

Controlling the Stage

By Christopher-Elizabeth

This Nightswimming Pure Research project was conducted at the Tarragon Theatre's Extraspace from January 18 to 20, 2023. Thank you to the Tarragon for its partnership and collaboration on Pure Research. This report published July 2023. For more information on Pure Research, please visit nightswimming.ca.

Research conducted by Christopher-Elizabeth, Brian Quirt, Gloria Mok and Nathaniel Hanula-James.

Introduction

I love video games. A group of designers, directors and programmers create a world where players are free to play. A game is incomplete without a player, who in a game is also the audience. My favourite games have interactive and responsive worlds, which instill a sense of liveness in the player.

I love theatre. A group of designers, a director, and technicians create a world in which actors can play. A production is incomplete without an audience. My favourite productions take full advantage of the presence of audience. Audiences can give prompts, ask questions, and join the stage. But what about controlling the stage? The lights? The sound? The Haze?

The reason we don't do this is simple: the equipment is expensive, there's a major learning curve, lighting boards look terrifying, blah blah blah. You know what's not terrifying? A Rockband guitar. An Xbox controller. A Wii Remote. These are all objects that are designed with ease of use in mind.

So what happens if we give the audience access to design tools, and unleash clowns upon them?

What does that relationship look like and is this pursuit worthwhile? Can it be fun for more than just the participant?

What new stories will this new technology allow us to tell and how can this enhance stories we're already familiar with? That is the research I took on in collaboration with Nightswimming and the Tarragon Theatre.

Language

Before we enter these newly charted waters, I just want to establish some of the language I'll be using in this report.

Audience

An observer of the story

Player

An audience member with agency to affect/interact with the world of the story

Improvisor

A performer performing without a script or without any particular style or character

Clown

A performer using their clown personas to tell a story

Part 1: The Experiment

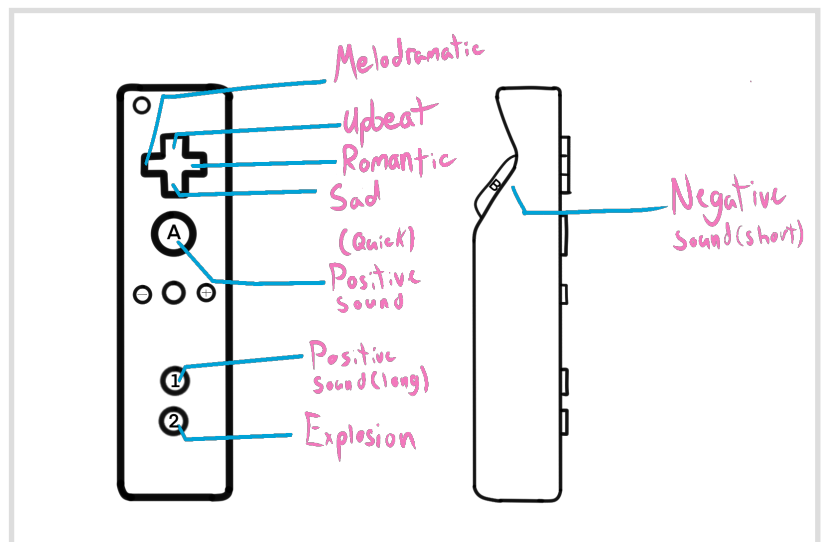
Day One: The Set-Up

Games

On the first day, after the equipment had been setup, myself and the clowns (Kendelle Parks, Kole Durnford, Michelle Blight, Jacob Willis, and Robert Morrison) started playing regular theatre games, just to get maximum play in, and to get acquainted with playing together in the space. We had a good game of Whoosh at the top, then we did scenes with actors making sound effects with their mouths. This was to help the actors get an idea of how the interactions with audience members would feel, and gave me a good idea of what type of sounds they gravitated towards as clowns, vs as improvisers and players.

After a few rousing rounds of mouth sounds, we tried giving the performers a scenario: beach volleyball. I jumped on the soundboard and we did regular sound improv. The soundboard had sounds and music from various videogames and movies. Partway through, Kole jumped on to

Wii Controller Mapping



play with me and the performers found themselves trying to get us to play them a specific sound again. So they would telegraph the actions they believed made us play the sound. Telegraphing is also very important in videogames: it gives observant players a fair chance to react appropriately to an action. For example: An enemy in a game might yell before attacking, giving the player time to block. Or in our case, the clowns would make fingers guns to get the audience to play the 'pew pew' sound from *Star Wars*.

