DESN22848 | Winter 2023
Interaction Design: Media, Motion, & the Body

TOUCH

DESIGN FOR LIVING SYSTEMS

Hanh Trang Do

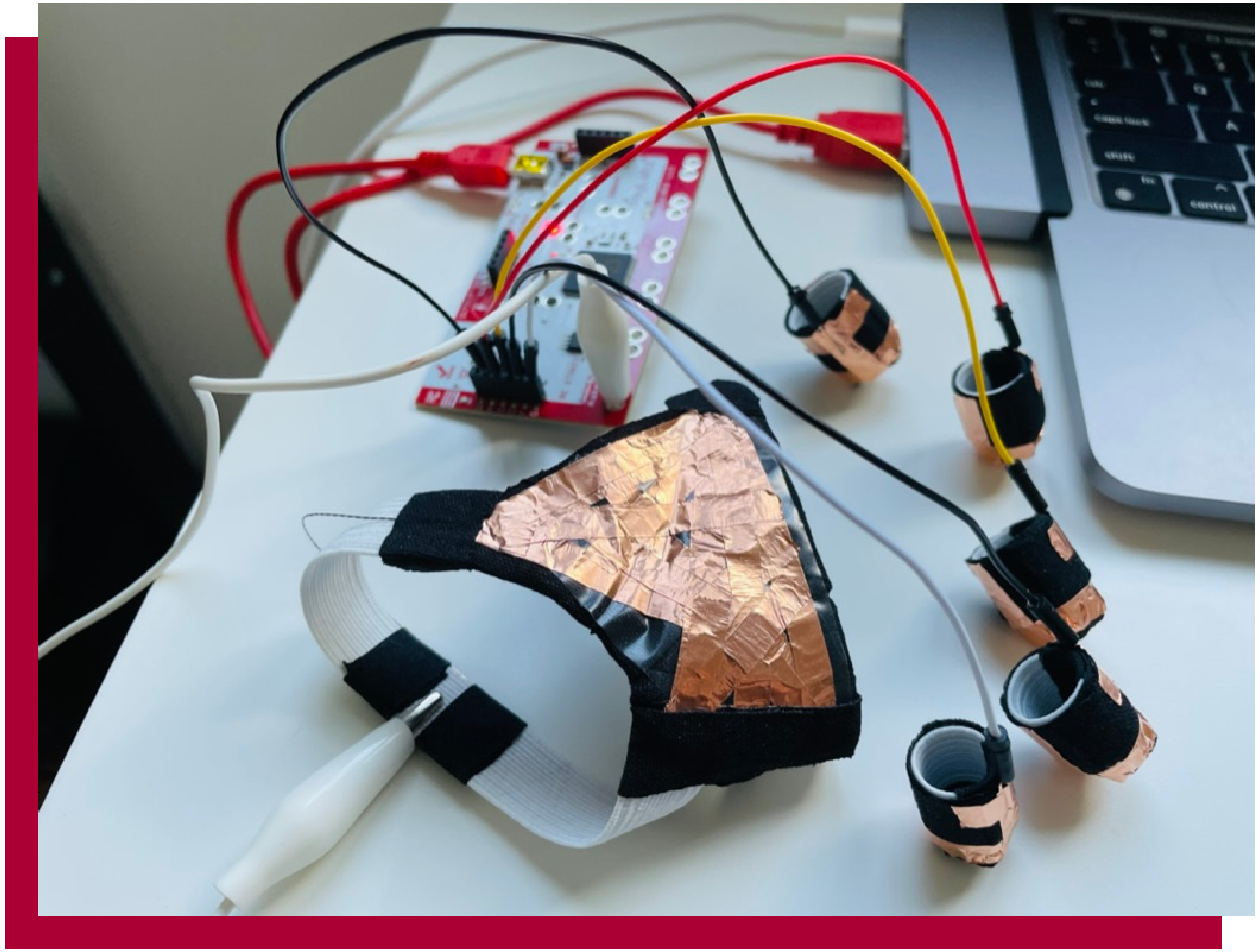
Submitted to Steve Hudak
April 17th, 2023

CONTENT

DESIGN STATEMENT

Design for Living Systems

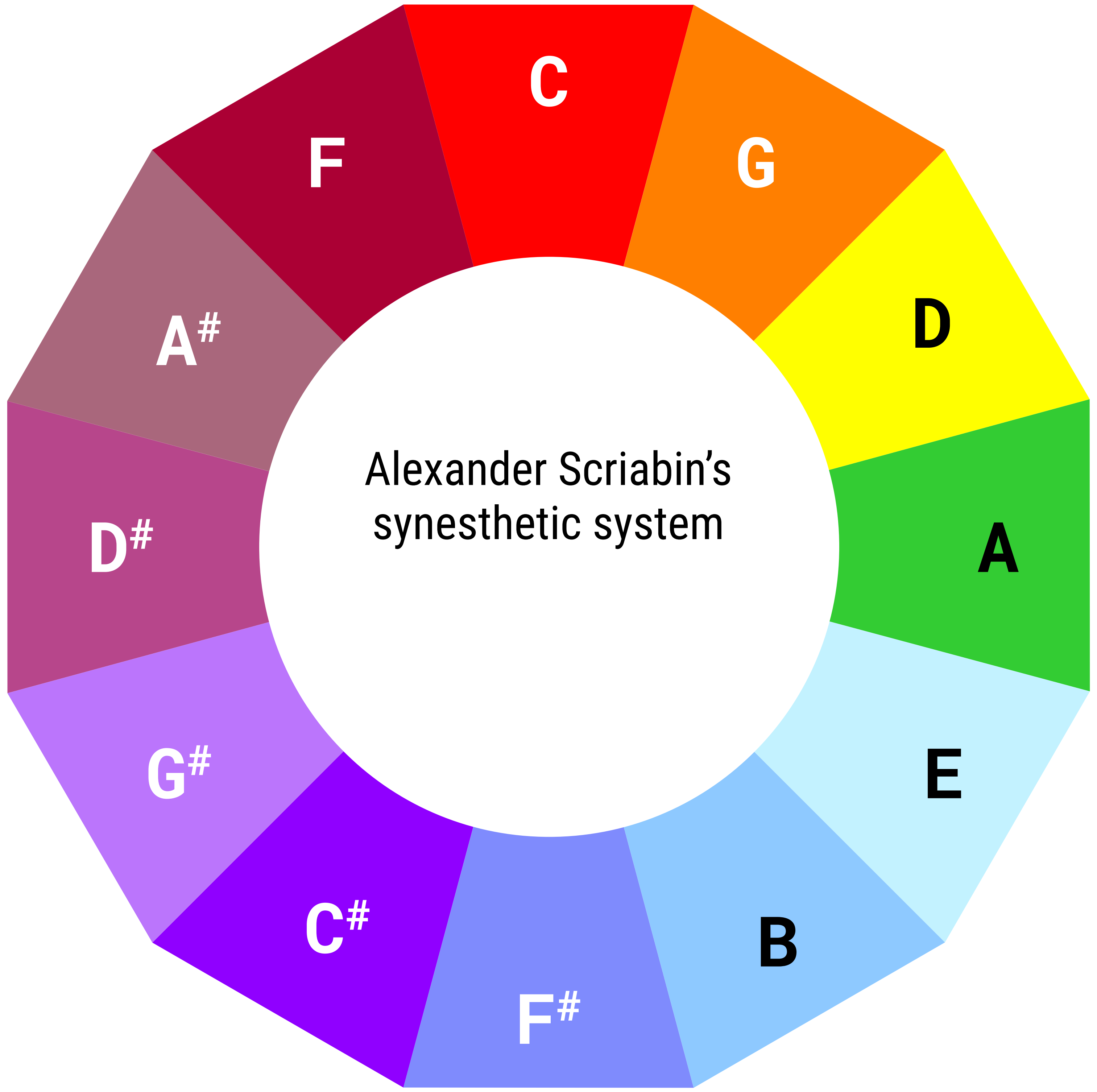
This project examines the connections between analog physical movements and WebGL visual output. Drawing from previous projects, conductive materials circuits and sound-based interactivity were integrated, combining physical computing with WebGL visualization through the use of the Makey Makey set, conductive materials, and p5 JavaScript library. The design process moved from divergent investigation to finalizing and implementing the physical and digital integration. The formstorming stage explored how visual and auditory elements can be interacted with using regular keystrokes; the provided tutorial videos and accompanying code serve as guidelines for this process. Physical prototypes serving as the analog input were created from common objects and conductive materials; a range of prototypes and interactions were experimented with. Once the direction for finalization was identified, further research into sound and colour representations were conducted, specifically sound-to-colour synesthesia and music chords. The final design draws from experimentation to connect hand movements to the experience of playing a musical instrument, achieved through visual and sound outputs triggered by interactions contained to one hand.



RESEARCH

Tutorial, Sourced Code, & Inspiration

Research were conducted into p5 visual and sound outputs, the basics of music notes and guitar chords, and sound-to-colour synesthesia. Alexander Scriabin's synesthetic system (the diagram below) were used for colour coding throughout the project. Diagrams detailing specific interactions and music notes are included in later pages of this document. Other JavaScript codes, not included in the sources cited on the right, were taken from weekly tutorials.



Chromesthesia, or sound-to-colour synesthesia

- Benteh. "Synesthesia: hearing colour, seeing sound." *Visual Squirrels*, 2014, https://visualsquirrels.net/visualisation/synesthesia-hearing-colour-seeing-sound/
- "Clavier à lumières." *Wikipedia*, https://en.wikipedia.org/wiki/Clavier_%C3%A0_lumi%C3%A8res

Music notes and guitar chords

- "All Guitar Chords." *All Guitar Chords*, https://www.all-guitar-chords.com/chords/index
- "Guitar tuning by ear." *Tuner-Online.com*, https://tuner-online.com/by-ear/
- "Five Basic Piano Chords You Can Start Playing Right Now." *Piano Secrets*, https://pianosecrets.com/basic-piano-chords/
- Marty. "Guitar Chord Theory." *The Acoustic Guitarist*, 2020, https://theacousticguitarist.com/guitar-chord-theory/
- Marty. "How To Learn All Notes On Guitar." *The Acoustic Guitarist*, 2019, https://theacousticguitarist.com/all-notes-on-guitar/
- "Midi Note Numbers and Center Frequencies." *Inspired Acoustics*, https://computermusicresource.com/midikeys.html
- O'Connor, Celeste. "9 Basic Guitar Chords Beginners Need To Know! With Photos To Help." *Musician's HQ*, https://musicianshq.com/9-basic-guitar-chords-beginners-need-to-know-with-photos-to-help/
- "Piano Basics: Notes, Octaves, Scales and Tones." *La Touche Musicale*, https://latouchemusicale.com/en/piano-basics/
- "Piano To Guitar String Notes." *GuitarSix*, https://guitarsix.com/blog/lessons/piano-to-guitar-string-notes/215/1/
- Play Guitars. "Mastering The 15 Most Important Guitar Chords For Beginners." *Play Guitars*, 2022, https://play-guitars.com/guitar-chords-for-beginners/

P5 Code

- "JavaScript Key Code." *Toptal,* https://www.toptal.com/developers/keycode
- monniqian. "Piano." *p5.js Web Editor*, https://editor.p5js.org/monniqian/sketches/TwckaaeGl
- mrbombmusic. "Keyboard Piano." *p5.js Web Editor*, https://editor.p5js.org/mrbombmusic/sketches/ryeLfZTd-

Sourced images used in this document were taken from Adobe Stock.

RESEARCH

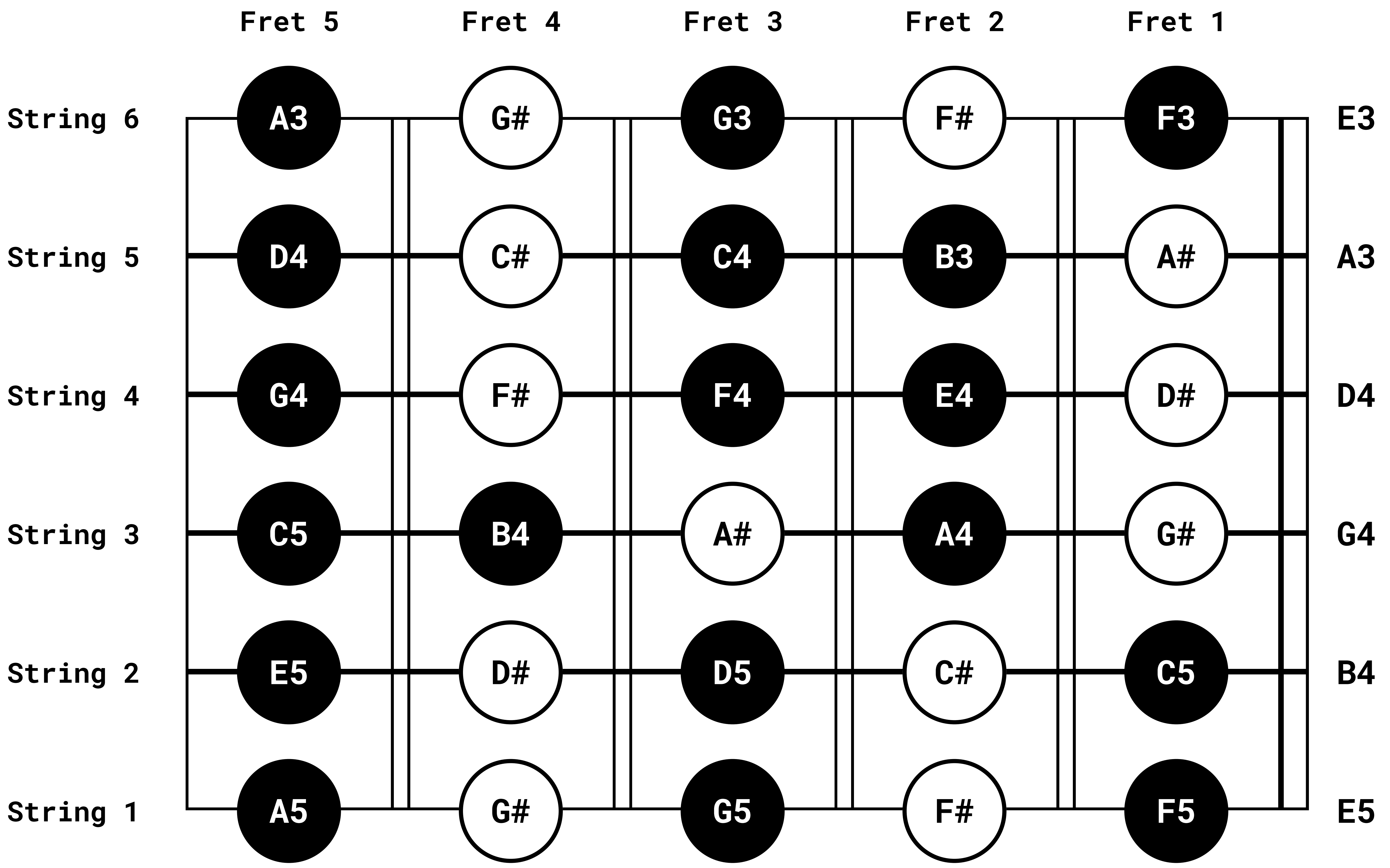
Guitar Chords in relation to Piano Octaves

The standard chord chart is turned 180° clockwise, resembling what people would see when facing the guitar player. The notation follows the ABC notation of C major scale as applied to a standard 88-key piano keyboard. The notation on the leftmost also follows the standard 6 string guitar tuning notes (i.e., the first string plays the E note on the 3rd octave). Filled circles indicate regular notes, and outlined circles indicate sharp notes. For example, pressing the 5th string at the 3rd fret would play the C4 note when strummed. C4 (middle C) is...

- The C note in the 4th octave
- The 4th C key in a standard 88-key piano
- Note number 60 in MIDI notation

Other notes stand for different keys, octaves, and MIDI values. Below are all notes included in the right diagram and their equivalence in MIDI (following the C4 convention):

- E3 = 52
- F3 = 53
- G3 = 55
- A3 = 57
- B3 = 59
- C4 = 60
- D4 = 62
- E4 = 64
- F4 = 65
- G4 = 67
- A4 = 69
- B4 = 71
- C5 = 72
- D5 = 74
- E5 = 76
- F5 = 77
- G5 = 79
- A5 = 81
- B5 = 83
- C# = 61
- D# = 63
- F# = 66
- G# = 68
- A# = 70



	Fret 5	Fret 4	Fret 3	Fret 2	Fret 1	
String 6	A3	G#	G3	F#	F3	E3
String 5	D4	C#	C4	B3	A#	A3
String 4	G4	F#	F4	E4	D#	D4
String 3	C5	B4	A#	A4	G#	G4
String 2	E5	D#	D5	C#	C5	B4
String 1	A5	G#	G5	F#	F5	E5

