

Designing CLIO, An Open-Source Toolkit for Museum Pop-Up Digital Interactives

Dillon Connelly, University of Washington Museology Master's Program, USA, **Joshua Frechette**, Central Seattle College, USA

Abstract

With the increasing prevalence of open-source technologies, museums have been provided with a plethora of low-cost tools to enhance exhibitions and educational programming at their institutions. While there are toolkits developed to tackle the digital interactive needs of museums, they often require a programming background to fully implement. CLIO, named after the Greek muse of history, is an open-source digital interactive toolkit that aims to lower the knowledge ceiling required to implement interactive exhibits by creating a series of activity templates that can be customized by anyone able to edit a website. Our demonstration will briefly introduce the goals of the CLIO project and discuss challenges in executing CLIO for the first time and possible future implementations of the technology, with a more detailed description of the project in the accompanying paper.

Keywords: digital interactives, open source, JavaScript, HTML, museum education



Designing CLIO

An Open-Source Toolkit for Museum Pop-Up Digital Interactives

Joshua Frechette

Designer and Developer

Dillon Connelly

University of Washington Museology Graduate Program

Introduction

With the increasing prevalence and availability of open-source technologies, museums have been provided with a plethora of low-cost tools to enhance exhibitions and educational programming at their institutions (Langer & Alderman, 2016). While there are tools developed to tackle the digital interactive needs of museums, they often require extensive knowledge of coding to fully utilize. We aim to rectify this with the creation of CLIO, an open-source toolkit for the creation of digital exhibit kiosks.

The CLIO toolkit is powered by web technologies such as HTML, CSS and JavaScript. Worldwide surveys of developers consistently show that scripting languages like JavaScript are the most commonly understood programming languages worldwide (Ramel, 2018; Stack Overflow Developer Survey, 2019), and museums are more likely to have staff familiar with web technologies than other forms of digital development tools (Price & James, 2018).

In this paper, we will describe the design and creation of the CLIO toolkit, which includes a 3D-printed touchscreen kiosk built for the BurkeMobile, a mobile pop-up mini-museum educational program at the Washington State Burke Museum of Natural History and Culture.

What is CLIO?

Project Overview and Background

We chose to name our project “CLIO,” after the Greek muse of history. Like the muses of old, we hope to inspire new creation and open innovation within the museum sector.

CLIO integrates existing low-cost electronics, 3D printing, and open-source software to create kiosk hardware that museums can easily replicate, either in batches or one at a time, by following the provided guides. Prototype kiosks were designed for mobility and offer a modular design that can be configured for integration into different environments, such as classrooms, cafeterias, libraries and community centers. The interface and activities utilize web-based technologies, allowing them to be used with virtually any display and input system with little to no modification of the underlying code.

When the kiosk is plugged in, it will automatically boot into “Facilitator Mode”, a graphical user interface that allows program facilitators to customize which “Activities” to display as part of their exhibition. Activities are collected into “Programs”, which can be loaded on-the-fly without the need to change hardware. Activity content is stored locally on the device, allowing the kiosk to be used across multiple exhibits without access to the internet. After the desired Activities are selected, the kiosk can enter “Exhibit Mode”; an alternate graphical user interface that displays the selected Activities as a carousel to be navigated by users.

Activities included in this toolkit range in complexity and customization options. There are simple interactive activities with very little customization, as well as more complex activities that can be extensively customized to assist in the creation of a narrative. Simple interactive games like quizzes, flashcards, and tile matching are included for younger audiences, while other activities like timelines enable older audiences to have a more in-depth informational experience. Activities can be sorted into customizable audience groups, allowing facilitators to tailor the experience more closely to their current audience. All activity templates included in the toolkit can be fully customized with new images and rich text descriptions using a plain text editor.

The kiosks used during pilot testing were constructed using a Raspberry Pi 3B+ computer, official Raspberry Pi 7” touchscreen and consumer 3D printer for a total unit build cost of \$125. This low-cost and modular hardware configuration allow the kiosk to be assembled and deployed with minimal resources, while offering the flexibility, accessibility and maintainability to engage a variety of audiences in different learning contexts.

Goals

Accessibility

CLIO was designed to create materials surrounding a collection that can be accessed and understood by the broadest possible user base. In practice, this manifests in many different ways; from clarity within the interface and accessibility features, to repairability and maintenance, and the inclusion of open-source technologies that anyone can improve.

When implementing an accessibility feature, it should serve the largest audience possible. The interface uses simple controls and established conventions for icons, menus, and user interactions to improve clarity and ease navigation (Reeves, 2015). The interface intentionally utilizes large on-screen controls to create a more accessible user experience, especially when used with a touchscreen (Dickerson, 2018). The interface employs client-side, rather than server-side, technologies making the kiosk more responsive than one limited by internet availability and bandwidth (Segue Technologies, 2013). Finally, the kiosk was designed to have a small physical footprint as not to overshadow the physical objects it is contextualizing (Damala, Ruthven, & Hornecker, 2019).

Community

The keystone of open-source technology is community involvement. Unlike proprietary technologies developed by private companies, the continued development of open-source code is not limited by budgets and business interests but the need for and use of that code by anyone worldwide.

As museums explore different ways to engage with and define themselves within communities, it is rare that they think of themselves as part of the global technology community. Our hope is that CLIO and other projects like it will empower museums to create their own technologies and share these tools as part of the global open-source development community.

The Toolkit

The CLIO toolkit includes an interface and templates for interactive activities that are built on web technologies, allowing them to work on any hardware and software that can host a web server (Figure 1). Templates for twelve different interactive activities are available (Table 1), as well as all necessary software, usage and activity editing guides, hardware recommendations, and 3D printable files for the kiosk enclosure and mounting system. A printable brochure is also provided to help facilitators get the kiosk setup quickly and into Exhibit Mode.

HARDWARE INDEPENDENT
GRAPHICAL USER INTERFACE

ACTIVITIES

JavaScript

INTERFACE

HTML, CSS, JavaScript, PHP, AJAX

PROTOTYPE KIOSK

SOFTWARE

Debian, Apache, Chromium, Python

HARDWARE

Raspberry Pi, Adafruit, 3D Printing

Figure 1: Toolkit Technologies- The interface and activities are web-based, allowing them to be used with any hardware capable of running a web server and browser. The CLIO toolkit also includes specifications for utilizing modular open-hardware and quickly producing a usable kiosk.

Activity Type	Description
Image Comparison	Superimpose two images with a draggable divider to show visual difference. This activity can also be configured to use hot areas which open full screen informational pop-ups.
Timeline	Navigate through a timeline with each slide having an image and rich text.
Tile Match	Designed to draw connections between two sets of objects, images, or ideas, these tiles can be customized to use text or an image and invites users to find a matching set. This activity can be configured to show informational pop-ups on a correct match.
Objects	Display objects from collections with images and rich text.
Slideshow	Navigate through a slideshow, with optional basement slides to further contextualize information.
Flashcards	Display topic cards that can be interacted with to provide more information.

<i>Image Gallery</i>	Display a grid of interactive images.
<i>True or False Quiz</i>	Display an image and question with binary answers. This activity can be configured to show different responses on correct and incorrect answers.
<i>Visual Thinking</i>	This inquiry-based learning activity can be used to highlight an image and provide a prompt so that the user can learn more about the things they find most interesting.
<i>Play Video</i>	Play and control a video.
<i>3D Render</i>	Manipulate renders of three-dimensional .stl objects.
<i>Rich Text with Narration</i>	Display rich text, which can include supplemental audio playback, such as an interview, narration or other form of voiceover.

Table 1: About the interactive activities- Activities included in this toolkit range in complexity and customization options.

Interface

When the kiosk is plugged in, it will automatically enter Facilitator Mode (Figure 2). This graphical user interface allows facilitators to change which Program is loaded, resume the previously used Activities and choose additional Activities they may wish to display as part of their exhibit. Facilitator Mode always has a menu bar visible at the top of the screen, displaying tabs relating to the core functionalities of the kiosk.

After the desired Activities are selected, the facilitator can enter Exhibit Mode (Figure 3) to temporarily disable further configuration of the kiosk. Exhibit Mode restricts kiosk users to Activities selected by the facilitator. By design, the kiosk must be fully restarted to return to Facilitator Mode. The kiosk can also be configured to boot directly into Exhibit Mode using activities specified within a configuration file.

Exhibit Mode displays the Activities selected by the facilitator as a carousel that users can navigate. Tapping on an activity will open it full screen and the user can return home at any time to access other Activities. Additionally, the kiosk will automatically return home after an amount of time without interaction.

The accessibility menu is always available on the kiosk screen during Facilitator Mode and Exhibit Mode, allowing both facilitator and user to customize the interface in a variety of ways. This includes font and paragraph settings, font faces that can improve character recognition for those with disorders like dyslexia (Rello, & Baeza-Yates, 2013), and different high contrast modes assisting those with visual impairment (W3C, 2016). These universal design principles not only allows more users to access the kiosk, but creates customization and usability options for deploying the kiosk in different programs and environments; a design benefit sometimes referred to as “the curb-cut effect” (Beagley, 2017). After the kiosk has been idle for a set amount of time, all accessibility options will be returned to their defaults.

The kiosk interface is a dedicated web application written using HTML, CSS, JavaScript, PHP and AJAX. All user-based interactions are handled dynamically through JavaScript resulting in a more responsive experience than a kiosk that must individually request, download and render new pages. Custom code for this project is created using jQuery, an open-source JavaScript library that simplifies the readability of complex JavaScript functions. The kiosk interface web application runs within the Chromium web browser with all access to the underlying operating system completely

disabled. Access to the kiosk filesystem is only possible by opening the kiosk case and entering the proper credentials with a physical keyboard.

