Emerging Music Nexuses: Technology, Video Games, and Coding

Jared O’Leary
Arizona State University
Avondale Elementary School District
What’s the plan?

- Three emerging music nexuses:
  - Music performance and Technology
  - Music and Video Games
  - Music && Coding
- Let’s Talk
How to reach the resources

- www.JaredOLeary.com
- Presentations
- Emerging Music Nexuses
Music performance and technology
Digital and hybrid musicianship

- youtube.com/OCPDMusic
- Playlists
  - Digital and Hybrid Musicianship

https://goo.gl/k5edaL
Augmenting an elementary ensemble
Augmenting an elementary ensemble

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)
Augmenting a community ensemble
Augmenting a community ensemble

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)
Some other presentations on this topic

- 21st century elementary ensemble director
- From large ensembles to video games: Technology in music education
- Multimedia ensemble: Performing live music with live video games
- Using technology to augment musicianship
- Using technology to augment teaching
Music and video games
Video game music and interactive audio

- Video Game Music
  - https://goo.gl/FSNAhB

- Interactive Audio
  - https://goo.gl/O5SOox

Presentation by Jared O’Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)
Percussion
Bass
Lead
Some other presentations on this topic

- 8-bit composition unit? Composing for old school video games
- Creating 8-bit
- Engaging with popular and participatory cultures: Implications for teaching and learning
- From large ensembles to video games: Technology in music education
- Interacting with 8-bit
- Modern video game projects
- Multimedia ensemble: Performing live music with live video games
- Old school video game projects
- Sonic modding: Modding video game music and sound
- Video games in music education
Upcoming publications on this topic


Music && coding
Music & coding

youtube.com/OCPDMusic

Playlists

Music & Coding

https://goo.gl/lbMvyt
live_loop :floor do
  bd_arr = [(ring 10, 5, 10), (ring 6, 10, 6)]
  4.times do
    bd_unquant = rrand(0, master_unquant)
    sleep bd_unquant / 2
    sample :drum_bass_hard, amp: rrand(0.75, 1.1), rate: rrand(0.999, 1.001) unless one_in(10) || count > 3
    sample :drum_bass_hard, amp: rrand(0.75, 1.1), rate: rrand(0.999, 1.001) if one_in(3) && count == 4
    sleep s = (bd_unquant / 2)
  end
end

live_loop :hat do
  hh_amp = (ring 0.4, 0)
  hh_arr = [(ring 20, 10), (ring 10, 12, 20, 12), (ring 5, 12), (ring 1, 8)]
  hh_t = (ring 2, 4)
  if one_in(2)
    sample :drum_blast_hard, amp: rrand(0.2, 0.7), rate: rrand(0.999, 1.001) if one_in(2)
  else
    sample :drum_blast_soft, amp: rrand(0.2, 0.7), rate: rrand(0.999, 1.001) if one_in(2)
  end
  32.times do
    if one_in(2) && count > 1
      2.times do
        hh_unquant = rrand(0, master_unquant)
        sleep hh_unquant / 2
        sample :drum_cymbal_closed, amp: (rrand(0.15, 0.4) + hh_amp.tick), rate: rrand(0.999, 1.001) unless one_in(hh_arr[count - 1].ti
        sleep t = (hh_unquant / 2)
This code will set the backdrop to a blank backdrop at the start of the program.

When the spacebar is pressed, this code will repeat the following sequence four times: Switch the costume and play a sound when it switches, then pause a small amount before moving to the next sound and costume in the sequence. Notice to make the last two sounds twice as fast. I had to make the wait twice as small. What other beats, sounds, or costumes, could you use to make this code?
// MARK: - Actions
leftJoiner.playOrPause(_ sender: AnyObject) {
    if player.isPlaying {
        player.pause()
        playPauseButton = UIButton(systemItem: .play, target: self, action: #selector(self.playOrPause(_:)))
    } else {
        player.play()
        playPauseButton = UIButton(systemItem: .pause, target: self, action: #selector(self.playOrPause(_:)))
    }
}

// Changes the lock of the playPauseButton
toolbar.items[2] = playPauseButton

// Updates the current time to the slider's value
leftJoiner.seek(to: currentTime)

// Changes the song when previous or next is pressed
leftJoiner.changeSong(_ sender: UIButton) {
    switch sender {
    case nextSongButton:
        currentSong = 1
        break;
    case previousSongButton:
        if we are less than two seconds through the song, this will simply restart the song by setting it to 0; otherwise, this will move to the previous song
        if player.currentTime > 2.0 {
            player.currentTime = 0
        } else {
            currentSong -= 1
        }
        break;
    default:
        print("Something didn't work in changeSong")
    }
}

// Goes to start or end of the song list depending on position in songlist
leftJoiner.currentSong = currentNumber + (currentSong == songlist.count) - 1

leftJoiner.nextSong() // Switches to the new song
leftJoiner.playOrPause() // Pauses the song so it will start playing with the next line of code

// MARK: - Functions
leftJoiner.updateSeekBar() {
    scrubSlider.value = Float(player.currentTime)
}

// This will make it go to a random song if the user shakes the player
leftJoiner.rotationEnd(_ motion: UIMotionEvent, with event: UIEvent) {
    if event.subtype == UIMotionEvent.rotationShake {
        player.stop() // Stops the music
    } else {
        ...
    }
}
Some music created by Jared O'Leary

This is the final project for an electronic studio course I took during my doctorate. I had to compose and synthesize a piece from scratch. The name is a reference to a composer, Henry Partch, who inspired this piece.
Some other presentations on this topic

- A k-8 nexus between music creation, sound design, and computer programming
- Beyond linear coding: Creating and innovating in arts-based programming
- Coding in the k-8 classroom
- Coding live music with Sonic Pi
- Sonic modding: Modding video game music and sound
- Starting an after school coding program
Let’s Talk

- www.JaredOLeary.com
- Presentations
- Emerging Music Nexuses

Presentation by Jared O'Leary and uses Creative Commons licensing Attribution-NonCommercial-ShareAlike (BY-NC-SA)