on Keyframe Structures

Keyheader3ds

Key temporal and spline attributes

typedef struct

{

ulong3ds time;

ushort3ds rflags;

float3ds tension;

float3ds continuity;

float3ds bias;

float3ds easeto;

float3ds easefrom;

} keyheader3ds;

ulong3ds time

The frame number the key occurs in. Default is 0.

ushort3ds rflags

A set of flags that define which spline terms are in use.

static const ushort3ds KeyUsesNone3ds = 0x00;static const ushort3ds KeyUsesTension3ds = 0x01;static const ushort3ds KeyUsesCont3ds = 0x02;static const ushort3ds KeyUsesBias3ds = 0x04;static const ushort3ds KeyUsesEaseTo3ds = 0x08;static const ushort3ds KeyUsesEaseFrom3ds = 0x10;

Default is KeyUseNone3ds.

float3ds tension

Spline tension value from -1.0–1.0. Default is 0.0.

float3ds continuity

Spline continuity value from -1.0–1.0. Default is 0.0.

float3ds bias

Spline bias value from -1.0–1.0. Default is 0.0.

float3ds easeto

Key ease to value from 0.0 –1.0. Default is 0.0.

float3ds easefrom

Key ease from value from 0.0 –1.0. Default is 0.0.

Camera Animation

Kfcamera3ds

Camera Animation

Camera animation key structure.

typedef struct {

char3ds name[11];

short3ds parent;

      ushort3ds flags1;

ushort3ds flags2;

      ulong3ds npkeys;

ushort3ds npflag;

keyheader3ds \*pkeys;

point3ds \*pos;

      ulong3ds nfkeys;

ushort3ds nfflag;

keyheader3ds \*fkeys;

float3ds \*fov;

      ulong3ds nrkeys;

ushort3ds nrflag;

keyheader3ds \*rkeys;

float3ds \*roll;

      short3ds tparent;

ulong3ds ntkeys;

ushort3ds ntflag;

keyheader3ds \*tkeys;

point3ds \*tpos;

ushort3ds tflags1;

ushort3ds tflags2;

} kfcamera3ds;

char3ds name[11]

Name of camera to animate. Defaults to empty string.

short3ds parent

Parent index. Obsolete and must be replaced with something that works before the product is released.

ushort3ds flags

Special node flags, any combination of the following:

ushort3ds flags2

Special node flags, any combination of the following:

static const ushortds ShowPath3ds = 1;Display the camera motion path.

ulong3ds npkeys

Count of position keys in track. Default is 0.

ushort3ds npflag

Track options for position track, can be one of the following:

static const ushort3ds TrackSingle3ds = 0;Track does not repeat or loop.

static const ushort3ds TrackLoop3ds = 3;Track loops.

static const ushort3ds TrackRepeat3ds = 3;Track repeats.

Default is TrackSingle3ds.

keyheader3ds \*pkeys

Array of spline path modifiers for the position keys. Must have same element count as npkeys.

point3ds \*pos

Array of position keys. Element count must be the same number as npkeys.

ulong3ds nfkeys

Count of FOV keys in track. Default is 0.

ushort3ds nfflag

Track options for FOV track. See npflag for their meanings.

keyheader3ds \*fkeys

Array of spline modifiers for the FOV keys. Element count must be the same as nfkeys.

float3ds \*fov

Array of FOV values. Element count must be the same number as nfkeys.

ulong3ds nrkeys

Number of roll keys in the track. Default is 0.

ushort3ds nrflag

Track options for roll track. Se npflag for their meanings.

keyheader3ds \*rkeys

Array of spline modifiers for camera roll. Element count must be the same number as nrkeys.

float3ds \*roll

Array of camera roll keys. Element count must be the same number as nrkeys.

short3ds tparent

Parent index for camera target.

ulong3ds ntkeys

Number of track position keys. Default is 0.

ushort3ds ntflag

Track options for target keys. See npflag for thier meanings.

keyheader3ds \*tkeys

Array of spline path modifiers for target keys. Element count must be the same number as ntkeys.

point3ds \*tpos

Array of target position keys. Element count must be the same number as ntkeys.

ushort3ds tflags1

Special node flags.

ushort3ds tflags2

Special node flags.

