Sound system

Description

Sound system based on FMOD library and uses FMOD event system to control behavior and properties of all sounds that can be played on game levels. All sounds and music are defined using FMOD Event Designer (external program that ships with FMOD library) and packaged into several sound bank files (*.fsb). All sound access is performed using predefined integer ID's and this ID list is exported by FMOD Event Designer during sound bank creation.

FMOD Event Designer can merge several sounds into single event, add various effects to it (chorus, reverberation, loop, etc.) and all this effects will be available during game play, so engine doesn't need to provide this functionality.

Associated classes and structures

CSoundSystem	Manager of all sound events. Wraps access to FMOD Event System.

class CSoundSystem

All sounds operations go through this class. CSoundSystem hides internal access to FMOD library.

Sources: SoundSys.h, SoundSys.cpp

Important methods

int Valid()

Return nonzero value if sound system initialized properly.

int Open()

Initialize sound system. Configure and setup FMOD Eent System, listener and general sound properties. Return nonzero value if initialization was successful.

int Close()

Deinitialize sound system. Return nonzero value if deinitialization was successful.

int LoadSoundEffects(const char *basedir, const char *fname)

Load sound bank from input path. Return nonzero value if loading was successful.

Parameters:
basedir – path to directory where required sounds bank reside.
fname – file name of required sound bank.

void* Play(int SampleID, const r3dPoint3D& Pos)

Play sound defined by sample ID. If required sound is 3D sound, than use position to place it into world. Return handle to played sound or NULL if input ID is invalid.

Parameters:
SampleID – sound event id. Use GetEventIDByPath function to obtain valid id.
Pos – world position of sound position. Ignored if sound is 2D.

void * Play2D(int SampleID)

Play 2D sound. Return handle to sound, or NULL if input id is invalid.

Parameters:
SampleID - sound event id. Use GetEventIDByPath function to obtain valid id.

bool SetSoundPos(void *handle, const r3dPoint3D& pos)

Set sound world position. If handle is 3D sound, do nothing. Return true if operation was successful.

Parameters:
handle – handle of sound to play. Valid handle can be obtained using Play, Play2D, Play3D functions.
pos – new position of sound in world space.

void * Play3D(int SampleID, const r3dPoint3D& Pos)

Play 3D sound. Return handle to sound, or NULL if input id is invalid.

Parameters:
SampleID – sound event id. Use GetEventIDByPath function to obtain valid id.

void SetPaused(void* handle, bool paused)

Pause or unpause sound.

Parameters:
handle - handle of sound to change paused state. Valid handle can be obtained using Play, Play2D, Play3D functions.
paused – sound paused state.

void Stop(void *handle)

Stop sound event playing.

Parameters:
handle - handle of sound to stop. Valid handle can be obtained using Play, Play2D, Play3D functions.

void Start(void *handle)

Start playing sound event from the beginning.

Parameters:
handle - handle of sound to start. Valid handle can be obtained using Play, Play2D, Play3D functions.

int GetState(void *handle) const

Get FMOD_EVENT_STATE of sound. For FMOD_EVENT_STATE description please consult FMOD documentation.

Parameters:
handle - handle of sound to get state of. Valid handle can be obtained using Play, Play2D, Play3D functions.

void Update(const r3dPoint3D &pos, const r3dPoint3D &dir, const r3dPoint3D& up)

Update sound system internal state. This is includes updating listener position and advancing states of all sound events.

Parameters:
pos – listener world position.
dir – listener forward direction.
up – listener up vector.

void Release(void *handle)

Release sound event and all associated data with it. To play sound again, call Play, Play2D or Play3D functions.

Parameters:
handle - handle of sound to release. Valid handle can be obtained using Play, Play2D, Play3D functions.

float GetSoundMaxDistance(void *handle) const

Return maximum sound audible distance in world space units.

Parameters:
handle - handle of sound to get max distance of. Valid handle can be obtained using Play, Play2D, Play3D functions.

float GetSoundMaxDistance(int id) const

Return maximum sound audible distance in world space units.

Parameters:
handle - handle of sound to get max distance of. Valid handle can be obtained using Play, Play2D, Play3D functions.

int GetEventIDByPath(const char *path) const

Return sound event ID. Path must be constructed using following pattern: “<sound bank name>/<path inside corresponding bank>”

Parameters:
path – path to sound event.

void Set3DAttributes(void *handle, const r3dVector *pos, const r3dVector *velocity, const r3dVector *orientataion) const

Set position, orientation and velocity of 3D sound event.

Parameters:
handle - handle of sound to get max distance of. Valid handle can be obtained using Play, Play2D, Play3D functions.
pos – position of sound in world space.
velocity – velocity of sound event.
orientation – orientation of sound event.

bool IsHandleValid(void *handle) const

Return true if input pointer is valid sound event handle.

Parameters:
handle – pointer to check.

bool Is3DSound(void *handle) const

Return true if input handle represents 3D sound event.

Parameters:
handle - handle of sound to get 3D property of. Valid handle can be obtained using Play, Play2D, Play3D functions.

bool IsAudible(void *handle, const r3dPoint3D &pos) const

Return true if input sound event will be audible at given world position.

Parameters:
handle - handle of sound check. Valid handle can be obtained using Play, Play2D, Play3D functions.
pos – world position to check.

const stringlist_t & GetSoundsList() const

Return list with paths to all loaded sounds.

void SetProperty(void *handle, FMOD_EVENT_PROPERTY propID, void *value)

Set FMOD property to specific sound event.

Parameters:
handle - handle of sound to set property to. Valid handle can be obtained using Play, Play2D, Play3D functions.
propID – property ID. Consuld FMOD SDK documentation for possible values.
va*l*ue – value of property.