Jigsaw Hardware

Table of Contents

Section 1: User Manual

System Overview

Figure 1: The Jigsaw Puzzle Simulator

This is the jigsaw puzzle device. It’s designed to be an accessible, easy-to-use console for the elderly, since real jigsaw puzzle pieces are small and difficult to manipulate. Once connected to the internet, it also is capable of multiplayer gameplay between it and other jigsaw puzzle devices.

Specifications:

- Voltage Rating: 12V
- Current Rating: 3A
- Dimensions: 15.75” x 12.5” x 8.4”
- Weight: ~8 lbs.

System Setup

Turning the Console On and Off

Turning the console on:

1. On the back panel, plug the provided power cord into the circular plug.
2. Connect other end to standard wall outlet
3. Once connected, console may boot up on its own, indicated by a soft click and the trackball lighting up
    a. If it does not, press the silver button on the same panel as the power plug
    b. If it still doesn't power on, try again with a different wall outlet
4. Wait until you see the Jigsaw Puzzle home screen
5. All done!

Turning the console off:

1. Press and hold the silver button on the back of the console for five seconds
2. If this does not work, unplug the console from the wall

Connecting to the Internet

⚠ ADMIN USE ONLY ⚠

Requires: USB keyboard (wired or wireless), Desired Wi-Fi name and password.

All commands will be in **bold** and surrounded by quotes. Type these in while excluding the quotes

1. Boot up machine (Plug in and press power button)
2. Select "Quit game" with built-in trackball and main button
3. Connect keyboard to USB port on lefthand side of appliance
4. Wait until bottom line of screen says "**capstone@raspberrypi:~ $"**
5. Type "**nmcli radio wifi on**"
6. Type "**nmcli dev wifi list**"
    a. You can now see a list of available networks
    b. Take note of your desired network's "SSID" column
    c. If you don't see your desired network, it isn't available to connect to
7. Type "**sudo nmcli --ask dev connect** "
    a. **"<SSID>"** replaced with Wi-Fi name
    b. If prompted, type in the Wi-Fi password and hit enter
    c. You may not see it being typed in. This is normal, the password is being entered
8. To check if the Wi-Fi is connected, type "**nmcli dev wifi list**” again
    a. The connected network will have an asterisk in the "**IN-USE**" column
9. Once it's connected, type "**sudo shutdown**" and the appliance will power off
10. Congrats, Wi-Fi is now set up!

How to Play

Controls

See Figure 1 for physical control references

- Moving the cursor: Roll the illuminated trackball
- Interacting with menu: Press the big red button while cursor is over desired option
- Volume up/down: White buttons on lefthand side of main panel
- Zoom in/out: Green buttons on righthand side of main panel
- Pick up piece: Press and release big red button while cursor is over desired piece
- Move piece: Roll trackball while piece is picked up
- Drop piece: Press and release big red button
- Pan camera: Press and release red button while not over piece to engage pan, move trackball to adjust, press and release red button again to disengage

---

Troubleshooting

Common errors:

- Game closes revealing the console (white text on black background)
- Game freezes and player is unable to continue playing
- Game ‘softlocks’: not frozen, but the user is stuck somewhere in the game they can’t get out of

Solution:

- Power off console and power console back on
    - Instructions for this can be found in “Turning the Console On and Off”

Section 2: Hardware Documentation

Introduction

Caring for our elders is a very important task. There are many ways we care for older generations in this country, and each has their own challenges. For example, nursing homes are a great way to ensure that the elderly get the care they need, and they stay in a community of others where they can continue their social lives and have fun. But this comes at the cost of leaving their lifelong home and living somewhere unfamiliar. Our client, Wilshire Community Services, uses a different approach. Wilshire cares for their clients from the comfort of each individual's own home, without having to move them to a nursing home. This approach doesn't give the client as much access to activities and social events, though. Wilshire clients live in separate homes and can sometimes feel isolated. Because of this, Wilshire is looking for ways to connect their clients with each other to create that feeling of community.