InitCameraMotion3ds

***Prototype:*** void InitCameraMotion3ds(kfcamera3ds \*\*cam, ulong3ds npkeys, ulong3ds nfkeys, ulong3ds nrkeys, ulong3ds ntkeys);

Initializes kfcamera3ds structure and optionally allocates memory.

Arguments:

kfcamera3ds \*\*cam

Address of a pointer to a kfcamera3ds structure. If pointer is NULL, then memory will be allocated for a new structure and the address stored in the pointer.

ulong3ds npkeys

Allocate memory for specified number of position keys. A value of 0 will leave the position key fields unchanged.

ulong3ds nfkeys

Allocate memory for specified number of FOV keys. A value of 0 will leave the FOV keys fields unchanged.

ulong3ds nrkeys

Allocate memory for specified number of roll keys. A value of 0 will leave the roll keys fields unchanged.

ulong3ds ntkeys

Allocate memory for specified number of target keys. A value of j0 will leave the target key fields unchanged.

Returns nothing.

ReleaseCameraMotion3ds

***Prototype:*** void ReleaseCameraMotion3ds(kfcamera3ds \*\*cam);

Frees memory allocated to a kfcamera3ds structure.

Arguments:

kfcamera3ds \*\*cam

The structured to be freed. After release, the pointer is set to NULL.

Returns nothing.

GetCameraNodeCount3ds

***Prototype:*** ulong3ds GetCameraNodeCount3ds(database3ds \*db);

Scans the database for camera nodes and returns a count of their number.

Arguments:

database3ds \*db

The database to scan for camera nodes.

Returns **ulong3ds** as the count of camera nodes.

GetCameraNodeNameList3ds

***Prototype:*** void GetCameraNodeNameList3ds(database3ds \*db, namelist3ds \*\*list);

Scans the database for camera nodes and returns a list of their names.

Arguments:

database3ds \*db

The database to scan for camera nodes.

namelist3ds \*\*list

The list of camera node names found. Point gets past to InitNameList3ds prior to use.

Returns nothing.

GetCameraMotionByIndex3ds

***Prototype:*** void GetCameraMotionByIndex3ds(database3ds \*db, short3ds index, kfcamera3ds \*\*kfcam)

Searches the database for the index(th) occurance of a camera node, and copies that node into camera motion structure. If the index is out of range, the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

ulong3ds index

The order of occurance of the node in the database. Valid range is 0 to one less than the count returned by GetCameraMotionCount3ds.

kfcamera3ds \*\*kfcam

The camera node structure that receives the settings. The pointer is passed to InitCameraMotion3ds prior to use.

Returns nothing.

GetCameraMotionByName3ds

***Prototype:*** void GetCameraMotionByName3ds(database3ds \*db, char3ds \*name, kfcamera3ds \*\*kfcam)

Seaches the database for a camera node of the given name and copies it into a camera node structure. If a the name is not found, then the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

char3ds \*name

The 1 to 10 character name to search for. The search is case sensitive.

kfcamera3ds \*\*kfcam

The camera node structure that receives the settings. The pointer is passed to InitCameraMotion3ds prior to use.

PutCameraMotion3ds

***Prototype:*** void PutCameraMotion3ds(database3ds \*db, kfcamera3ds \*kfcam)

Copies the camera motion settings from the structure into the database. Camera nodes with the same name are replaced with the new settings.

Arguments:

database3ds \*db

The database to copy the camera node into.

kfcamera3ds \*kfcam

The camera node being added to the database.

Returns nothing.