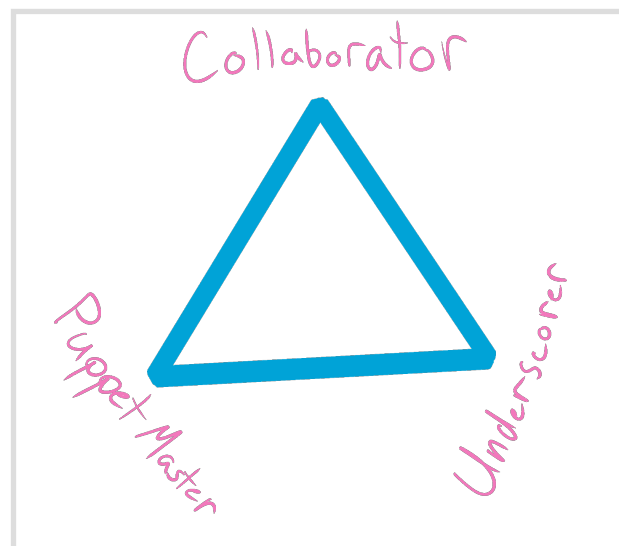
After this we realized that the soundboard had too many sounds, so we moved onto to using video game controllers, starting with the Wiimote! We also decided to try as many variations as possible for the rest of the day including:

- Playing games in the void
- Having sounds be diegetic (existing within the world of the scene, ie. an alarm)
- Having sounds be non-diegetic (existing outside the world of the scene, ie. a score)
- Having the performer be aware of the audience
- Letting audience know the scene
- Not letting the audience know the scene
- Giving the clown a controller
- Creating a moment where the performer discovers the player
- Only using two sounds
- Only using music
- Giving clown props
- Having two clowns in one scene

After doing all these variations we noted a few discoveries:

- The concept can be understood through watching
- Only a handful of distinct sounds were necessary to keep the scene interesting, which makes the Wiimote the perfect controller for the job
- The performance is just as compelling to watch as to play
 - In each performance we get to watch audience and performer build a language together, and we watch the two create rules for this game once together
- Every variation we tried had its own merits
- Hostility is fun and can go both ways
- It's helpful to introduce sounds one at a time, but not strictly necessary

- Audiences can give power to the prop by imbuing it with sound, but the performer is the one who gives it life
 - Costume pieces lend themselves well to this exercise, it's like a video game power up
- If there's more than one clown, more prep work has to be done between the two of them before going to an audience, this is because it's harder to build a shared language between three people than it is two
- This is a fun idea worth pursuing
- There is a triangular spectrum of the ways in which the player interacts with the performer
- Rapport building is important
- Clowns require a certain amount of gentleness from audience
- Performers need rules to follow in order to be successful, regardless of what exactly they are
- The same sound can be imbued with many different meanings, and the meanings of the



sounds can have an arc of their own throughout a single performance.

We've also been left with a number of questions:

- How can we nudge the audience toward the collaborative?
- Which rules allow things to succeed more?
- What type of scenes are satisfying to watch play out?

Since we made so much progress, on day two we started working with lights!

Day Two: Lights

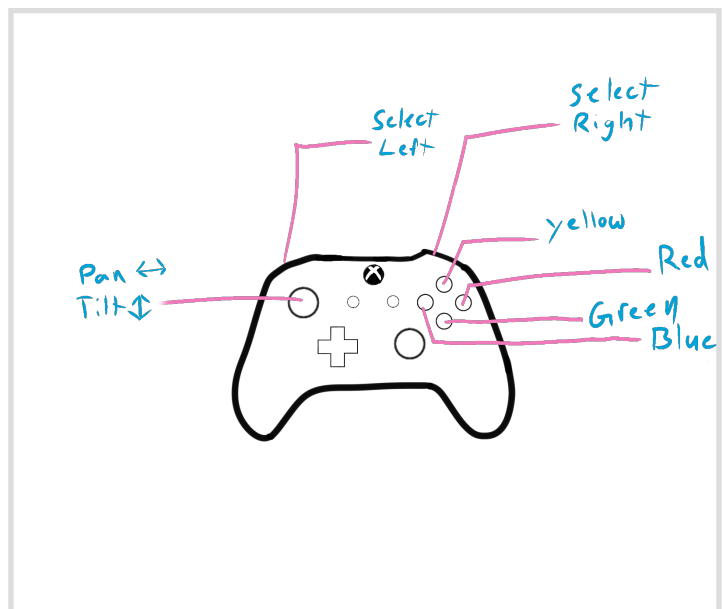
In the morning, I began programming the lights to work with the controller. Unfortunately I never could quite get the motion controls on the Wiimote to work, but I was able to control the moving lights with the Xbox controller and WOW, I've never felt cooler. Each button would change the colour of the light, and the joystick would move the light. We tried out a few games with the light after our warmup, mostly around how the light affects the mood or genre of the story. But first! We had our first play tester come along! See Section 6.1 for a more detailed breakdown of how it went.

