Sound

Haeyong Chung
Fall 2017

Slides contain examples from the official Corona docs as well as the textbook.

Sound

- We will use some sound effects.
- No need for require() function.
- Use http://www.bfxr.net/ for creating and download game sound effects

Audio Handles

- audio.loadSound() and audio.loadStream()
  - returns ***audioHandle*** to be used in playing the sound
  - audio.loadSound(): loadSound loads in sound files
  - audio.loadStream(): loadStream loads them as a stream:
    - used for long files to save memory
    - loads blocks at a time
    - cannot be shared across channels
- Examples:
  - local soundEffect = audio.loadSound("chime.wav")
  - audio.play(soundEffect)

Playing Sounds

- audio.play(audioHandle [, options])
  - returns selected channel
  - Options table can include:
    - channel : select 1 to 32. default=0=auto
    - loops: repeat x times.
      - 0=no repeition. 1=repeat 1 more, for total 2 times.
      - 2=repeat 2 more, for total 3 times.
    - duration: duration including repeats. (ms)
    - fadein: (ms)
    - onComplete: callback function pointer
      - It will pass back an event table to the callback function:
        - event.name
        - event.channel
        - event.handle
        - event.completed
- audio.pause(<channel # >)
  - Empty input parameter→ pause ALL channels

Playing Sounds (continued)

- Handles

```
local soundTable = {
    shootSound = audio.loadSound( "shoot.wav" ),
    hitSound = audio.loadSound( "hit.wav" ),
    explodeSound = audio.loadSound( "explode.wav" ),
}
```

- When shooting

```
audio.play( soundTable["shootSound"] );
```

- When target is hit

```
audio.play( soundTable["hitSound"] );
```

- When enemy explodes

```
audio.play( soundTable["explodeSound"] );
```

Audio Format

- Supports different audio formats:
  - .wav
  - .mp3
  - .mp4
  - .acc,
  - .ogg

Channel & Volume

- Audio Channels:
  - Supports 32 channels.
  - You can reserve certain channels using:
    - audio.reserveChannels(#channels)
    - audio.isChannelActive(), audio.isChannelPlaying(), and audio.isChannelPaused()
- Control Volume
  - a decimal representation of 0%-100% to the audio.setVolume() API
  - audio.setVolume(0.5)
  - audio.setVolume(0.5, { channel=1 })

Control Audio

- audio.pause
- audio.resume
- audio.rewind
- audio.seek()
- audio.stop()
- audio.stopWithDelay()
- audio.dispose(audioHandle)

Sound

- We will use some sound effects.
- No need for require() function.
- Use http://www.bfxr.net/ for creating and download game sound effects

Audio Handles

- audio.loadSound() and audio.loadStream()
  - returns ***audioHandle*** to be used in playing the sound
  - audio.loadSound(): loadSound loads in sound files
  - audio.loadStream(): loadStream loads them as a stream:
    - used for long files to save memory
    - loads blocks at a time
    - cannot be shared across channels
- Examples:
  - local soundEffect = audio.loadSound("chime.wav")
  - audio.play(soundEffect)

Playing Sounds

- audio.play(audioHandle [, options])
  - returns selected channel
  - Options table can include:
    - channel : select 1 to 32. default=0=auto
    - loops: repeat x times.
      - 0=no repeition. 1=repeat 1 more, for total 2 times.
      - 2=repeat 2 more, for total 3 times.
    - duration: duration including repeats. (ms)
    - fadein: (ms)
    - onComplete: callback function pointer
      - It will pass back an event table to the callback function:
        - event.name
        - event.channel
        - event.handle
        - event.completed
- audio.pause(<channel # >)
  - Empty input parameter→ pause ALL channels

Playing Sounds (continued)

- Handles

```
local soundTable = {
    shootSound = audio.loadSound( "shoot.wav" ),
    hitSound = audio.loadSound( "hit.wav" ),
    explodeSound = audio.loadSound( "explode.wav" ),
}
```

- When shooting

```
audio.play( soundTable["shootSound"] );
```

- When target is hit

```
audio.play( soundTable["hitSound"] );
```

- When enemy explodes

```
audio.play( soundTable["explodeSound"] );
```

Audio Format

- Supports different audio formats:
  - .wav
  - .mp3
  - .mp4
  - .acc,
  - .ogg

Channel & Volume

- Audio Channels:
  - Supports 32 channels.
  - You can reserve certain channels using:
    - audio.reserveChannels(#channels)
    - audio.isChannelActive(), audio.isChannelPlaying(), and audio.isChannelPaused()
- Control Volume
  - a decimal representation of 0%-100% to the audio.setVolume() API
  - audio.setVolume(0.5)
  - audio.setVolume(0.5, { channel=1 })

Control Audio

- audio.pause
- audio.resume
- audio.rewind
- audio.seek()
- audio.stop()
- audio.stopWithDelay()
- audio.dispose(audioHandle)

NEXT Agenda

- OOP
  - Inheritance, overriding, etc. in LUA