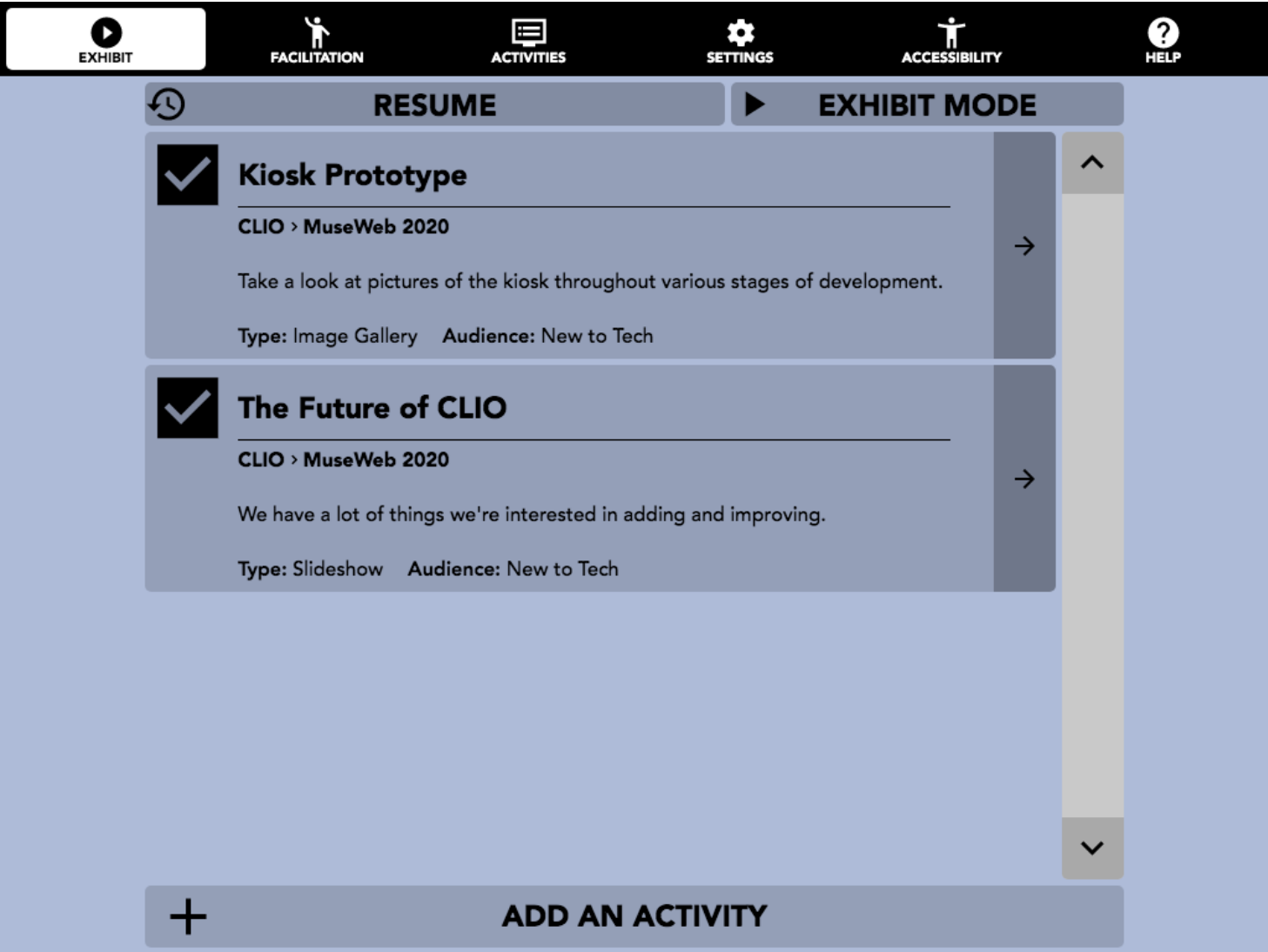
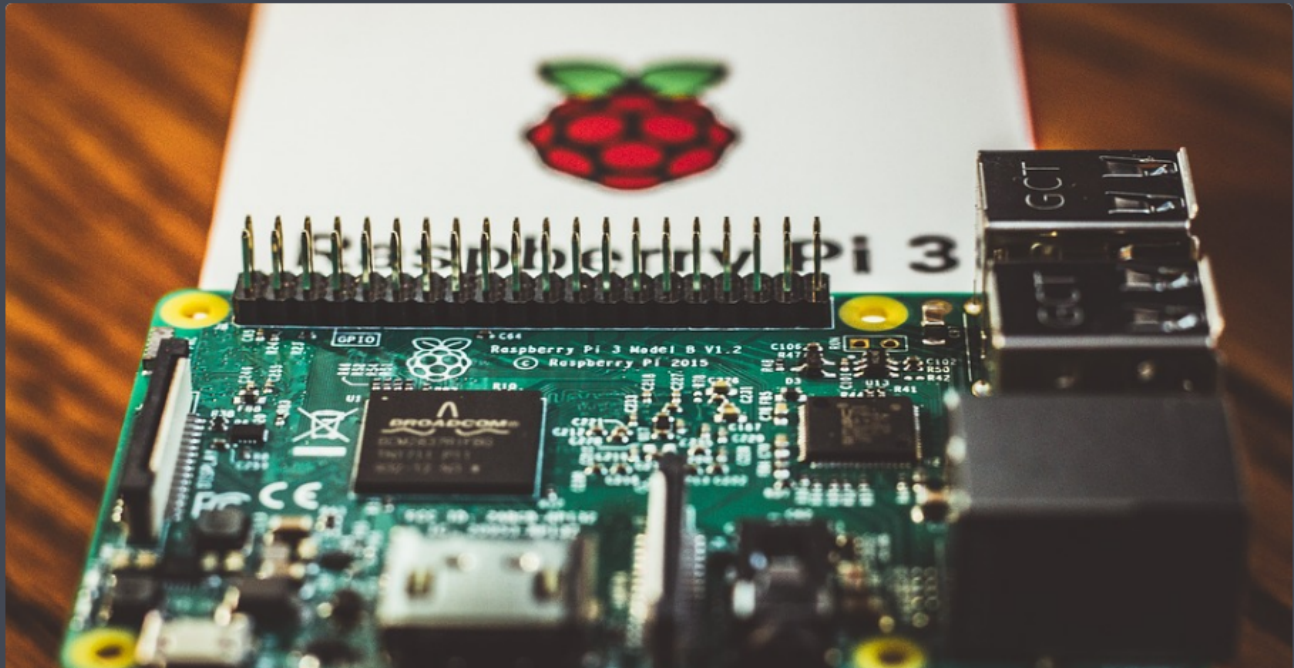


Figure 2: Facilitator Mode- Upon startup, the kiosk allows facilitators to select the Activities they wish to display in Exhibit Mode as part of their program.



Kiosk Hardware



Figure 3: Exhibit Mode- After Activities are selected, Exhibit Mode restricts users to Activities selected by the facilitator. Returning to Facilitator Mode requires a restart of the kiosk.

Activities

IMAGE COMPARISON

This activity superimposes two images over each other, with a divider that can be dragged to show the visual difference between them (Figure 4). By default, this activity will generate small buttons at the specified coordinates that can be used to open full screen informational displays with rich text content. The buttons can also be hidden, creating customizable hot areas that open the corresponding informational pop-up when touched (Figure 5). You can also choose to enable a “hint” button that will reveal these hot areas to the user when tapped (Figure 6). This activity can be used to compare two disparate images, overlay new information on an image, or show different views of the same object.

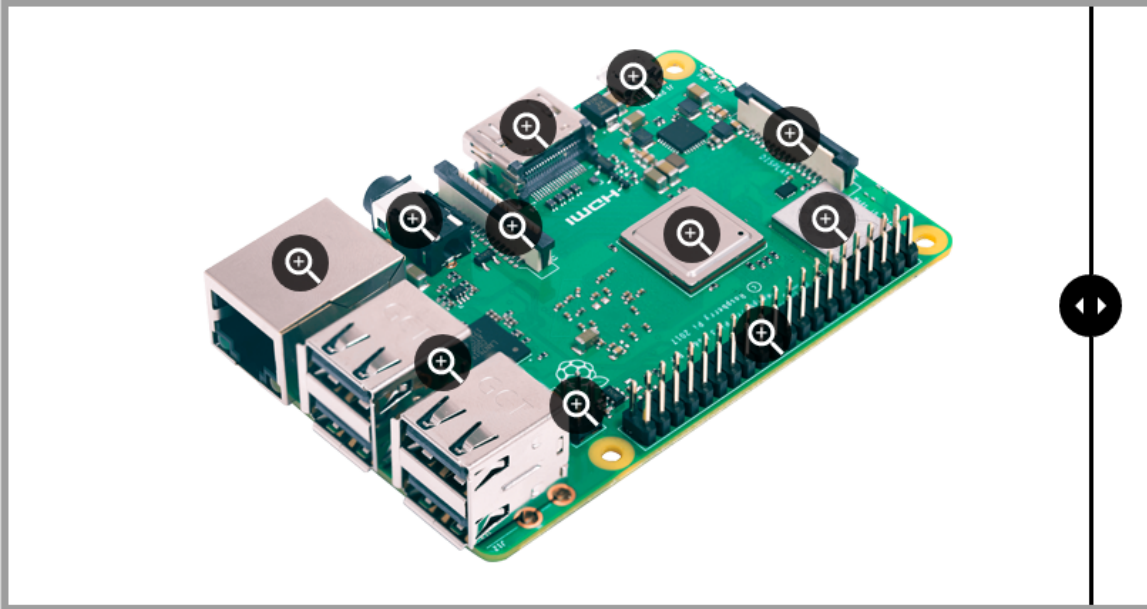


Figure 4a: Raspberry Pi Specifications “Compare Image” activity. By default, the tooltips are rendered as small magnifying glass icons.

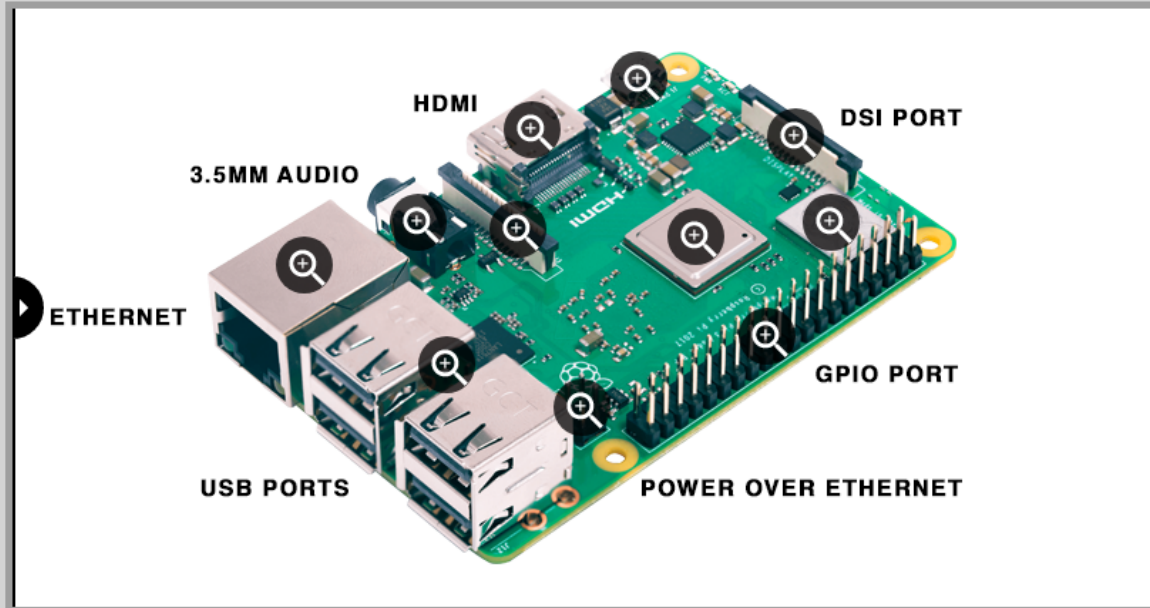


Figure 4b: Raspberry Pi Specifications “Compare Image” activity. The divider can be dragged to show the visual differences between the two images.



System-On-A-Chip

This microcomputer utilizes an ARM-based processor, which is more closely related to the processor in a cell phone than a traditional desktop processor. They were designed for power efficiency and passive heat management.

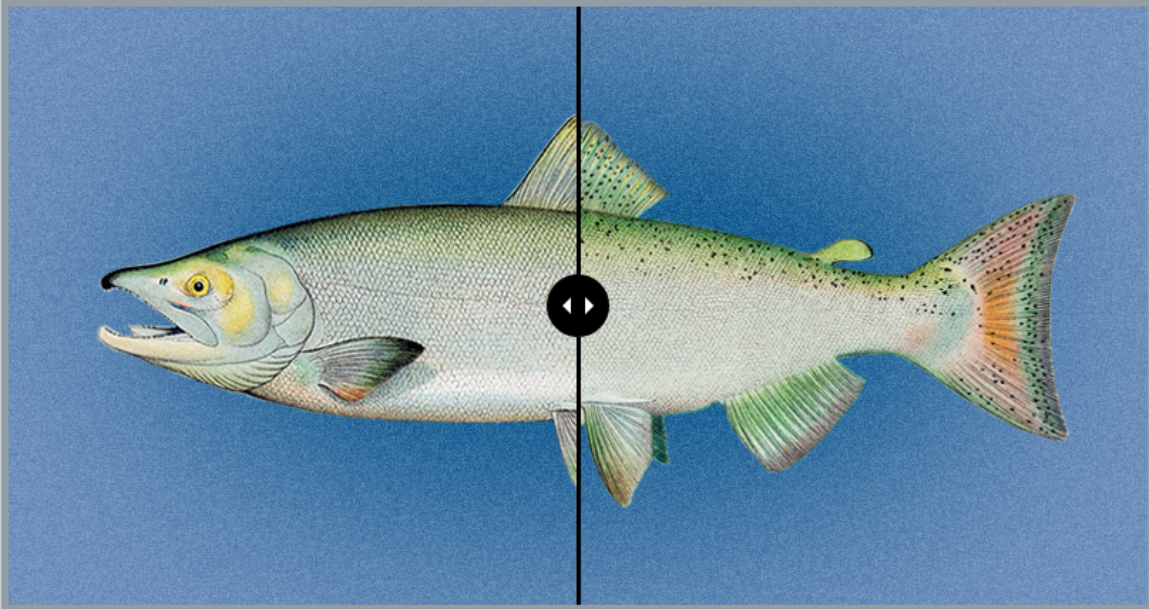


Figure 4c: Raspberry Pi Specifications “Compare Image” activity. Full screen informational pop-ups are utilized within many activities as a way to invite users to explore the information they find most interesting.



BACK

SPOT THE DIFFERENCE



Sockeye Salmon

Chinook Salmon



Figure 5: Spot the Difference “Compare Image” activity. Invisible hot areas can be configured to open full screen informational pop-ups when touched. In this example, dragging the slider over the image reveals the visual difference between the Sockeye and Chinook salmon.