So with both moving lights being controllable, we made up a game: if you're in a light, you have to adjust your intention to whatever mood the light is. The performers were allowed to move between lights to deliver a line how they liked, but I could also follow them with the light, or change the light as soon as they moved into a new one.

- Red = Angry
- Blue = Sad
- Yellow = Happy
- Green = Disgusted

I chose these colours because they aligned with the colours of the buttons on the xbox controller, and I only just realized now that they're the same moods as the characters from Pixar's inside out. As a game, this was a LOT of fun, but it didn't bring a lot to the table dramaturgically. I stand by this being a great warm-up exercise though, and I would really recommend bringing this to the rehearsal hall, classroom or improv games. The changing of the colours didn't quite advance the story, although it did give the audience plenty of agency to interact with the performers. I went in with the idea that the light could act as a third scene partner, treating it like a character on its own. We didn't get a chance to explore this concept very deeply. I think because it's so abstract, it would need a full day to figure out rules for this on its own. But we didn't have a day, so I took Brian's advice and instead designed different lighting looks to map onto the controller, which would be available the very next day.

Xbox Controller Mapping V1



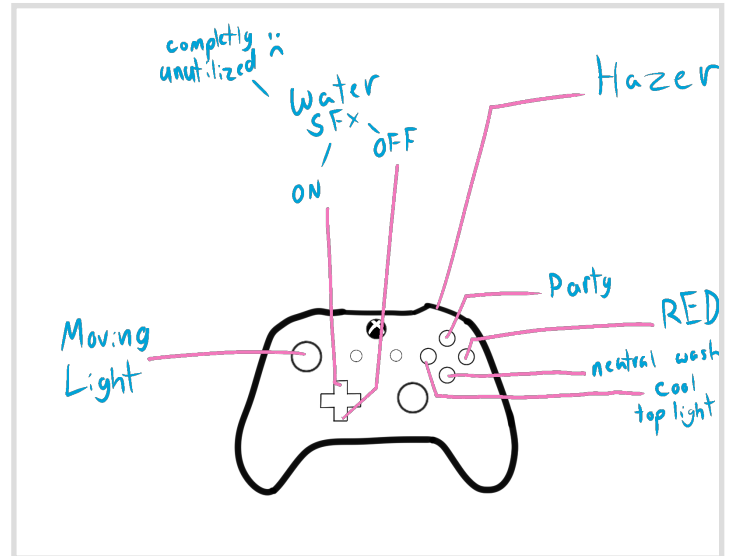
At this point, the xbox controller could only affect the colour of the moving lights

Xbox Controller Mapping V2

After our lunch break, our very first audience came in for play testing! After giving them a demo of the equipment, we jumped into scenes using just the Wiimote with sound effects.

Day Three: Refining and sharing

The final day we spent refining. The day started with designing scenes for the Xbox controller, since the moving light wasn't effective as a tool on its own. The scenes were distinct from each other and some came with effects. There were a lot more tools to play with now. This was also a lot simpler to program, especially with touch OSC. We later discovered the moving light could strobe, we don't know why, but it was used by an audience member quite confidently.



After playing with the lights we tried more variations of our game and established some rules we would stick to when the audience comes in later:

We would do four scenes for each audience

1. Single Sound Player and One Clown
2. Sound and Lighting Players and One Clown
3. Single Sound Player and Two Clowns
4. Sound and Lighting Players and Two clowns.

Part 2: Dramaturgical Concepts

Relationships

Audience reception was fantastic! Every member was excited to play, and they all found the Wiimote very easy to use. If they were confused before the start, they understood what they could do after watching the first player. Watching the performer interact with the audience this way was like watching kids playing a game, building a language and dancing. Each clown and player had a unique language between them using just the sound effects. Audiences felt playful and acted childish, both in a very kind way but also in a 'toddler learning a swear word' kind of

way. (They know that this thing will get a reaction out of the clown every time.) Some audience members found comfort in the anonymity of the performance since they got to interact without all eyes being on them.