Project Overview

As a solution to Wilshire's problem, a Cal Poly psychology professor, Dr. Bartlett, reached out to students looking for a capstone project. Our team designed a stand-alone system for Wilshire's clients, focusing on accessibility and simplicity that runs a video game. This game is a simple jigsaw puzzle simulator, designed for users on computers to click and drag pieces around. It is meant to act as a replacement for physical jigsaw puzzles, because small physical pieces can be difficult to manipulate. Additionally, the game includes a multiplayer aspect, so users can connect with each other to play together across multiple devices. Starting play is as simple as plugging the device in, and there are only two necessary controls: a trackball mouse to move a cursor on screen, and a button to pick up and place pieces. Other settings are available as well, including zoom in/out and volume up/down, which aren't required to interact with to play the game but will make the user's experience better. Dr. Bartlett and her students have been essential in the design process, connecting us with Wilshire's clients and giving us feedback on how we can continually improve our product.

Deliverables

Our final product is a self-contained appliance that boots into a jigsaw puzzle application which can be played without any external peripherals. From the user's point of view, it should be a black box that just works and doesn't require any equipment or setup besides plugging into an outlet. An integrated screen displays the game with a cursor that can be moved with a trackball and clicked with an arcade-style button. Other functionality like volume and zoom will also have labeled buttons near the trackball and main clicking button. We have created and deployed 3 of these devices, with 2 more nearly finished.

Goals and Objectives

This project has one overarching goal: simplicity. Every facet of this project should be as simple as we can make it. It should be simple for the user to set up and play. It should be easy and cheap to make more devices. For the user, the design's simplicity comes from how we configure the controls. We met with Dr. Bartlett and discussed some ideas for what kind of user-interfacing hardware would work best. She informed us that our elderly customers will all be fully physically and mentally capable, so we should focus on making the system familiar, and after that emphasize ease of access.

The device will also need to have speakers to give the users audible feedback as well as visual. Lastly and most importantly, the device had to be multiplayer-capable. The main goal of this project is to create a sense of community and connection among the Wilshire residents; it's not just creating a jigsaw puzzle game. This task relied mainly on the software team for communication between boxes, but we had to ensure that our hardware could connect to the internet, which was possible through each box's central processor.

Engineering Specifications

This device runs on a Raspberry Pi 5, which interfaces with all of our peripherals, including user controls and the monitor. The Raspberry Pi runs on 5V and draws up to 3A of current, while the monitor runs on 12V and draws 3A. The box takes in a 12V 3A power input to accommodate all components. The chassis is 15.75 inches wide, 12.5 inches long, and 8.4 inches tall, weighing approximately 8 pounds.