Ambient Light Animation

Kfambient3ds

Ambient Light Animation Structure

Ambient light animation keys for color.

typedef struct {

char3ds name[11];

short3ds parent;

      ushort3ds flags1;

ushort3ds flags2;

      ulong3ds nckeys;

ushort3ds ncflag;

keyheader3ds \*ckeys;

fcolor3ds \*color;

} kfambient3ds;

char3ds name[11]

Ambient light name should always be "$AMBIENT$"

short3ds parent

Parent node. Not used.

ushort3ds flags1

Various flags. Will be replaced with something better.

ushort3ds flags2

ulong3ds nckeys;

Count of color keys in track. Default is 0.

ushort3ds ncflag;

Track options for color track, can be one of the following:

static const ushort3ds TrackSingle3ds = 0;Track does not repeat or loop.

static const ushort3ds TrackLoop3ds = 3;Track loops.

static const ushort3ds TrackRepeat3ds = 3;Track repeats.

Default is TrackSingle3ds.

keyheader3ds \*ckeys;

Array of spline path modifiers for the color keys. Must have same element count as nckeys.

fcolor3ds \*color;

Array of color keys. Element count must be the same number as nckeys.

InitAmbientLightMotion3ds

***Prototype:*** void InitAmbientLightMotion3ds(kfambient3ds \*\*light, ulong3ds nckeys)

Initializes kfambient3ds structure and optionally allocates memory.

Arguments:

kfambient3ds \*\*light

Address of a pointer to a kfambient3ds structure. If pointer is NULL, then memory will be allocated for a new structure and the address stored in the pointer.

ulong3ds nckeys

Allocate memory for specified number of color keys. A value of 0 will leave the color key fields unchanged.

Returns nothing.

ReleaseAmbientLightMotion3ds

***Prototype:*** void ReleaseAmbientLightMotion3ds(kfambient3ds \*\*light);

Frees memory allocated to a kfambient structure.

Arguments:

kfambient3ds \*\*cam

The structured to be freed. After release, the pointer is set to NULL.

Returns nothing.

GetAmbientLightMotionByName3ds

***Prototype:*** void GetAmbientLightMotion3ds(database3ds \*db, kfambient3ds \*\*light)

Seaches the database for a ambient light node and copies it into a ambient light node structure. If a node is not found, then the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

kfambient3ds \*\*light

The ambient light node structure that receives the settings. The pointer is passed to InitAmbientLightMotion3ds prior to use.

PutAmbientLightMotion3ds

***Prototype:*** void PutAmbientLightMotion3ds(database3ds \*db, kfambient3ds \*light)

Copies the ambient light motion settings from the structure into the database. Any existing ambient light node is replaced with the new settings.

Arguments:

database3ds \*db

The database to copy the ambient light node into.

kfambient3ds \*light

The ambient light node being added to the database.

Returns nothing.

Mesh Object Animation

Kfrotkey3ds

Rotation Key

typedef struct {

float3ds angle;

float3ds x;

float3ds y;

float3ds z;

} kfrotkey3ds;

float3ds angle

Angle of rotation in degrees. Defaults to 0.0.

float3ds x

X component of rotation axis. Defaults to 0.0.

float3ds y

Y component of rotation axis. Defaults to 0.0.

float3ds z

Z component of rotation axis. Defaults to 0.0.

Kfmorphkey3ds

Morph Object Key

typedef struct {

char3ds name[11];

} kfmorphkey3ds;

char3ds name[11]

Name of morph target mesh. Mesh must have same number of vertices as base object.