The audience can be guided toward specific actions or to act a specific way depending on how the performer decides to interact with them. For example, the performer might offer large obvious gestures, like a finger gun, a big stomp or a thumbs up. The actor maintains plenty of agency in the situation, because the audience has control over the space and not the actor, it is up to the actor to entertain the audience based on the offers they receive – antagonistic, collaborative, ambient or otherwise. Clown and Bouffon lend themselves well to that type of interaction, since the desire to entertain and the ability to gracefully (or funnily) fail are embedded in those forms.

Player-Performer Relationship

We noticed a triangular spectrum of ways the audience interacts with the performer: they are either collaborative, puppetmasters or they stick to scoring the action.

- Collaborator
 - Working together to create something magical
 - Enhancing what the performer already brings to the table
 - Laughing together
 - Co-op mode
- Puppetmaster
 - Conditioning the clown to act a specific way
 - Can be antagonistic
 - Performer can rebel
 - Can feel like a match between two adversaries
 - Sound directly effects action
- Underscorer
 - Sticking to the musical cues to enhance a scene
 - A lot of fun in cutting the song off at pivotal points
 - Music greatly affects a scene that's been prepared in advance
 - Easiest way to interact
- Language
 - At the start of every performance someone is responsible for making an offer, and their partner will make a counter-offer. Eventually these lead to sounds having specific connotations: generally there's a "carrot" sound and a "stick" sound; that is to say a punishment and a reward. The reward is often being strived for, particularly in puppetmaster type relationships, similar to how a clown works for applause. This desire was seen when an offer was repeatedly made by the performer but the player refuses to engage, to them. The basic rules of every scene are established in the first 90 seconds, but by the end the rules are subverted, or they evolve. Often at the end of a scene the player will give the clown the sound that they wanted the entire time. It's all very clear what the intentions are, and the scenes often feel like duets between friends or rivals.

Inter-Audience Relationships

Audience members were connected with each other when they were playing together, and there was a responsibility that the players felt to put on a good show for everyone else. The first scenes can really set the mood for the rest of the performers, so giving them an idea of what's possible is important.

Part 3: Technical Concepts

• MIDI

- How digital music instruments talk to each other, and computers. It can send generalized messages, such as a key on a keyboard, and an action will happen whenever that particular key is pressed. The most common form is using a digital keyboard to tell a computer to play a note. However, MIDI is much more versatile. In theatre it's most often used to sync lighting looks to specific moments in music. The first thing I did with MIDI was to try connecting different looks to specific drums. However, it's good for musicals, but it had a hard time keeping up with my amazing drum solos.

• OSC

- This brings me to OSC. It can do everything MIDI can do but instantly, and do much more. Most (if not all) modern lighting and sound boards can use OSC. What makes it better is that there are predefined messages, and it runs strictly on a network (over wifi). With MIDI, every message has to be defined on the receiving end, and it's limited to whatever the developer thinks should be controlled by MIDI. OSC uses specific commands that can easily be adjusted by the device sending commands.

• Networking

- Networking can be very complicated in practice, but quite simple to explain. Every device has a unique number on a network, and these can directly talk to each other. Most importantly for our use, we can have some devices tell another device what to do. Like when you tell your wireless printer to print from your computer. So long as all the devices are on the same network, and they all know each other's unique number(s) they can communicate, so long as they speak the same language. (Like OSC or even MIDI)
 - Unique numbers
 - The MOST important troubleshooting tip is to ensure that all the numbers match, and that you note what they are BEFORE you start connecting things.
 - The numbers in question for this project:
 - IP address
 - OSC Receiving Port
 - OSC Sending Port
 - Channel numbers

• UX Design

- Ensuring that functions of a device are intuitive and clearly communicated to the user. The Wiimote controller is a great example of good UX because every button is different, so it's easy to remember what each button does.

• Ludonarrative

- Ludonarrative is the story told by the way you interact with elements of a videogame.
 - Ludonarrative Dissonance: when the way you interact with a game contradicts the story being told.
 - For example: a player being told they can go anywhere, but there are invisible walls and locked doors everywhere.
 - Ludonarrative Harmony: When the way you interact with a world and the story being told are in agreement.
 - For example: a player being told the world they inhabit is cruel and unforgiving, and the gameplay is extremely difficult.

Design Considerations

- Lighting
 - Have strong distinct lighting looks.
 - Colour and movement are fun, but they aren't enough on their own.
- Sound
 - Different sounds can have different meanings, and you will be surprised by how audiences use them.
 - You don't need a lot of sounds, but they all need to be distinct.
 - Toggling for music is a lot easier to do than holding a button down.
 - There are many ways to map the controller, it can be as simple or complicated as you wish, there's room to play with sound placement and speed.
 - If there were multiple players, one could control sound effects and the other can score. This would free up space on the controller for more manipulation.
 - Additionally, the nunchuck could be added for more buttons on the Wiimote.

