

There's
beer!

```

// ENEMY THINKING
// Enemies are always spawned
// with targetplayer = 1, threshold = 0
// Most monsters are spawned unaware of all players,
// but some can be made pre-aware
//
// P_SoundTraverse
// called by P_SoundBlocks
// Recursively traverses adjacent sectors,
// sound blocks, and lines for sound traversal
//
mobj_t* soundtarget;
int validcount;
int soundblocks;
int sec;
int check;
int other;
int i;
int linecount;
int openrange;
int sidenum[2];
int sidsel;
int subsector;
int sector;
int flags;
int lemmiter;
int target;
int emmitter;

// P_SoundTraverse
// If a monster yells at a player,
// it will alert other monsters to the player.
//
void P_NoiseAlert(mobj_t* target,
                  mobj_t* emmitter)
{
    soundtarget = target;
    validcount++;
    P_RecursiveSound(lemmiter > subsector > sector, 0);
}

// P_CheckMolePath
// Checks if a path exists between two sectors
// through adjacent sectors, sound blocks, and lines
int P_CheckMolePath(mobj_t* target,
                     mobj_t* check)
{
    if (check == target)
        return 1;
    else
        return 0;
}

// P_SoundBlocks
// Checks if a sound block is valid
// based on its valid count and soundtraversed
int P_SoundBlocks(mobj_t* sec)
{
    if (sec->validcount == validcount
        && sec->soundtraversed <= soundblocks+1)
        return 1;
    else
        return 0;
}

// wake up all monsters in this sector
if (sec->validcount == validcount
    && sec->soundtraversed <= soundblocks+1)
{
    return; // already flooded
}

sec->validcount = validcount;
sec->soundtraversed = soundblocks+1;
sec->soundtarget = soundtarget;

for (i=0 ; i<sec->linecount ; i++)
{
    check = sec->lines[i];
    if (! (check->flags & ML_TWOSIDED))
        continue;

    P_LineOpening (check);

    if (openrange <= 0)
        continue; // closed door

    if ( ! (sidesel = check->sidenum[0]) || sector == sec)
        other = sidesel = check->sidenum[1] > sector;
    else
        other = sidesel = check->sidenum[0] > sector;

    if (check->flags & ML_SOUNDBLOCK)
    {
        if (!soundblocks)
            P_RecursiveSound (other, 1);
    }
    else
        P_RecursiveSound (other, soundblocks);
}

r_names[i];
usage;
}

// P_NoiseAlert
// If a monster yells at a player,
// it will alert other monsters to the player.
//
void P_NoiseAlert(mobj_t* target,
                  mobj_t* emmitter)
{
    soundtarget = target;
    validcount++;
    P_RecursiveSound(lemmiter > subsector > sector, 0);
}

```

RailsBridge