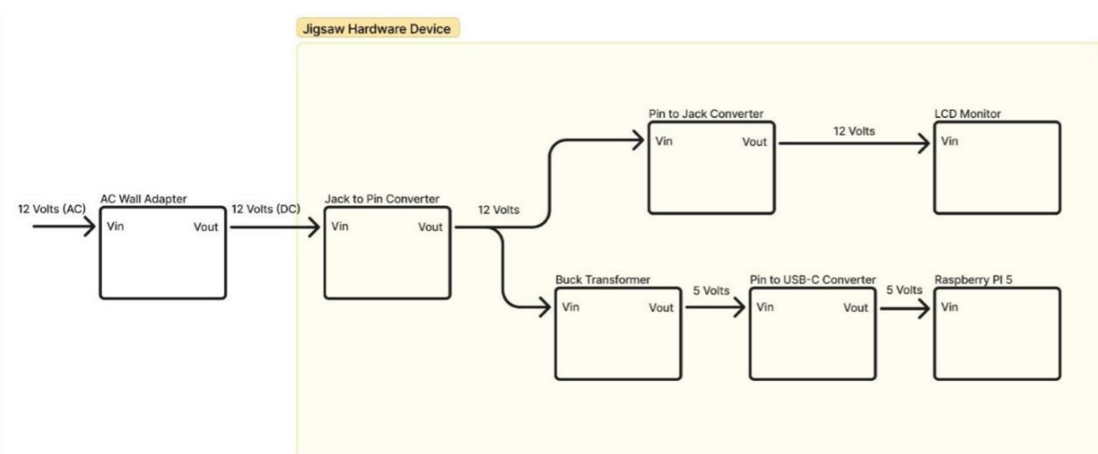
Design Development

We ultimately decided on an arcade machine-style layout for the hardware implementation of the jigsaw puzzle game. We designed the device to use a large 17.3” monitor to account for differing levels of user vision, as well as large buttons with a satisfying ‘click’ and a trackball to ensure that older adults with minor motor skill issues or arthritis could use the device without any major problems. On the software side of the implementation, we also wanted to ensure that non-technically savvy users could use the device. This led to the design decision to have the jigsaw puzzle game, and all the accompanying scripts required to allow for player inputs, execute on boot, which would allow the users to begin playing the jigsaw puzzle game immediately after turning the device on rather than having to manually run the executable files.

System Architecture

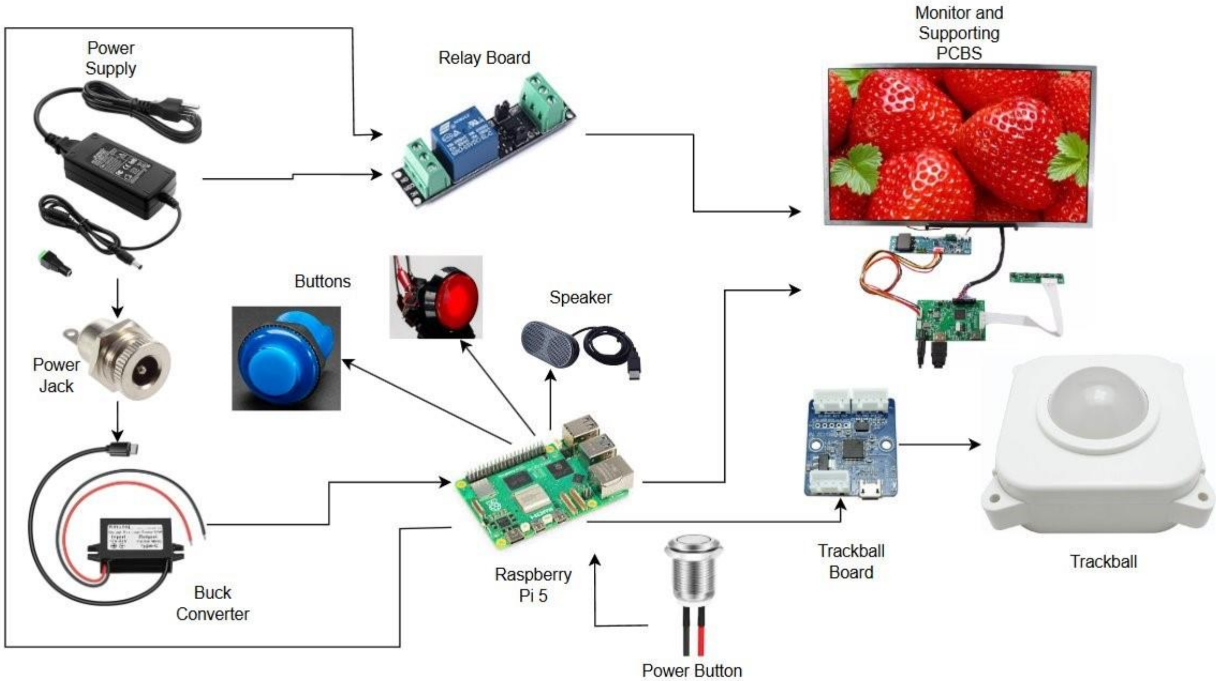
A subsystem of note is our power system. One of our constraints is that we can only have a single power cable coming out of the device. The trouble there is that our screen and our Raspberry Pi require two different voltages, 12V and 5V respectively. To solve this problem, we have a 12V power cable coming out of the device which is then split into the screen and a buck down switch. The buck down switch takes in 12V of DC power and outputs 5V via a USB-C connector that plugs directly into the Raspberry Pi.