Part 4: Findings

- This is an inherently fun way for audiences, designers and performers to engage with each other
- Every variation we've tried was fun, but our favourites were the ones when actors had a scene that could be disrupted/enhanced, or a prop to explore.
- The Wiimote is the most user-friendly controller
 - Different people with different familiarities with video games have different desires for options. (i.e. the more experienced player Jack would have liked to be able to manipulate speed and pitch of audio samples, but the less experienced Andrea was happy with the options)
 - The buttons have different shapes and are distributed over a long and ergonomic surface
- Anything can be a controller
- Motion controls are significantly harder to program than anticipated, and if you want stable software, do not update your operating system
- Specificity is key: in action and design
- It takes about 90 seconds for players to gain an understanding with performers.
- Actors need to acknowledge what the player contributes in some way; this can be tricky and can get overwhelming with pairs who don't have a plan
- Watching the players struggle was just as fun as watching the performers
- A solid tutorial on the controls goes a long way, and the first scene sets the tone
- For performers, it's easier to react to sound than lights

Part 5: Next Steps

- Find way to allow players to memorize sounds faster, like a projected Heads Up Display showing them what button they're pressing
- Discover ways to weave this game into a compelling story while retaining spontaneity and audience participation

- Create an easy-to-use version of this tech so more people can play with it
- Create wearables for actors to be able to control specific sounds when needed
- Do more variations on these games, maybe bring them into some improv groups.

Performance Descriptions

6.1 Dean vs Kendelle

Notes:

For this performance, I was actually just really excited to have an outside eye, so I forgot to take good notes, and didn't quite have a system for doing so yet, but here's what I wrote down:

RULE: Sound only, one clown

- Dean mostly played songs instead of sound effects, played along with Kendelle
- Holding down the button to play music was frustrating because their hand would slip and they'd have to restart the song again.
- Dean asked questions about the controllers as opposed to trying them out on their own
- Aside from that, the controllers were intuitive

6.2 Dean vs Robert

RULE: Sound only, one clown w/ prompt

- **Prompt:** Triathlon on the moon
- Robert wasn't satisfied with the offering that Dean was giving him, so Robert asked for a specific sound, while telegraphing an action, this helped the two develop a language

- Dean was more compelled to rebel against the performers' wishes since they had a better understanding of what was possible.
- There was more action for Robert to do with a prompt than without one
- By the end of the performance, the two felt close and they had a grand finale of sounds and callbacks
- The rules of this impromptu game were clear, as Robert was very good about telegraphing actions and making requests early on. When the rules are established it's more enticing to break them
- Music was used to progress the story

Lessons

- Offers that are in the world are more compelling than when they're from the audience
- The clown should avoid controlling the controller, demanding from the audience, you have to be able to roll with the punches the player gives you

6.3.1 Justin vs Jacob

Rule: Sound Only, one clown

- They quickly developed a clear language with the sound bank, some sounds were positive, some were negative, and the songs were distinct enough to create different moods.
- Switched controllers partway through

6.3.2 Olivia vs Jacob

- The language created with Justin was completely different with Olivia, despite the scene, and sound effects being the same
- Controls felt intuitive
- Felt powerful

6.4 Anne vs Michelle

Rule: Sound only, one clown, with prop (piece of plastic)

- Didn't identify as a gamer, was nervous at first
- Did a test with everything before beginning
- Found games with sounds
 - Would make exploding sounds when Michelle walked
 - Teased Michelle with music
- Gravitated toward using music over sound effects
- Never felt like she was out of sounds
- Eventually was endeared to Michelle, and even gave Michelle the remote without being prompted.

Lessons

- The novelty beats the fear
- The prop brought as much to the table as a prompt
- Music happens when the clown doesn't do much
- Clown will NOT leave the scene unless forced to
- The first player sets the tone of the audience. Because Justin was gentle, everyone else felt like they should be gentle as well.

