Music in the Time of COVID
Coping Mechanisms for Network Performance

Rob Hamilton Ph.D.
Associate Professor of Music and Media
Rensselaer Polytechnic Institute
Department of the Arts
robhamilton.io
hamilr4@rpi.edu
Ensemble Nonlinear @ Rensselaer Polytechnic Institute

- Electronic performing **ensemble**/seminar
- **Multi-channel** real-time performance
- Client-Server configurations
- Raspberry **Pi** hardware
- **ChucK**, Pure Data, Processing, XR software
WHERE WE WANTED TO BE...
2021 - WHERE WE WANTED TO BE...
2021 - WHERE WE WANTED TO BE...

CORETET virtual instruments for the 21st century
2021 - WHERE WE ARE...
2021 - WHERE WE ARE...
Ensemble Nonlinear @ Pre-COVID

**CONDUCTOR**
- Multi-channel audio server
- Visuals/Interface display
- Conducting gesture

**LOCAL PERFORMERS**
- Raspberry Pi + keyboard
- ChucK Keystroke Speaker
- Open Sound Control
- Speaker

**ETHERNET SWITCH**
Ensemble Nonlinear @ Pre-COVID

**PERFORMERS**
* Raspberry Pi + keyboard
* ChucK Keystroke Speaker
* Open Sound Control
* Speaker

**AUDIO SERVER**
* Multi-channel audio server
* Visuals/Interface display
* Conducting gesture

**ETHERNET SWITCH**

**OSC**

8-channel audio
Ensemble Nonlinear @ During-COVID

**AUDIO SERVERs**
- ChucK/Pure Data
- Multi-port Input

**PERFORMERS**
- Distanced Ensemble of Student Performers
- Multiple Operating Systems (Win, OSX, Linux)
- Home networks
- Wifi + Ethernet + Mobile (!)
**Ensemble Nonlinear @ During-COVID**

**CONDUCTOR**
- ZOOM HOST
  - Video Sharing
  - Processing GUIs
- AUDIO SERVERs
  - ChucK/Pure Data
  - Multi-port Input
  - Processing GUIs

**ZOOM Stream**
- Stereo audio output
- Variable latency
- Visual Feedback/GUIs

**ZOOM Recording**
- MONO audio capture
- Cloud storage

**PERFORMERS**
- Distanced Ensemble of Student Performers
- Multiple Operating Systems (Win, OSX, Linux)
- Home networks
- Wifi + Ethernet + Mobile (!)

**Screen Share**
- Stereo

**OSC**
Non-Specific Taiko Gamelan Fusion

By Ge Wang and Perry Cook (arr. R. Hamilton)

ChucK + Processing
Jude
By Lennon and McCartney (arr. R. Hamilton)
ChucK + Pure Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Date:</th>
<th>Chord:</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samson</td>
<td>34:1/4</td>
<td>2/39</td>
<td></td>
</tr>
<tr>
<td>Walter</td>
<td>33:3/3</td>
<td>1/39</td>
<td>G 0 G 0</td>
</tr>
<tr>
<td>Matthew</td>
<td>0:1/3</td>
<td>8/39</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>87:1/3</td>
<td>3/39</td>
<td>G 02 G 0</td>
</tr>
<tr>
<td>Garrett</td>
<td>55:2/3</td>
<td>3/39</td>
<td>G 448 G 0</td>
</tr>
<tr>
<td>Ben</td>
<td>39:1/3</td>
<td>3/39</td>
<td>G 32 G 0</td>
</tr>
<tr>
<td>Emily</td>
<td>68:4/4</td>
<td>2/39</td>
<td>Db 072 Db 0</td>
</tr>
<tr>
<td>Chelsea</td>
<td>0:1/3</td>
<td>8/39</td>
<td></td>
</tr>
<tr>
<td>Matthew</td>
<td>46:4/4</td>
<td>2/39</td>
<td>C 28 G 0</td>
</tr>
<tr>
<td>Maxwell</td>
<td>46:3/3</td>
<td>3/39</td>
<td>G 48 G 0</td>
</tr>
<tr>
<td>Bosmas</td>
<td>184:4/4</td>
<td>2/39</td>
<td>Ib 0 G 0</td>
</tr>
<tr>
<td>Kristof</td>
<td>63:1/3</td>
<td>3/39</td>
<td>G 58 G 0</td>
</tr>
</tbody>
</table>

