

Foxconn Frequency (no.3)

by Remy Siu (蕭逸南)

Introduction

Foxconn Frequency (no.3)—for three visibly Chinese performers was commissioned by the PuSh International Performing Arts Festival and premiered on 31 January 2018.

I have been working on the *Foxconn Frequency* series for the last five years (2013–2018), and this work is intended to be the final entry. The thematic through line of the *Foxconn Frequency* series has always been to examine mass manufacturing and piano pedagogy through the lens of globalization and the contemporary ‘Chinese’ situation. In every entry, the pianists’ skill and labour has been subverted in one way or another through the use of software. Each work in the series approached this subversion differently.

The first entry decoupled the pianists’ labour from the sound output and featured three performers. The second entry introduced simple real-time game mechanics for one performer. The third entry featured three performers again, while greatly expanding the game mechanics and exploring the possibilities of the performance as a process to create unique objects.

For entries two and three, the series features the poetry of Xu Lizhi (徐立志). He was a Foxconn worker who took his own life on 30 September 2014 at the age of 24. His poems were published online and translated posthumously.

In *Foxconn Frequency (no.3)*, his entire body of published work (at the time) was entered into the system, the software controlling all input from the performers and output to the audience. During the performance, the audience sees several of his works in their entirety, as well as lines chosen by the computer to display at various points. My article “Real-Time Game Mechanics in Theatre,” which also appears in this issue of *Canadian Theatre Review*, addresses the journey of this project in detail.

Production Artists

Project Lead Remy Siu (remysiu.com)
Creators Hong Kong Exile (Natalie Tin Yin Gan, Milton Lim, Remy Siu), with Vicky Chow, Paul Paroczai, and Matt Poon
Performers Natalie Tin Yin Gan, Vicky Chow, and Andrei Chi Lok Koo

About the Creators

Hong Kong Exile is an interdisciplinary arts company and registered non-profit organization based in Vancouver. The company is made up of three emerging contemporary artists: Natalie Tin Yin Gan, Remy Siu, and Milton Lim. The three met as students at Simon Fraser University’s School for the Contemporary Arts and have

been consistently creating and producing work since 2011 when the company was formed. They focus on pushing the boundaries of interdisciplinary process and creation (hongkongexile.com).

On the Provided Score Images

The following images are from the system that runs *Foxconn Frequency (no.3)*. The system was built in TouchDesigner, a node-based visual programming language developed in Toronto, Ontario (!). Within TouchDesigner, one can also use the Python programming language for scripting. Much of *Foxconn Frequency (no.3)*’s real-time game mechanics are built using Python within TouchDesigner.

Kim McLeod, of *Canadian Theatre Review*, asked for script excerpts for this article. Since this work does not have a script, my next thought was to submit a ‘score’ excerpt. But a score also does not exist in the traditional sense, so I thought to represent the system by showing the reader what it looks like to me when I am working in it.

In some of the images, you can see that there are actually scores with Western music notation. These are what the performers see in general testing. As seen in the images, these scores are living in the system’s ‘memory,’ to be called when needed.