6.5 Jonnie vs Kole

Rule: Sound only, one clown, with prop

- Jonnie was sitting front and centre
- Prop became a vessel for sound to come out of
- Kole verbally reprimanded Jonnie and asked for specific sound
- Kole talked to Jonnie quite a bit

- Sound made a relationship possible, creating objectives for both Kole and Jonnie
- Jonnie enjoyed the lack of fourth wall

Lessons

- The demonstration before was helpful
- Knowing who's in control makes it easy for performer to get what they want, since the performer can direct their attention and energy towards the player

6.6 Jenna vs Robert

RULE: Sound only, one clown, with prop

- Sound happened when Robert engaged with the prop
- The relationship with the prop was antagonistic
- Jenna felt like part of the creative process
- They found their own language
- They built trust over the course of the scene

Lessons

- Performers really only need 3 clear gestures to get going
- We need a time limit on these scenes
- There's no discomfort or anxiety when engaging with performers this way
- Power feels equal
- Brought out audiences' playfulness

6.7 Aaron vs Kendelle

- Felt like he was contributing to the performance by giving bits and obstacles to overcome
- Desired a way to get more sound options
- Had fun purposely rejecting offers

6.8 Sofia + Irene vs Robert

- Robert interacted with audience immediately
- Irene's controller didn't work for the first little bit
- Quick to be influenced by sounds
- Good at creating gestures to get certain sounds
- Frantically allowing design to influence performance + strong desire to play with audience
- "I enjoyed him asking for the thing, and I wanted to give him what he wanted"
- Collaborator - Sofia
- Forcing the scene, bending and breaking their will -Irene
- Irene would have liked to manipulate the clowns voices in some way.

6.9 Mike vs Michelle + Kole

- rejected offers in favour of their scene
- Michelle created a game quickly by doing big gestures that Mike could react to
- Michelle and Mike were working against Kole who insisted on their predetermined scene
- Michelle would touch their foot and Mike would press the laser button, she would then use this to piss off Kole's clown
- Mike: controllers, but couldn't remember where all the sounds were off the bat, but made interesting discoveries when they pressed wrong button
- Physical offers Michelle made the most visible physical offers

- Desire to control volume
- Encourages Andrea who doesn't play much games

6.10 Andrea + Josh vs Jacob + Kendelle

- had an oven mitt
- Josh used the hazer a lot, and they reacted to it, trying to stop the hazer
- There was a lot of information for everyone to absorb,
- Andrea would give some sounds to Kendelle but not Jacob
- When the lights hit they HIT very strong, but not on their own
- Josh strobed the red light
- They really committed to the music sometimes
- Andrea was not a gamer
- Josh: felt like he had a responsibility to do the story justice, but still had fun. Developed a visual language, felt like offers went back and fourth
- Prefers lights, as a visual person, very fun affect atmosphere vs sound is affecting the actors
- Josh liked having very specific tools to create, a limited amount, it allowed for more discovery and creativeness. "What is the funniest thing I could do right now"
- Aaron thinks this a good tool for designers who don't know Qlab to build with choreographers and the same for designers who don't really know how to work the board

6.11 Annasofie vs Michelle

- Played a lot more with musical offers
- Setup musical roles and scenes for Michelle to interact with her prop, the sound would trigger from prop in the performer's world
- A game wasn't found more towards the end of the session
- Spent a lot of time testing the waters.
- Felt collaborative

- Desired an option to fade sounds in and out

6.12 Jack + Jordan vs Kendelle

- After finding a game rather quickly, the players began instigating a lot
- Jack and Jordan really worked well together, usually against Kendelle
- Kendelle responded very well, taking in all of the stimulus, and turning on the audience for this rude behaviour
- Jack found a lot of joy when they were able to sync up with Jordan
- I also want to note that the two were not sitting near each other, nor did they know each other before this.
- Jack desired motion controls

6.13 Shireen vs Jacob + Kole

- Performers had a scene prepared and Shireen underscored it
- The SFX were integrated into their story
- Jacob was good at making requests for sounds, Shireen knew exactly what to do
- Shireen was actively trying to advance the narrative, but also instigated when she wanted to.

6.14 Alex + Antel vs Robert + Michelle

- Performers had a scene prepared, and weren't very reactive to sound
- Alex forgot the controls sometimes, but never felt uncomfortable
- Alex wanted to support the scene without ruining what the actors prepared
- Alex wanted moving lights
- Antel was nervous, and felt caught between two songs
- Felt great whenever he was able to sync up with the scene

Part 7: Instruction Manual

Ingredients

• Software

- TouchOSC
- ETC EOS
- Logic Pro or Ableton Live
- Gamepad-tool

• Hardware

- Videogame controller
- Computer (ideally with bluetooth capabilities)
- ETC Lighting console or ETC Nomad
- Mayflash dolphin bar
- Lights
- Router
- Ethernet cable

• Optional

- iPad or other tablet
- WLED
- Pixel Tape
- Microcontroller

Step 1 – Download and Install all Software

Step 2 – Build Network

While you're waiting for your software to download, create your A/V Network. I used a TP-Link portable router, but any Wi-Fi router will do. First power on your router, and connect to it with your computer. The default password is usually near the barcode (if not, try admin). While you're in your network settings, write down your IP address. Next connect your router to the lighting console with an Ethernet cable.