Kfmesh3ds

Mesh Animation

typedef struct {

char3ds name[11];

short3ds parent;

ushort3ds flags1;

ushort3ds flags2;

      point3ds pivot;

char3ds instance[11];

point3ds boundmin;

point3ds boundmax;

      ulong3ds npkeys;

short3ds npflag;

keyheader3ds \*pkeys;

point3ds \*pos;

      ulong3ds nrkeys;

short3ds nrflag;

keyheader3ds \*rkeys;

kfrotkey3ds \*rot;

      ulong3ds nskeys;

short3ds nsflag;

keyheader3ds \*skeys;

point3ds \*scale;

      ulong3ds nmkeys;

short3ds nmflag;

keyheader3ds \*mkeys;

kfmorphkey3ds \*morph;

      ulong3ds nhkeys;

short3ds nhflag;

keyheader3ds \*hkeys;

      float3ds msangle;

} kfmesh3ds;

char3ds name[11];

Name of mesh to animate. Special name of "$$$DUMMY" is used for a dummy object. Defaults to empty string.

short3ds parent

Parent index. Obsolete and must be replaced with something that works before the product is released.

ushort3ds flags

Special node flags, any combination of the following:

ushort3ds flags2

Special node flags, any combination of the following:

static const ushortds ShowPath3ds = 1;Display the object motion path.

point3ds pivot

Rotation pivot point for object node. Defaults to 0.0, 0.0, 0.0.

char3ds instance[11]

Name of object node instance.

point3ds boundmin

Point defining the upper, left, and front extents of the object

point3ds boundmax

Point defining the lower, right, and back extents of the object.

ulong3ds npkeys

Count of position keys in track. Default is 0.

ushort3ds npflag

Track options for position track, can be one of the following:

static const ushort3ds TrackSingle3ds = 0;Track does not repeat or loop.

static const ushort3ds TrackLoop3ds = 3;Track loops.

static const ushort3ds TrackRepeat3ds = 3;Track repeats.

Default is TrackSingle3ds.

keyheader3ds \*pkeys

Array of spline path modifiers for the position keys. Must have same element count as npkeys.

point3ds \*pos

Array of position keys. Element count must be the same number as npkeys.

ulong3ds nrkeys

Count of rotation keys in track. Default is 0.

short3ds nrflag

Track options for rotation track. See npflag for their meanings.

keyheader3ds \*rkeys

Array of spline path modifiers for the rotation keys. Must have same element count as nrkeys.

kfrotkey3ds \*rot

Array of rotation keys. Element count must be the same number as nrkeys.

ulong3ds nskeys

Count of scale keys in track. Default is 0.

short3ds nsflag

Track options for scale track. See npflag for their meanings.

keyheader3ds \*skeys

Array of spline path modifiers for the scale keys. Must have same element count as nskeys.

point3ds \*scale

Array of scale keys. Element count must be the same number as nskeys.

ulong3ds nmkeys

Count of morph keys in track. Default is 0.

short3ds nmflag

Track options for morph track. See npflag for their meanings.

keyheader3ds \*mkeys

Array of spline path modifiers for the morph keys. Must have same element count as nmkeys.

kfmorphkey3ds \*morph

Array of morph keys. Element count must be the same number as nmkeys.

ulong3ds nhkeys

Count of hide keys in track. Default is 0.

short3ds nhflag

Track options for hide track. See npflag for their meanings.

keyheader3ds \*hkeys

Array of hide keys. Spline modifiers are ignored. Element count must be the same number as nhkeys.

float3ds msangle

Angle for morph smoothing angle. Corresponds to the Angle field in the Keyframer/Object/Morph/Options/Morph Options dialogue. Defaults to 24.0

InitObjectMotion3ds

***Prototype:*** void InitObjectMotion3ds(kfmesh3ds \*\*obj, ulong3ds npkeys, ulong3ds nrkeys, ulong3ds nskeys, ulong3ds nmkeys, ulong3ds nhkeys);

Initializes kfmesh3ds structure and optionally allocates memory.

Arguments:

kfmesh3ds \*\*obj

Address of a pointer to a kfmesh3ds structure. If pointer is NULL, then memory will be allocated for a new structure and the address stored in the pointer.

ulong3ds npkeys

Allocate memory for specified number of position keys. A value of 0 will leave the position key fields unchanged.

ulong3ds nrkeys

Allocate memory for specified number of rotation keys. A value of 0 will leave the rotation key fields unchanged.

ulong3ds nskeys

Allocate memory for specified number of scale keys. A value of 0 will leave the scale key fields unchanged.

ulong3ds nmkeys

Allocate memory for specified number of morph keys. A value of 0 will leave the morph key fields unchanged.

ulong3ds nhkeys

Allocate memory for specified number of hide keys. A value of 0 will leave the hide key fields unchanged.