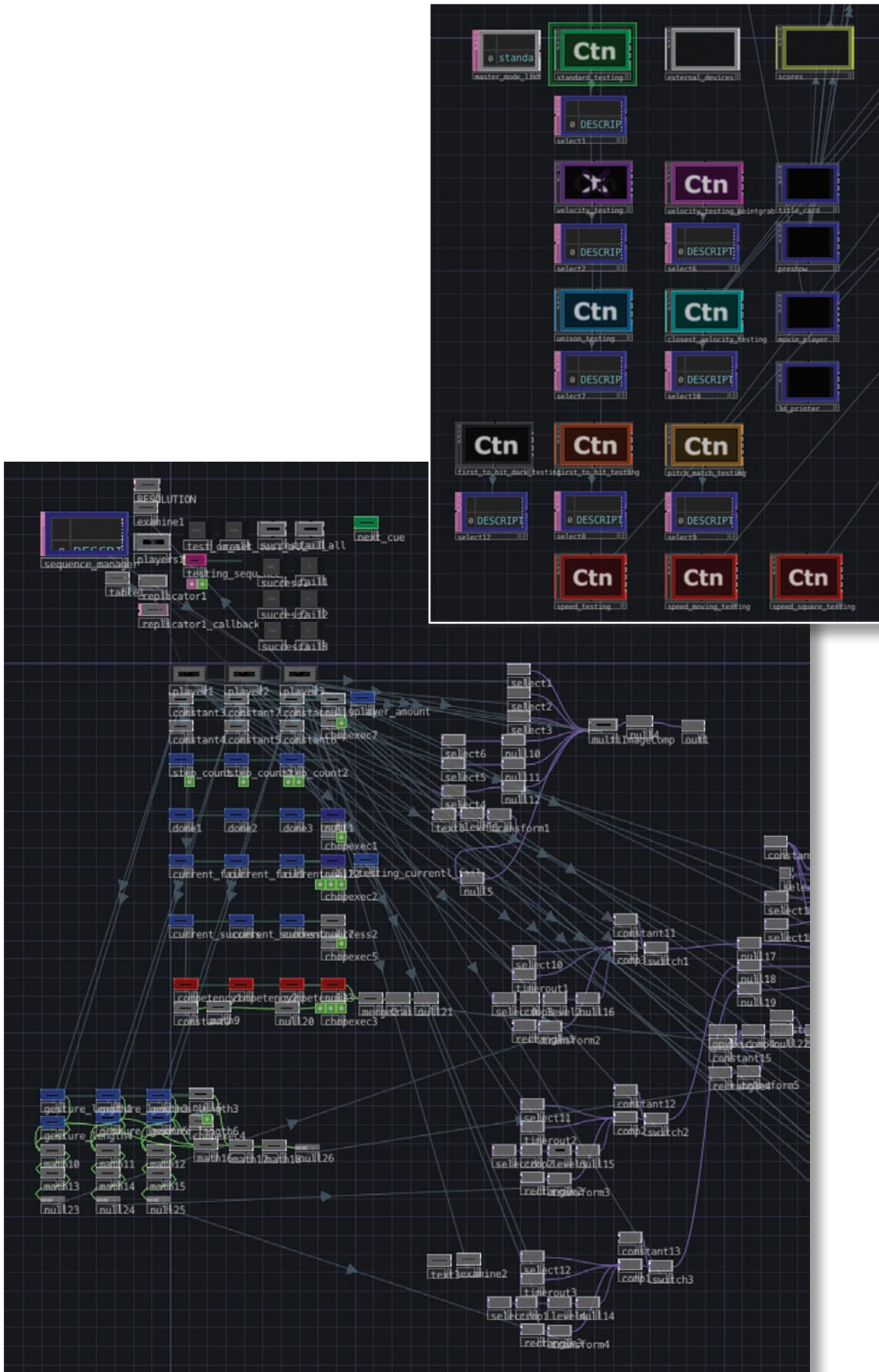
As anyone with programming experience might notice by looking at my TouchDesigner network and Python code, this process can get pretty messy. In many ways, the process of *Foxconn Frequency (no.3)* is informed by my experiences in devised theatre. In our rehearsals, I wanted to follow a promising trajectory if it suddenly appeared—for a split second—in front of us. This meant diving into the code and doing things quickly as opposed to cleanly.

One can see this too, in the list/table of cues and various parameters. Some like, “FILLER,” seem tacked on. That’s because they were. It was probably not the best place to put this kind of parameter, but it was born out of the necessity for speed. Because oh my god, the performers look cold and agitated, we better get this thing on its feet in the next ten minutes.

