Playing Sounds inside XNA

Game Design Experience
Professor Jim Whitehead
February 23, 2009
Announcements

• Quadtree homework assignment
  ▶ Due today
  ▶ Hand in to box by my office (E2 273), or in class

• Partially operational game prototype
  ▶ Due Friday, February 27
  ▶ Need to demonstrate:
    • An XNA project (25%)
    • A few objects, including a player object (25%)
    • The ability to move the player object, even if in a limited way (25%)
    • Some ability for the player object to interact with their world (firing, jumping, picking up objects, etc.) (25%)
  ▶ Submit on CDROM, USB Drive, or URL to Subversion project
Sound in games

• Think about truly memorable games
  ► They almost always have excellent background music and sound effects
  ► Legend of Zelda, PacMan, Katamari Damacy, Little Big Planet, Radiant Silvergun
  ► Music and artwork style combine to create an overall tone, or mood, for a game
  ► Done well, this substantially enhances the overall gameplay experience
Finding/Making Sounds

• Where can you find music to use in your game?
  ► Reminder: there is this legal framework called Copyright Law
  ► Creative Commons: use licenses that may allow free, non-commercial use
    • http://creativecommons.org/

• Sites with Creative Commons licensed music
  ► New Grounds, Jamendo
    • http://www.newgrounds.com/audio/
    • http://www.jamendo.com/en/creativecommons/
    • Look for “Attribution, Non-commercial”
      – “No Derivative Works” is OK, so long as you don’t modify
  • If you use in your game, make sure you provide attribution
    – Put name of artist in your game (About page, splash screen, etc.)
    – Is polite to send them an email telling them about the use—will make them jazzed
• Find someone to create music for you
  ► Music student at UCSC, for example
• It has never been cheaper to create high quality music
  ► Instruments, microphones, mixing technology are all at historically low prices
  ► Has led to a proliferation of music
  ► Biggest problem: finding an audience
  ► Games provide a good audience
  ► Sales of many videogames larger than most music album sales
  ► For many musicians, might have larger audience for video game soundtrack than for traditional album
Finding/Making Sounds (cont’d)

• Use your voice!
  ▶ Your voice is wonderfully adaptable and expressive

• Consider:
  ▶ Record a raw voice clip
  ▶ Bring into an editing software suite
  ▶ Tweak/filter/alter until it suits your game
  ▶ Can do much worse…

• Tools
  ▶ Audacity
    • http://audacity.sourceforge.net/
    • Free, open source sound recorder/editor
  ▶ FL Studio (grown-up commercial version of Fruity Loops)
    • http://flstudio.image-line.com/documents/what.html
Playing Sounds in XNA

• Two ways
  • Hard (but powerful) way
    ► XACT audio tool
      • Cross-platform audio creation tool
    ► Many neat features
    ► Edit volume, pitch, looping of sound clips
    ► Can easily group together sound clips
  • Easy (and 95% sufficient) way
    ► Use Simplified Sound API
    ► Can start, stop, and pause sound playing
    ► Much, much easier to use
Simple Sound API

- Two ways to play music
  - As a song
    - Good for background music, or other long sounds
  - As a sound effect
    - Good for short duration sounds
XNA Simple Sound API

- Supported music types: wav, wma, mp3
- Add sound into project Contents folder
  - Audio files treated like other files in content pipeline
  - Copy sound file into project Contents folder
  - Right-click on Contents folder inside Visual Studio C# Express
    - Add → Existing Item … select audio file you just copied in
  - Will now be visible inside Visual Studio
- Need to double-check the Content Processor
  - Sound Effect – XNA Framework – sound effects
  - Song – XNA Framework - songs
XNA Song API

• Create a variable of type Song
  ▶ Used to load songs via the content pipeline
  ▶ Song mySong;

• Load sound file
  ▶ mySong = Content.Load<Song>("{name of song file without extension}")

• To play a sound, call Play() method on MediaPlayer object
  ▶ MediaPlayer.Play(mySong);

• To pause/resume, call Pause() / Resume() on MediaPlayer object
  ▶ MediaPlayer.Pause(); // no argument
  ▶ MediaPlayer.Resume(); // no argument
XNA Sound Effect API

- Create a variable of type SoundEffect
  - Used to load sounds via the content pipeline
  - `SoundEffect soundEffect;`

- Load sound file
  - `soundEffect = Content.Load<SoundEffect>("{name of sound file without extension}");`

- To play a sound, call `Play()` method on `SoundEffect` object
  - Returns a `SoundEffectInstance` object
  - Can use this to stop, pause, and restart sound
  - `SoundEffectInstance soundEffectInstance = soundEffect.Play();`
Demo of Song and Sound Effect API

// Demo of use of Songs and Sound Effects inside XNA

• Caution: Treating a song as a sound effect can lead to very long compile times
  ▶ Solution: keep sound effects short