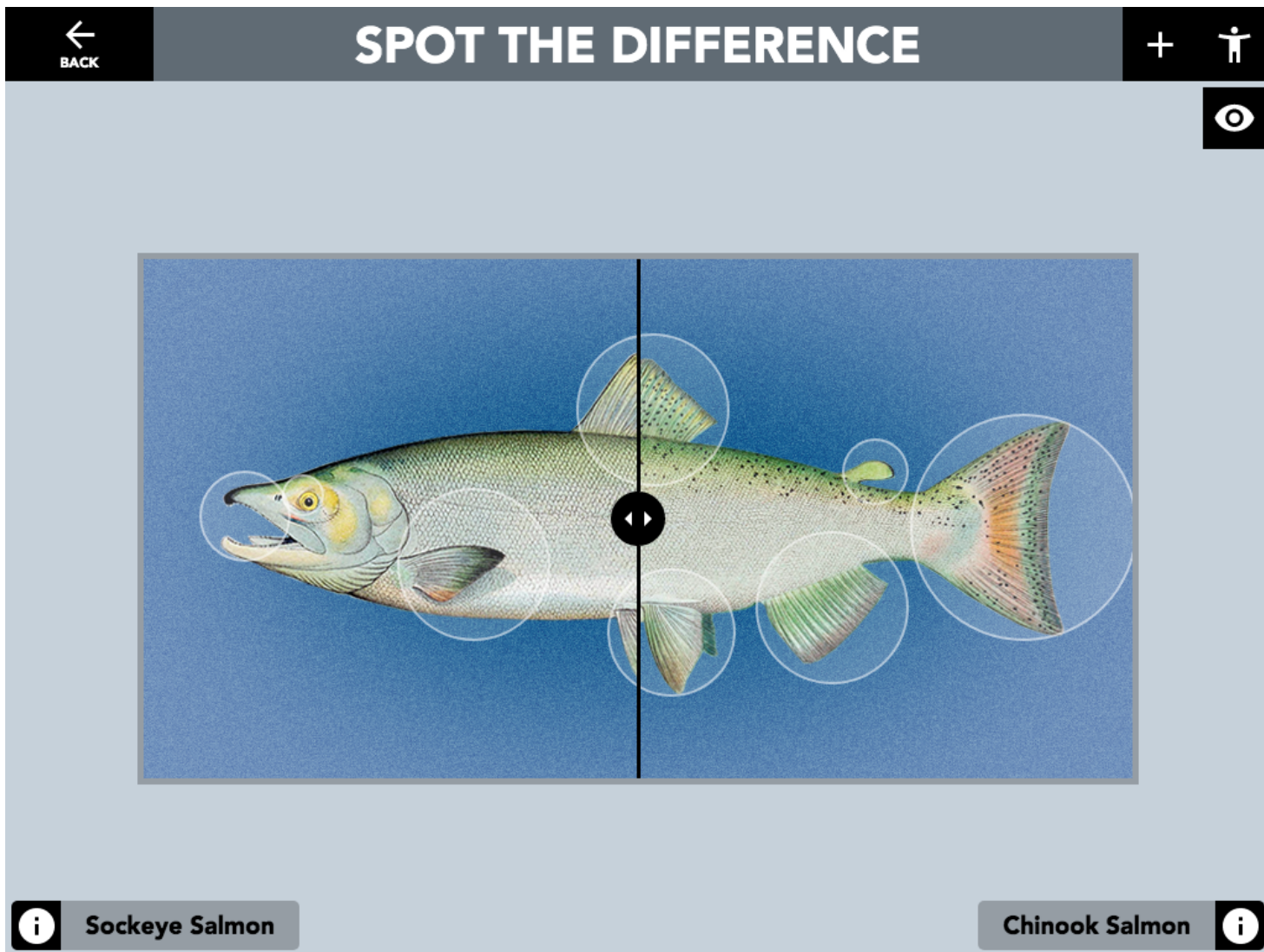


Figure 6: Spot The Difference “Compare Image” activity. A “hint” button can be enabled that allows users to reveal the hidden hot areas when touched.

TIMELINE

This activity allows users to move through a timeline with each slide containing an image and rich text. The timescale, number of dates, and date format are all customizable, allowing for the creation of micro- or macroscale timelines (Figure 7).

BACK

THE CLIO TEAM

+

November 18, 2017

We spent a six month internship at the Roger Williams Natural History Museum in Providence, RI creating a small digital exhibit, named *"Habitable Worlds: Searching for Life in our Solar System and Beyond"*.

The kiosks used Raspberry Pi 3B+ microcomputers and televisions to create digital exhibits that used a keyboard number pad. It utilized PiPresents, written by Ken Thompson, an open-source Python utility to create interactive exhibits.

11.18.2017

9.6.2018

Figure 7: CLIO Team “Timeline” activity. The time scale used by the activities is customizable, allowing micro- and macroscale timelines.

TILE MATCH

Designed to draw connections between two sets of objects, images, or ideas, these tiles can be customized to have a title, text and an image (Figure 8). One tile from each side can be selected at a time and a matching pair displays a full screen pop-up with customizable rich text to provide further information to the user about their answer.

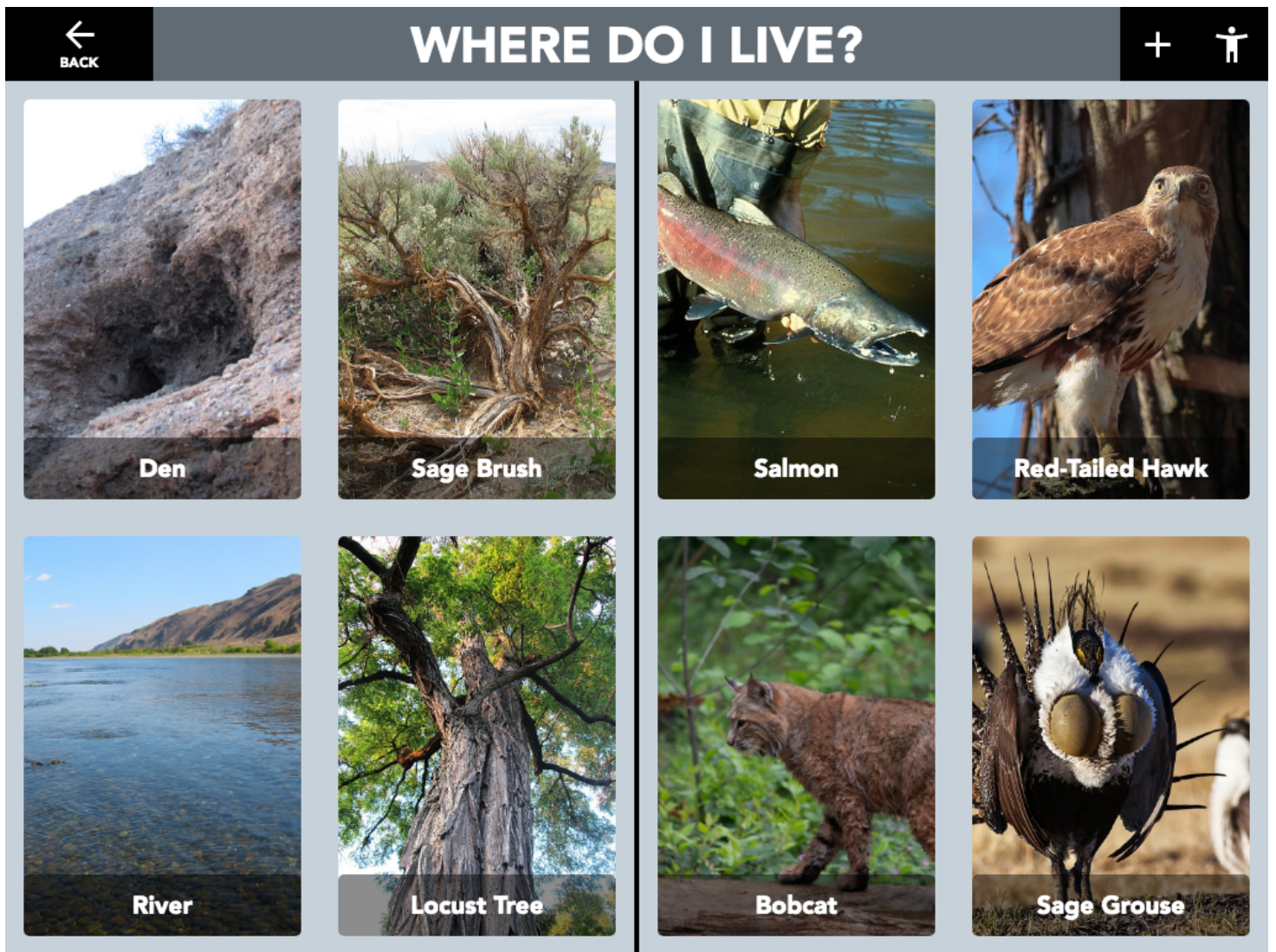


Figure 8: Where Do I Live? “Tile Match” activity. When customizing the activity, program facilitators can select which tile in the left group matches with which tile on the right.

OBJECTS

Objects from museum collections can be listed, each with a title, subtitle, description and image (Figure 9). Tapping on an object will open it in a full screen pop-up that can be scrolled through to show additional images of the object, if available, as well as a description (Figure 10). Zooming in will hide the description, allowing the user to pan around a full screen image. This activity is designed to draw connections between a museum’s “home” collections and the objects on display in outreach programming.

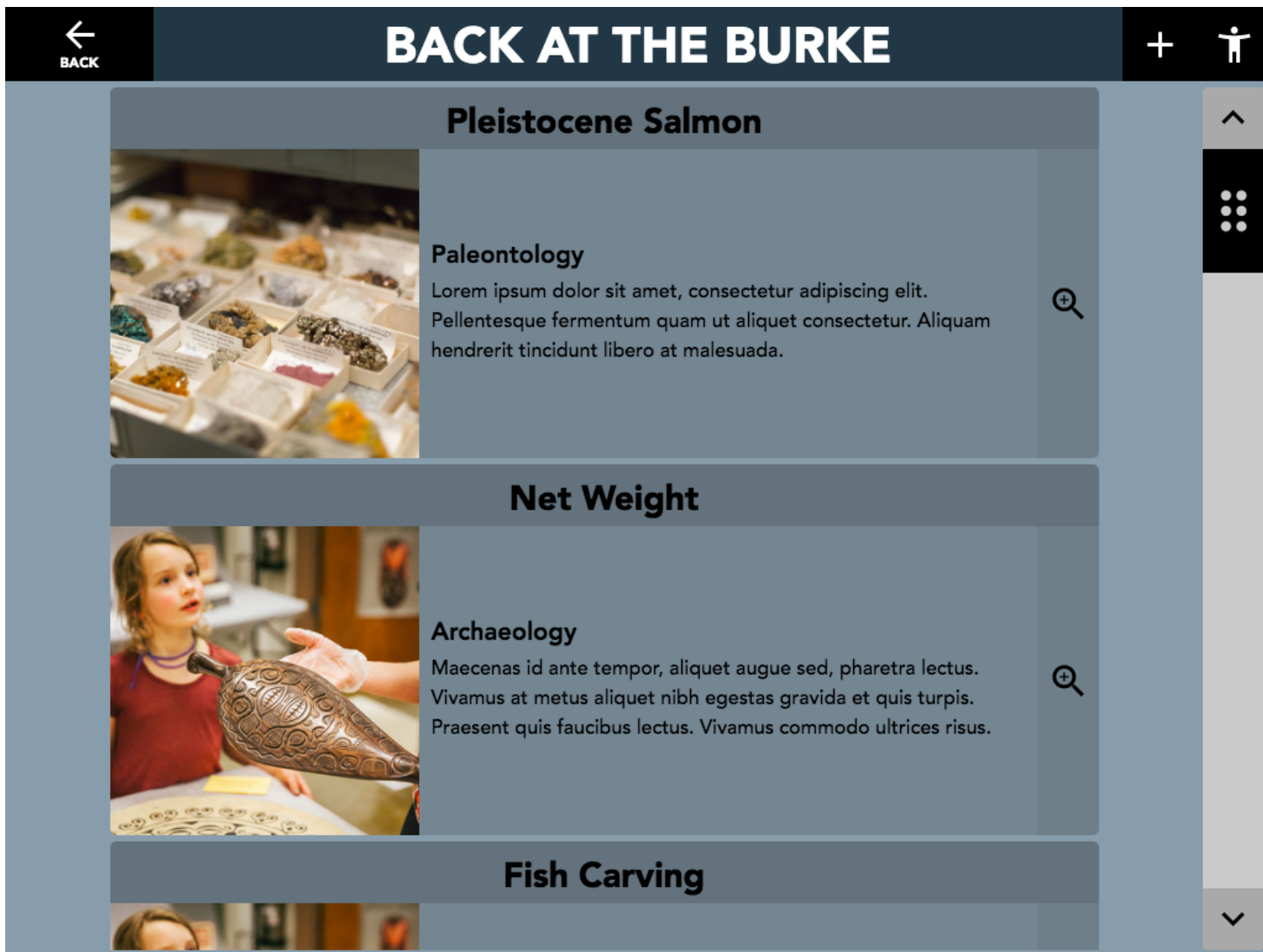


Figure 9: Back at the Burke “Object” activity.

