Subject: Re: [REQUEST]Sound on turret rotate
Posted by danpaul88 on Fri, 28 Dec 2012 20:30:45 GMT
View Forum Message <> Reply to Message

OK, so I figured the existing script had some fundamental flaws in it's implementation so I wrote a new, much simpler to use, version.

http://doc.tiberiantechnologies.org/scripts.dll/classdp88__turret_sound.html

If you want to edit scripts.dll to start using it now, before the next scripts release, here's the code. Please don't ship any .dll files to anyone with this in, since we don't want people running non-versioned (and therefore, non-debuggable) client side code, but you can use it for testing or in a server side mod.

Toggle Spoiler

dp88_custom_timer_defines.h:

```
#define DP88_TIMER                      0xDB000000
#define MISC                            0x00000000
#define TIMER_TURRETSOUND               (DP88_TIMER|MISC|0x04)          //!< Used by dp88_turretSound to test for turret rotation
```

dp88_misc.h:

```
/*!
* \brief Turret Rotation Sound Effect
* \author Daniel Paul (danpaul88@yahoo.co.uk)
* \ingroup scripts_sound
*
* This script plays a 3d sound at a vehicles turret bone when that bone is being rotated and stops
* the sound when the rotation stops. The sound will be looped whilst the turret is being rotated.
*
* \note
*   This script uses the difference between the vehicle rotation and the turret bone rotation to
*   determine if the turret is rotating. This means simply aiming in one direction and spinning on
*   the spot will cause the sound to be played, since the turret is rotating relative to the vehicle
*
* \param Sound_Preset
*   The name of a 3D sound preset to be played whilst the turret is rotating
* \param Min_Differential_Rad
*   The minimum difference in the turret rotation, in radians, to be considered as "rotating", this
*   helps to filter out tiny movements caused by driving along uneven terrain.
*/
class dp88_turretSound : public ScriptImpClass
{
```

```
protected:
  void Created ( GameObject* pObj );
  void Timer_Expired ( GameObject* pObj, int number );
  void Custom ( GameObject* pObj, int type, int param, GameObject* pSender );

  float Get_Turret_Facing ( class RenderObjClass* pRenderObj );
  void Play_Sound ( GameObject* pObj );
  void Stop_Sound ( GameObject* pObj );

  float m_lastFacing;
  int m_nSoundId;

  /*! \name Cached Script Parameters */
  /*! @{ */
  float m_nMinDifferential;
  /*! @} */
};
```

dp88_misc.cpp:

```
#include "VehicleGameObj.h"
#include "RenderObjClass.h"

void dp88_turretSound::Created ( GameObject* pObj )
{
  if ( VehicleGameObj* vObj = pObj->As_VehicleGameObj() )
  {
    m_nMinDifferential = Get_Float_Parameter("Min_Differential_Rad");
    m_lastFacing = Get_Turret_Facing(vObj->Peek_Model());
    m_nSoundId = -1;
    Commands->Start_Timer(pObj, this, 0.5f, TIMER_TURRETSOUND );
  }
  else
  {
    Console_Output ( "[%d:%s:%s] Critical Error: This script is only compatible with vehicle game
objects. Destroying script...\n", Commands->Get_ID(pObj),
Commands->Get_Preset_Name(pObj), this->Get_Name() );
    Destroy_Script();
  }
}

// ----------------------------------------------------------------------------------------------

void dp88_turretSound::Timer_Expired ( GameObject* pObj, int number )
{
  if ( number == TIMER_TURRETSOUND )
  {
```

```
    if ( VehicleGameObj* vObj = pObj->As_VehicleGameObj() )
    {
      float newFacing = Get_Turret_Facing(vObj->Peek_Model());

      // Check if we are rotating - ignore tiny rotation amounts
      bool bRotating = ( abs(newFacing-m_lastFacing) > m_nMinDifferential );

      if ( m_nSoundId == -1 && bRotating)
        Play_Sound(pObj);

      else if ( m_nSoundId != -1 && !bRotating )
        Stop_Sound(pObj);

      m_lastFacing = newFacing;
    }

    // Restart timer - runs even whilst playing sound so we can loop an uninterrupted sound
    // if the turret is still rotating when the sound completes
    Commands->Start_Timer(pObj, this, 0.5f, TIMER_TURRETSOUND );
  }
}

// ----------------------------------------------------------------------------------------------

void dp88_turretSound::Custom ( GameObject* pObj, int type, int param, GameObject* pSender )
{
  if (type == CUSTOM_EVENT_SOUND_ENDED && param == m_nSoundId)
  {
    // We will allow the timer to stop the sound if necessary, since this might trigger
    // on the same engine tick, thus checking our facing against the previous timer
    // facing could produce a false-positive for "stopped rotating"
    Play_Sound(pObj);
  }
}

// ----------------------------------------------------------------------------------------------

float dp88_turretSound::Get_Turret_Facing ( RenderObjClass* pRenderObj )
{
  if ( pRenderObj )
  {
    Matrix3D vehicleTransform = pRenderObj->Get_Transform();
    Matrix3D transform = pRenderObj->Get_Bone_Transform("turret");
    float offset = abs(vehicleTransform.getRotationZ()-transform.getRotationZ());
    return offset;
  }

  return 0.0f;
```

```
}

// ----------------------------------------------------------------------------------------------

void dp88_turretSound::Play_Sound ( GameObject* pObj )
{
  m_nSoundId = Commands->Create_3D_Sound_At_Bone(Get_Parameter("Sound_Preset"),
pObj, "turret");
  Commands->Monitor_Sound(pObj, m_nSoundId);
}

// ----------------------------------------------------------------------------------------------

void dp88_turretSound::Stop_Sound ( GameObject* pObj )
{
  Commands->Stop_Sound(m_nSoundId,true);
  m_nSoundId = -1;
}

// ----------------------------------------------------------------------------------------------

ScriptRegistrant<dp88_turretSound> dp88_turretSound_Registrant(
  "dp88_turretSound",
  "Sound_Preset:string,"
  "Min_Differential_Rad=0.25:float"
);
```