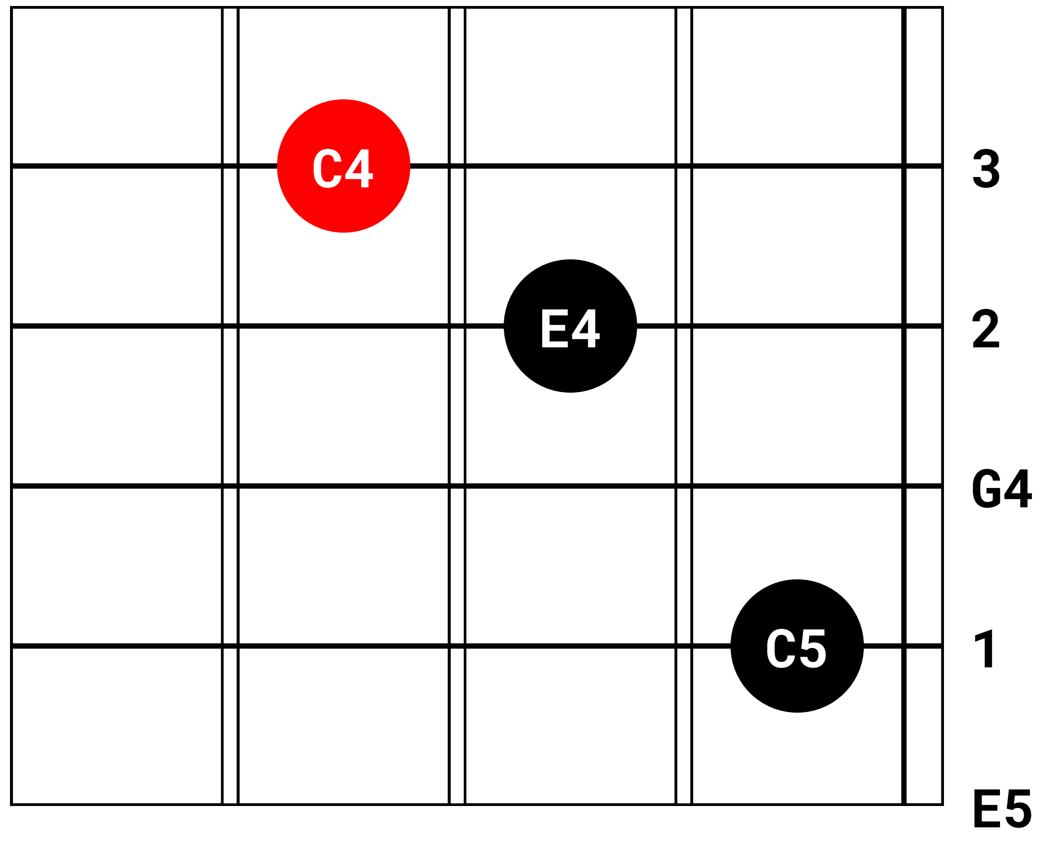
RESEARCH

Guitar Chords in relation to Piano Octaves – Major Chords

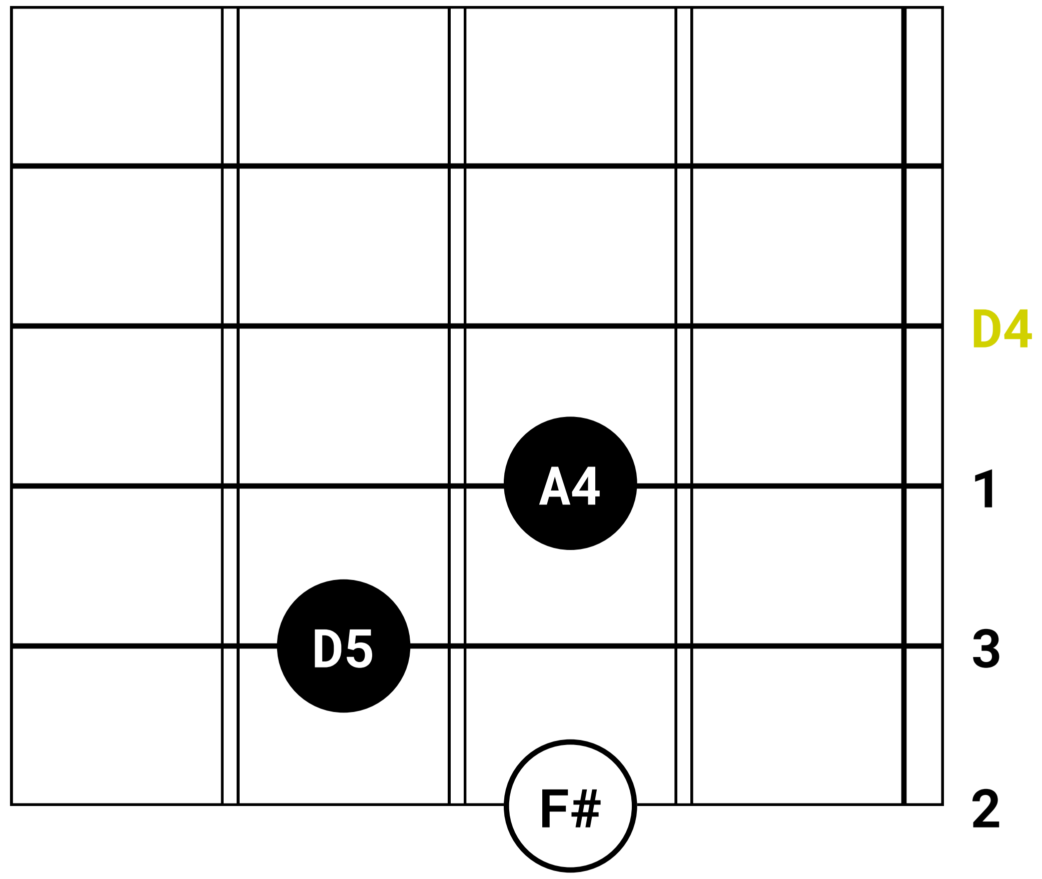
The diagrams demonstrate how the major chords can be played, except for B major. The note on the string indicates which string and fret the finger is on; standard root notes are coloured. Strings without a note or number on the right are not played. The numbers on the side indicate which finger is supposed to press on the string:

- 1 = index finger
- 2 = middle finger
- 3 = ring finger
- 4 = little finger

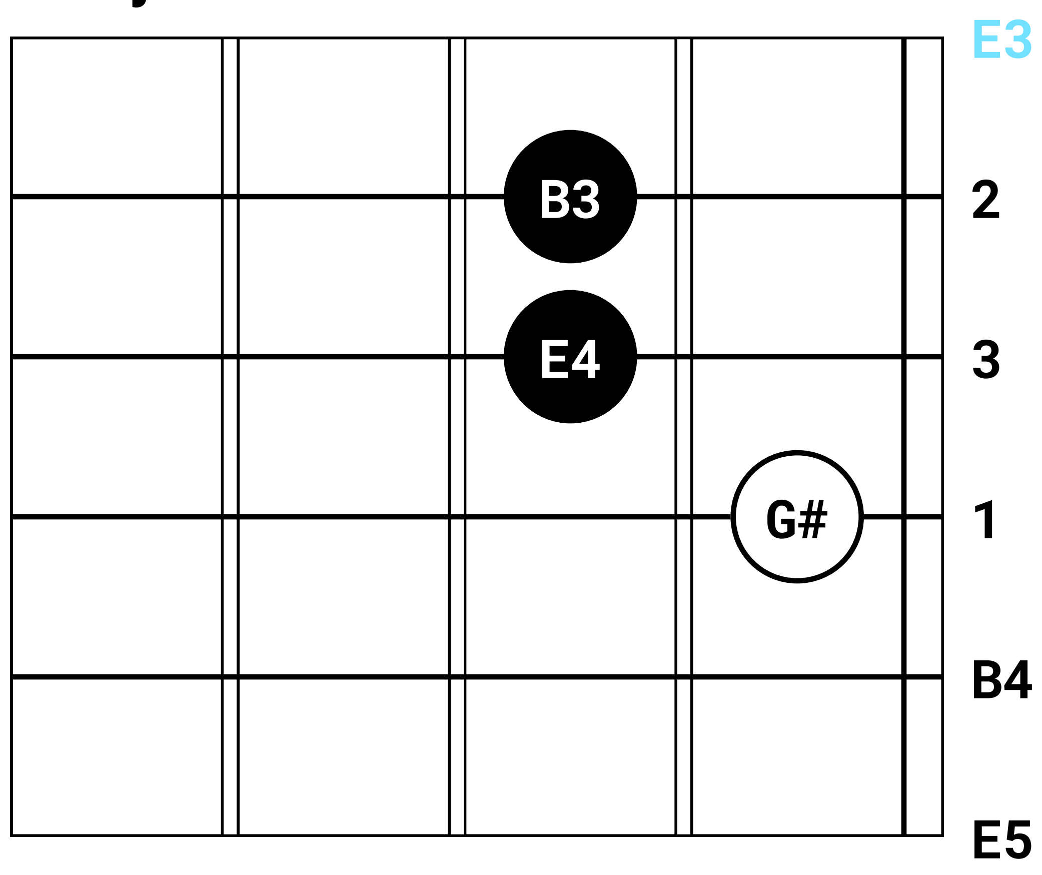
C major



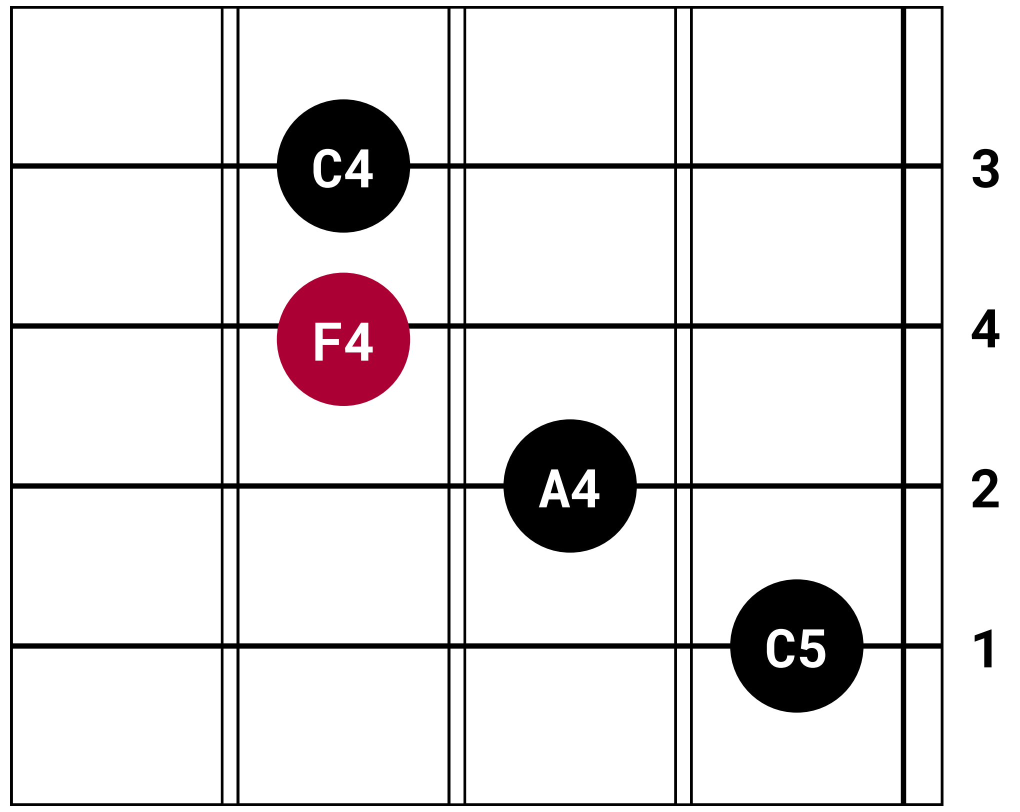
D major



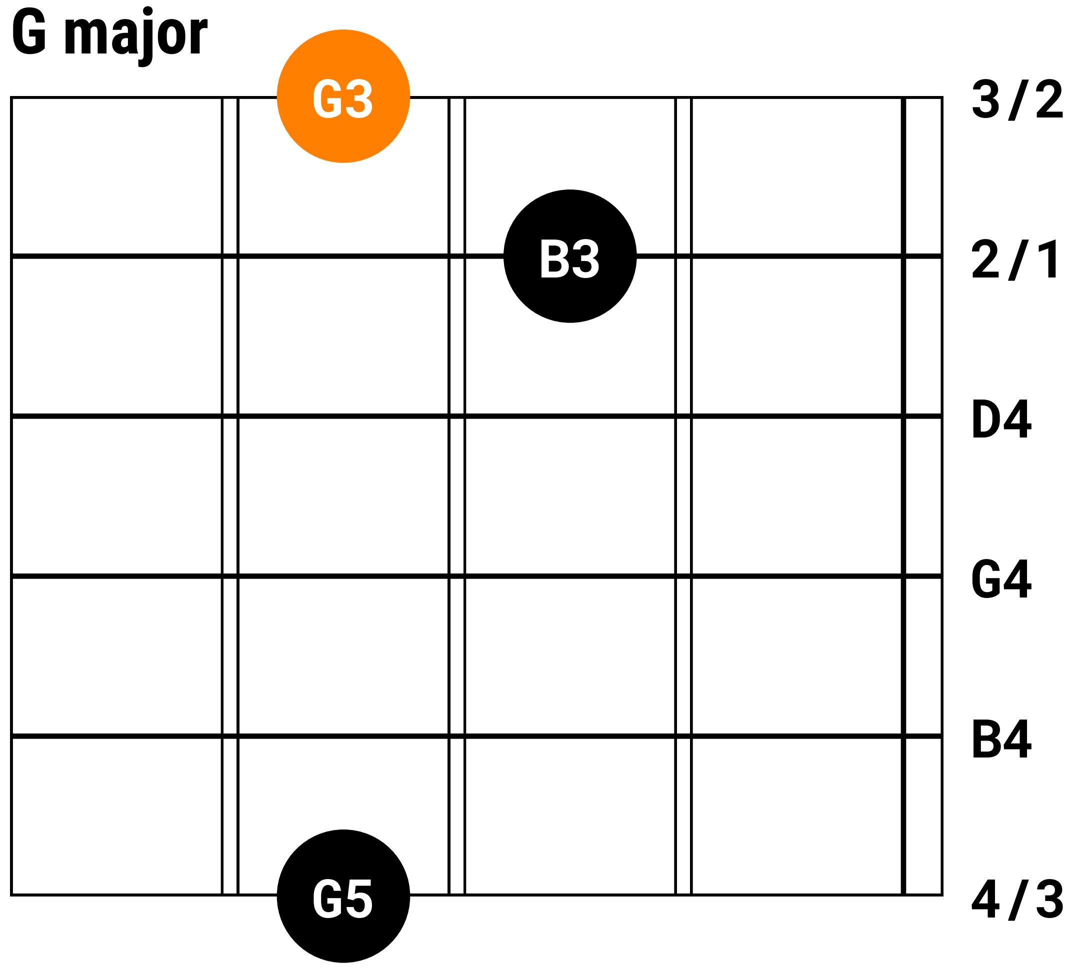
E major



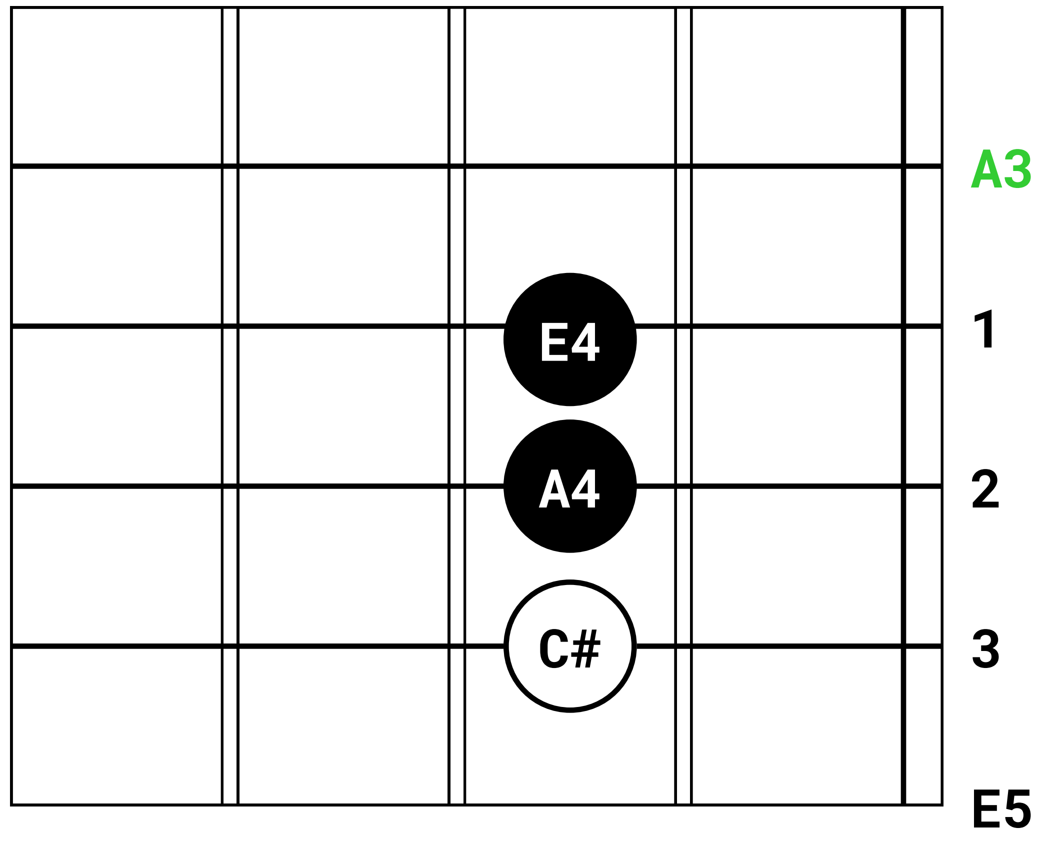
F major



G major



A major



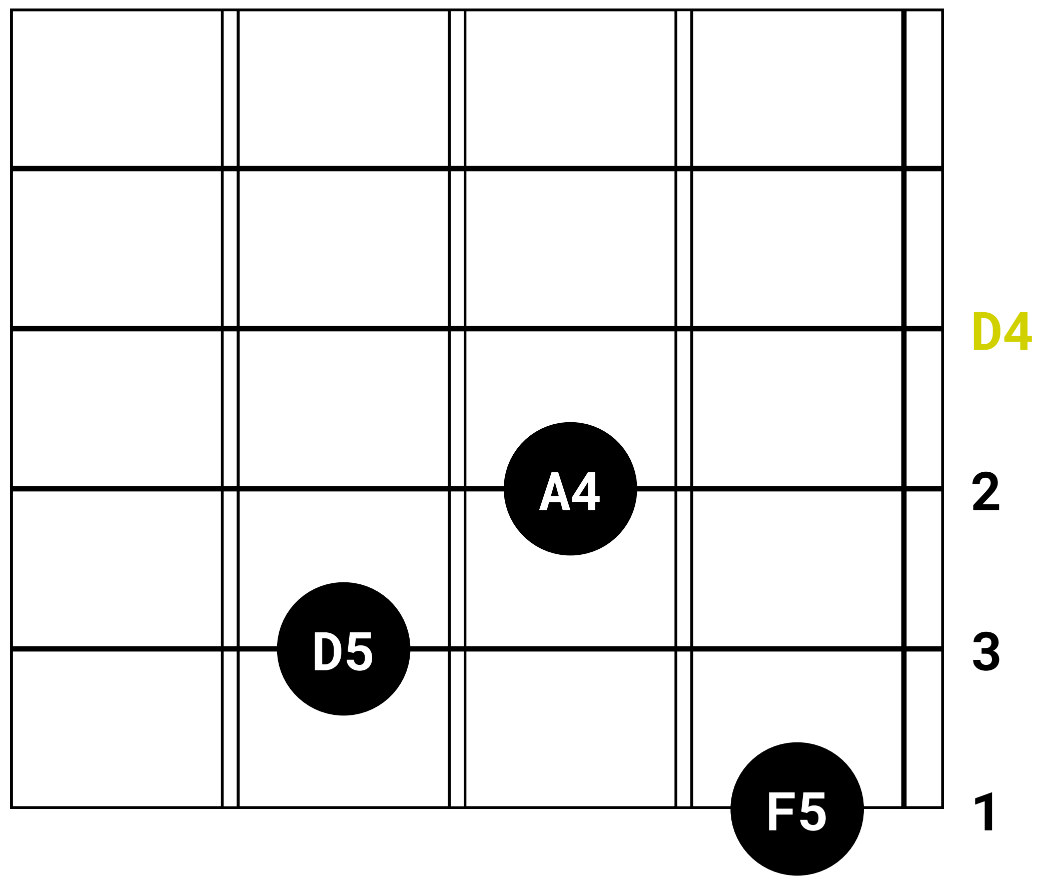
RESEARCH

Guitar Chords in relation to Piano Octaves – Minor Chords

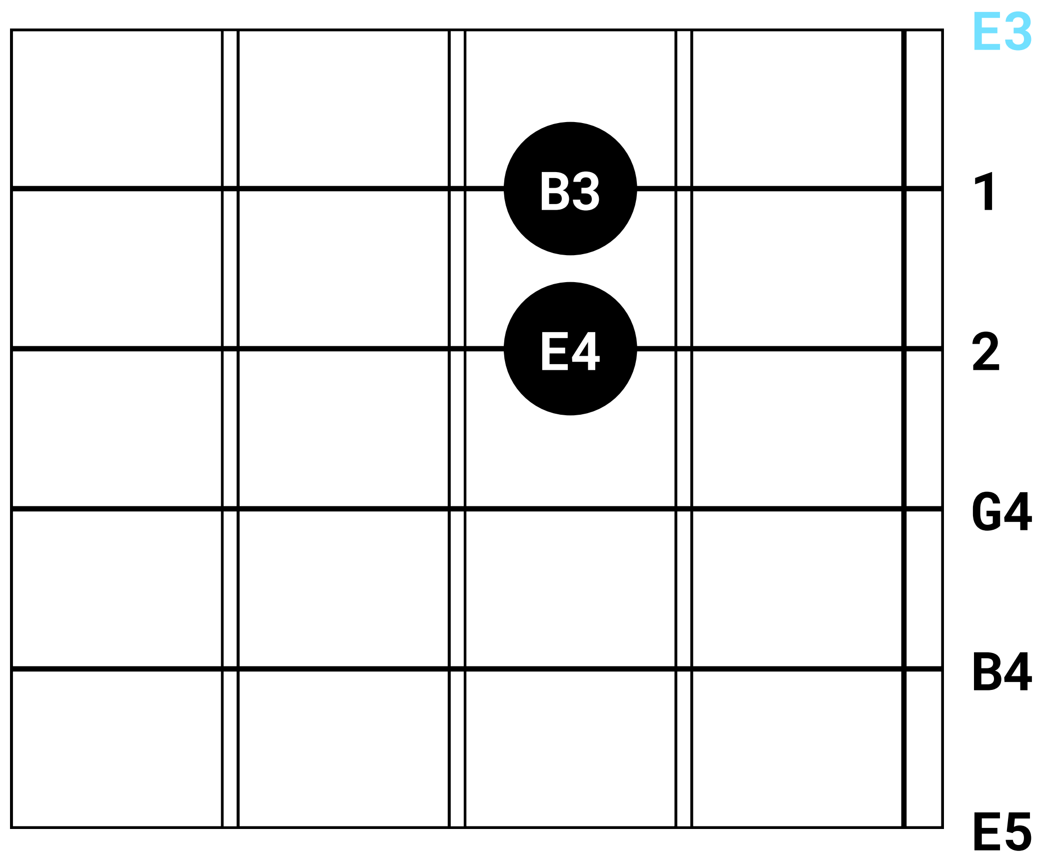
The diagrams demonstrate how 3 common minor chords can be played. The note on the string indicates which string and fret the finger is on; standard root notes are coloured. Strings without a note or number on the right are not played. The numbers on the side indicate which finger is supposed to press on the string:

- 1 = index finger
- 2 = middle finger
- 3 = ring finger
- 4 = little finger

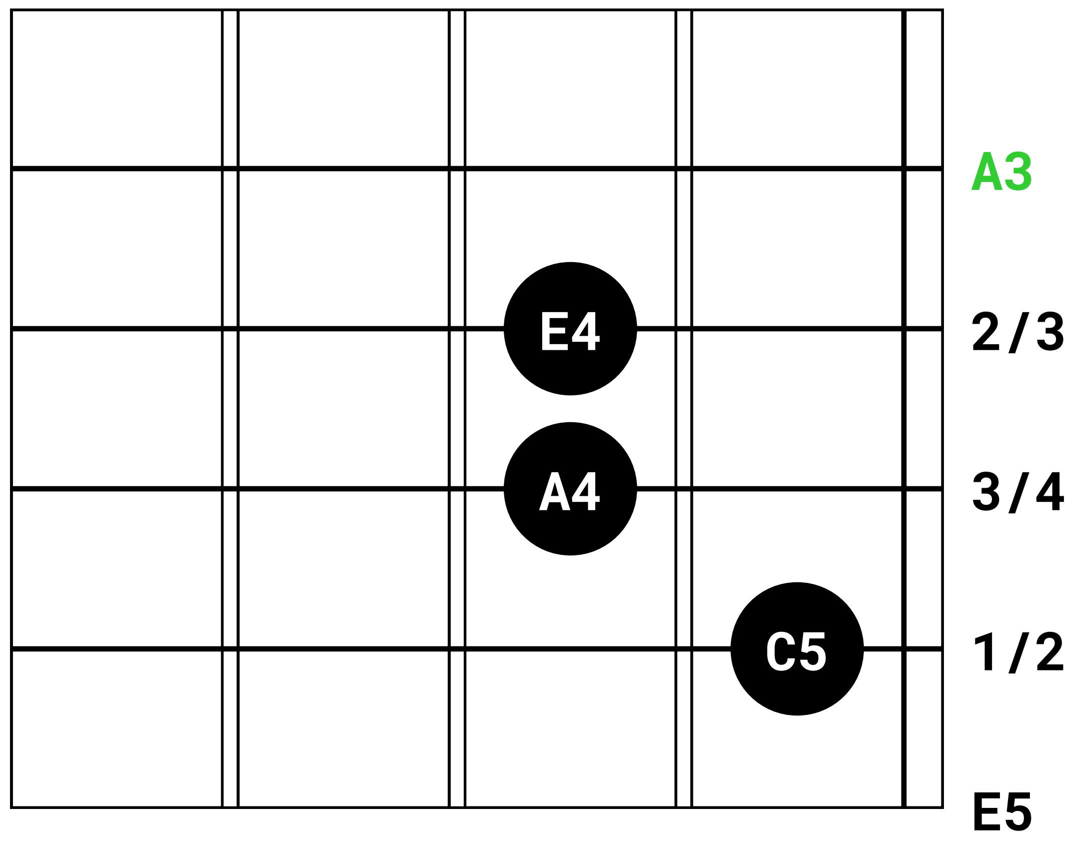
D minor



E minor



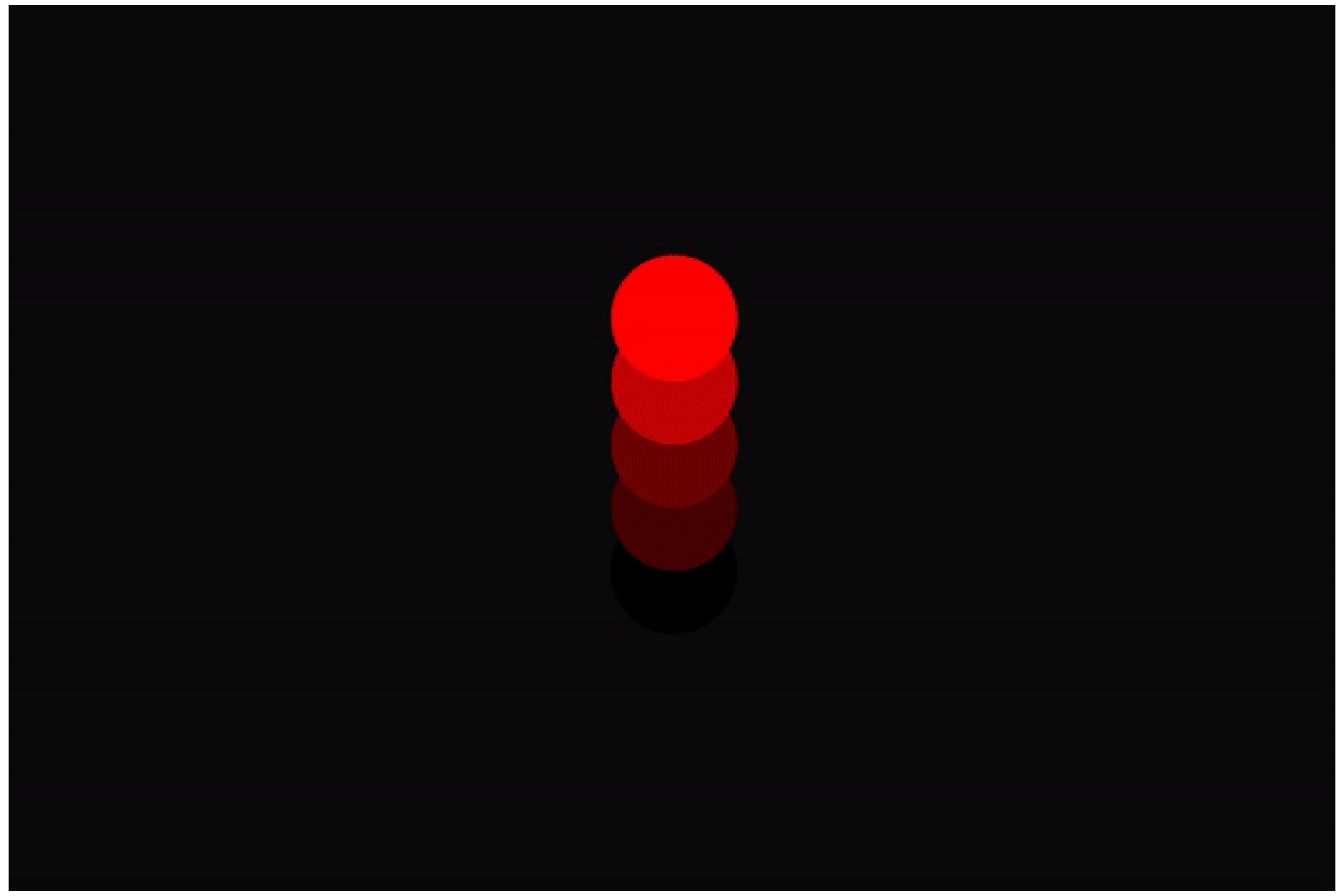
A minor

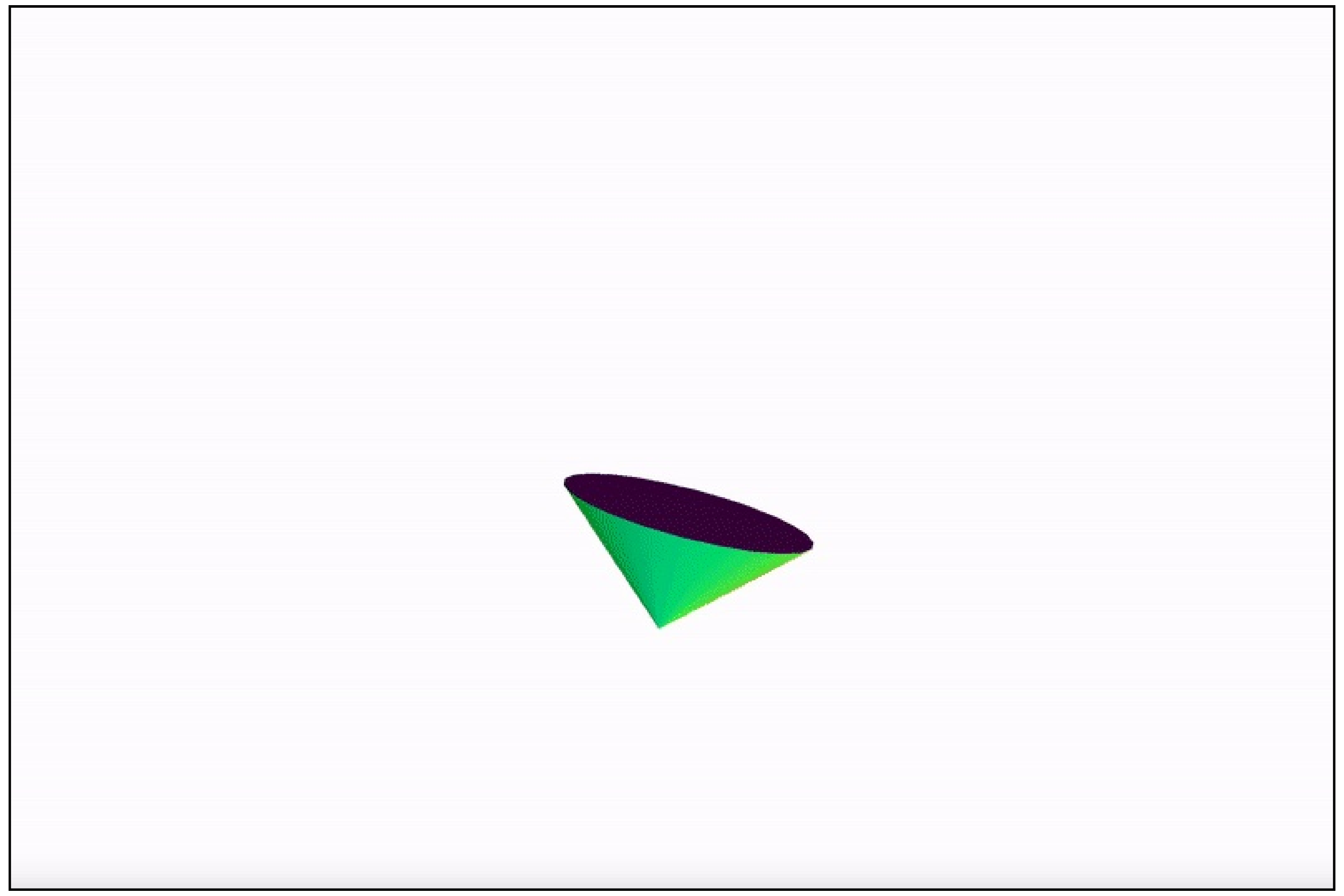


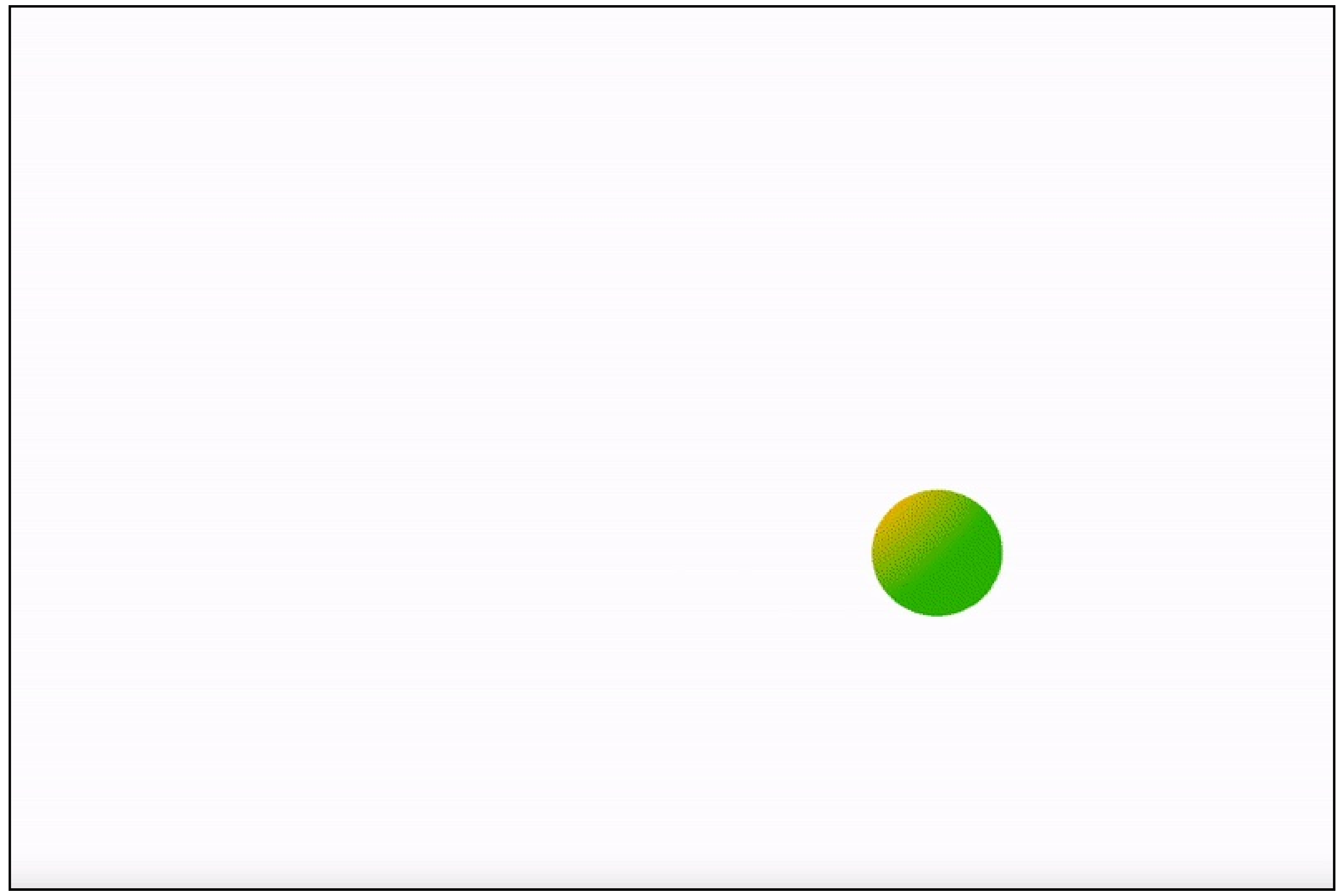
FORMSTORMING

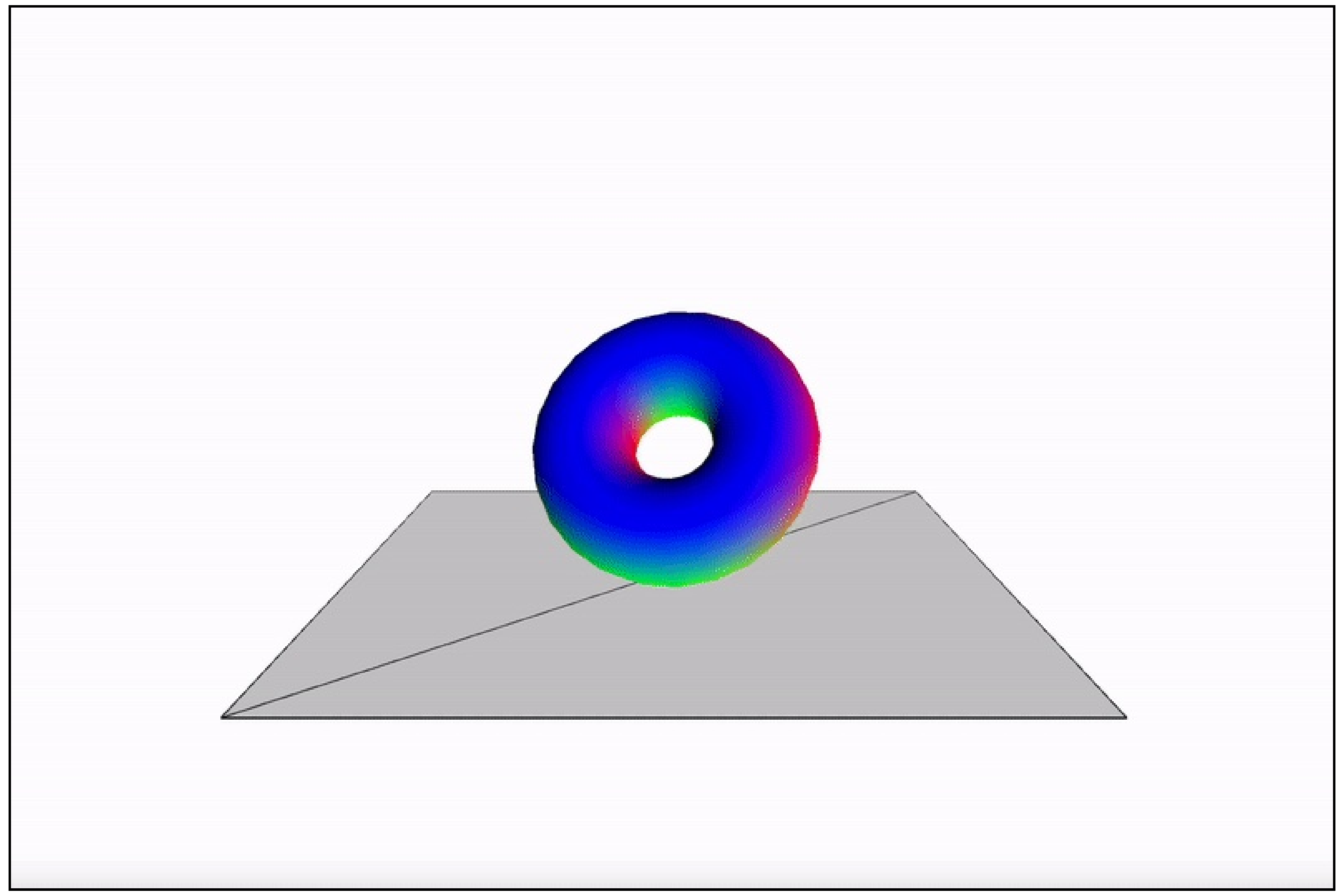
P5 Keyboard Interaction

The p5 JavaScript library allows onscreen visual elements to be control once the code has been run. Keystrokes can be used to control a digital object, whether by changing the direction it is moving in or rotating a 3D object (*week 9, image 18*). These controls can be done in increments, i.e. each press move the object by a fraction, or the object keeps moving until the key is released. The background can also be adjusted to create special effects for each movement (*week 9, image 2*). The direction and rotation controls can be combined to simulate a realistic control of a 3D object; for example, a ball rolling in the directions of the arrow keys (*week 9, image 25*). The keystroke-triggered functions can also control the camera to simulate movement around elements (*week 9, image 21*).



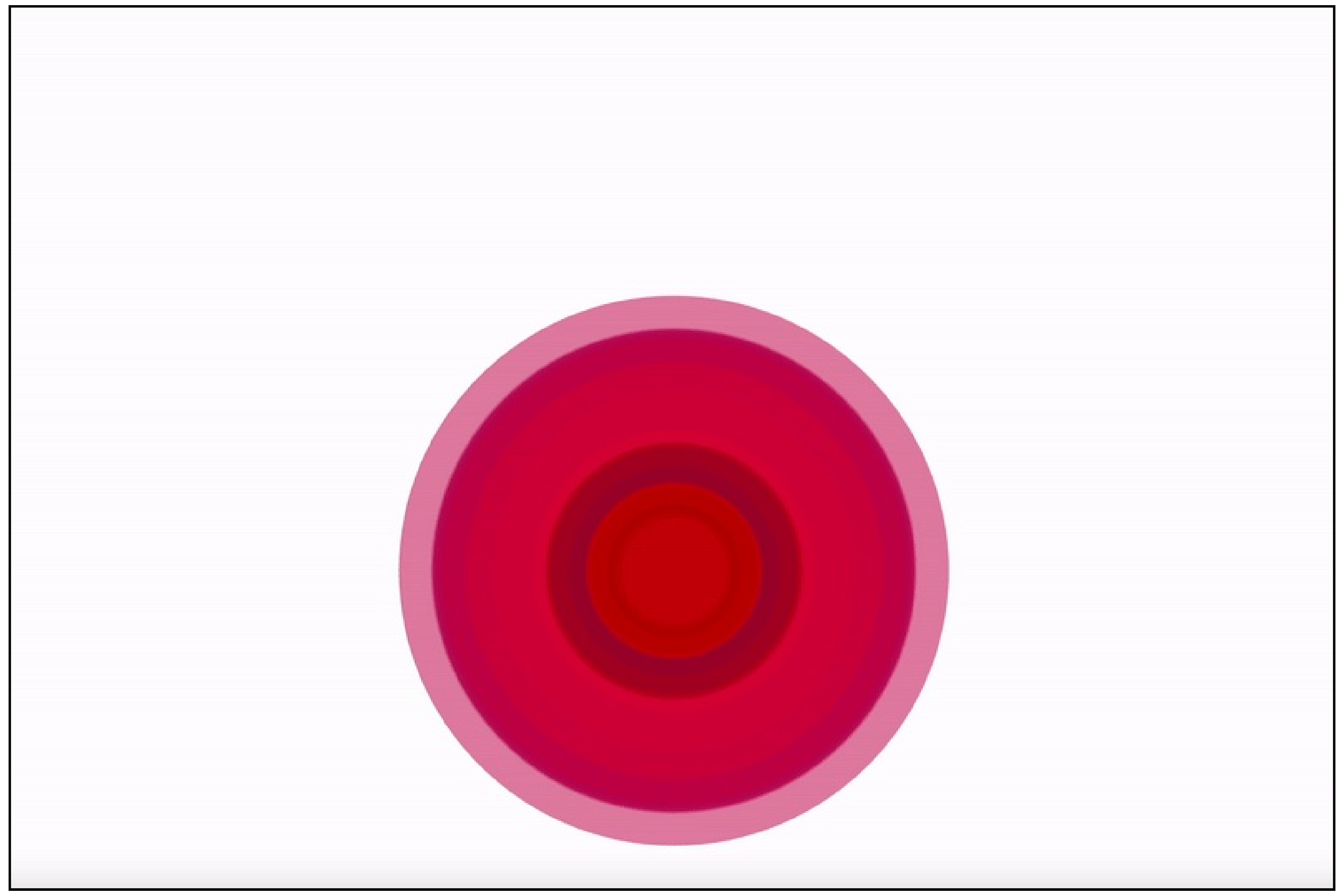




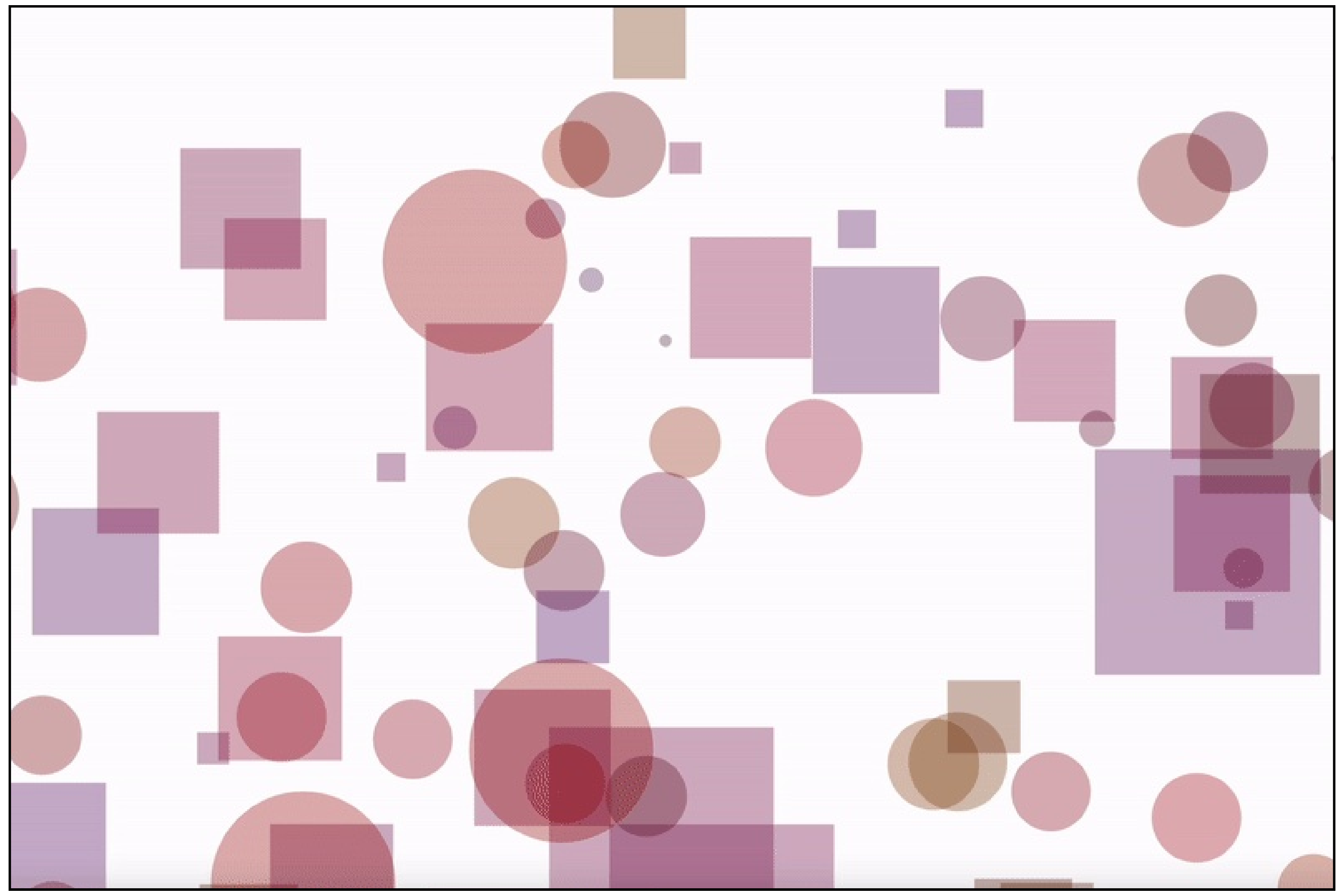


FORMSTORMING

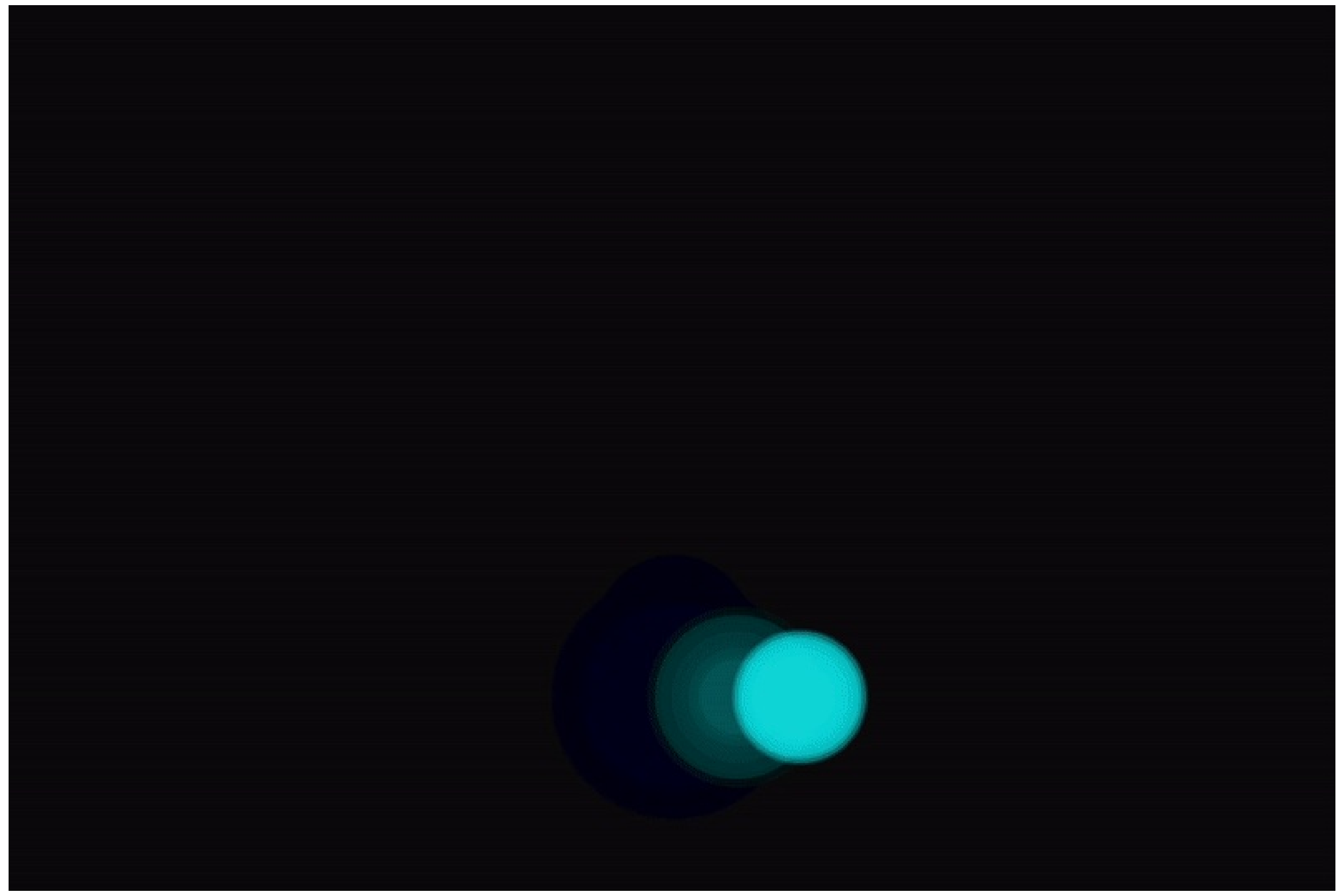
P5 Keyboard Interaction



The previous project explored the oscillator as an audio source. Aside from the oscillator, this project examined the polysynth as a mean of generating music notes. Similarly, MIDI numbers are used to play the corresponding notes; for example, number 60 plays the middle C (do). Using the code provided alongside the weekly tutorial, visualization of said polysynth was explored (*week 9, image 5*).



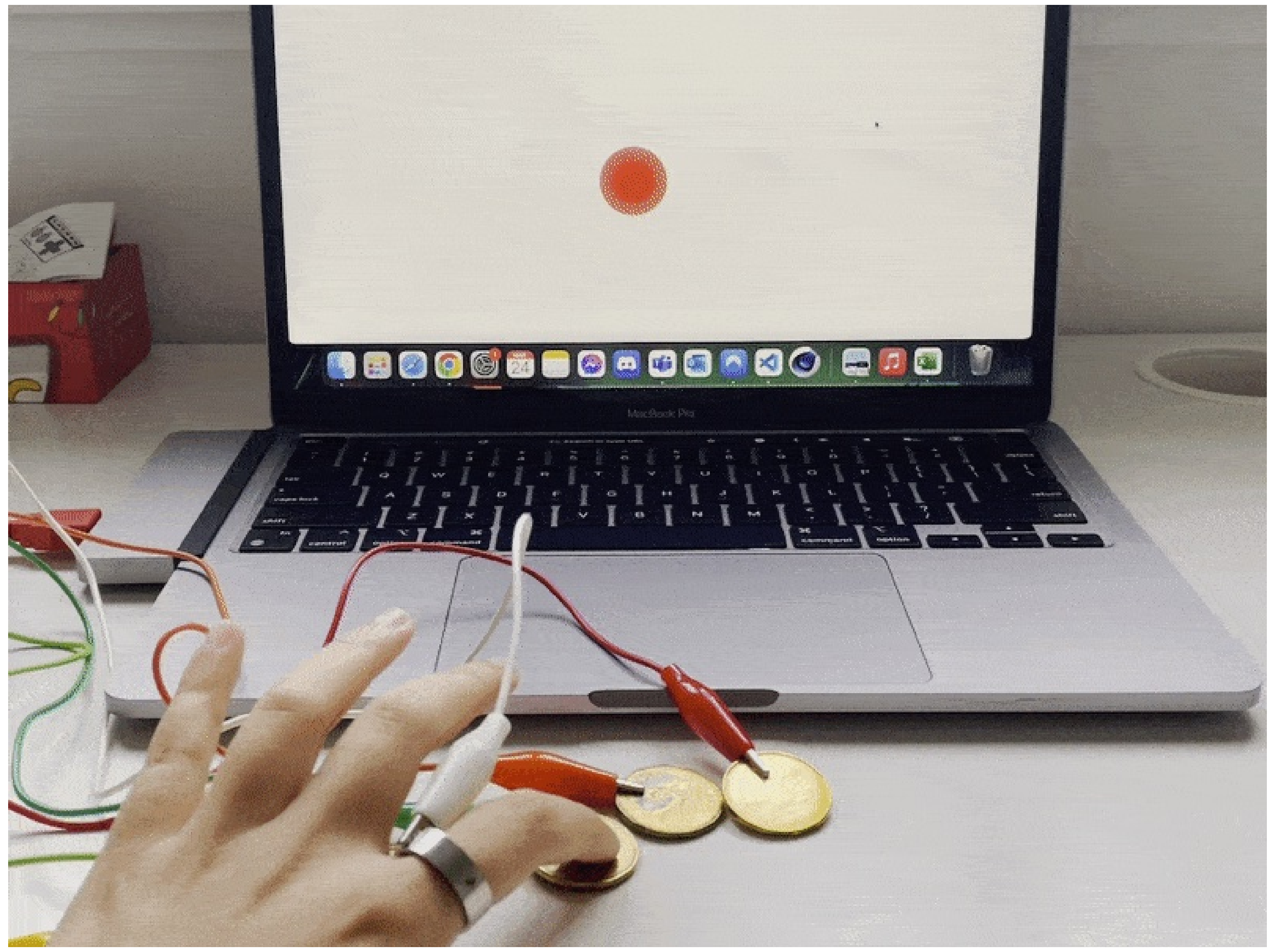
Since each keystroke changed the note being played, said keystroke may change the visualization to a predetermined colour, or change its shape. Iterations in the weekly activity show shapes being randomly generated around the screen, their colours and shapes dependent on the key being pressed (*week 9, image 8*). This can be interpreted as the note determining the shapes and colours.