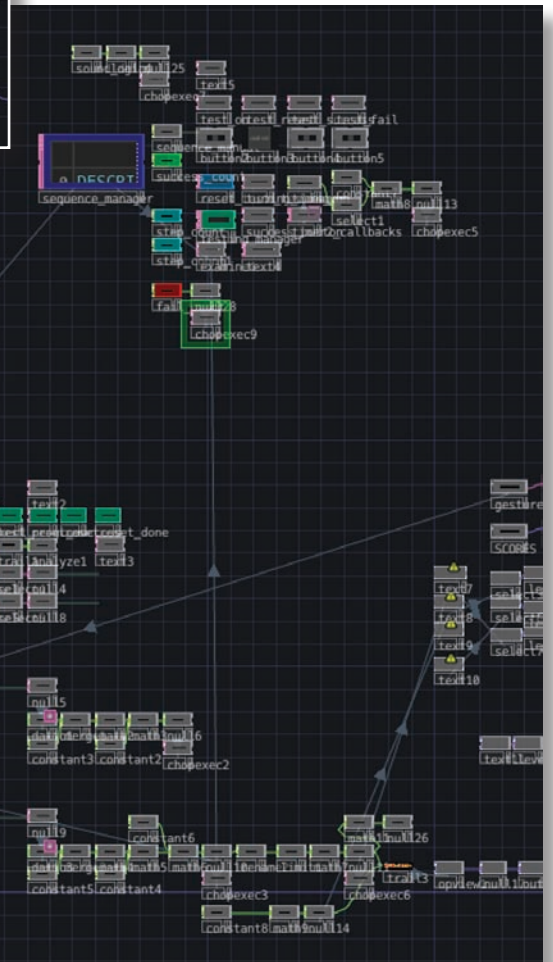
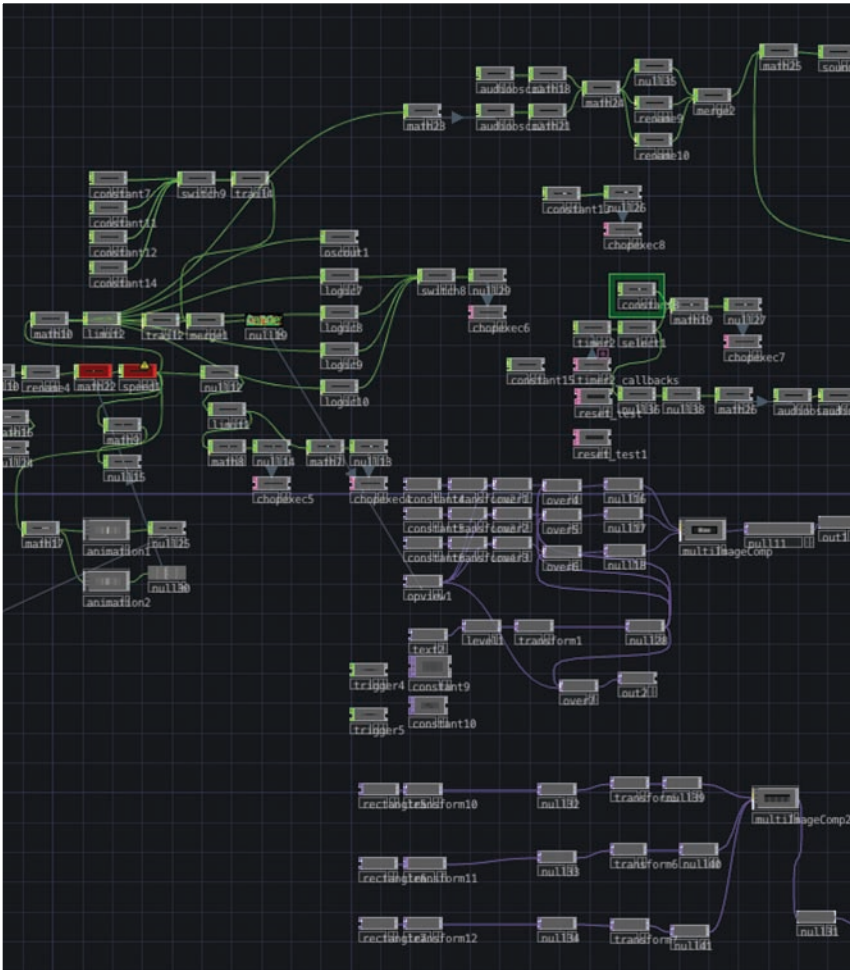
About the Creator

Remy Siu (蕭逸南) is a composer and new-media artist based in Vancouver, British Columbia. He is the co-Artistic Director of Hong Kong Exile. He actively creates with Mahaila Patterson-O’Brien, Theatre Replacement, and Howie Tsui. During 2017–2019, he will be one of two composers-in-residence with the National Arts Centre Orchestra (Canada). He is also currently curator-in-residence at Contemporary Musiking Hong Kong. In 2019, he was nominated for the Gaudeamus Award.