2.1 Connecting your console to the network

Okay so this requires precision, so follow these steps exactly.

2.1.1

Power on Ion Console, and immediately exit, this will bring you to the shell, where you can configure network settings.

2.1.2.

Click settings then network and scroll down, write down the IP address and ensure subnet mask is set to 255.255.0.0 OSC and UDP Strings & OSC. Then hit accept and open EOS (depending on what you're using it'll say offline or online)

2.1.3 Now go to setup>settings>>system>show control>OSC. Turn on OSC RX and OSC TX, and type in your computer's IP address under "OSC UDP TX IP Address". Write down the "OSC UDP RX Port" and the "OSC UDP TX Port" numbers. While we're here, let's go to MIDI and check MSC Receive and MSC Transmit just for fun.

2.1.4.

Return to the home page on your console and go to Setup>Diagnostics. This will open up a page that'll be helpful when troubleshooting.

2.2 — Connecting TouchOSC to the console

Now open TouchOSC on your computer, and hit the button on the top that looks like two chains, then click OSC and enter the following data:

Host — IP address of your console

Send Port — UDP RX port number

Receive Port — UDP TX port number

Now here's the test, open up Eos_TouchOSC.touchosc in touchOSC and enter a number. If you've followed the steps correctly, the number you pressed will appear in the text box!! Congratulations, that was the hardest part. If nothing happens, double check all the numbers and ensure they're correct. I lost about 20 minutes trying to figure out what was wrong, and it turns out, it was just the subnet mask was off by a single digit, it's probably something small like that.

STEP 3 — Mapping your controller for EOS

For ease, I'm going to explain how to setup an XBOX controller, because regardless of what controller you end up using, you will have to do this. Now this all depends on what you want to do with your controllers, so I'll lay out a few options.

3.1 GO

Go to page 5 in your TouchOSC file, Double click, and then click the GO button.

Now on the sidebar on the left, scroll down to messages, and then hit the + button, then select gamepad.

Click the new message it will be brown and should say "GAMEPAD Button_A -> text" and pick whichever button you like on inputs. Let's stick with A for go, since it makes sense.

3.2. Stop/Back

Now we can do the same thing with the stop back button, but this time let's pick B

3.3 Fire specific cue

Now this one requires a bit more work, but it isn't hard.

3.3.1. Click the plus button on the top lefthand corner of the screen, and select button. Move it wherever you want

3.3.2 Now go to messages of that button, delete the MIDI one and add a gamepad, this time we'll select X.

3.3.3. Hit the plus button again, but this time, add an OSC message, now open up the new OSC Message and go to address.

3.3.4. Delete everything in the address field, and add a constant, then a textbook that says constant will appear below arguments. In that text box type /eos/cue/1/fire (1 can be replaced by any cue number.)

3.3.5. Now hit the play button at the top of the screen. And you're good to go

3.4 Fader control

Okay so now here's the most advanced part.

3.4.1. Go to fader page, and select a fader, now go to messages and add a gamepad message

3.4.2. Now just select stick_left_x this is the horizontal axis on the left thumbtack of the controller.

Now it could be as simple as that, but you may be wondering why the fader keeps returning to zero. If that doesn't bother you, stop here, things are about to get messy.

So, the fader is following exactly where the joystick is, and joysticks always return to the centre position. My solution, create a lock to lock the light in place.

3.4.3. Download my file Light Controller. Or follow these instructions **exactly**.

3.4.4. Add a fader, and two XY axis, and two buttons. This can be done in no particular order, but here's the instructions for each element

Button 1

1. Add two gamepad messages. The first:
 1. TRIGGER_RIGHT ->Tag
 1. Open gamepad, and in the row that says "Target, you will hit VALUE, and then select property, now go to the previous box and select tag
 2. BUTTON_STICK_LEFT- > x
 1. Open message, and change scale to 1 and 0
2. Under script, enter this:

```
function onReceiveGamepad(input, value, connections)
if value == 1 then
self.values.x = ((self.values.x -1) * -1)
```

```
end
end
```

```
function onValueChanged(value)
if(value == "x") then
if(self.values.x==1) then
-- root.children.gamepad.properties.lockX = true

else
-- root.children.gamepad.properties.lockX = false
end
end

end

end
```