Returns nothing.

ReleaseObjectMotion3ds

***Prototype:*** void ReleaseObjectMotion3ds(kfmesh3ds \*\*obj);

Frees memory allocated to a kfmesh3ds structure.

Arguments:

kfmesh3ds \*\*obj

The structured to be freed. After release, the pointer is set to NULL.

Returns nothing.

GetObjectNodeCount3ds

***Prototype:*** ulong3ds GetObjectNodeCount3ds(database3ds \*db);

Scans the database for mesh nodes and returns a count of their number.

Arguments:

database3ds \*db

The database to scan for mesh nodes.

Returns **ulong3ds** as the count of mesh nodes.

GetObjectNodeNameList3ds

***Prototype:*** void GetObjectNodeNameList3ds(database3ds \*db, namelist3ds \*\*list);

Scans the database for mesh nodes and returns a list of their names.

Arguments:

database3ds \*db

The database to scan for mesh nodes.

namelist3ds \*\*list

The list of mesh node names found. Pointer gets past to InitNameList3ds prior to use.

Returns nothing.

GetObjectMotionByIndex3ds

***Prototype:*** void GetObjectMotionByIndex3ds(database3ds \*db, short3ds index, kfmesh3ds \*\*obj)

Searches the database for the index(th) occurance of a mesh node, and copies that node into mesh motion structure. If the index is out of range, the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

ulong3ds index

The order of occurance of the node in the database. Valid range is 0 to one less than the count returned by GetObjectMotionCount3ds.

kfmesh3ds \*\*obj

The mesh node structure that receives the settings. The pointer is passed to InitObjectMotion3ds prior to use.

Returns nothing.

GetObjectMotionByName3ds

***Prototype:*** void GetObjectMotionByName3ds(database3ds \*db, char3ds \*name, kfmesh3ds \*\*obj);

Seaches the database for a mesh node of the given name and copies it into a mesh node structure. If a the name is not found, then the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

char3ds \*name

The 1 to 10 character name to search for. The search is case sensitive.

kfmesh3ds \*\*obj

The mesh node structure that receives the settings. The pointer is passed to InitObjectMotion3ds prior to use.

PutObjectMotion3ds

***Prototype:*** void PutObjectMotion3ds(database3ds \*db, kfmesh3ds \*obj);

Copies the mesh motion settings from the structure into the database. Object nodes with the same name are replaced with the new settings.

Arguments:

database3ds \*db

The database to copy the mesh node into.

kfmesh3ds \*obj

The mesh node being added to the database.

Returns nothing.

Omnilight Animation

Kfomni3ds

Omnilight Animation Tracks

typedef struct {

char3ds name[11];

short3ds parent;

      ushort3ds flags1;

ushort3ds flags2;

      ulong3ds npkeys;

ushort3ds npflag;

keyheader3ds \*pkeys;

point3ds \*pos;

      ulong3ds nckeys;

ushort3ds ncflag;

keyheader3ds \*ckeys;

fcolor3ds \*color;

} kfomni3ds;

char3ds name[11];

Name of omnilight to animate. S

short3ds parent

Parent index. Obsolete and must be replaced with something that works before the product is released.

ushort3ds flags

Special node flags, any combination of the following:

ushort3ds flags2

Special node flags, any combination of the following:

static const ushortds ShowPath3ds = 1;Display the object motion path.

ulong3ds npkeys

Count of position keys in track. Default is 0.

ushort3ds npflag

Track options for position track, can be one of the following:

static const ushort3ds TrackSingle3ds = 0;Track does not repeat or loop.

static const ushort3ds TrackLoop3ds = 3;Track loops.

static const ushort3ds TrackRepeat3ds = 3;Track repeats.