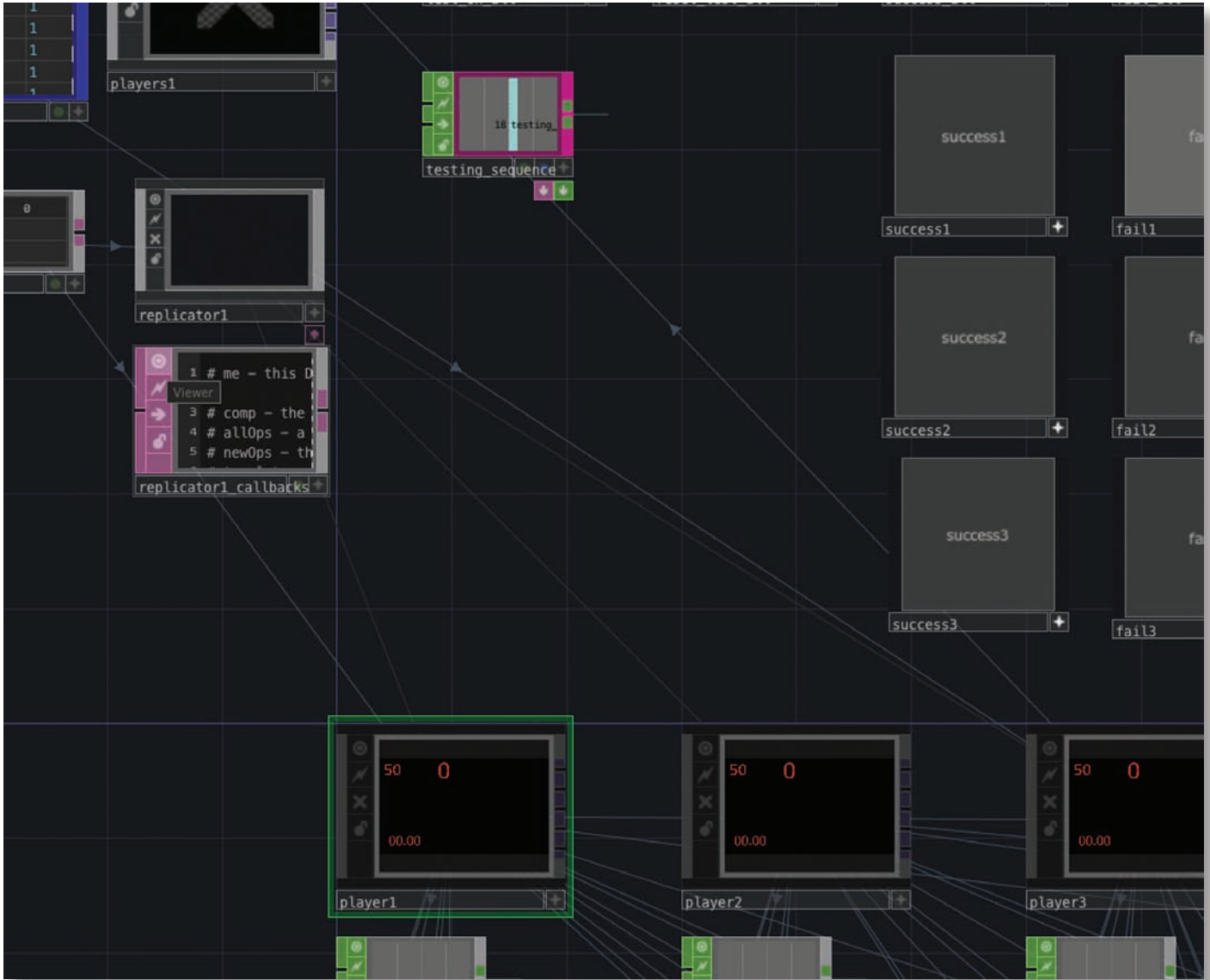
\$_[protocol]://ctr.utpjourmals.press/doi/pdf/10.3138/ctr.178.011 - Remy Siu <remysiu@gmail.com> - Tuesday, March 19, 2019 8:43:38 AM - IP Address:81.149.38.197



