

Lighting Mode:

ls_dm_dull	*- Non-additive simplistic lighting.*
ls_dm_normal	*- Additive realistic lighting.*
ls_dm_vivid	*- Additive lighting with brighter color mixtures.*