Default is TrackSingle3ds.

keyheader3ds \*pkeys

Array of spline path modifiers for the position keys. Must have same element count as npkeys.

point3ds \*pos

Array of position keys. Element count must be the same number as npkeys.

ulong3ds nckeys;

Count of color keys in track. Default is 0.

ushort3ds ncflag;

Track options for color track, can be one of the following. See npflag for their meanings.

keyheader3ds \*ckeys;

Array of spline path modifiers for the color keys. Must have same element count as nckeys.

fcolor3ds \*color;

Array of color keys. Element count must be the same number as nckeys.

InitOmniLightMotion3ds

***Prototype:*** void InitOmniLightMotion3ds(kfomni3ds \*\*light, ulong3ds npkeys, ulong3ds nckeys)

Initializes kfomni3ds structure and optionally allocates memory.

Arguments:

kfomni3ds \*\*obj

Address of a pointer to a kfomni3ds structure. If pointer is NULL, then memory will be allocated for a new structure and the address stored in the pointer.

ulong3ds npkeys

Allocate memory for specified number of position keys. A value of 0 will leave the position key fields unchanged.

ulong3ds nckeys

Allocate memory for specified number of color keys. A value of 0 will leave the color key fields unchanged.

Returns nothing.

ReleaseOmniLightMotion3ds

***Prototype:*** void ReleaseOmniLightMotion3ds(kfomni3ds \*\*light)

Frees memory allocated to a kfomni3ds structure.

Arguments:

kfomni3ds \*\*light

The structured to be freed. After release, the pointer is set to NULL.

Returns nothing.

GetOmniLightNodeCount3ds

***Prototype:*** ulong3ds GetOmniLightNodeCount3ds(database3ds \*db)

Scans the database for omni light nodes and returns a count of their number.

Arguments:

database3ds \*db

The database to scan for mesh nodes.

Returns **ulong3ds** as the count of mesh nodes.

GetOmniLightNodeNameList3ds

***Prototype:*** void GetOmniLightNodeNameList3ds(database3ds \*db, namelist3ds \*\*list)

Scans the database for omni light nodes and returns a list of their names.

Arguments:

database3ds \*db

The database to scan for omni light nodes.

namelist3ds \*\*list

The list of omni light node names found. Pointer gets past to InitNameList3ds prior to use.

Returns nothing.

GetOmniLightMotionByIndex3ds

***Prototype:*** void GetOmniLightMotionByIndex3ds(database3ds \*db, short3ds index, kfomni3ds \*\*light)

Searches the database for the index(th) occurance of a omni light node, and copies that node into omni light motion structure. If the index is out of range, the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

ulong3ds index

The order of occurance of the node in the database. Valid range is 0 to one less than the count returned by GetOmniLightMotionCount3ds.

kfomni3ds \*\*light

The omni light node structure that receives the settings. The pointer is passed to InitOmniLightMotion3ds prior to use.

Returns nothing.

GetOmniLightMotionByName3ds

***Prototype:*** void GetOmniLightMotionByName3ds(database3ds \*db, char3ds \*name, kfomni3ds \*\*light)

Seaches the database for a omni light node of the given name and copies it into a omni light node structure. If a the name is not found, then the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

char3ds \*name

The 1 to 10 character name to search for. The search is case sensitive.

kfomni3ds \*\*light

The omni light node structure that receives the settings. The pointer is passed to InitOmniLightMotion3ds prior to use.

PutOmniLightMotion3ds

***Prototype:*** void PutOmniLightMotion3ds(database3ds \*db, kfomni3ds \*light)

Copies the omni light motion settings from the structure into the database. Object nodes with the same name are replaced with the new settings.

Arguments:

database3ds \*db

The database to copy the omni light node into.

kfomni3ds \*light

The omni light node being added to the database.

Returns nothing.