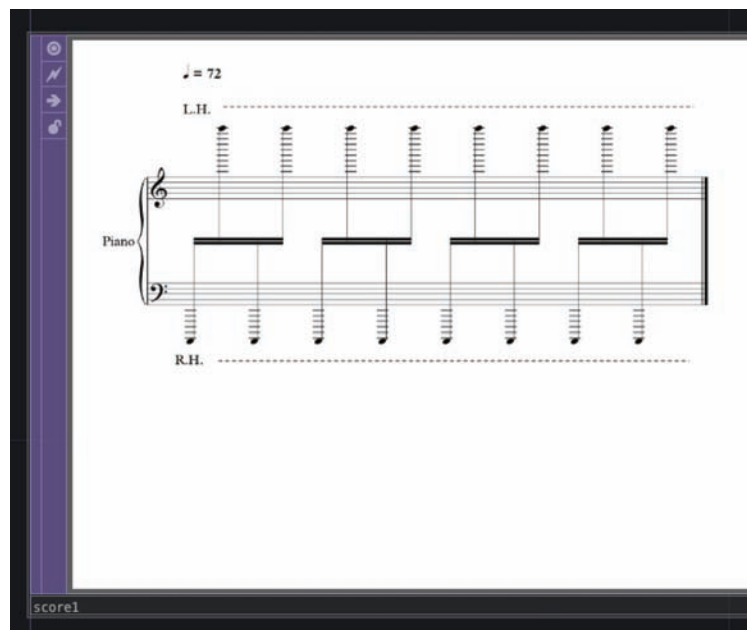
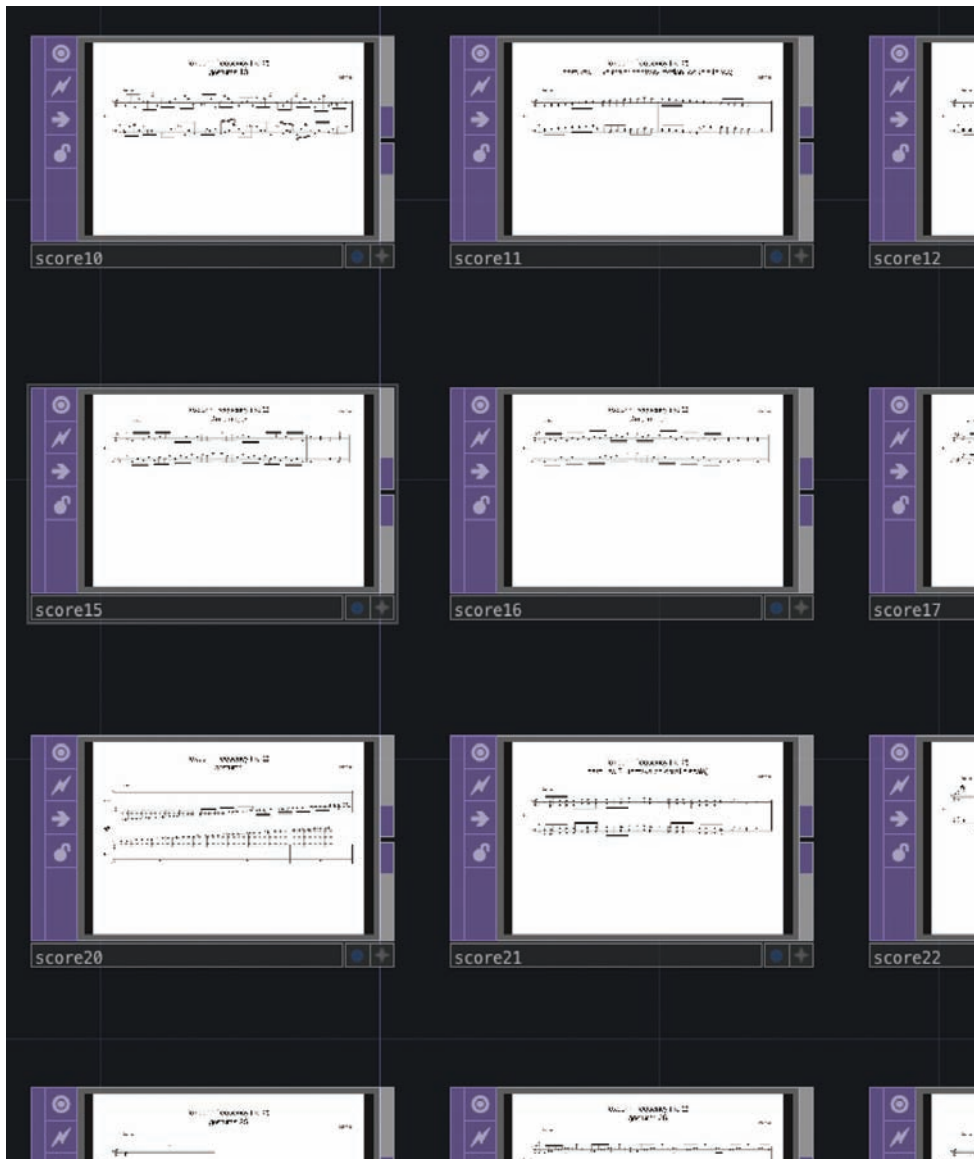
\$_[protocol]://ctr.utpjournals.press/doi/pdf/10.3138/ctr.178.011 - Remy Siu <remysiu@gmail.com> - Tuesday, March 19, 2019 8:43:38 AM - IP Address:81.149.38.197