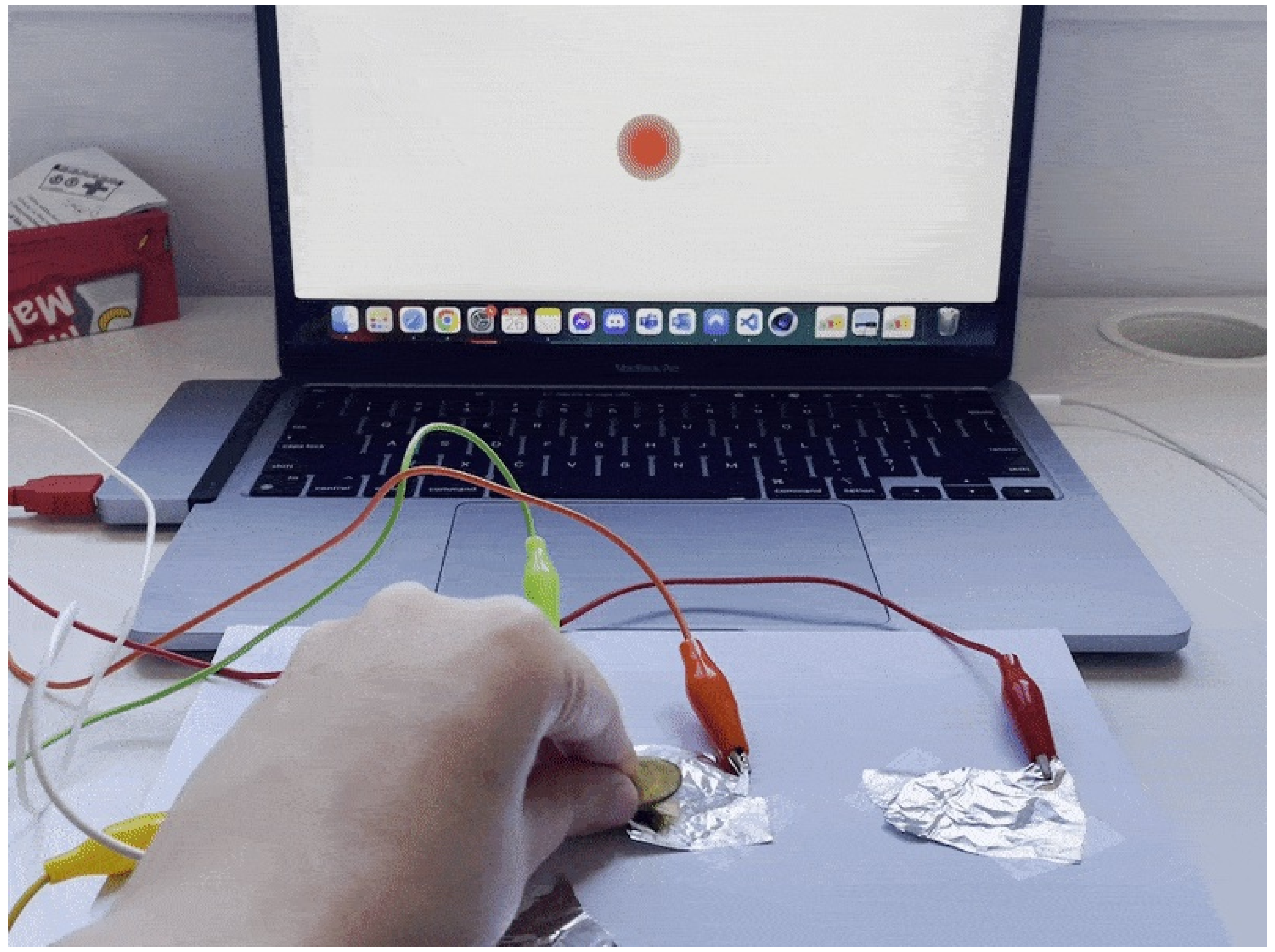
The keystroke controls code from previous experimentation is integrated to control the polysynth visualization (*week 9, image 16*). Each arrow key determined the direction and the note being played. This can be used as the basis of a game, in which the sound changes (in volume or pitch) when the player gets closer to their target, or the changing notes serving as the only mean of direction.

FORMSTORMING

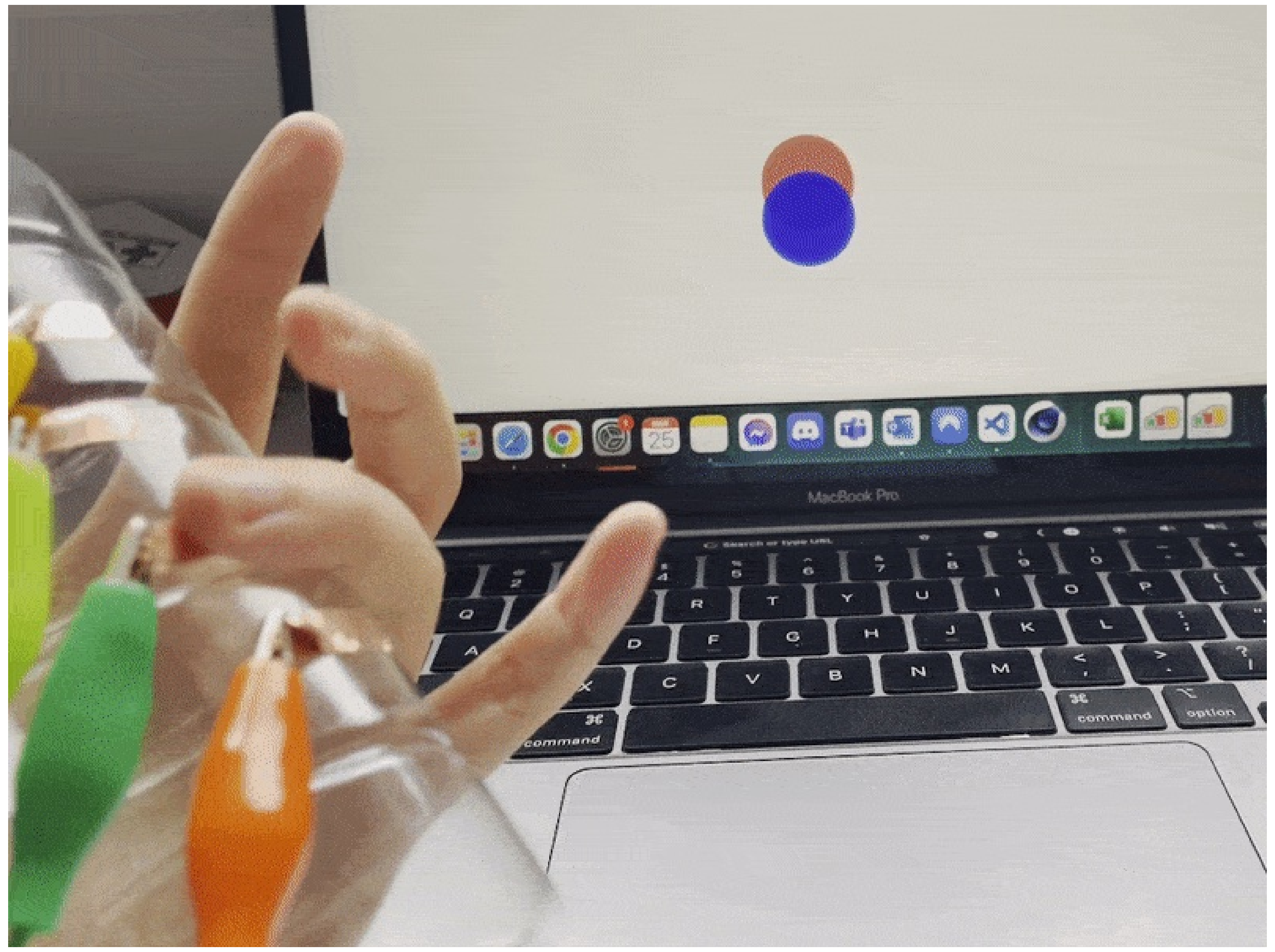
Makey Makey Interaction



Random objects were collected and used to experiment with the physical component of this project. Drawing from a first year project, coins were connected to the Makey Makey board with alligator clips. The alligator clip connected to the ground was clipped onto a ring; whoever wears the ring can interact with the coins to trigger WebGL output (*week 10, image 1*). This prototype allows for quick experimentation with different objects as the analog input.



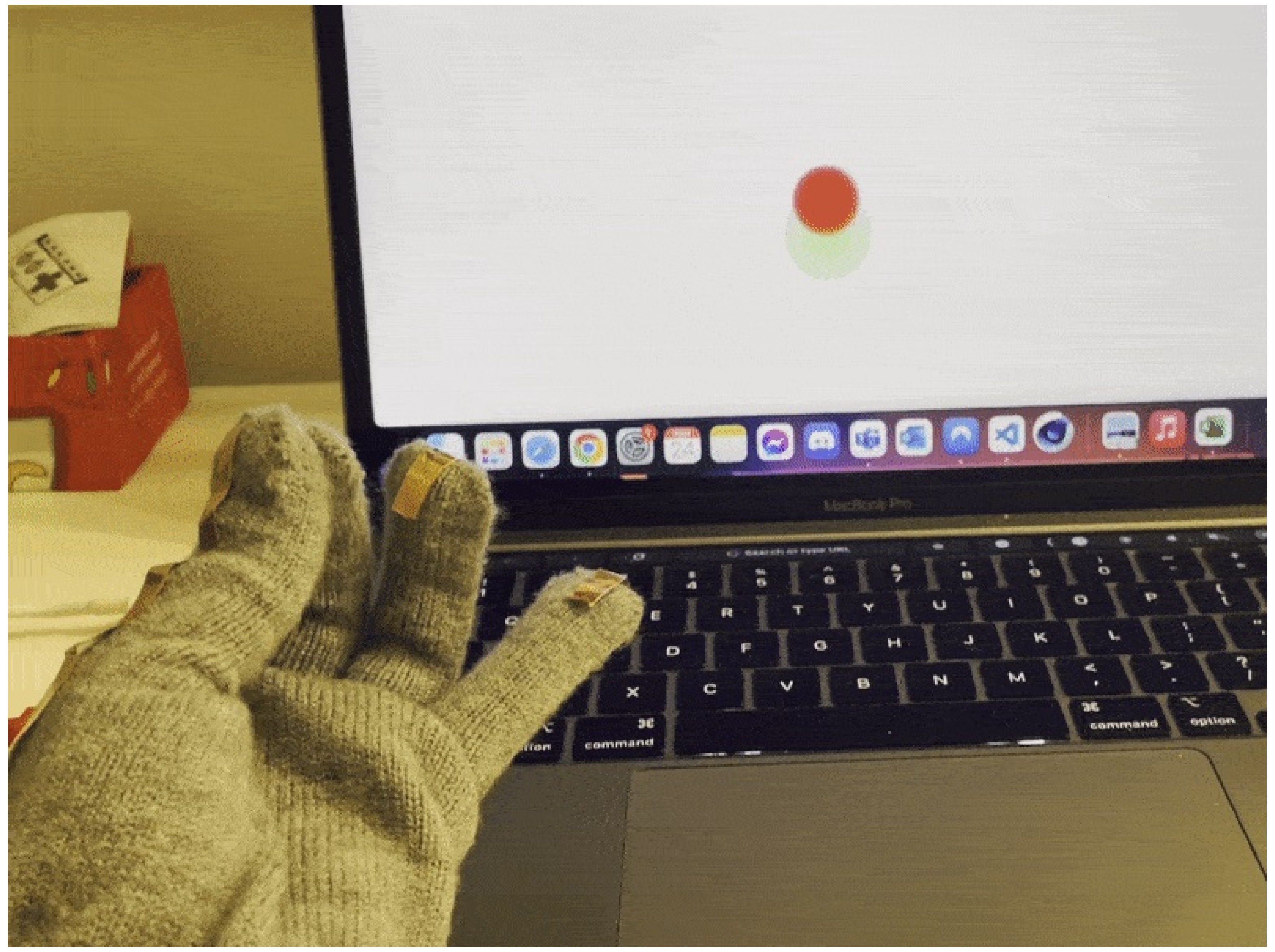
Strips of copper tape or aluminium foil can be taped to a surface and a grounded object can be used to interact with them (*week 10, image 25*). This is a simple but effective approach to a custom game controller. The formstorming images show a coin or a can being grounded and interacting with said surface; however, non-conductive objects can be covered in aluminium foil or copper tape and used to interact with said surface.



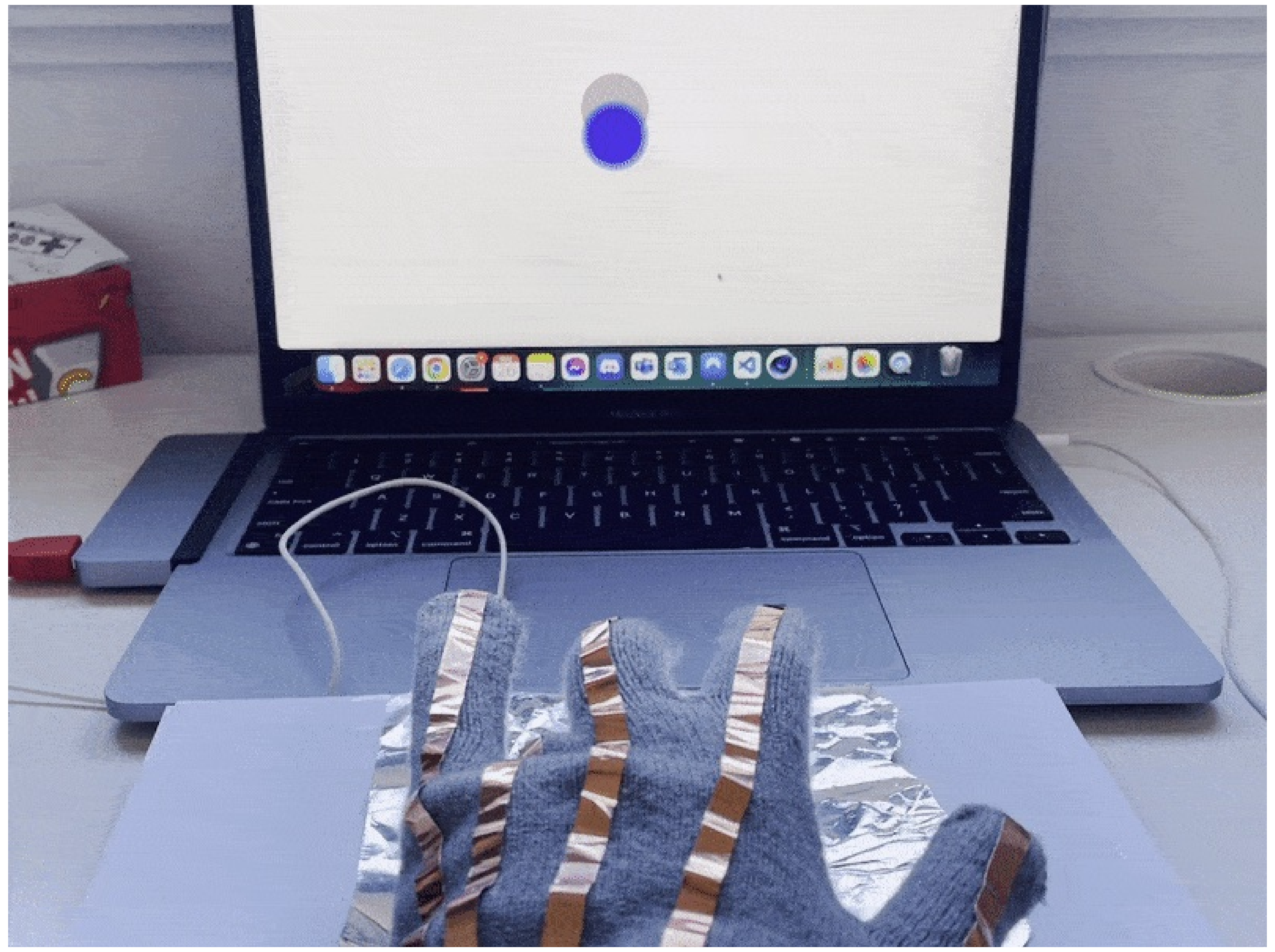
Alternatively, the user can be grounded and interact with different keystroke. Image 16 from week 10 formstorming shows alligator clips taped to a bottle and the user executing different functions by touching the corresponding tapes. This resembles either a game control or playing an instrument. If it is used as the physical component, there needs to be a better arrangement of interactive parts.

FORMSTORMING

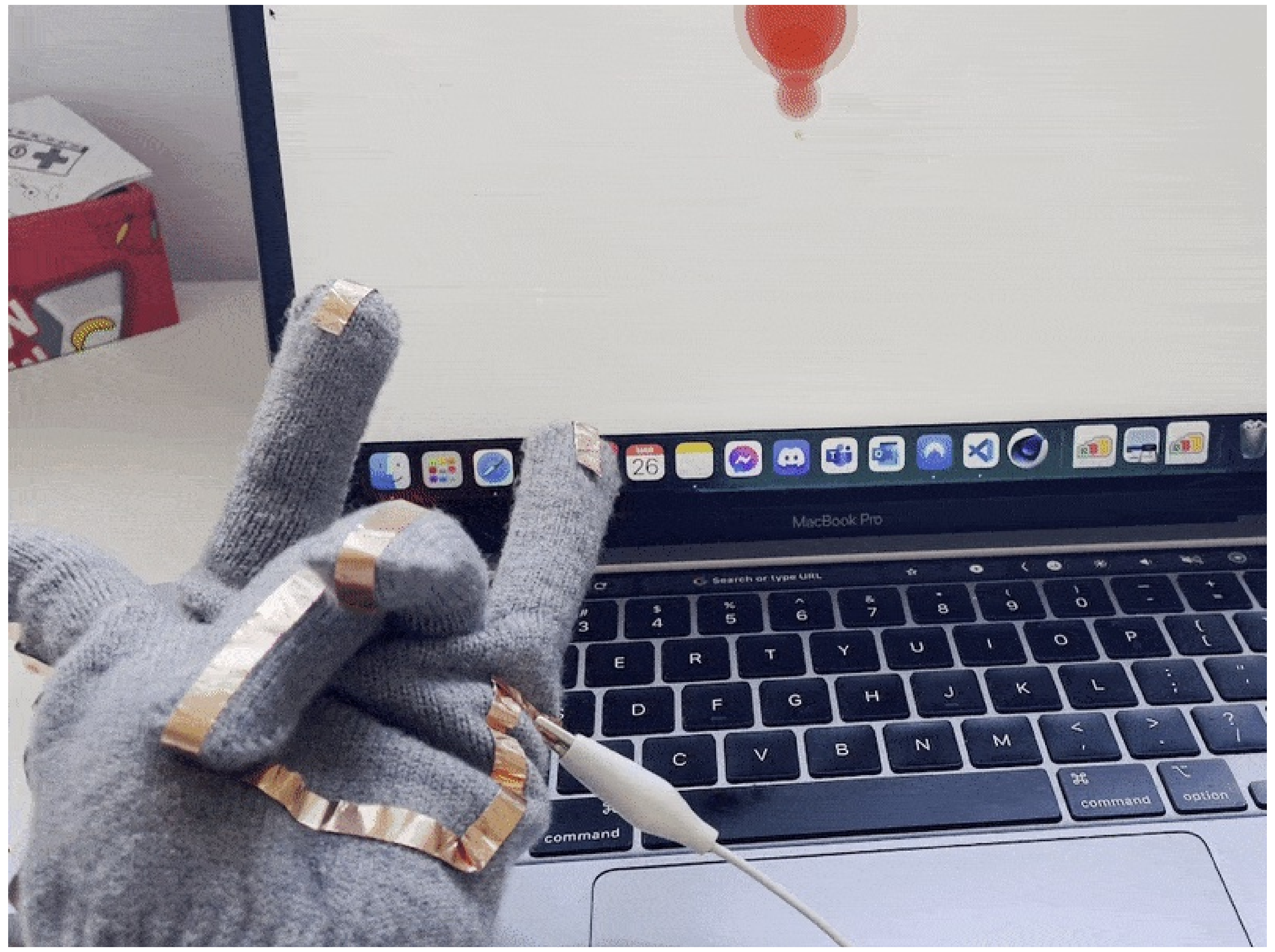
Makey Makey Interaction



A glove was used to further prototype hand interactions. Strips of copper tape were stuck onto the glove and connected to the Makey Makey board. Initially, the thumb is grounded and each of the other fingers was connected to a separate keystroke. Touching the thumb to a finger executes the corresponding function (*week 10, image 6*). While this was effective, the number of functions was limited.