This activity is designed to highlight objects from the museum’s “home” collection that are related to objects in the pop-up exhibition.



This is the geology image.

Figure 10: Back at the Burke “Object” activity. Tapping on an object opens a collection of images related to the object, as well as a brief description for each image.

SLIDESHOW

This activity allows users to navigate through a slideshow, with optional “basement” slides. Each slide is individually customizable, including the title, content and background, which can be either a color, image or video (Figure 11). A slide can have as many basement slides as desired, which exist “beneath” that slide and allows users to navigate down to see additional information (Figure 12). Tapping the “overview” button allows users to interact with a grid layout of all the available slides.

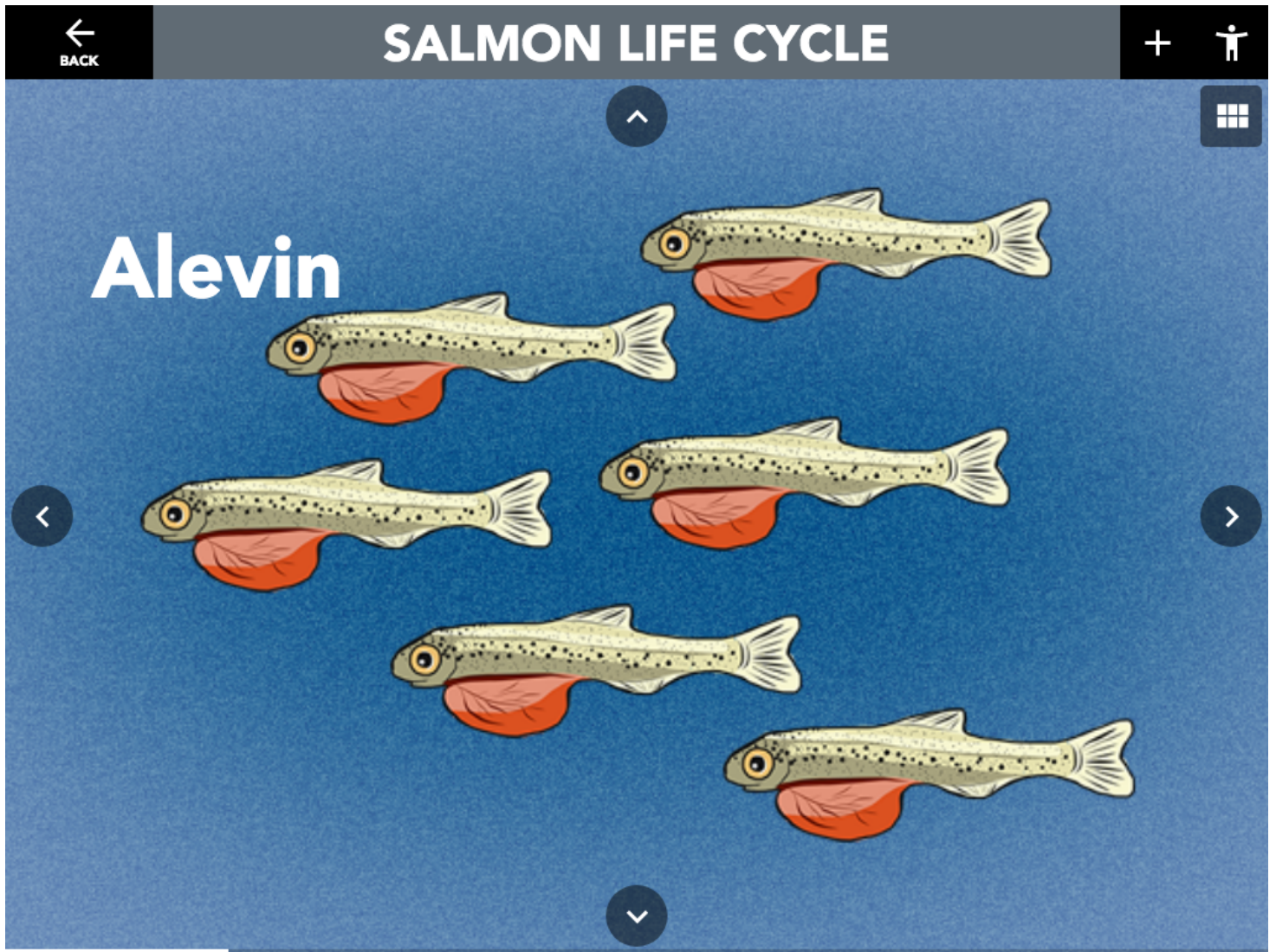


Figure 11: Salmon Life Cycle "Slideshow" activity.

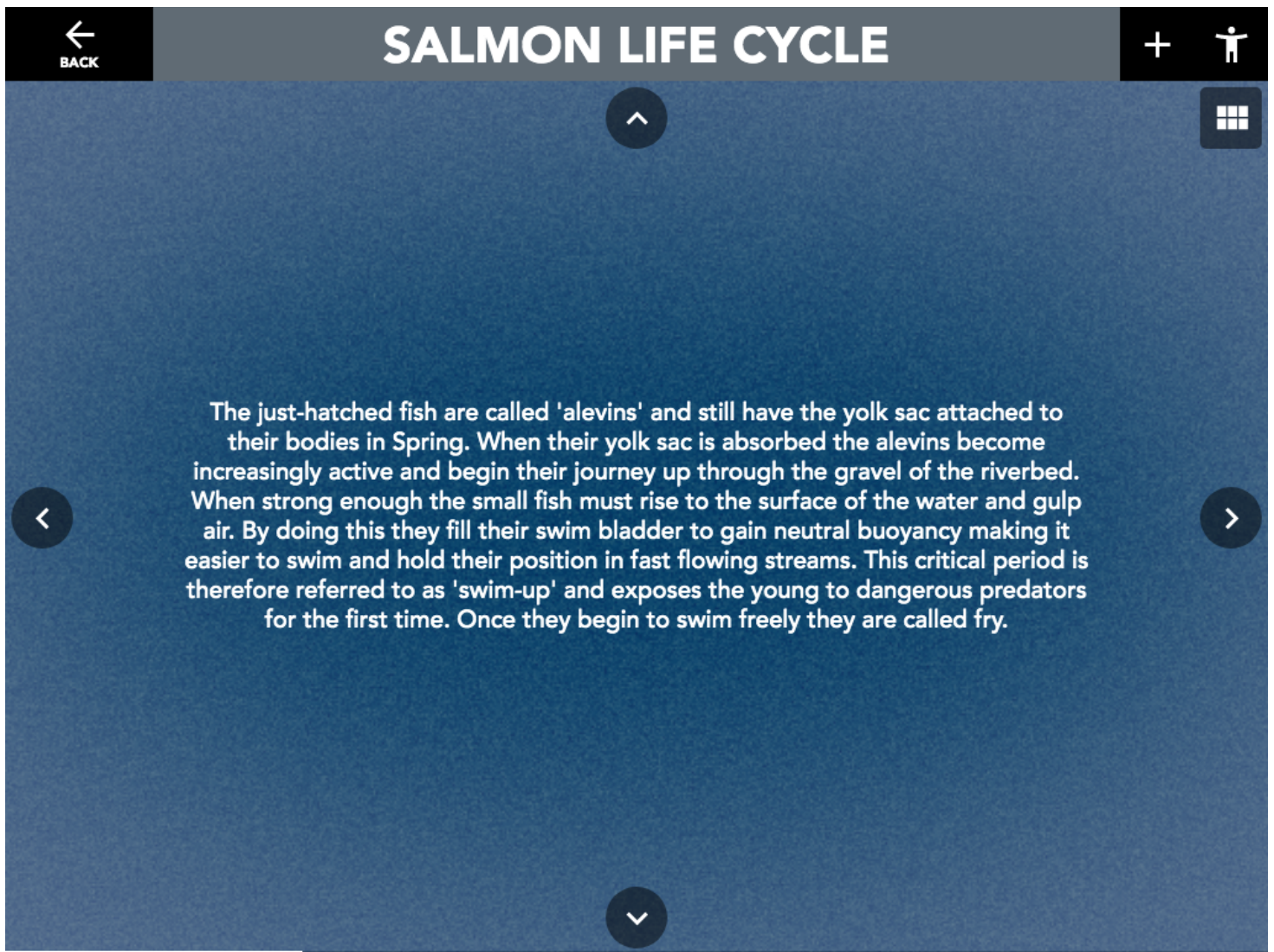


Figure 12: Salmon Life Cycle “Slideshow” activity- basement slides exist beneath slides so users can navigate down to read more in-depth information about the slide they have selected.

FLASHCARDS

This activity displays topic cards that, when tapped, will pull up a brief description (Figure 13). Each card can be configured with additional content that will open in a fullscreen popup.

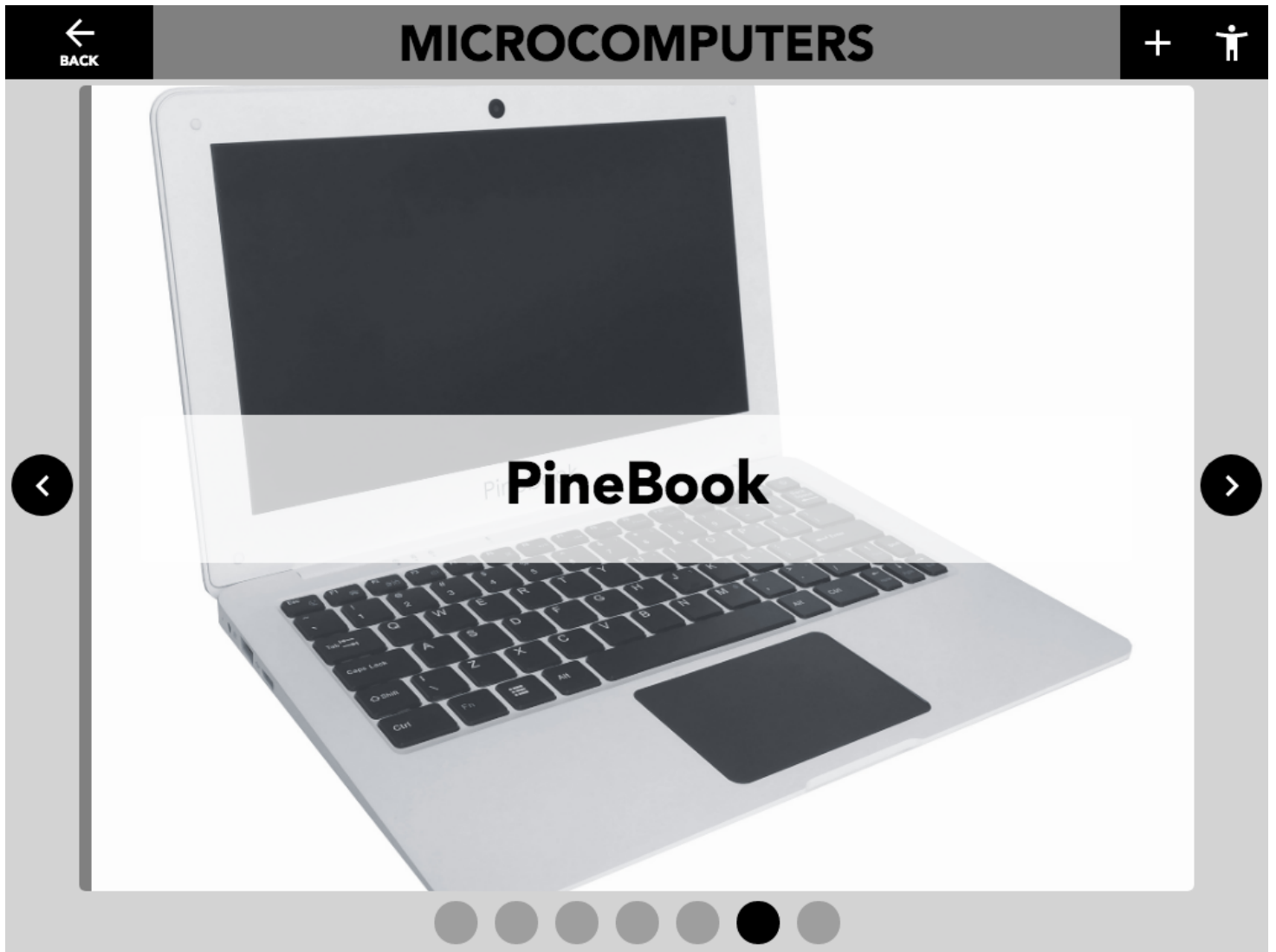


Figure 13a: Single Board Computer “Flashcards” activity.