NOTE: VOLUMES ARE DOWN BY DEFAULT!!!
<table>
<thead>
<tr>
<th>Name</th>
<th>Port</th>
<th>Gain (0-6)</th>
<th>Width (ms)</th>
<th>Position (samp)</th>
<th>Pitch/Speed</th>
<th>Pitch (rand)</th>
<th>Width (rand)</th>
<th>Position (rand)</th>
<th>Key Value</th>
<th>Pan (-1 - 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kosmas</td>
<td>7112</td>
<td>0.285</td>
<td>28.028</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Up (1082)</td>
<td>0.300</td>
</tr>
<tr>
<td>Kris</td>
<td>7112</td>
<td>0.314</td>
<td>89.888</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Up (1082)</td>
<td>-0.300</td>
</tr>
<tr>
<td>maxk</td>
<td>7104</td>
<td>0.000</td>
<td>80.000</td>
<td>0.000</td>
<td>4.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>j (74)</td>
<td>0.000</td>
</tr>
<tr>
<td>Dennem</td>
<td>7109</td>
<td>0.418</td>
<td>80.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>3.846</td>
<td>0</td>
<td>Up (1082)</td>
<td>0.700</td>
</tr>
<tr>
<td>Maxim</td>
<td>7106</td>
<td>0.556</td>
<td>84.800</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Up (1082)</td>
<td>0.500</td>
</tr>
<tr>
<td>Ben</td>
<td>7110</td>
<td>0.000</td>
<td>143.268</td>
<td>0.000</td>
<td>0.250</td>
<td>2.000</td>
<td>2.959</td>
<td>4584455</td>
<td>Test (8)</td>
<td>-0.700</td>
</tr>
<tr>
<td>Walter</td>
<td>7102</td>
<td>1.586</td>
<td>89.888</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Right (1079)</td>
<td>-0.500</td>
</tr>
<tr>
<td>Garrett</td>
<td>7105</td>
<td>0.335</td>
<td>80.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Up (1082)</td>
<td>0.250</td>
</tr>
<tr>
<td>Samson</td>
<td>7101</td>
<td>0.345</td>
<td>80.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>5.000</td>
<td>0</td>
<td>Up (1082)</td>
<td>-0.800</td>
</tr>
<tr>
<td>Emily</td>
<td>7107</td>
<td>0.000</td>
<td>53.205</td>
<td>0.000</td>
<td>0.250</td>
<td>0.000</td>
<td>3.846</td>
<td>0</td>
<td>s (83)</td>
<td>0.800</td>
</tr>
</tbody>
</table>

```
0 1 2 3 4 5 6 7 8 9 Set position of read from file  - Step through file =

↓ Decrease/Increase Gain ↑  ⇐ Decrease/Increase Grain Duration ⇩

SHIFT Decrease/Increase Random Grain Duration RETURN

[ Decrease/Increase Random Position ]

Q: 0 Set random position of read from file
W: 200
E: 2K
R: 20K
T: 40K
Y: 80K
U: 100K
I: 7/9 file length
O: 8/9 file length
P: end file length

G: 1.0 Set pitch (speed) playback
F: 0.75
D: 0.5
A: 0.25
H: 2.0
J: 4.0
K: 8.0
L: 16.0

Z: 0 Set pitch randomness
X: 1.0
C: 2.0
V: 3.0
B: 4.0
N: 5.0
M: 6.0

, Decrease/Increase Randomness of Pitch .
```
MUSICAL PROBLEMATICs

- **Spatialization**? What’s that…
- Network reliability / variable latencies
- OS issues, remote support/maintenance
- Gain staging
MUSICAL SOLUTIONS

- Stereo streams with GUI feedback (Spatialization? What's that...)
- Flexible compositional timing (Network reliability/variable latencies)
- **KISS** on the Client-side (OS issues/support/maintenance)
- Prayer (Gain staging)
MUSICAL FUTURES?
MUSIC is a universal experience, bringing us together across huge cultural and physical distances. With Coretet, the joy of playing music has been brought to virtual reality, allowing beginners and experts alike to create and share beautiful music together.

CORETET is a virtual reality instrument and networked performance environment allowing users to play music either alone or across the network. Choose from five instrument presets - like cello, bass or violin - or create your own instrument of the future. Record your solo performances and share them online. Or better yet, connect to other users in real-time across the network to form your own string “coretet”.

Coming Christmas 2021 for Oculus Rift and Oculus Quest
Formed by researchers from Stanford University’s Center for Computer Research in Music and Acoustics (CCRMA), PROCEEDE combines bleeding-edge audio processing technologies and game design to craft the musical game systems of the future.

Design and development on Coretet is led by Dr. Rob Hamilton (Professor of Music and Media, Rensselaer Polytechnic Institute), an internationally-recognized expert in the field of mobile and VR instrument design. As a member of the team that launched Smule, Inc. Dr. Hamilton designed the core music systems that powered industry leading hit mobile music titles such as Ocarina, I Am T-Pain, Magic Piano, and Sing Karaoke/Smule. Art direction and modeling for Coretet is overseen by Chris Platz (Zynga, Smule).

At the heart of Coretet lies the PROCEEDE Audio Engine, using data from the virtual instrument itself to procedurally generate the sound of a bowed string. Alongside the native synthesis and audio processing capabilities of Epic’s UNREAL ENGINE 4, the Proceede Audio Engine uses a process called physical modeling to accurately reproduce the sound and feel of bowed or plucked strings.

TIMELINE

Coretet has already been successfully prototyped and battle-tested in live networked concert performances around the globe (US, Austria, France, Mexico) using a suite of Oculus Rifts. A single player version of Coretet is in development currently with plans to ship on the Oculus Store by Christmas, 2020.

A fully-networked version of the software and an accompanying in-app purchase layer for licensed song content are planned for release respectively in Summer and Fall 2021.
QUEST

Coretet was designed and developed using the Oculus Rift. And with the Rift, the core Coretet experience has been validated by professional and amateur musicians around the globe. With an eventual goal to leverage the Oculus Quest’s hand-tracking functionality, Coretet running on the Quest using the Oculus Touch controllers already offers an amazing experience, giving users the ability to play freely, without restriction.

Players can sit or stand to play and can interact with their instruments using Oculus Touch controllers or their hands in Quest hand-tracking mode. During networked play, players using Quest can freely turn and face their collaborators without restriction.

MONETIZATION

We plan on charging $5.99 for the initial solo-player launch of Coretet. At that time, Coretet will include a set of free-to-play musical scores firmly in the public domain.

With the multi-player release of Coretet, we anticipate increasing the application cost to $9.99. And with the initial release of downloadable content including new instrument skins and sounds, musical scores to purchase, we plan on introducing both a la carte item purchasing or monthly subscriptions. As we solidify relationships with music publishers, we anticipate the application becoming free-to-play with the primary income stream shifting to subscription to our licensed catalog of hit songs.
Music in the Time of COVID
Coping Mechanisms for Network Performance

Rob Hamilton Ph.D.
Associate Professor of Music and Media
Rensselaer Polytechnic Institute
Department of the Arts
robhamilton.io
hamilr4@rpi.edu