To utilize all five keystrokes available (one for each finger), alternative interactions were experimented with. Each finger was connected to a separate keystroke and interacted with different conductive objects: a coin (*week 10, image 7*), a surface covered in aluminium foil (*week 10, image 12*), or a soda can (*week 10, image 25*). Nonconductive objects can be covered in aluminium foil to be interacted with.



Another placement was experimented with by taping directly onto the hand. The appropriate tape arrangement was then applied onto the glove, having the palm grounded and each finger connected to a keystroke. Touching a finger to a palm execute the function (*week 10, image 22*). Overall, containing the interactions to a hand seems to have more potential for further exploration.

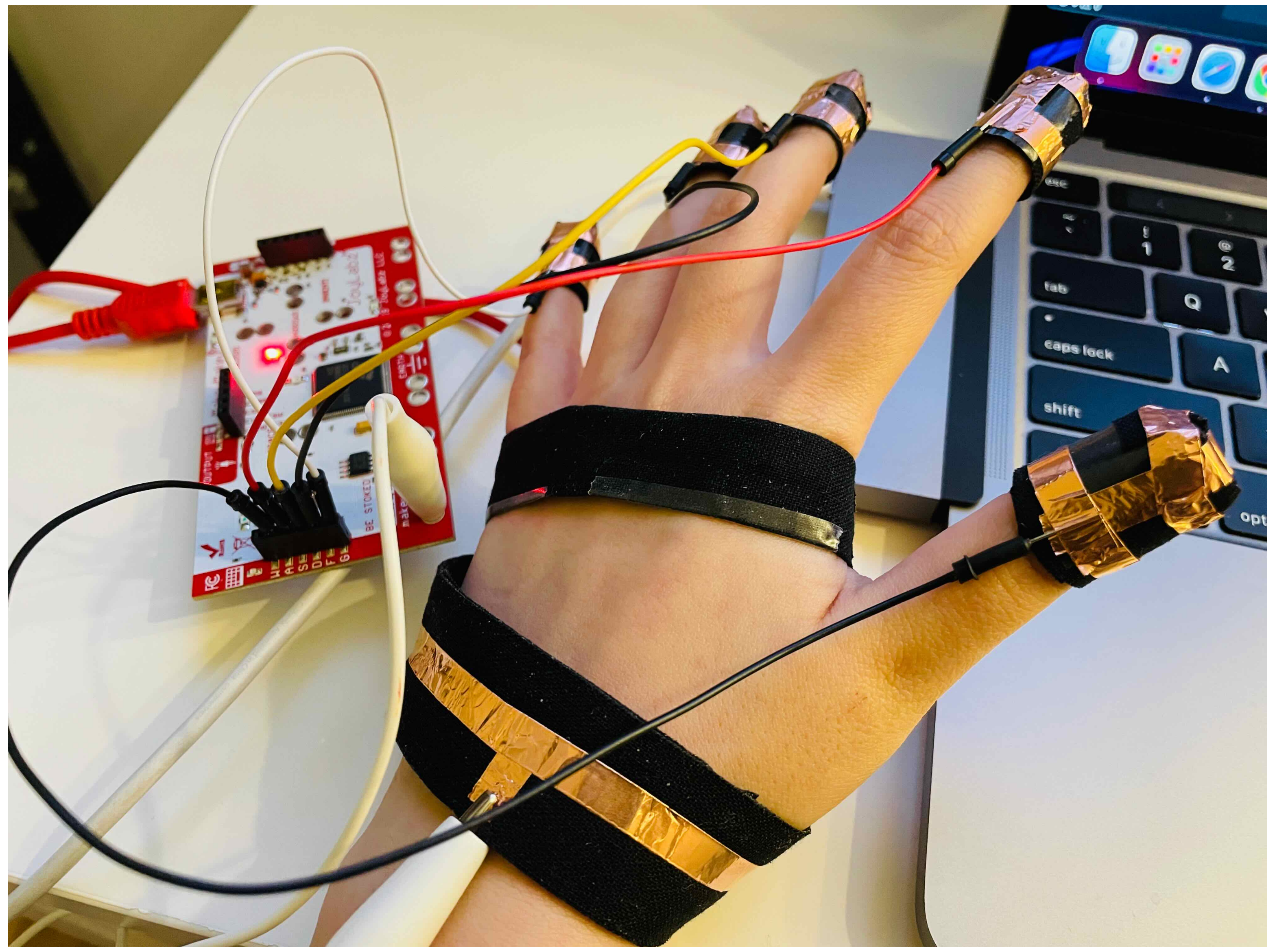
FORMSTORMING

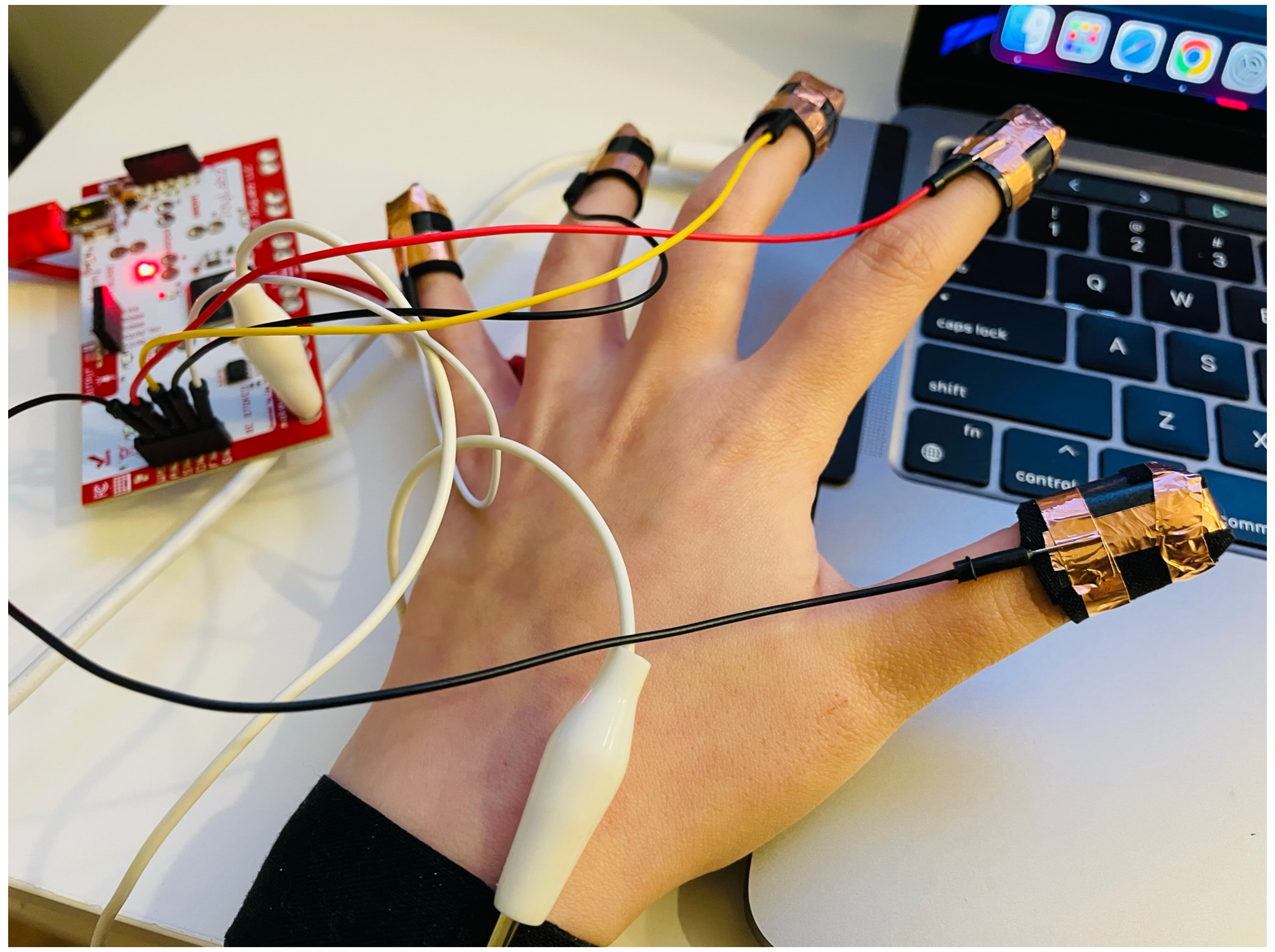
Integrated P5/Makey Prototype

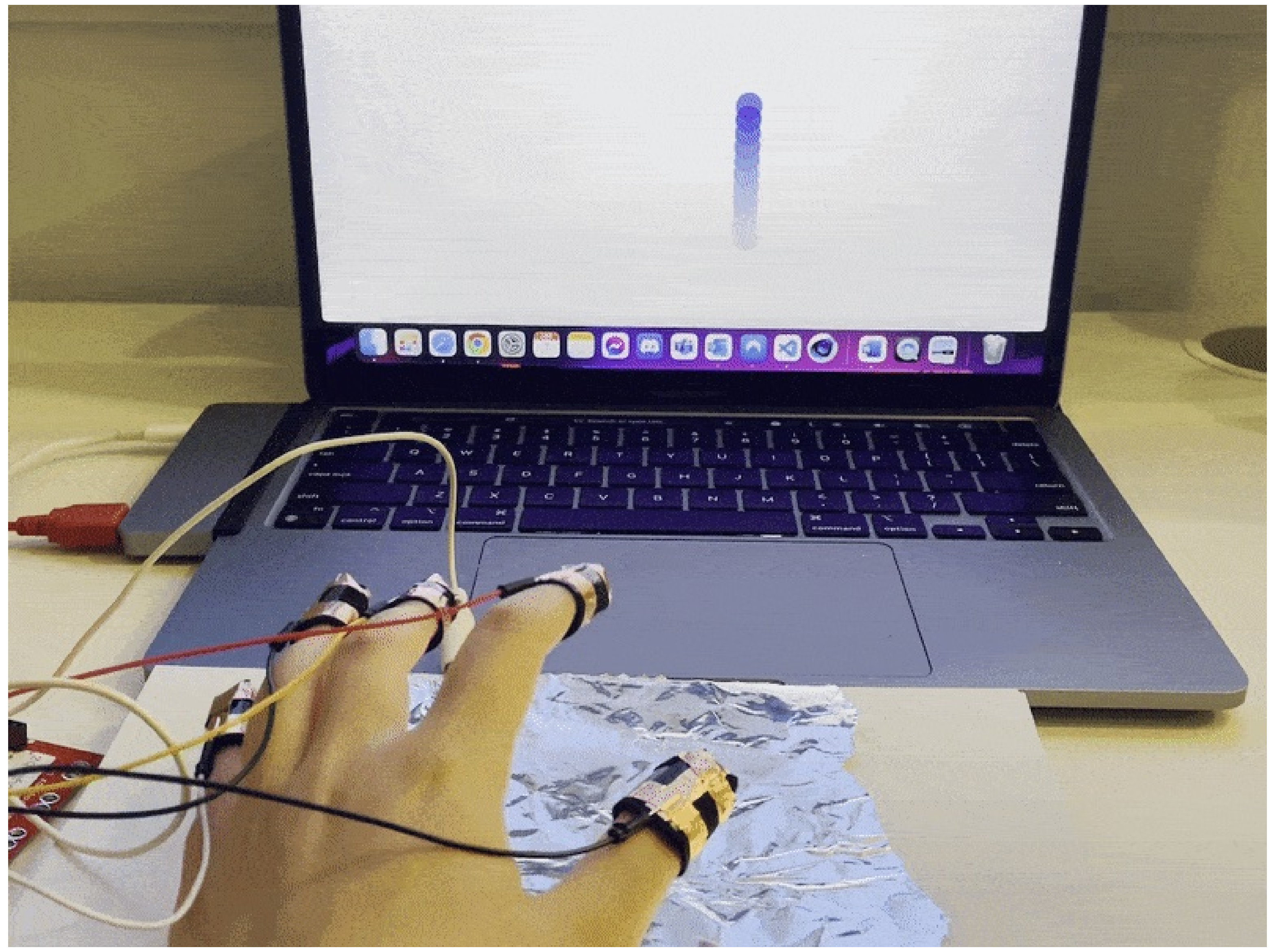
The physical prototype used from here on is the glove, in which the palm is grounded and each finger is connected to a different keystroke. By touching a finger to the palm, the corresponding function is executed (*week 11, image 8*). Instead of using copper tape on a regular glove, a custom prototype was made out of hockey tape for easy removal and a better fit. The prototype consists of a strap across the palm and five finger covers. The palm strap is clipped to the ground of the Makey Makey board. Each finger is connected to the back of the board as follows:

- Thumb to the A key
- Index finger to S key
- Middle finger to D key
- Ring finger to F key
- Little finger to G key

A previous prototype replaced the grounded palm with a wristband that had copper tape inside. Ideally, when the finger covers touch any uncovered part of the hand, the keystroke functions would be triggered; however, the connectivity was not reliable. Another alternative saw the finger covers interacting with a grounded surface (*week 11, image 5*). While this was the most stable prototype, there is less potential for further exploration.