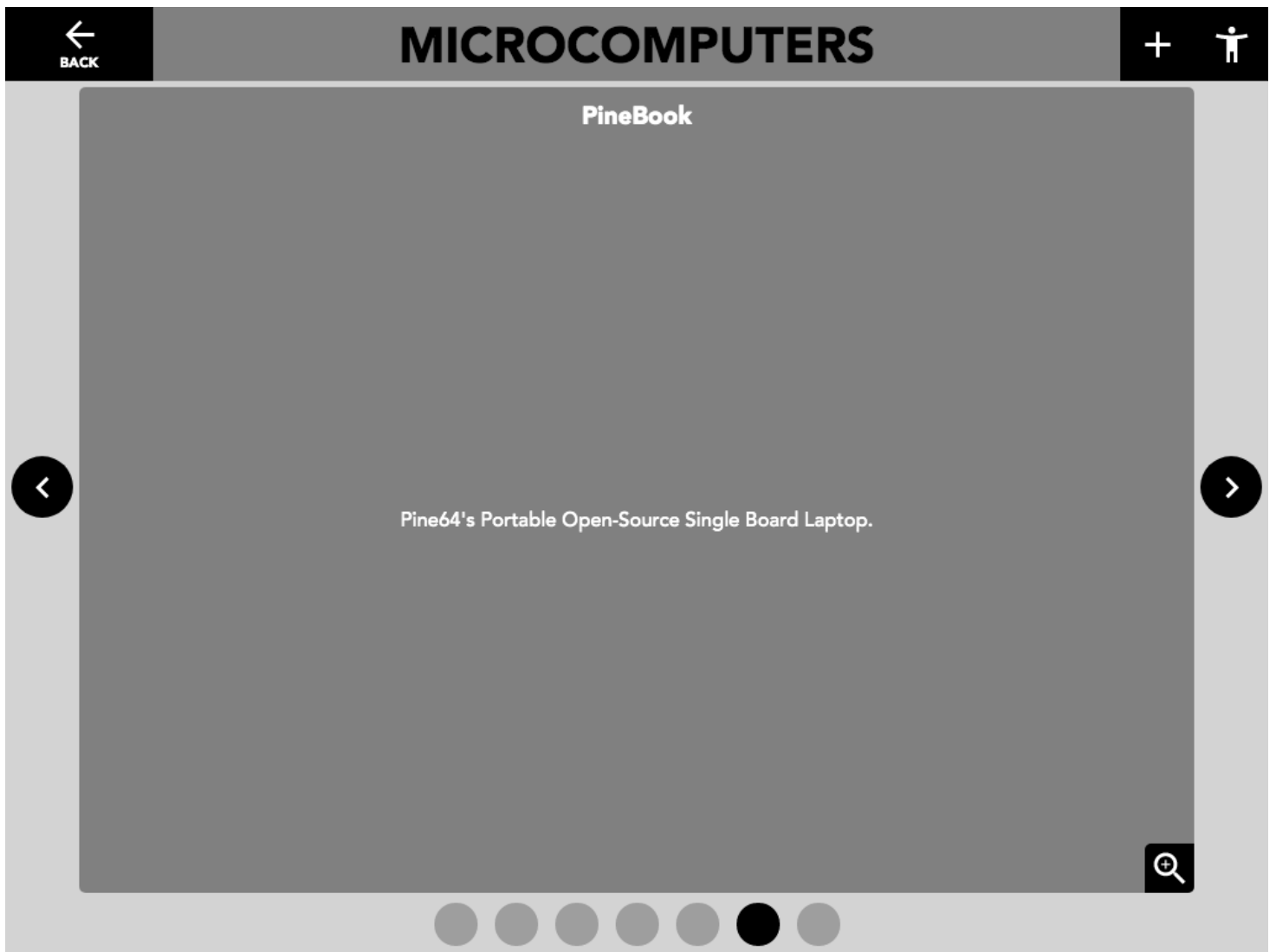


Figure 13b: Single Board Computer “Flashcards” activity. Tapping on a flash card will pull up a brief description and the option for the user to pull up further information.

IMAGE GALLERY

A collection of images can be combined into a gallery that users can look through, with brief descriptions for each image (Figure 14).



Figure 14: Kiosk Prototypes “Image Gallery” activity.

TRUE OR FALSE QUIZ

Though the template is called “True or False Quiz,” the activity can be customized to prompt users to select from any binary set of choices (Figure 15). The buttons can be configured with the desired wording, or left blank (Figure 16). Each time the user answers, they are shown a full screen prompt with a configurable response (Figure 17). Depending on their answer, the buttons allow them to retry the question or move on to the next one (Figure 18).

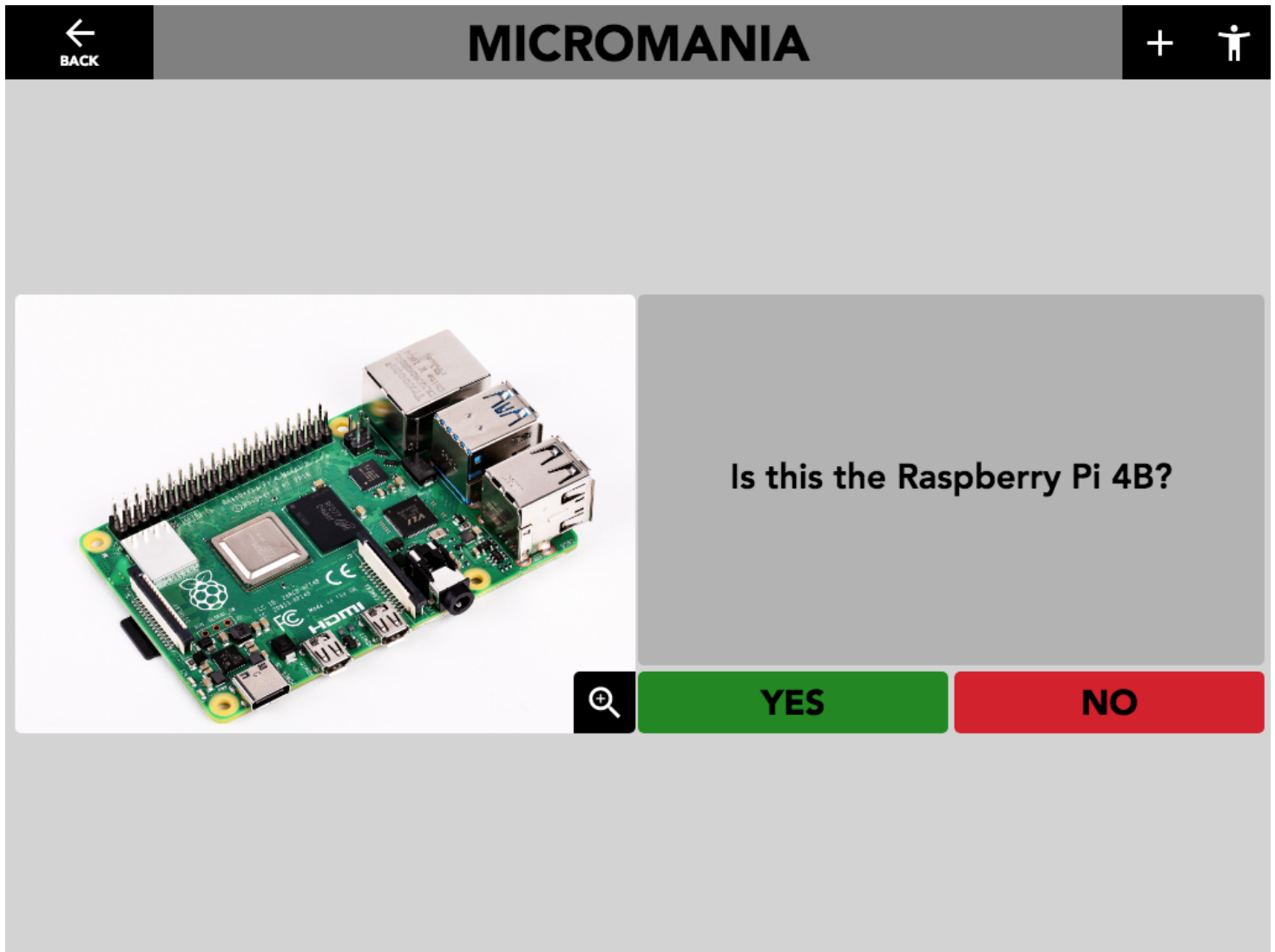


Figure 15: Micromania "True or False Quiz" activity.

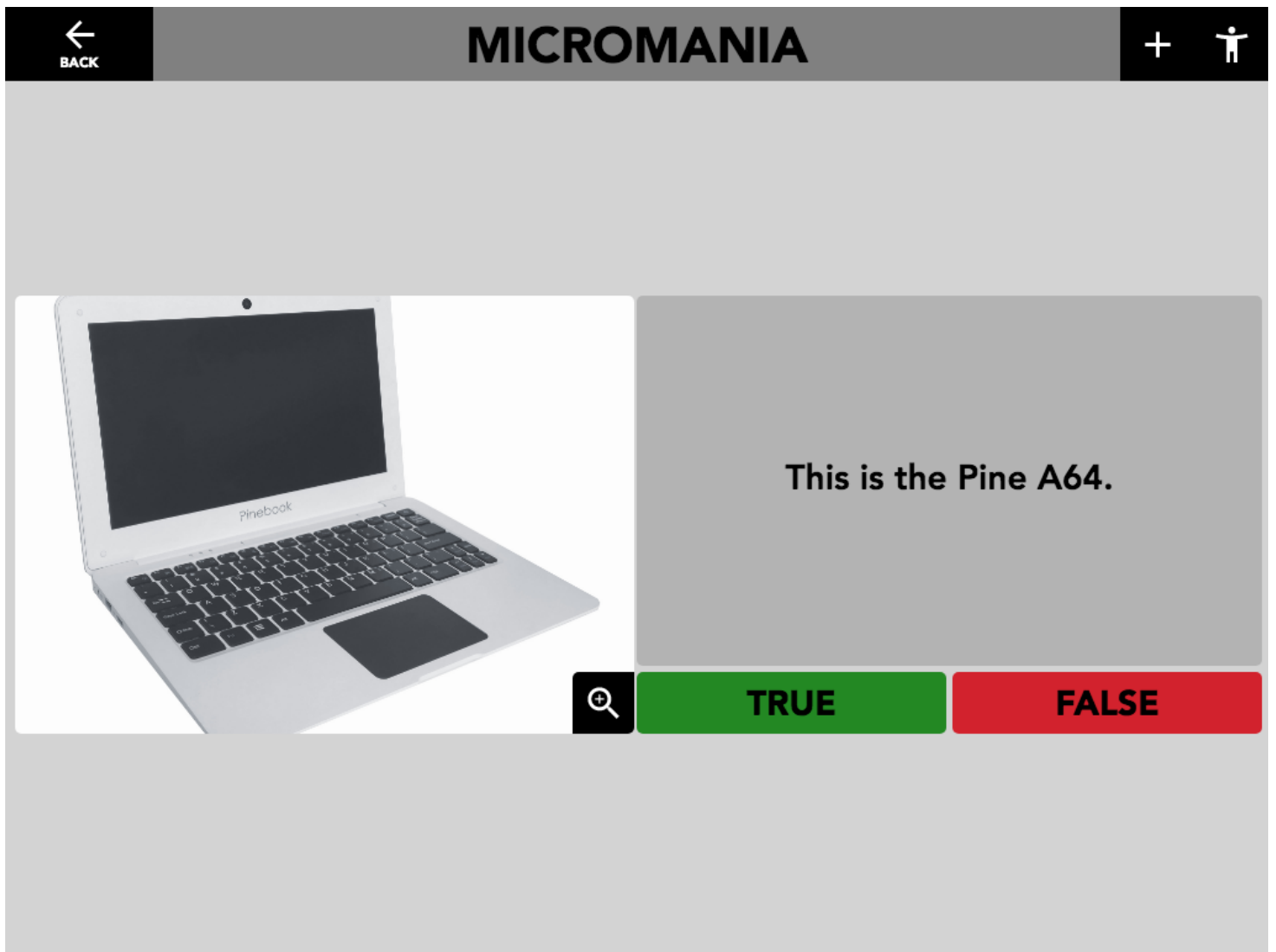


Figure 16: Micromania “True or False Quiz” activity- button text can be configured with different wording or left blank.