Figure 1: Block diagram of power subsystem



Hardware Block Diagram

Figure 2: Illustrated block diagram of all components and connections



Component Overview

- Large button.
  - Control for the user to pick up and place puzzle pieces. Its size and central placement are designed to emphasize its importance to the user. This button is one of the components mounted front and center for the user to interface with.
- Trackball.
  - Control for the user to move the cursor around on the screen. It's intuitive, easy to use, and allows for very fine position adjustment. It can also light up, which is a helpful indicator that the device is powered on.
- Trackball circuit board.
  - Adapts the serial trackball input to a USB output, allowing the trackball to be recognized by the Raspberry Pi as a mouse.

- 2x Smaller zoom buttons.
  - Controls for the user to zoom the puzzle in and out. These controls aren't necessary for gameplay but are useful. They are to the right of the main controls and clearly marked.
- 2x Smaller volume buttons.
  - Controls for the users to adjust the volume of the device. These controls aren't necessary for gameplay but are useful. They are to the left of the main console and clearly marked.
- On/Off button.
  - Button on the back of the device to power it on and off. Press once to power on, press and hold for 5 seconds to power off. When pressed, the button shorts two pins on the Raspberry Pi, which initiates a shutdown / power on signal. On shutdown, it allows the device to safely shut down.
- 17.3-inch LCD screen.
  - Monitor to display the Jigsaw Puzzle game. It is mounted at a 30º angle to be easily visible to users sitting at a table or playing the game in their lap.
- Supporting PCBs.
  - Two PCBs to power the monitor and display the Jigsaw Puzzle game. An additional (optional) PCB can be plugged in to access the monitor's menu.
- Raspberry Pi 5.
  - Central processor for the Jigsaw Puzzle game. Sends HDMI data to the monitor and interfaces with all peripherals.
- Heat sinks.
  - Copper heat sinks that adhere to components on the Raspberry Pi to dissipate the excessive heat it generates.
- HDMI to micro-HDMI cable.
  - Cable to connect the video output from the Raspberry Pi (micro-HDMI) to the monitor (HDMI)

- Micro-USB to USB cable
  - Cable to connect the trackball PCB output (micro-USB) to the Raspberry Pi (USB)
- SD card.
  - Memory card for the Raspberry Pi containing its operating system and program data.
- 12V buck down transformer.
  - Device to step down input voltage from 12V to 5V to power the Raspberry Pi via USB-C.
- 12V AC adapter.
  - Power adapter to plug in to a wall outlet and power the Jigsaw Puzzle game console.
- Power jack.
  - Component mounted on the back of the device to secure the connection to the wall power adapter.
- Relay board.
  - Component to control power to the monitor. It is open while the Raspberry Pi is powered on, and it closes when the Raspberry Pi is shut off.
- Speaker.
  - Provides an additional sensory input for the user. It plays a ‘ding’ when a piece is placed correctly.
- Wooden chassis / 3D-printed PCB mounts
  - Laser-cut frame for the device comprised of 6 wooden panels connected by metal angle brackets. Designed with holes to securely mount all hardware. PCBs are secured to 3D-printed plates, which are then fixed to the chassis.
- Rubber feet
  - Allows for the frame to be elevated slightly to allow airflow beneath the Raspberry Pi. Also helps keep the frame from sliding around while being used.

Software Algorithms

All software algorithms are limited to a script that runs on the Raspberry Pi on boot. We have a Python script that activates the trackball and buttons along with the executable file which runs the jigsaw puzzle application. Each file is fully self-contained and located within a shared directory. The script that runs on boot is written in bash and executes those files. The Python script waits for a voltage to appear on one of 5 pulled-down pins which are connected to a 5V pin via one of the buttons. When a monitored pin receives a voltage, a keystroke is executed, corresponding to whichever button was pressed using the built-in library gpio-key. It runs as a system administrator so one of the keystroke options is able to be a mouse click.