Button 2

1. Repeat what you did for button 1, but instead enter this under script:

```
function onReceiveGamepad(input, value, connections)

if value == 1 then

self.values.x = ((self.values.x -1) * -1)
```

```
end
```

end

```
function onReceiveGamepad(input, value, connections)
```

```
  if value == 1 then
```

```
    self.values.x = ((self.values.x - 1) * -1)
```

```
  end
```

```
end
```

```
function onValueChanged(value)
```

```
  if(value == "x") then
```

```
    if(self.values.x==1) then
```

```
      --root.children.gamepad.properties.lockX = true
```

```
    else
```

```
      -- root.children.gamepad.properties.lockX = false
```

```
    end
```

```
  end
```

```
end
```

XY1

1. Create two Gamepad messages
 - A. Change STICK_LEFT_X → tag
 1. Change target to tag,
 2. Under the XY row change response from absolute to relative

- B. STICK_LEFT_Y → tag
 - 1. Repeat the previous steps, this time choosing STICK_LEFT_Y
- 2. Create two local messages.
 - A. LOCAL x → xy2 x
 - 1. Under target press the eye dropper and then click on xy2
 - 2. Select x for the target
 - B. LOCAL y → xy2 y
 - 1. Change the following values
 - 1. Trigger = Y - ANY
 - 2. Source = y - VALUE
 - 3. Select xy2 and select y for target

3. Enter the following under script:

```
function onReceiveGamepad(input, value, connections)
  if(input==0) then
    if(root:findByName("lock_x_toggle", true).values.x == 0) then
      self.values.x = value +0.5
    end
  end
end
```

```
if(input==1) then
  if(root:findByName("lock_y_toggle", true).values.x == 0) then
    self.values.y = (value -0.5) * -1
  end
end
end
```

```
root:findByName("lock_x_toggle", true)
```

XY2

- 1. Create one (or two) Local messages
 - A. Makes sure the values are
 - 1. Trigger = x - ANY
 - 2. Source = x - VALUE
 - 3. Target = x - [fader you want to control on the x axis]
 - 1. Use the eyedropper tool and click on the fader you want to modify
 - B. Make sure the values are
 - 1. Trigger = y - ANY
 - 2. Source = y - VALUE
 - 3. Target = x -[fader you want to control with the y axis]

Now you have two options to lock your fader in place, you can either hold the Right trigger to move it, and then it will lock upon release, or you can do the same with the joystick button. You can change this in step 1, these are just the two mappings I found the most intuitive for a moving light.

STEP 4 – Mapping your controller for Logic (or any sampler of your choice)

1. Create a button in touchOSC for each button on your controller, and assign a midi value to it

A. Note that touchOSC midi values are in their raw hexadecimal form, so you'll need to figure out how that translates into notes, I've attached a table below.

2. Open your sampler and match the notes to the ones you made in TouchOSC,

A. there's usually a quick remap button so you can quickly flash thru the controls

3. Drag the sounds you want to use onto the sampler, and ensure your start times are where you want them to be

4. Adjust the settings of the sample playback, some sounds you'll want to play all the way (one-shot) and other you may only want to play as it's being held down (classic)

A) To make a toggle, like I did with music, you have to set insert this script onto the desired button

```
function onReceiveGamepad(input, value, connections)
```

```
  if value == 1 then
```

```
    self.values.x = ((self.values.x -1) * -1)
```

```
  end
```

```
end
```

5. Run TouchOSC and you should be good to go.

A. Note there's a plethora of interesting things you can do with this if you get creative with synths and effects. You can use your joysticks and pressure sensitive triggers as dials

for settings in any plug-in, and because they're pressure sensitive, they can be used for aftertouch.

Additional Resources

Setting ups LED fixtures in your own home

<https://kno.wled.ge/basics/compatible-hardware/>

MIDI Decimal to Notes

<http://www.somascape.org/midi/basic/notes.html>

MIDI Decimal to Note

Note	Octave										
	-2	-1	0	1	2	3	4	5	6	7	8
C	0	12	24	36	48	60	72	84	96	108	120
C# / Db	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
D# / Eb	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F# / Gb	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G# / Ab	8	20	32	44	56	68	80	92	104	116	
A	9	21	33	45	57	69	81	93	105	117	
A# / Bb	10	22	34	46	58	70	82	94	106	118	—
B	11	23	35	47	59	71	83	95	107	119	

Chart Retrieved from <http://www.somascape.org/midi/basic/notes.html>