This is the Raspberry Pi 4B Single Board Computer.

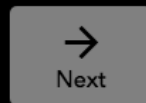


Figure 17: Micromania “True or False Quiz” activity- correct answers show the “correct” response that was configured, as well as a button to move to the next question.

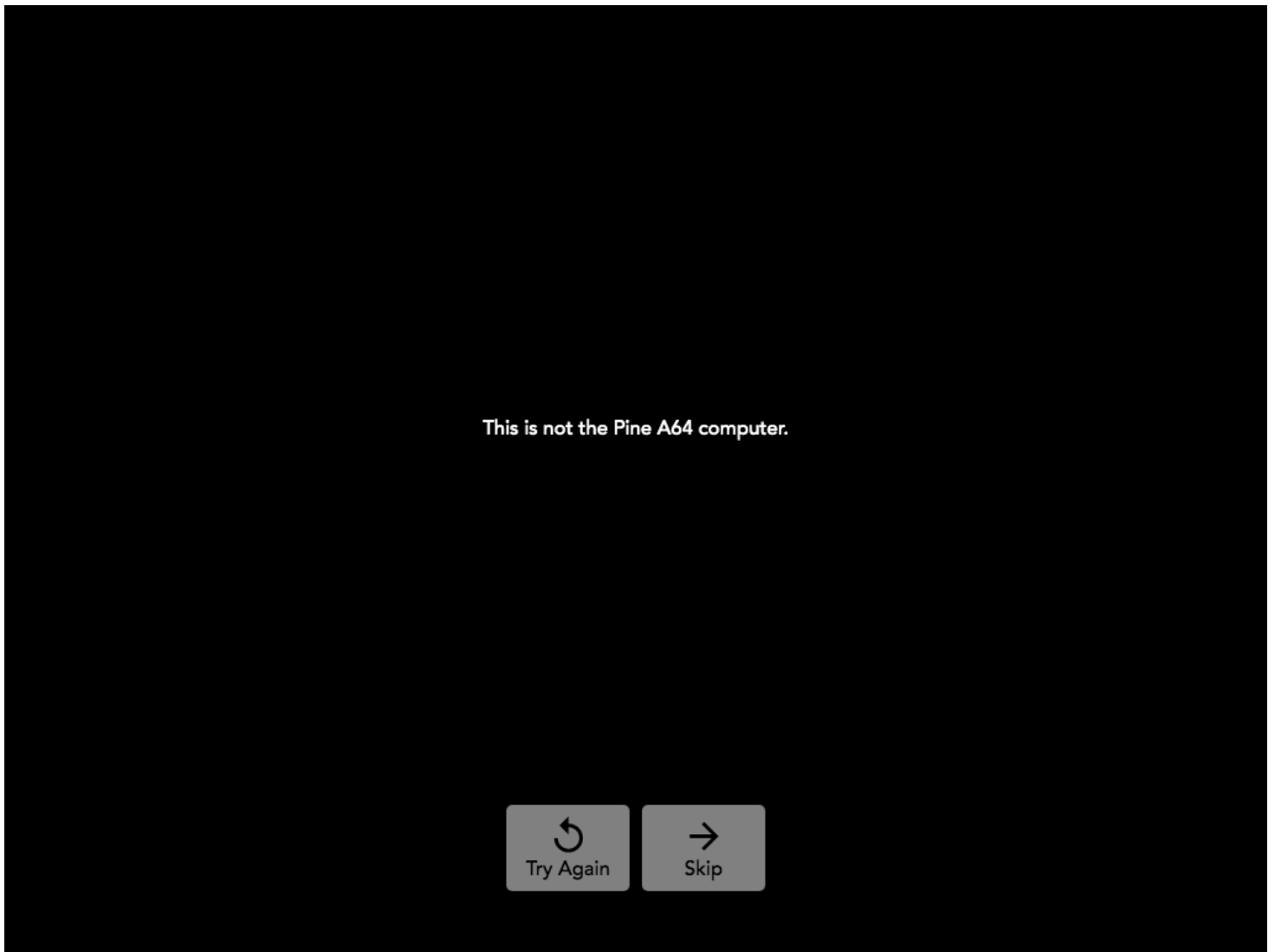


Figure 18: Micromania “True or False Quiz” activity- incorrect answers also show individually customized responses and allows users to move to the next question or try again.

VISUAL THINKING

This inquiry-based learning activity (Wolpert-Gawron, 2016) can be used to highlight a collection of images and provide a prompt so that the user can learn more about the things they find most interesting (Figure 19).

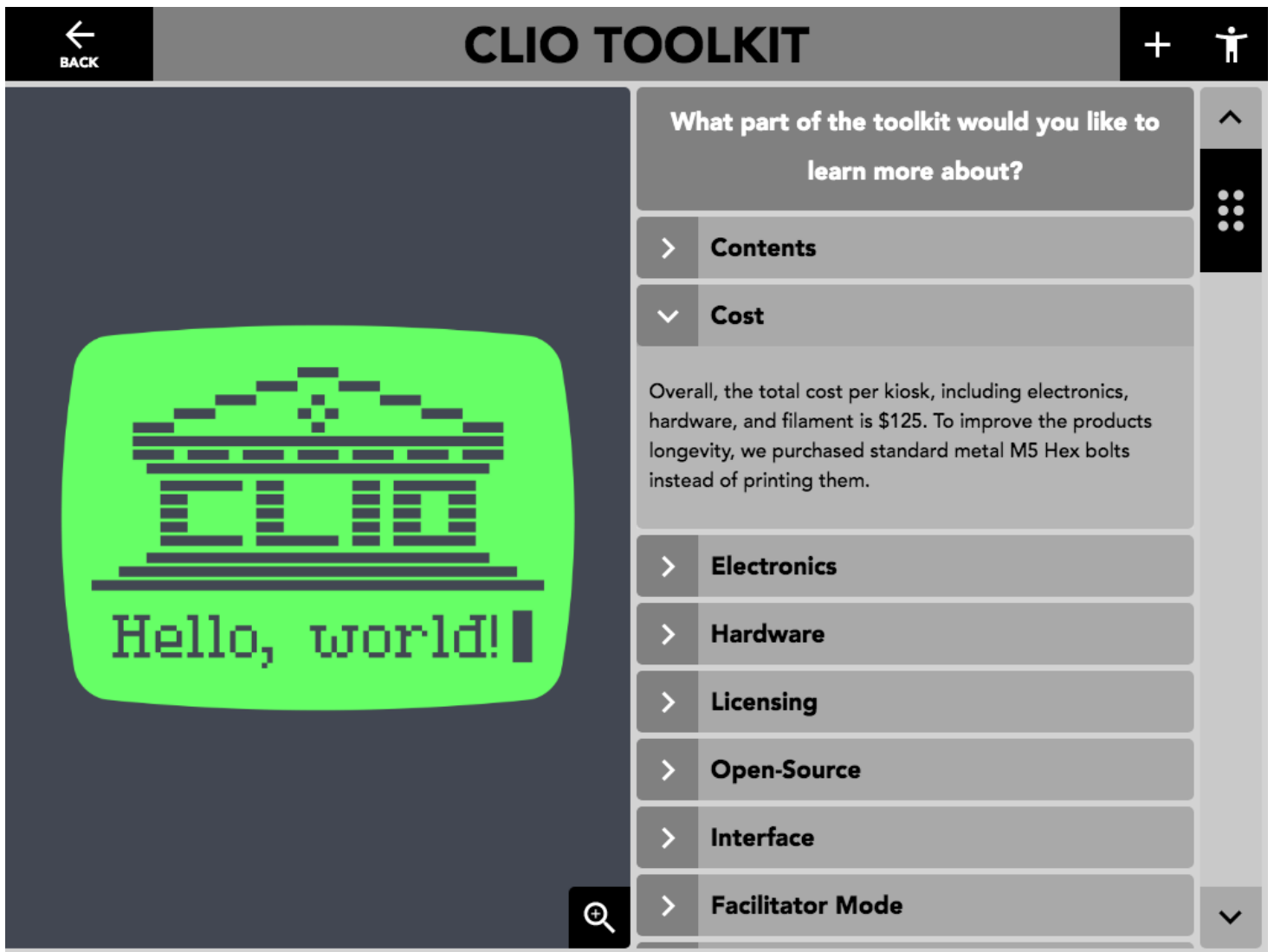


Figure 19: CLIO Toolkit “Visual Thinking” activity. The right side of the screen lists individual bullet points that can be expanded to provide information about that topic.

PLAY VIDEO

This activity plays an MP4 video and can be configured to disable the audio or control bar (Figure 20). Without speakers, this activity could be used to play short animated infographic videos.



Figure 20: Life on Mars “Play Video” activity.

3D RENDER

This activity allows the user to select between a list of objects to load into the frame (Figure 21). Each object can have a customizable rich text description that is opened in a full screen pop-up. This activity can render 3D object files saved in .stl format.

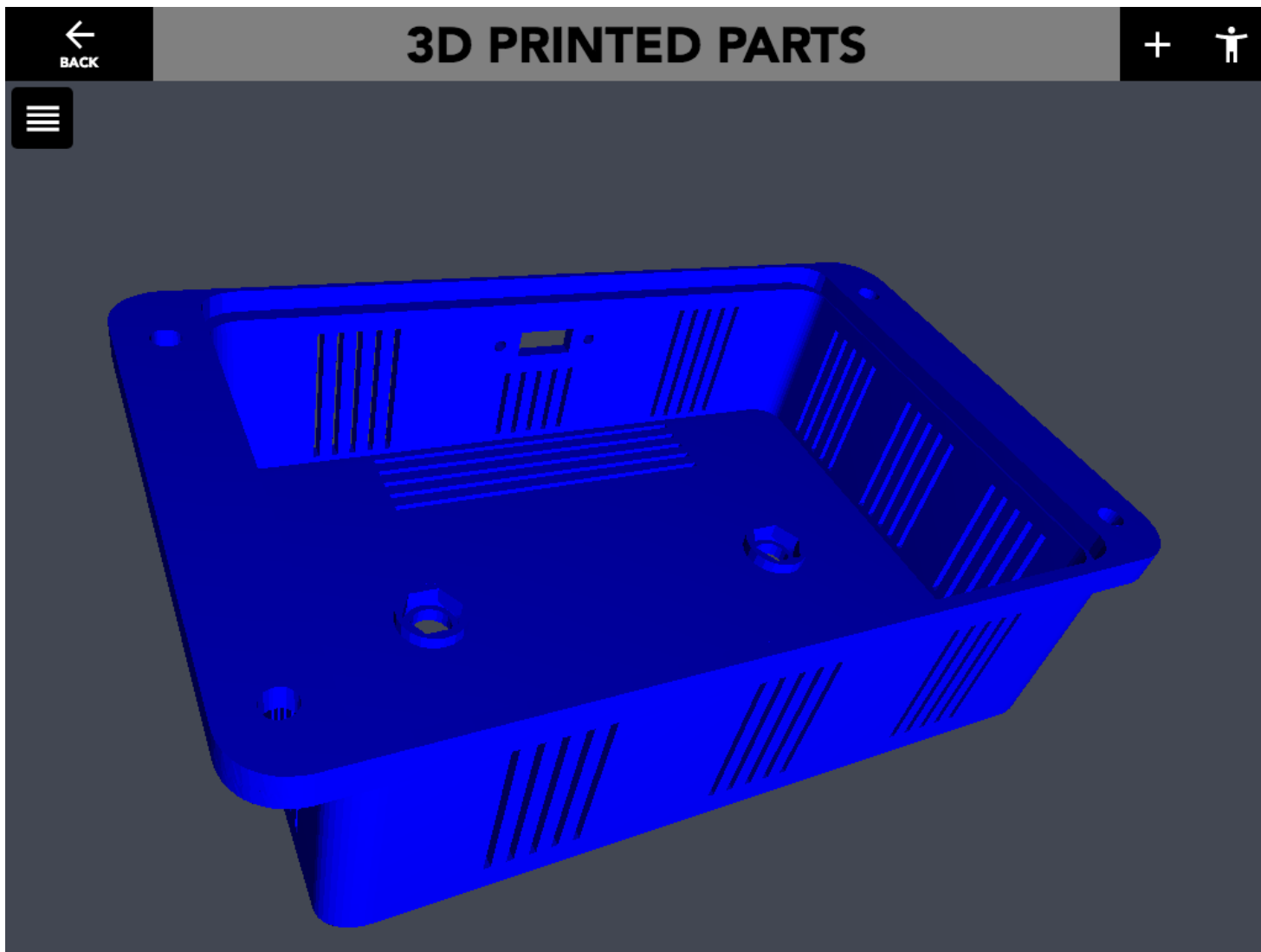


Figure 21a: 3D Printed Parts “3D Render” activity.