Spotlight Animation

Kfspot3ds

Spotlight Animation Settings

typedef struct {

char3ds name[11];

short3ds parent;

      ushort3ds flags1;

ushort3ds flags2;

      ulong3ds npkeys;

ushort3ds npflag;

keyheader3ds \*pkeys;

point3ds \*pos;

      ulong3ds nckeys;

ushort3ds ncflag;

keyheader3ds \*ckeys;

fcolor3ds \*color;

      ulong3ds nhkeys;

ushort3ds nhflag;

keyheader3ds \*hkeys;

float3ds \*hot;

      ulong3ds nfkeys;

ushort3ds nfflag;

keyheader3ds \*fkeys;

float3ds \*fall;

      ulong3ds nrkeys;

ushort3ds nrflag;

keyheader3ds \*rkeys;

float3ds \*roll;

      short3ds tparent;

ulong3ds ntkeys;

ushort3ds ntflag;

keyheader3ds \*tkeys;

point3ds \*tpos;

ushort3ds tflags1;

ushort3ds tflags2;

} kfspot3ds;

char3ds name[11];

Name of spot light to animate.

short3ds parent

Parent index. Obsolete and must be replaced with something that works before the product is released.

ushort3ds flags

Special node flags, any combination of the following:

ushort3ds flags2

Special node flags, any combination of the following:

static const ushortds ShowPath3ds = 1;Display the object motion path.

ulong3ds npkeys

Count of position keys in track. Default is 0.

ushort3ds npflag

Track options for position track, can be one of the following:

static const ushort3ds TrackSingle3ds = 0;Track does not repeat or loop.

static const ushort3ds TrackLoop3ds = 3;Track loops.

static const ushort3ds TrackRepeat3ds = 3;Track repeats.

Default is TrackSingle3ds.

keyheader3ds \*pkeys

Array of spline path modifiers for the position keys. Must have same element count as npkeys.

point3ds \*pos

Array of position keys. Element count must be the same number as npkeys.

ulong3ds nckeys;

Count of color keys in track. Default is 0.

ushort3ds ncflag;

Track options for color track, can be one of the following. See npflag for their meanings.

keyheader3ds \*ckeys;

Array of spline path modifiers for the color keys. Must have same element count as nckeys.

fcolor3ds \*color;

Array of color keys. Element count must be the same number as nckeys.

ulong3ds nhkeys;

Count of hot spot keys in track. Default is 0.

ushort3ds nhflag;

Track options for hot spot track, can be one of the following. See npflag for their meanings.

keyheader3ds \*hkeys;

Array of spline path modifiers for the hot spot keys. Must have same element count as nhkeys.

float3ds \*hot;

Array of hot spot keys. Element count must be the same number as nhkeys.

ulong3ds nfkeys;

Count of falloff keys in track. Default is 0.

ushort3ds ncflag;

Track options for falloff track, can be one of the following. See npflag for their meanings.

keyheader3ds \*fkeys;

Array of spline path modifiers for the falloff keys. Must have same element count as nfkeys.

fcolor3ds \*fall;

Array of falloff keys. Element count must be the same number as nfkeys.

ulong3ds nrkeys

Number of roll keys in the track. Default is 0.

ushort3ds nrflag

Track options for roll track. Se npflag for their meanings.

keyheader3ds \*rkeys

Array of spline modifiers for spot roll. Element count must be the same number as nrkeys.

float3ds \*roll

Array of spot roll keys. Element count must be the same number as nrkeys.

short3ds tparent

Parent index for spot light target.

ulong3ds ntkeys

Number of track position keys. Default is 0.

ushort3ds ntflag

Track options for target keys. See npflag for thier meanings.

keyheader3ds \*tkeys

Array of spline path modifiers for target keys. Element count must be the same number as ntkeys.

point3ds \*tpos

Array of target position keys. Element count must be the same number as ntkeys.

ushort3ds tflags1

Special node flags.

ushort3ds tflags2

Special node flags.