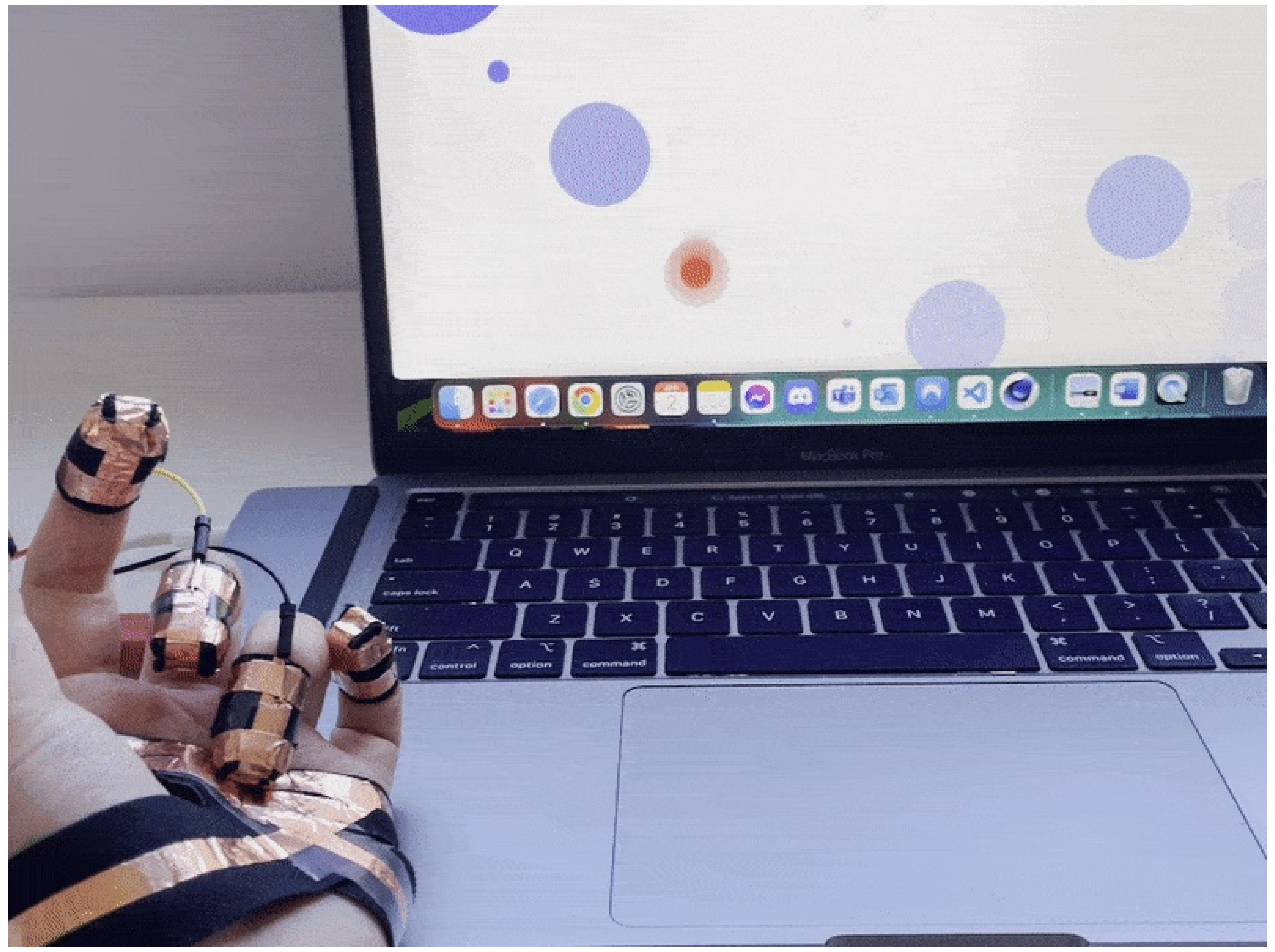




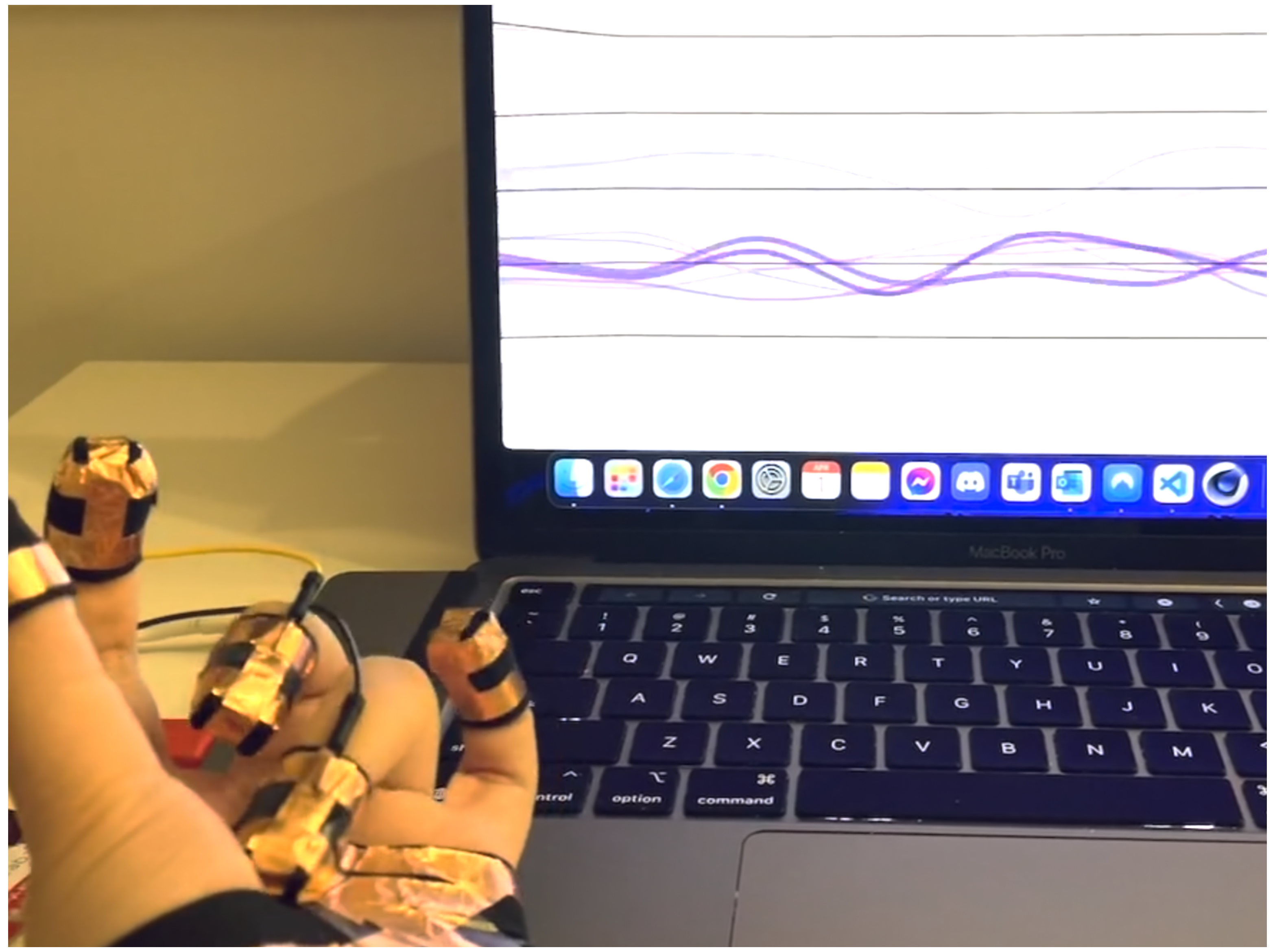


FORMSTORMING

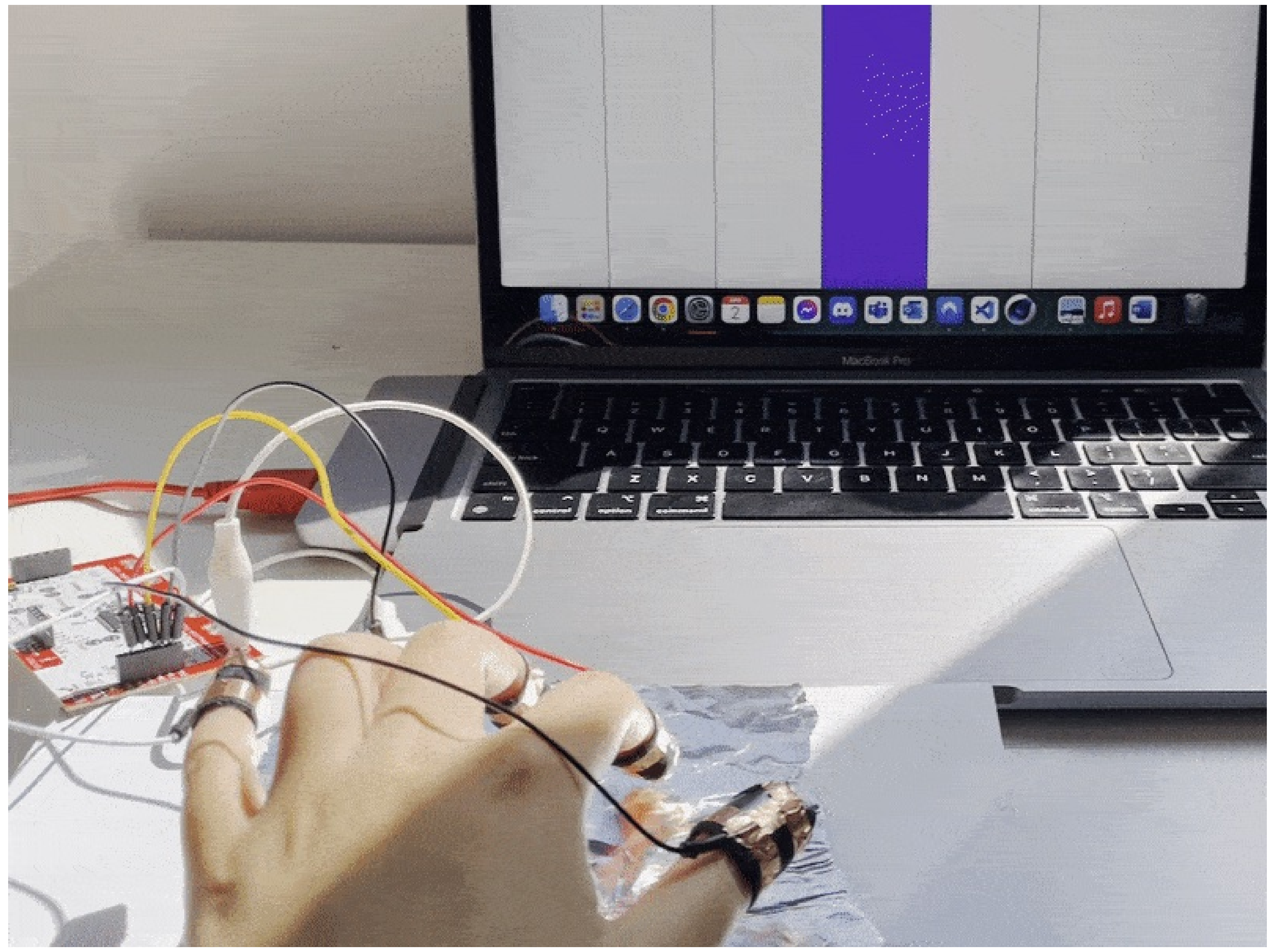
Integrated P5/Makey Prototype



The physical component was first used to experiment with controlling the polysynth visualization. The functions were rearranged to accommodate for the physical restrictions (*week 11, image 8*): the middle finger moved the element up, ring finger moved it down, index finger moved it to the left, little finger moved it to the right, and the thumb moved it in a random direction. The colours were coded based on Scriabin's synesthetic system corresponding to the notes being played (*week 11, image 10*).



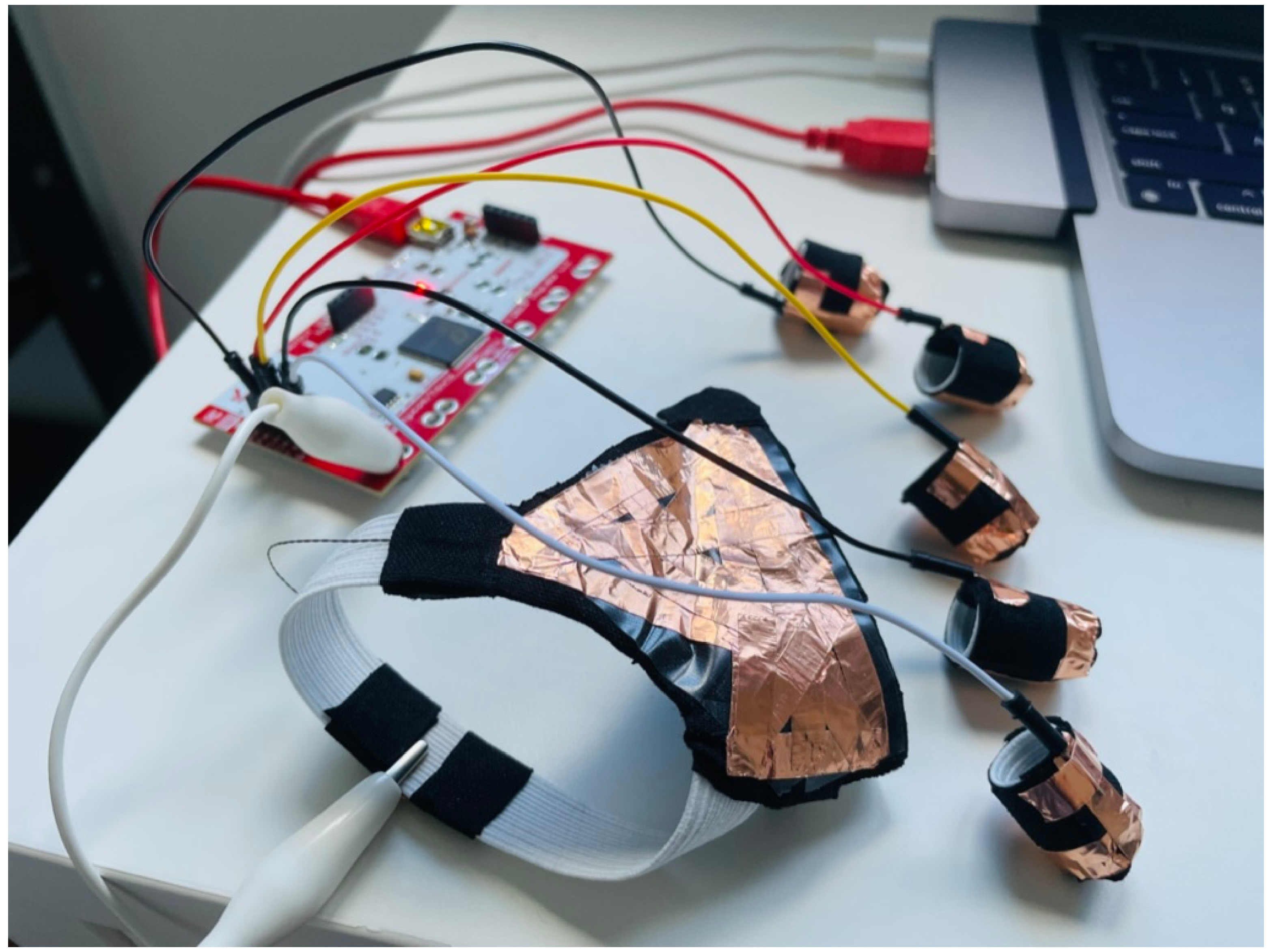
With the interactions contained to the movement of a hand, using the physical component for a musical instrument feels suitable. Incorporating sourced code, coloured strings were used to simulate a guitar being played. When a finger touched the palm, the corresponding string was made to "vibrate" and played a note (*week 11, image 15*). Each finger would play a different string. Chromesthesia was utilized to connect the visual elements to the musical notes, allowing different visualizations to be explored.



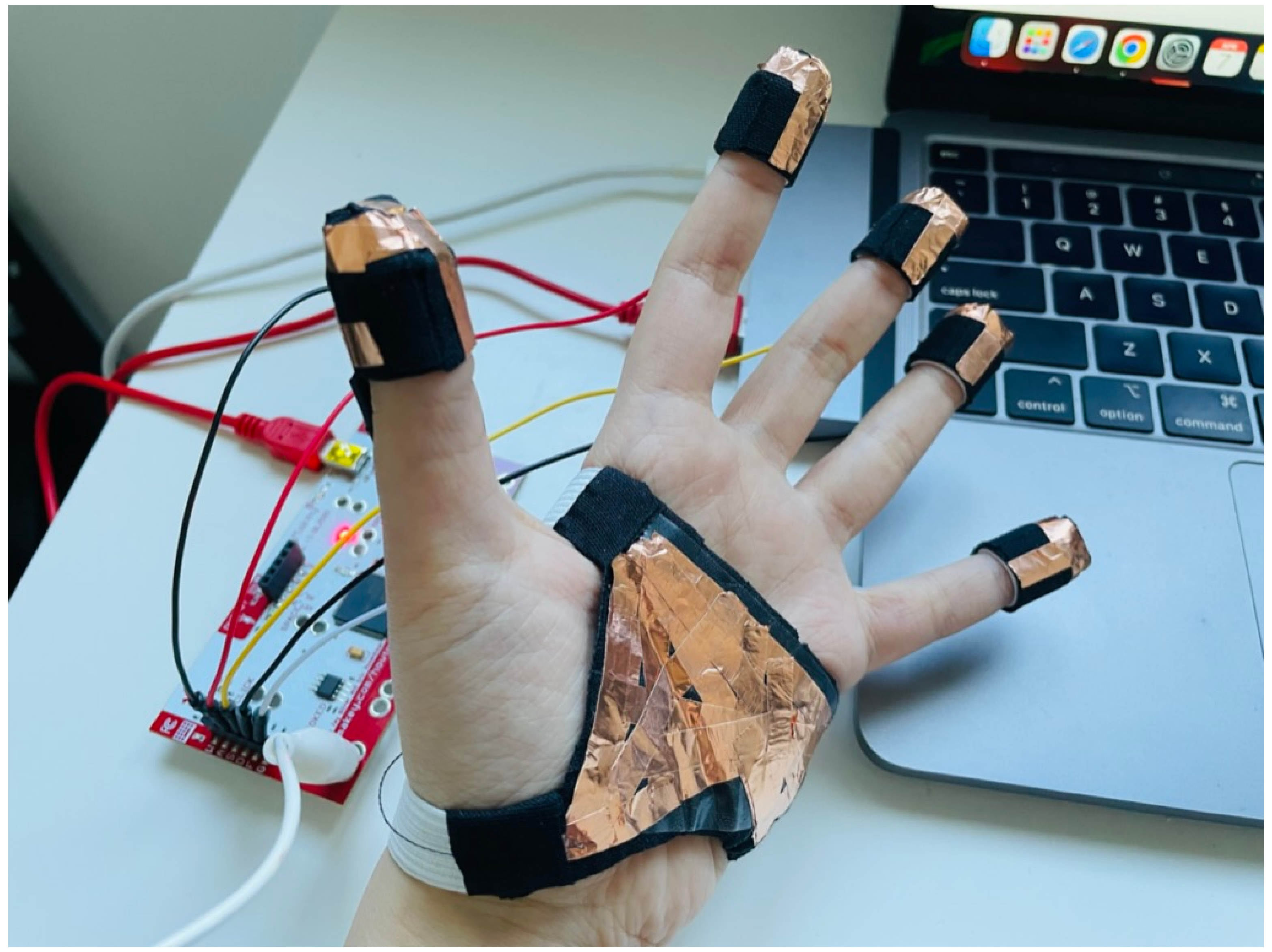
An alternative instrument was a piano; two different sourced examples were used to explore this option. The first one mapped the note envelope across the width of the screen but only played white keys. The second used both black and white keys but was more difficult to adapt to the movement of a hand (*week 11, image 22*). This option would be more suitable with a physical component that better resembles a piano (*week 11, image 25*); for example, strips of copper tape each connected to a different letter.

DESIGN PROCESS

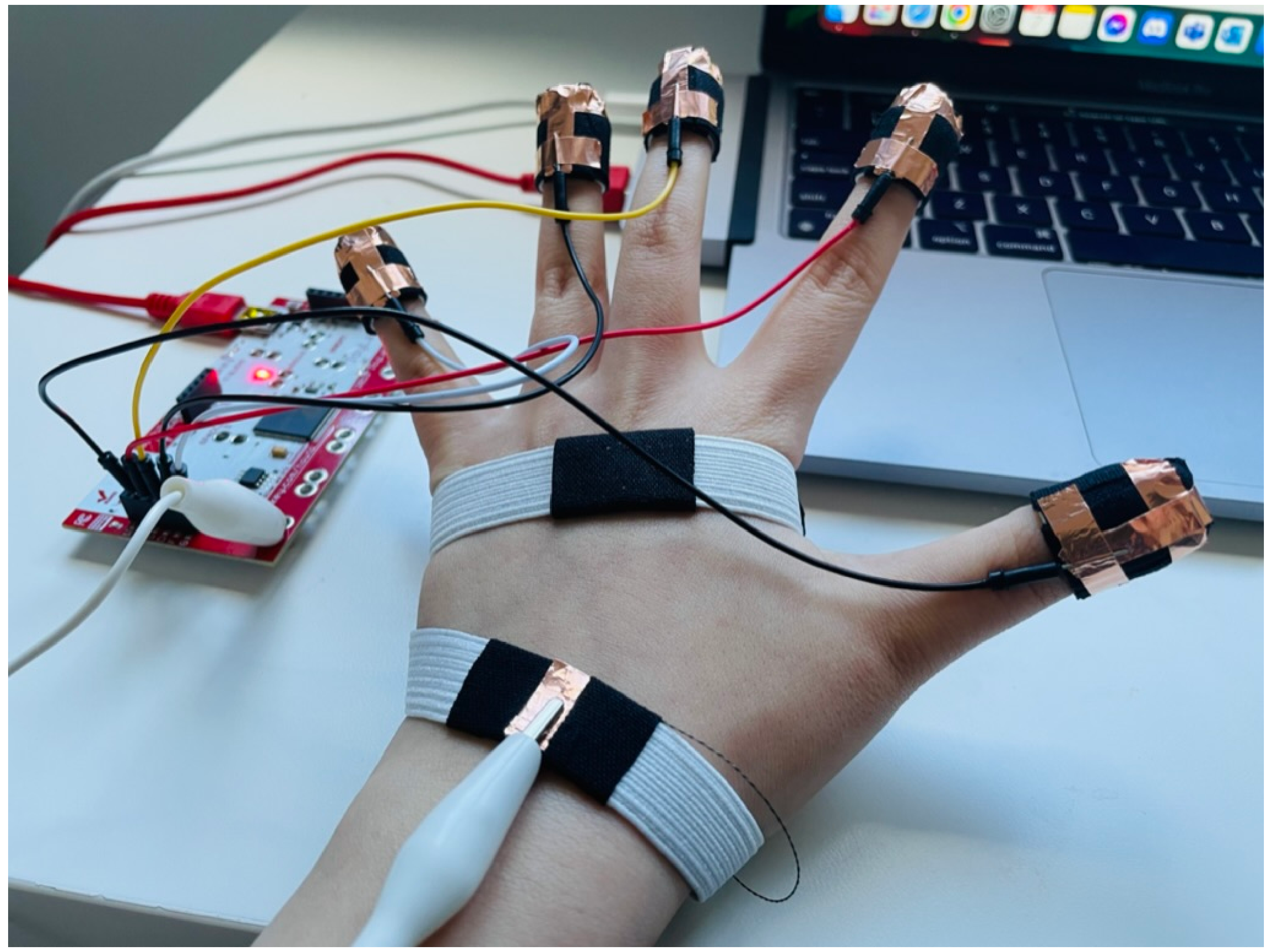
Physical Prototype



The hockey tape strap and finger covers were replaced with elastic bands for a better fit. Ideally, they should have been sewn to best maintain the elasticity, with conductive fabric used for the conductive parts. However, without proper resources, the bands were taped together to create the strap and covers. The prototype might not suit a wide range of hand sizes, but it fits better than previous prototypes.