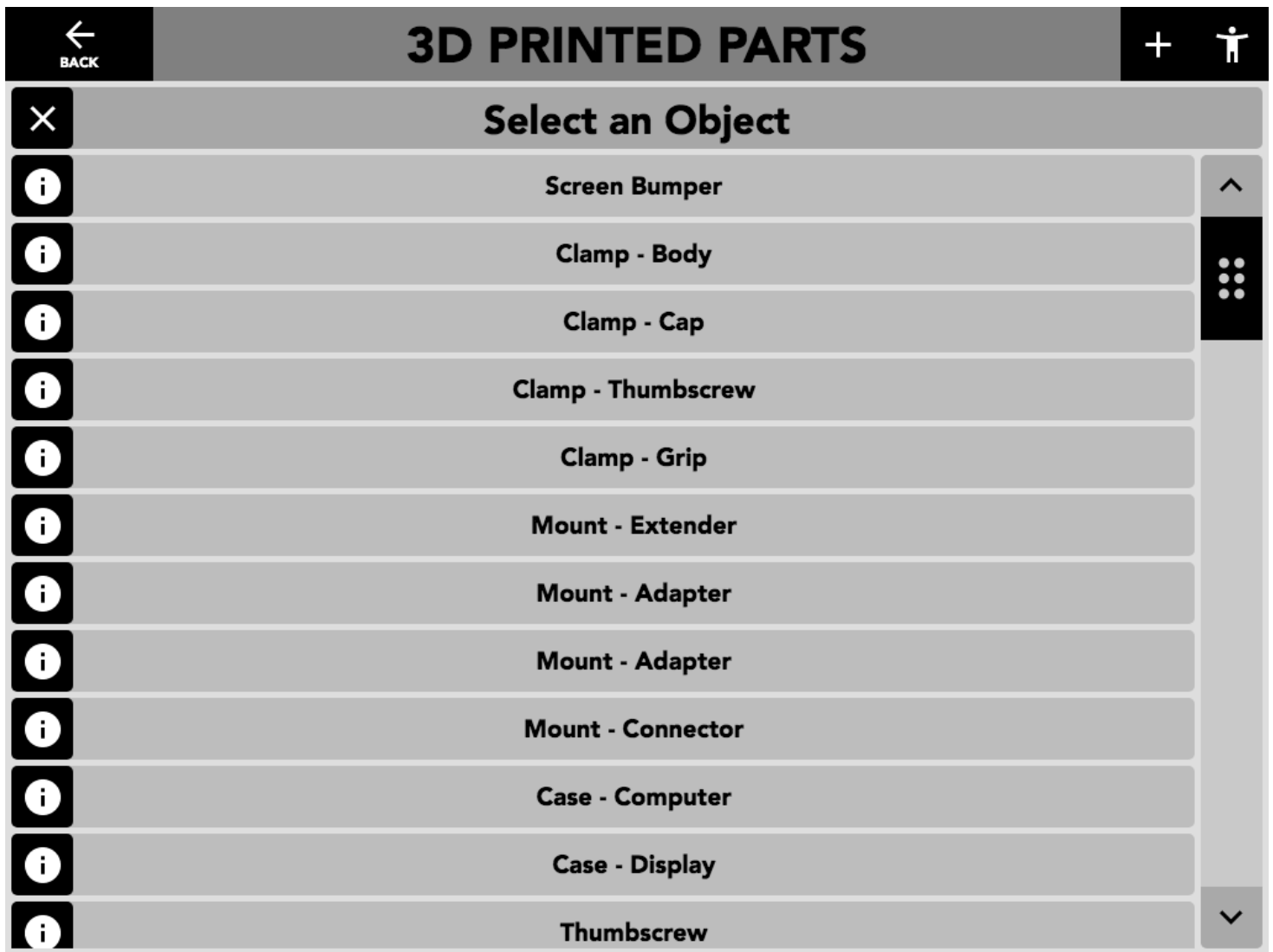


Figure 21b: 3D Printed Parts “3D Render” activity. Tapping the menu icon opens up an overlay that lists all of the available objects and any available information about them.

RICH TEXT WITH NARRATION

This activity can be used to provide the user with rich text. Audio playback can also be included, such as an interview, narration or other form of voiceover, or disabled altogether (Figure 22).



Figure 22: Designing CLIO “Rich Text with Narration” activity.

Activity Configuration

All the content required for an activity to operate is stored on the kiosk within the web application’s “Content” directory. This directory contains all the Programs that are available to be loaded on-demand from the settings menu. Activity content is stored as a JSON (*JavaScript Object Notation*) file within a Program’s directory. The JSON file format allows data to be saved in a way that is easily readable and writable by both humans and computers. Multimedia files, such as images, videos or object renders, are also stored within a Program’s directory. Creating new activities can be done by copying over the desired media files and editing a few points of data from a template using a JSON editor or plain text editor.

Paragraph and text formatting is stored using a lightweight markup language, closely based on the BBCode tag system. This allows formatting information, such as bold, italic and underlined text, to be stored within a plain text format like JSON. For example, storing “[b]Example[/b]” within a configuration file would cause the interface to render everything between the “[b]” and “[/b]” tags as bold, i.e. **Example**. This markup system currently supports paragraphs; headings;

subheadings; bold, italic, underline, and strikethrough text; ordered, unordered and nested lists; images and image sets; and quote blocks.

Software

PHP and Python are utilized in user interactions that require commands to be executed by the hardware's operating system, such as changing display brightness. PHP and Python scripts created for use during our initial pilot testing with the Raspberry 3B+ hardware are included in the toolkit. Additionally, an SD card image is provided that comes with all necessary software, files and scripts required to quickly replicate a kiosk.

Hardware

The toolkit is designed to be hardware- and resolution-independent so that it can be used with any compatible computer, display and input device. Web-based technologies are widely supported by most operating systems through the use of a web browser, allowing institutions with smaller technology budgets to instead repurpose hardware such as unused and secondhand computers.

Pilot Testing of CLIO

Overview

The prototype of a CLIO exhibit kiosk is currently being pilot tested as part of a pop-up educational outreach program through the Washington State Burke Museum of Natural History and Culture. Three kiosks are being created for the Burke Museum's Education department. One kiosk will be integrated into a mobile pop-up exhibit about interconnected ecological networks in spring and summer of 2020. Two additional kiosks are being fabricated for Burke Education to customize with activities for future inclusion in programming.

The project is being designed and evaluated using established frameworks for the design and evaluation of digital museum interactives (Dubois, Bortolaso, Bach, Duranthon, & Marmont, 2011). Evaluation will use the MUSETECH museum technology evaluation framework (Damala, Ruthven, & Hornecker, 2019), with evaluation research being conducted to assess the impact of the kiosk on the museum professionals, program participants, and the Burke Museum as an institution. Evaluation is planned in four phases: an evaluation of the user interface with Burke Museum volunteers, a critique session at the MuseWeb 2020 conference, a focus group and training session with Burke Museum education staff, and several rounds of visitor testing on site at the Burke Museum.

At the conclusion of our pilot test, all of the guides, designs, code, and evaluation data created for the CLIO toolkit will be made available through an online software repository and on the CLIO project website (<http://www.cliomuseums.org>).

Software

The kiosk prototype uses a Debian Linux distribution known as DietPi for the operating system and runs a Chromium browser operating in kiosk mode.

Hardware

The CLIO kiosk prototype is designed to have a mobile and modular form factor. All hardware is easily replaceable in the event of hardware damage or failure. It is constructed using off-the-shelf parts and will not require soldering. Raspberry Pi hardware facilitates easy cloning of the kiosk system, allowing the institution to create additional kiosks as demand dictates.

For the pilot test, all required parts were fabricated using a Creality CR-10 Mini. This includes a case, articulating arm, clamp and bumper-style screen protector (Table 2). To improve the products longevity, we purchased standard metal M5 Hex bolts instead of printing them. Overall, the total cost per kiosk, including electronics, hardware, and filament is \$125 (Table 3). The final kiosk utilizes many designs that are available through online repositories under permissive licenses (Figure 23).

Part	Filament	Amount	Print Time
Protective Screen Bumper	TPU	82g	10h
Clamp – Body	PLA	22g	3h
Clamp – Cap	PLA	2g	15m
Clamp – Threaded Thumbscrew	PLA	8g	1h 30m
Clamp – Anti-Slip Grips	TPU	1g	12m
Articulating Arm – 90mm Straight Extender	PLA	11g	1h 30m
Articulating Arm – Male to Female Connector	PLA	4g	45m
Articulating Arm – Male to Male Connector	PLA	3g	30m
Articulating Arm – Computer Case Connector	PLA	14g	2h
Case – Computer	PLA	118g	17h
Case – Display	PLA	41g	5h
M5 Hex Thumbscrews	PLA	6g	1h
Total:		312g	47h 12m

Table 2: Estimates for print time and required filament. Times may vary based on printer model and print settings. Above are estimates for the Creality CR-10 Mini.

Item	Unit Price	Qty	Cost
------	------------	-----	------

Raspberry Pi 3B+	\$29.95	1	\$29.95
Raspberry Pi Official 7" Touchscreen	\$64.00	1	\$64.00
MicroUSB Power Supply	\$7.50	1	\$7.50
Panel-Mount MicroUSB Extension	\$5.00	1	\$5.00
Right Angle MicroUSB Adapter	\$4.50	1	\$4.50
229g PLA Filament per Kiosk (1kg spool)	\$27.00	0.229	\$6.18
83g TPU Filament per Kiosk (1kg spool)	\$27.00	0.083	\$2.24
M5 x 15mm Hex Nut and Bolt	\$0.80	2	\$1.60
M5 x 25mm Hex Nut and Bolt	\$0.90	4	\$3.60
Total Unit Cost:			\$124.57

Table 3: Total unit cost per individual kiosk including electronics, hardware, and filament.



Figure 23a: Our design uses a clamp to attach to tables, bookshelves and other similar surfaces.



Figure 23b: Kiosk starts in Facilitator Mode to allow Activities to be selected.



Figure 23c: After Activities are selected, the kiosk can be placed in Exhibit Mode.



Figure 23d: Users can navigate into an Activity using the carousel and return home at any time.