InitSpotLightMotion3ds

***Prototype:*** void InitSpotLightMotion3ds(kfspot3ds \*\*spot, ulong3ds npkeys, ulong3ds nckeys, ulong3ds nhkeys, ulong3ds nfkeys, ulong3ds nrkeys, ulong3ds ntkeys);

Initializes kfspot3ds structure and optionally allocates memory.

Arguments:

kfspot3ds \*\*spot

Address of a pointer to a kfspot3ds structure. If pointer is NULL, then memory will be allocated for a new structure and the address stored in the pointer.

ulong3ds npkeys

Allocate memory for specified number of position keys. A value of 0 will leave the position key fields unchanged.

ulong3ds nckeys

Allocate memory for specified number of color keys. A value of 0 will leave the color key fields unchanged.

ulong3ds nhkeys

Allocate memory for specified number of hot spot keys. A value of 0 will leave the hot spot key fields unchanged.

ulong3ds nfkeys

Allocate memory for specified number of falloff keys. A value of 0 will leave the falloff key fields unchanged.

ulong3ds nrkeys

Allocate memory for specified number of roll keys. A value of 0 will leave the roll key fields unchanged.

ulong3ds ntkeys

Allocate memory for specified number of target position keys. A value of 0 will leave the target position key fields unchanged.

Returns nothing.

ReleaseSpotLightMotion3ds

***Prototype:*** void ReleaseSpotLightMotion3ds(kfspot3ds \*\*spot);

Frees memory allocated to a kfspot3ds structure.

Arguments:

kfspot3ds \*\*spot

The structured to be freed. After release, the pointer is set to NULL.

Returns nothing.

GetSpotLightNodeCount3ds

***Prototype:*** long3ds GetSpotLightNodeCount3ds(database3ds \*db);

Scans the database for spotlight nodes and returns a count of their number.

Arguments:

database3ds \*db

The database to scan for spotlight nodes.

Returns **ulong3ds** as the count of mesh nodes.

GetSpotLightNodeNameList3ds

***Prototype:*** void GetSpotLightNodeNameList3ds(database3ds \*db, chunklist3ds \*\*list);

Scans the database for spotlight nodes and returns a list of their names.

Arguments:

database3ds \*db

The database to scan for spotlight nodes.

namelist3ds \*\*list

The list of spotlight node names found. Pointer gets past to InitNameList3ds prior to use.

Returns nothing.

GetSpotLightMotionByIndex3ds

***Prototype:*** void GetSpotLightMotionByIndex3ds(database3ds \*db, ulong3ds index, kfspot3ds \*spot);

Searches the database for the index(th) occurance of a spotlight node, and copies that node into spotlight motion structure. If the index is out of range, the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

ulong3ds index

The order of occurance of the node in the database. Valid range is 0 to one less than the count returned by GetSpotLightMotionCount3ds.

kfspot3ds \*\*spot

The spotlight node structure that receives the settings. The pointer is passed to InitSpotLightMotion3ds prior to use.

Returns nothing.

GetSpotLightMotionByName3ds

***Prototype:*** void GetSpotLightMotionByName3ds(database3ds \*db, char3ds \*name, kfspot3ds \*\*spot);

Seaches the database for a spotlight node of the given name and copies it into a spotlight node structure. If a the name is not found, then the structure remains unchanged.

Arguments:

database3ds \*db

The database being searched.

char3ds \*name

The 1 to 10 character name to search for. The search is case sensitive.

kfspot3ds \*\*spot

The spotlight node structure that receives the settings. The pointer is passed to InitSpotLightMotion3ds prior to use.

PutSpotLightMotion3ds

***Prototype:*** void PutSpotLightMotion3ds(database3ds \*db, kfspot3ds \*spot);

Copies the spotlight motion settings from the structure into the database. Object nodes with the same name are replaced with the new settings.

Arguments:

database3ds \*db

The database to copy the spotlight node into.

kfspot3ds \*spot

The spotlight node being added to the database.