Similarly to the previous prototype, the palm strap was covered in copper tape and connected to the Makey Makey board with an alligator clip, and each finger cover was connected to a separate letter key. Rather than covering most of the finger cover, only the necessary parts have copper tape to prevent an unintended function executing when two fingers accidentally touch.

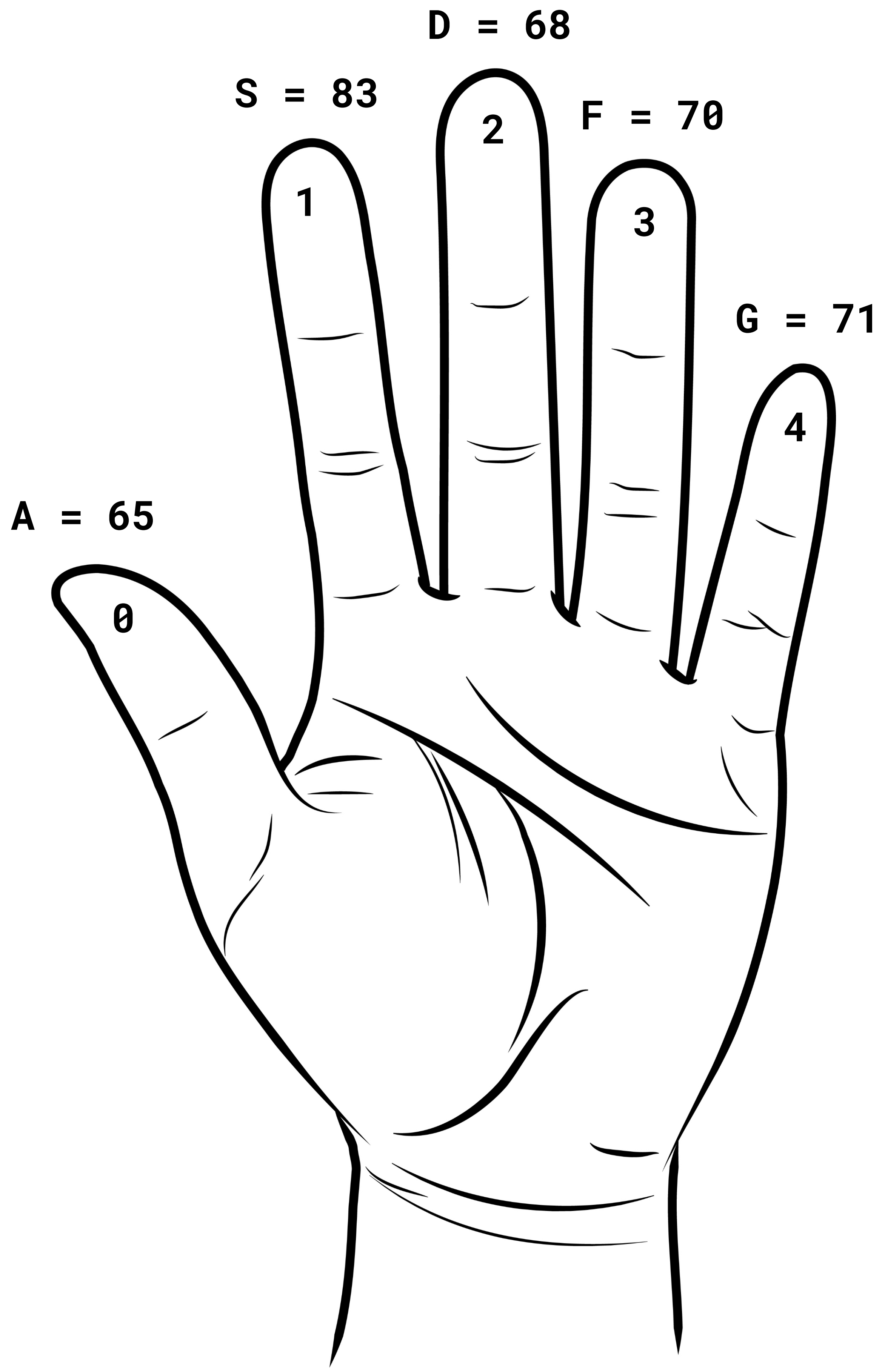


The wires were later replaced with longer ones to be less restrictive. Since an appropriate mobile compartment for the Makey Makey board could not be made with the available resources, it is either placed on a surfaced or held up. Due to the length of the wires, even after replacing them, the prototype has to be used with the palm facing up. Having the palm faced down further limits the range of movement.

DESIGN PROCESS

Guitar Chords to WebGL

Using the letter keys (on the back of the Makey Makey board) makes it easier to touch each finger to the palm. Each letter corresponds to a key code in JavaScript. The position of the root note and overall hand placement determine which finger(s) would play that chord.



C major chord

- Letter key: F (ring finger)
- Notes: C4, E4, G4, C5, E5
- Strings: 1 to 5

E major chord

- Letter keys: S (index finger) & D (middle finger)
- Notes: E3, B3, E4, G#, B4, E5
- Strings: 1 to 6

G major chord

- Letter key: D (middle finger)
- Notes: G3, B3, D4, G4, B4, G5
- Strings: 1 to 6

D minor chord

- Letter keys: D (middle finger) & G (little finger)
- Notes: D4, A4, D5, F5
- Strings: 1 to 4

A minor chord

- Letter keys: D (middle finger) & F (ring finger)
- Notes: A3, E4, A4, C5, E5
- Strings: 1 to 5

D major chord

- Letter keys: S (index finger) & G (little finger)
- Notes: D4, A4, D5, F#
- Strings: 1 to 4

F major chord

- Letter key: G (little finger)
- Notes: C4, F4, A4, C5
- Strings: 2 to 5

A major chord

- Letter key: S (index finger)
- Notes: A3, E4, A4, C#, E5
- Strings: 1 to 5

E minor chord

- Letter keys: S (index finger) & F (ring finger)
- Notes: E3, B3, E4, G4, B4, E5
- Strings: 1 to 6

The next page details the interaction that would trigger the chord, the corresponding letter key(s), the colours of the strings, and the notes being played. The number in brackets next to each note is the MIDI note used to play that note; notes without that value are not played in the code, even though they are played in real life.

C major	Ring finger	F key
X	⬛	no colour
C4 (60)	🟥	rgb(255, 0, 0)
E4 (64)	🟦	rgb(195, 242, 255)
G4 (67)	🟧	rgb(255, 127, 0)
C5	🟥	rgb(255, 0, 0)
E5	🟦	rgb(195, 242, 255)

D major	Index & little fingers	S & G keys
X	⬛	no colour
X	⬛	no colour
D4 (62)	🟨	rgb(255, 255, 0)
A4 (69)	🟩	rgb(51, 204, 51)
D5	🟨	rgb(255, 255, 0)
F# (66)	🟦	rgb(127, 139, 253)

E major	Index & middle fingers	S & D keys
E3	🟦	rgb(195, 242, 255)
B3	🟦	rgb(142, 201, 255)
E4 (64)	🟦	rgb(195, 242, 255)
G# (68)	🟪	rgb(187, 117, 252)
B4 (71)	🟦	rgb(142, 201, 255)
E5	🟦	rgb(195, 242, 255)

F major	Little finger	G key
X	⬛	no colour
C4	🟥	rgb(255, 0, 0)
F4 (65)	🟥	rgb(171, 0, 52)
A4 (69)	🟩	rgb(51, 204, 51)
C5 (72)	🟥	rgb(142, 201, 255)
X	⬛	no colour

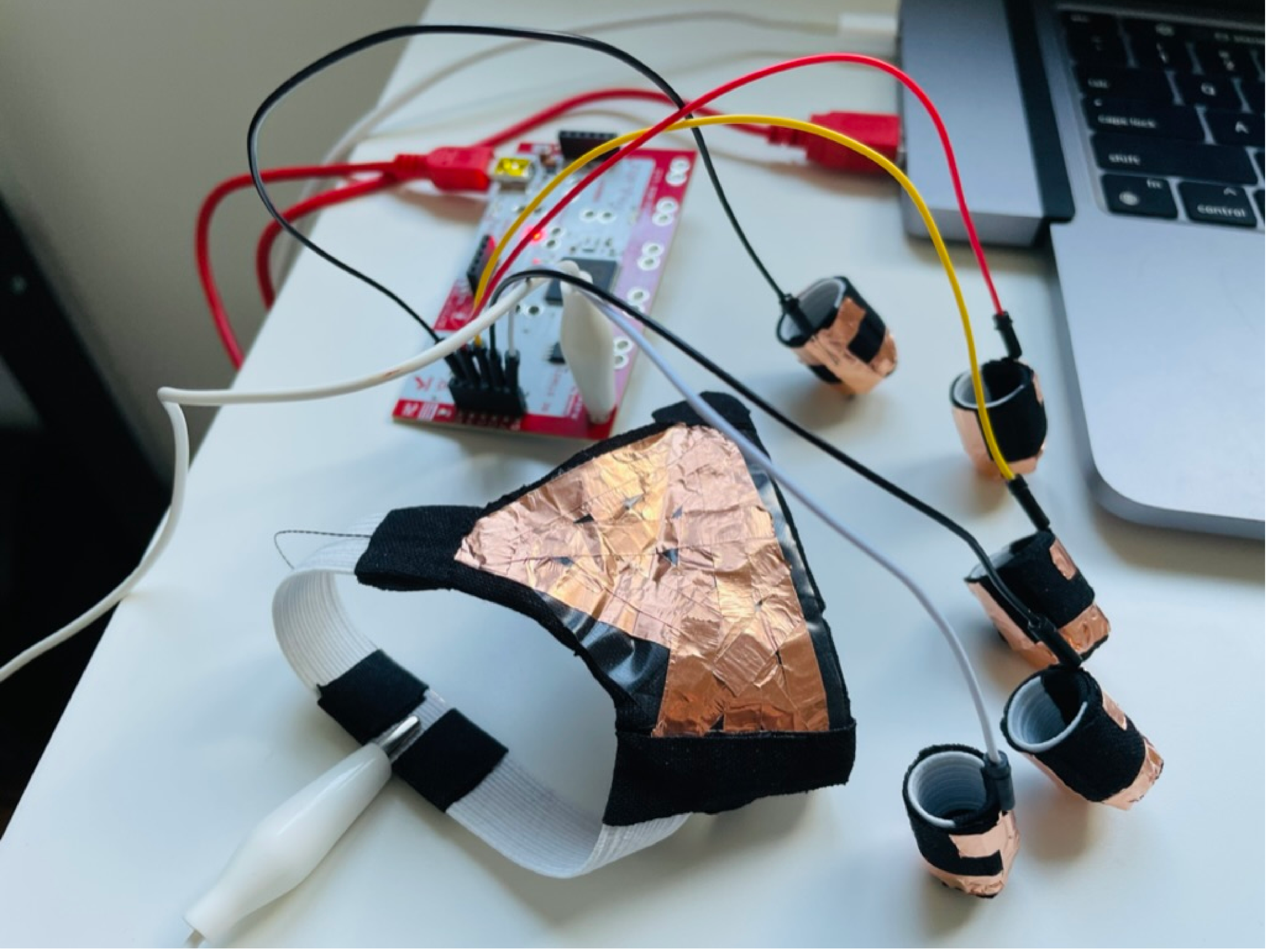
G major	Middle finger	D key
G3 (55)	🟧	rgb(255, 127, 0)
B3 (59)	🟦	rgb(142, 201, 255)
D4 (62)	🟨	rgb(255, 255, 0)
G4	🟧	rgb(255, 127, 0)
B4	🟦	rgb(142, 201, 255)
G5	🟧	rgb(255, 127, 0)

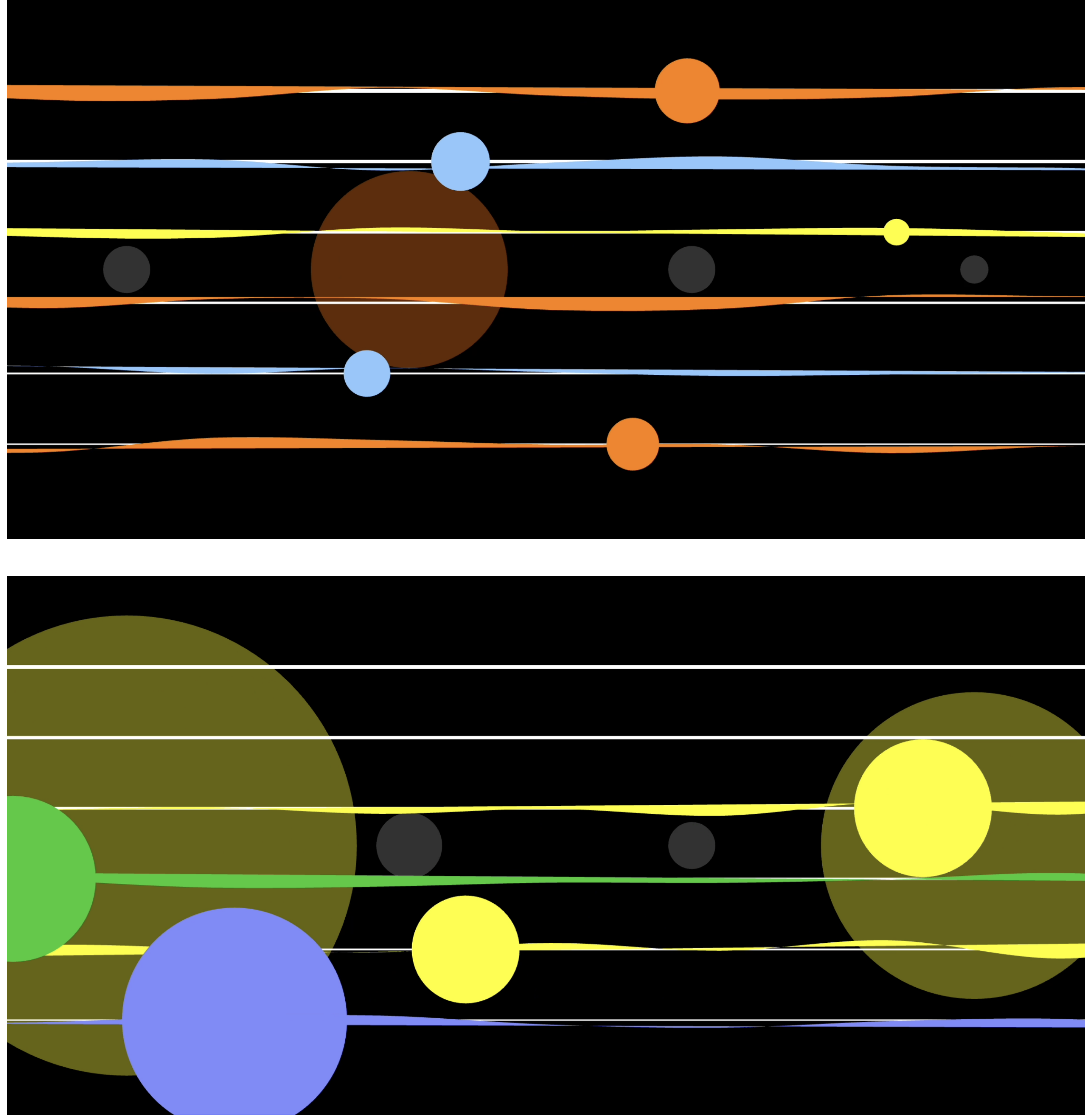
A major	Index finger	S key
X	⬛	no colour
A3 (57)	🟩	rgb(51, 204, 51)
E4 (64)	🟦	rgb(195, 242, 255)
A4	🟩	rgb(51, 204, 51)
C# (61)	🟪	rgb(183, 70, 139)
E5	🟦	rgb(195, 242, 255)

D minor	Middle & little fingers	D & G keys
X	⬛	no colour
X	⬛	no colour
D4 (62)	🟨	rgb(255, 255, 0)
A4 (69)	🟩	rgb(51, 204, 51)
D5	🟨	rgb(255, 255, 0)
F5 (77)	🟥	rgb(171, 0, 52)

E minor	Index & ring fingers	S & F keys
E3	🟦	rgb(195, 242, 255)
B3	🟦	rgb(142, 201, 255)
E4 (64)	🟦	rgb(195, 242, 255)
G4 (67)	🟧	rgb(255, 127, 0)
B4 (71)	🟦	rgb(142, 201, 255)
E5	🟦	rgb(195, 242, 255)

A minor	Middle & ring fingers	D & F keys
X	⬛	no colour
A3	🟩	rgb(51, 204, 51)
E4	🟦	rgb(195, 242, 255)
A4 (69)	🟩	rgb(51, 204, 51)
C5 (72)	🟥	rgb(255, 0, 0)
E5 (76)	🟦	rgb(195, 242, 255)





FINAL DESIGN

Synesthetic Guitar

WebGL output:
https://ixd544.phoenix.sheridanc.on.ca/MediaMotion&Body/MM&B_Module_3/images/final/

Weekly activity:
https://ixd544.phoenix.sheridanc.on.ca/MediaMotion&Body/MM&B_Module_3/

The analog physical input takes the form of a custom glove that allows the wearer to display visual and auditory outputs based on closing and opening different fingers. The WebGL output simulates a guitar being played using visualization of a regular guitar and the perception of sound and movement. The notes and chords being played are colour-coded based on Alexander Scriabin's synesthetic system.

Each note corresponds to a specific colour, playing a chord triggers a combination of colours and movements. The colour on a string indicates that it would be strummed on a physical guitar and which note is being played. The circles in the background indicate which finger(s) is pressing down.