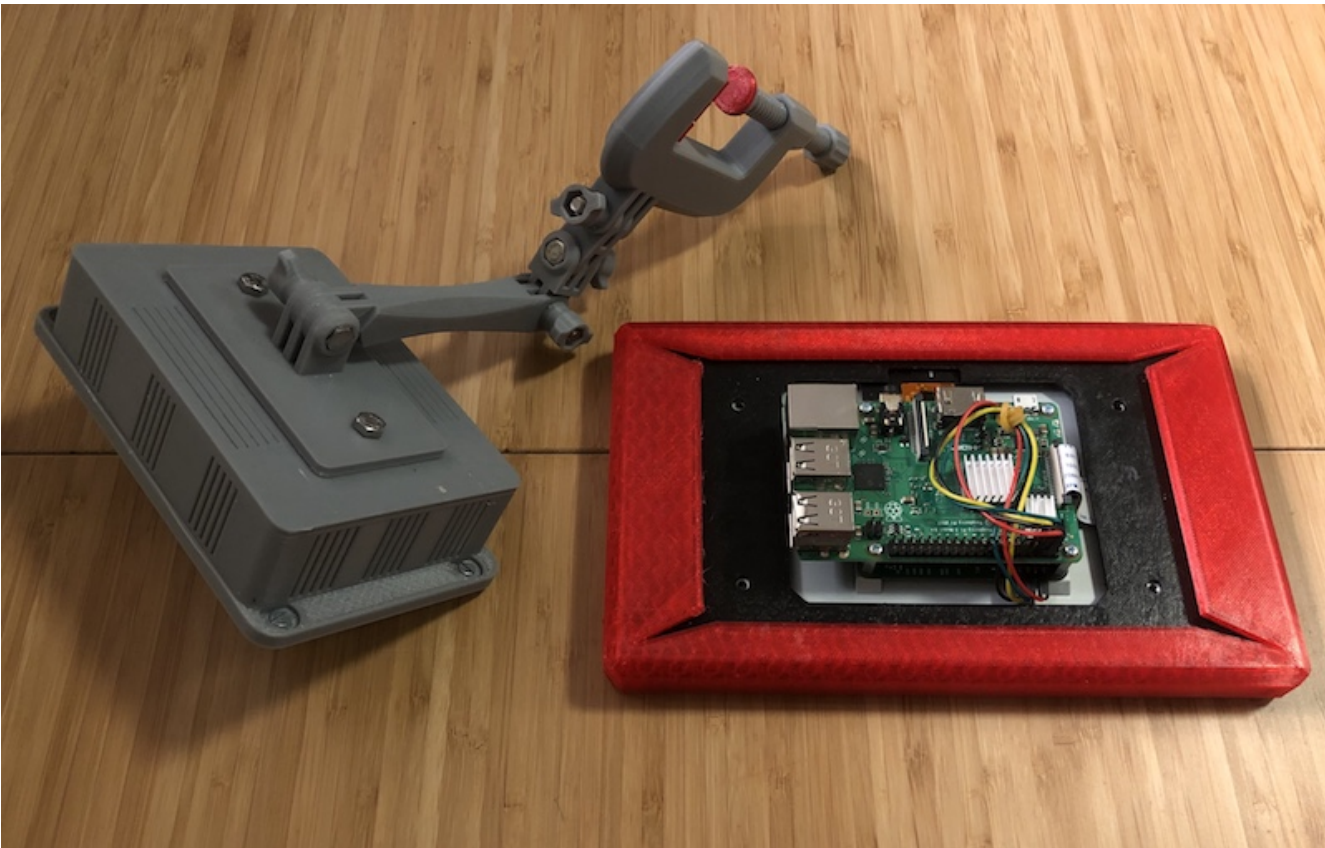


Figure 23e: The kiosk is easy to take apart and components are easily replaced.

The Future of CLIO

The CLIO toolkit is open-source, so any museum or organization can use and customize its content free of charge. As such, we hope to empower museum professionals to create their own digital content for both in-house and outreach programming. The versatile nature of the electronics, combined with the power of 3D printing technology, allows the CLIO toolkit to be customized into form factors other than an exhibit kiosk. It could be used to create a portable tablet, laptop, wall-mounted kiosk, or even as a display integrated into an immersive exhibit environment. The pilot phase of this project has shown that the creation and customization of open-source tools is not only possible, but feasible for museums of all sizes. When considering the future of CLIO, the possibilities for future projects and research are limitless.

Accessibility

While already designed for accessibility, there will always be further improvements that can be made. The inclusion of headphones or on-board speakers would improve accessibility for a wide range of audiences. It would improve multimedia playback and open new avenues for accessibility, such as interface feedback and learning Activities that better utilize video and audio assets. Testing would need to be completed to understand the ambient noise generated by these exhibits.

Content

Activity customization currently requires editing text-based files, but further tools could more easily allow those without computer knowledge to update an activity's content. Designing a web-based platform with a graphical user interface for Activity creation would allow facilitators to create Activities and preview how they would behave on a kiosk. Implementing a user-friendly content creation system opens up a wide range of options, such as the inclusion of language support and improved rich text markup. Additional interactive activity types are also being explored.

Easily the most important part of utilizing open-source and open-access materials is to credit the original content creators. Along with a content creation wizard, we would like to improve the implementation of content attributions to be sure that credit is given where credit is due.

Community

Tools like CLIO are only as good as the community that interact with and shape them. Sharing information and seeking community input is a large part of the power of open-source tool sets. In order to help foster a vibrant and active CLIO community, we would like to create a web blog and online community to facilitate the inclusion of open-source technologies into museums and related institutions. Additionally, this community-focused website could be used to host forums, guides and a repository for additional activity types.

Static Museum Context

The current toolkit is designed for and tested in a pop-up programming environment, but future work with CLIO could investigate ways to further integrate this toolkit into a static museum context. Possibilities include a kiosk to provide supplemental information about curated exhibit content or for use as in evaluation data gathering. Activity usage and duration could also be collected to give a broader overview of kiosk interaction. Additionally, utilizing an open-source NFC/RFID toolkit could potentially add a tactile level of interaction, where visitors are invited to hold an RFID-embedded object to a scanner and tap a button on the kiosk to pull up related activities.

Technical Improvements

CLIO is currently designed to be hardware-independent, but future versions of CLIO could take deeper advantage of the Raspberry Pi hardware through further integration of open-source Python tool sets. Allowing the interface to control system level functions, such as enabling or disabling internet connectivity or utilizing the Raspberry Pi's customizable GPIO inputs, could provide a secure way to implement features like wireless content updates or Arduino-powered motor, sound or lighting exhibit triggers.

Acknowledgements

This project would not have been possible without the financial support of the University of Washington Resilience Lab’s Compassion Seed Grant and the University of Washington Museology Graduate Program’s paid internship pilot. Funding student research and projects fosters innovation and we have been incredibly lucky to have these support options available as we developed CLIO. We also wish to thank Katharine Canning and Pamela Maldonado from the Burke Museum of Natural History and Culture’s Education department for their mentorship and partnership over the last year as we have developed CLIO.

Open-source projects like CLIO are built upon the countless hours of work that open-source developers dedicate to creating tools for everyone. Listed below are the open-source hardware and software tools used to create CLIO (tables 4 – 8).

Hardware	
Raspberry Pi Foundation	https://www.raspberrypi.org/
Adafruit	https://www.adafruit.com/

Table 4: Hardware was sourced through two community-driven manufacturers and distributors.

Case	
Modular Mounting System	https://www.thingiverse.com/thing:2194278
Modular Mounting System – Clamp Grips	https://www.thingiverse.com/thing:3189384
Modular Mounting System – Case Mount (modified)	https://www.thingiverse.com/thing:2837458
M5 Thumb Screw Cap	https://www.thingiverse.com/thing:3843319/files

Table 5: Some 3D parts were designed or modified for this project, while others were sourced through online repositories. Sources for parts that were not custom designed for the toolkit are listed in the table above.

Software	
DietPi Debian	https://dietpi.com/
Apache Server	https://www.apache.org/
Chromium	https://www.chromium.org/
Unclutter	https://packages.debian.org/jessie/x11/unclutter
Matchbox Window Manager	https://packages.debian.org/sid/matchbox-window-manager

Table 6: All software used for this project is available through the Debian package manager.

Interface	
HTML5 Boilerplate	https://html5boilerplate.com/
Material.io Icons	https://material.io/resources/icons/?style=baseline
jQuery	https://jquery.com/
BBCode Parser	http://frug.github.io/js-bbcode-parser/
Fancybox	http://fancyapps.com/fancybox/3/
mCustomScrollbar	https://github.com/malihu/malihu-custom-scrollbar-plugin
Alertify	https://alertifyjs.com/
Select2	https://select2.org/
Owl Carousel	https://owlcarousel2.github.io/OwlCarousel2/
Hammer	https://hammerjs.github.io/

Table 7: The interface is rendered dynamically using client-side technologies.

Activities	
TwentyTwenty	https://zurb.com/playground/twentytwenty
jPlayer	http://jplayer.org/
Three	https://threejs.org/
Madeleine	https://github.com/JinJunho/Madeleine.js/
Timelinr	https://github.com/juanbrujo/jquery-Timelinr
Reveal	https://github.com/hakimel/reveal.js

Table 8: Similar to the interface, Activities are rendered dynamically through JavaScript.

The CLIO Team

Dillon Connelly is a museum studies graduate student at the University of Washington studying the effects of open-access and open-source technology in museums through a lens of equity and access. Joshua Frechette is a designer and developer that believes that accessibility is about more than just function, but is itself a way of creating community collaboration.

Bibliography

Beagley, L. "Curb Cut Thinking: An Argument for Universal Design". Published January 4, 2017. Consulted February 15, 2020. <https://ui.asu.edu/blog/curb-cut-thinking>

Damala, A., & Ruthven, I., & Hornecker, E. "The MUSETECH Companion : Navigating the Matrix". Published January 2019. Consulted February 15, 2020. <https://www.semanticscholar.org/paper/The-MUSETECH-Companion-%3A-Navigating-the-Matrix-Damala-Ruthven/1b2f4976105a4491469d321da9036f3bfa2087f2>

Dickerson, Z. (2018). "Size matters! Accessibility and Touch Targets." Last updated April 4, 2018. Consulted February 15, 2020. <https://medium.com/@zacdicko/size-matters-accessibility-and-touch-targets-56e942adc0cc>

Dubois, E., Bortolaso, C., Bach, C., Duranthon, F., & Blanquer-Maumont, A. "Design and Evaluation of Mixed Interactive Museographic Exhibits". Published October 2011. Consulted February 15, 2020. https://www.researchgate.net/publication/259714620_Design_and_Evaluation_of_Mixed_Interactive_Museographic_Exhibits

Langer, M. & Alderman, J. "Open hardware belongs in your museum." MW2016: Museums and the Web 2016. Published January 15, 2016. Consulted December 27, 2019. <https://mw2016.museumsandtheweb.com/paper/open-hardware-belongs-in-your-museum/>

Price, K. & James, D. "Structuring for digital success: A global survey of how museums and other cultural organizations resource, fund, and structure their digital teams and activity." MW18: MW 2018. Published January 31, 2018. Consulted December 29, 2019. <https://mw18.mwconf.org/paper/structuring-for-digital-success-a-global-survey-of-how-museums-and-other-cultural-organisations-resource-fund-and-structure-their-digital-teams-and-activity/>

Ramel, D. (2018). "Developer Economics Survey: Data Science, Machine Learning Are Most-Wanted Skills." Retrieved December 29, 2019, from <https://adtmag.com/articles/2018/09/24/developer-economics-survey.aspx>.

Reeves, J. (2015). "Making Simple Ideas Simpler." Last updated March 31, 2015. Consulted February 15, 2020. <https://www.uxbooth.com/articles/making-simple-ideas-simpler/>

Rello, L., & Baeza-Yates, R. "Good Fonts for Dyslexia". Published 2013. Consulted February 15, 2020. http://dyslexiahelp.umich.edu/sites/default/files/good_fonts_for_dyslexia_study.pdf

Segue Technologies. (2013). "Client-Side vs. Server-Side Code: What's the Difference?" Last updated May 1, 2013. Consulted February 15, 2020. <https://www.seguetech.com/client-server-side-code/>

Stack Overflow Developer Survey 2019. (2019). Retrieved December 29, 2019. Consulted February 15, 2020. <https://insights.stackoverflow.com/survey/2019>.

W3C. "Accessibility Requirements for People with Low Vision". Published March 17, 2016. Consulted February 15, 2020. <https://www.w3.org/TR/low-vision-needs/#text-contrast>

Wolpert-Gawron, H. "What the Heck Is Inquiry-Based Learning?" Published on August 11, 2016. Consulted February 15, 2020.