\$_[protocol]}://ctr.utpjournals.press/doi/pdf/10.3138/ctr.178.011 - Remy Siu <remysiu@gmail.com> - Tuesday, March 19, 2019 8:43:38 AM - IP Address:81.149.38.197



\$_[protocol]://ctr.utpjournals.press/doi/pdf/10.3138/ctr.178.011 - Remy Siu <remysiu@gmail.com> - Tuesday, March 19, 2019 8:43:38 AM - IP Address:81.149.38.197





```

1 show_order = op('show_order')
2 cue_number = me.digits
3 poems_op = op('poems')
4 movie_op = op('movie_player')
5 printer_op = op('3d_printer')
6
7 filler = show_order[ cue_number , 'FILLER' ]
8
9 if filler == 0:
10
11     poems_display = show_order[ cue_number , 'POEMS' ]
12     poems_speed = show_order[ cue_number, 'POEMS_SPEED' ]
13     poems_move = show_order[ cue_number, 'POEMS_MOVE' ]
14
15     poems_op.store('poems_display', poems_display)
16     poems_op.store('poems_speed', poems_speed)
17     poems_op.store('poems_move', poems_move)
18
19     movie_play = show_order[ cue_number, 'MOVIE_PLAY' ]
20     movie_op.store('movie_play', movie_play)
21
22     printer_time = show_order[ cue_number, 'PRINTER_TIME' ]
23     printer_op.store('printer_time', printer_time )
24
25     keyboards_on = show_order[ cue_number , 'KEYS_ON']
26     keyboards_amp = show_order[ cue_number, 'KEYS_AMP']
27     me.parent().store('keyboards_on', keyboards_on)
28     me.parent().store('keyboards_amp', keyboards_amp)
29
30     audio_amp = show_order[ cue_number, 'AUDIO_AMP' ]
31     me.parent().store('audio_amp', audio_amp)
32
33     movie_amp = show_order[ cue_number, 'MOVIE_AMP' ]
34     me.parent().store('movie_amp', movie_amp)
35

