Music Jukebox

Robb Sherwin's *Cryptozookeeper* has the largest Hugo soundtrack to date. After each song ended, one would be picked at random to start at the next turn. I added a similar "jukebox" system to Roodylib.

```
#set USE_JUKEBOX
```

turning on the jukebox system

First, make `song` objects of all of the songs for your game.

```
song zombiecrap "Zombie Crap"
{
      artist "Ben Parrish"  ! optional property
      file "zombie"
      length 4 14
      in jukebox
}
```

song example

Notice the length property. The first field is the song length in minutes, and the second one is the seconds.

Assuming you're going to want all of your songs bundled in one resource file, you'll also have to define a `MUSIC_RESOURCE_FILE` constant.

```
constant MUSIC_RESOURCE_FILE "gamemus"

resource "gamemus"
{
      "zombie.mp3"
}
```

defining the MUSIC_RESOURCE_FILE constant

The way the jukebox system works is that it always plays the eldest child of the jukebox. Once played, it moves it back to the jukebox so it's the youngest child, continually looping. If you'd like the songs to be shuffled each time someone plays the game, add this code:

```
object music_shuffler
{
      in init_instructions
      execute
      {
            if not CheckWordSetting("undo")
                  MixObjects(jukebox)
      }
}
```

music shuffling code

Now, all that's left is turning on the jukebox! You can control it with these routines:

`PlayJukebox` – Turns the jukebox on and plays songs continuously.

`StopJukebox` – Stops the jukebox.

`NowPlaying` – Says either “There is no song currently playing.” or “<song name> by <artist> is currently playing.”

`PlaySong( songfile , loop)` – This will play a song class object (as if it were in the jukebox) and update audio.current_music when it is over, in case there’s any time you want to know when a song is over but aren’t actually using the jukebox.