```

```

dat_testing_manager_td_53738_42.py x
1 import random
2
3 player = me.parent()
4 parent = me.parent(2)
5 audio_player = me.parent(3).op('audio_player' + str(me.parent().digits))
6
7 sequence_manager = op('sequence_manager')
8 testing_sequence = player.fetch('testing_sequence')
9 testing_step = player.fetch('testing_step')
10 testing_step_prev = player.fetch('testing_step_prev')
11 testing_fail = player.fetch('testing_fail')
12 testing_success = player.fetch('testing_success')
13 global_reset = player.fetch('testing_global_reset')
14 gesture = op('gesture_output')
15 row_length = gesture.numRows
16 col_length = gesture.numCols
17 chord_next = []
18 timewait = parent.fetch('timewait')
19 sequence_length = sequence_manager[ testing_sequence, 'LENGTH' ]
20
21 def main ():
22
23     testing_on = player.fetch('testing_on')
24     op('colour_box').par.index = 0
25     player.store('test_display', 0)
26     player.store('key_result', 1)
27
28     white_box = sequence_manager[ testing_sequence, 'WHITE_BOX' ]
29     parent.store('white_box', white_box )
30
31     get_coop_setting()
32     timewait()
33     success_condition()
34     get_gesture_length()
35
36     # IF PLAYER_ID is in PLAYERS_LIST
37     # RUN TESTING
38     # ELSE, INCREMENT SEQUEUNCE STEP ??
39
40     if testing_on > 0:
41
42         current_fail = parent.fetch('testing_current_total_fail')
43         #print('current_fail', current_fail)
44
45         if current_fail == 1:
46             multiplayer_fail()
47
48         if current_fail == 0:
49             linear_player()
50
51         player_on = player.fetch('testing_player_on')
52         fail_forward()
53
54         if player_on > 0:
55
56             player.op('switch2').par.index = 1
57             player.op('switch3').par.index = 1
58             player.store('testing_competency_on', 1)
59
60             first_step = testing_fail + testing_success
61
62             if first_step == 0:
63                 gesture_get()
64                 #print('one')
65
66             if first_step > 0:
67                 gesture_get()
68                 #print('two')
69
70
71

```

	0	1	2	3	4
0 DESCRIPTION	FILLER	AUDIO_AMP	KEYS_ON	KEYS_A	
1 SPEED	0	1	0,0,0	0	0
2 *****TITLE_CARD	0	0	0,0,0	0	0
3 *****DOWN_GOOGLEMAPS	0	1	0,0,0	0	0
4 1p - Vicky Intro	0	1	0,0,0	0	0
5 *****LEFT_GOOGLEMAPS	0	1	0,0,0	0	0
6 1p - Nat Intro	0	1	0,0,0	0	0
7 *****RIGHT_GOOGLEMAPS	0	1	0,0,0	0	0
8 1p - Matt Intro	0	1	0,0,0	0	0
9 *****IN_GOOGLEMAPS	0	0.9	0,0,0	0	0
10 *****3D PRINTER INTERLUDE	0	1	0,0,0	0	0
11 1p - dif players, normal gesture, 6 length, ff	0	1	0,0,0	0	0
12 1p - dif players, short gesture, 4 length, ff	0	1	0,0,0	0	0
13 2p - Nat + Matt, sync, ff, short, 1	0	1	0,0,0	0	0
14 2p - Vicky + Matt, sync, ff, short, 1	0	1	0,0,0	0	0
15 2p - Nat + Vicky, sync, ff, short, 1	0	1	0,0,0	0	0
16 1p - Nat, nff, sp, 5 length	0	1	0,0,0	0	0
17 3p - co-op, 3	0	1	0,0,0	0	0
18 3p - FIRST TO HIT, 15, ben track float87 **KEYBOARD ON**	0	0.5	1,1,1	1	1
19 1p - matt, sp, nff, 5 **	0	1	0,0,0	0	0
20 1p - vicky, sp, nff, 3	0	1	0,0,0	0	0
21 2p - UNISON, nat + matt, 1	0	1	0,0,0	0	0
22 1p - sp, ff, vicky, 1	0	1	0,0,0	0	0
23 2p - UNISON, nat + matt, 1	0	1	0,0,0	0	0

show_order