Mechanical Design

The mechanical design of the jigsaw puzzle hardware device primarily consists of its chassis, which will contain all of the interactive and electrical components required to operate the device. We utilized Onshape, a free CAD software tool, to create an initial design prototype for the chassis of the hardware device. This drafting process included measuring the physical dimensions of all the device components to ensure that they would fit. For the internal components housed within the device chassis, we have added mounts that securely hold the systems in place to prevent possible damage when moving the device. Throughout this quarter, we laser cut, tested, and refined this model, resulting in the finalized design seen in Figure 3.

Figure 3: Onshape CAD design of the jigsaw hardware testing chassis.

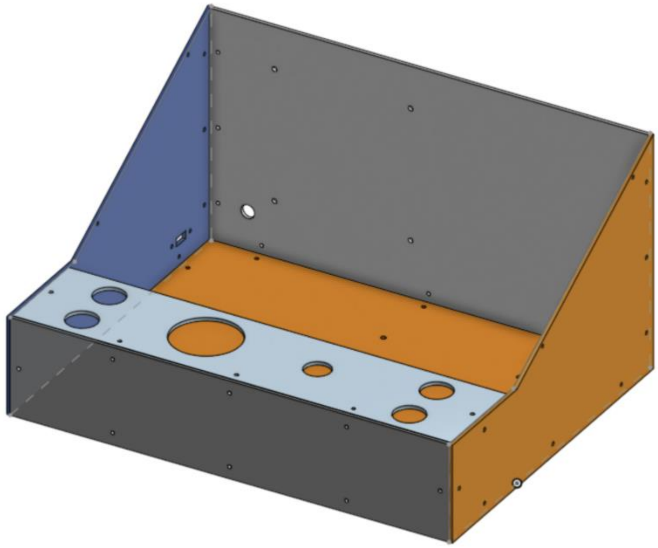


Figure 4: Top view of the current wooden chassis and components.

Figure 5: Side view of the current wooden testing chassis, with mounted components.

Future Improvements

As this school year comes to an end, we looked back at the requirements we defined for our project back in the Fall and reflected on what can be improved in the future

- Device connects the people of the Wilshire community through multiplayer features.
    - Multiplayer features have been partially implemented in software but not thoroughly tested on our hardware devices.
- Five functional devices.
    - We currently have four functional devices, and one more nearly done but not complete. It's missing an HDMI cable.
- Chassis button plate is not very stable.
    - The small wooden plate that the buttons are attached to is flimsy; it could benefit from a spacer inside the box supporting its weak end.
- Heatsinks are not completely effective
    - Several heatsinks are affixed on the Raspberry Pi, but the board still gets very hot. It can run without fail, but the heat it generates can be felt from the outside of the box, which over time, and if left on for several hours, could pose a problem.
- The extra click on boot
    - Currently when the device boots up, the game does launch into full screen. However, the Raspi OS does not focus on the program. The program will react to mouse location (buttons highlight when moused over), but the first click will not be sent to the device. This first click tells the OS that we want to bring the game into focus and for all future button presses should be sent to the game. We have not been able to find a way to bring the software into focus automatically, so an additional click is needed to accomplish this task.

- All hardware within the chassis is secured
  - Large physical components are secured, but wires connecting them are free to move around and get tangled. Better wire management would help hardware organization as well as preventing any leads from disconnecting. The braided cable coming out of the monitor is also very sketchy: it's extremely short and unplugs very easily.
- Speaker Integration
  - We currently have one speaker, and it's not well-integrated into the system. Implementing a speaker and ensuring its output volume is adjustable by the two volume buttons is yet to be completed.
- Display Aspect Ratio
  - For some reason, two of the three currently functional boxes boot into a 4 by 3 aspect ratio instead of a 16 by 9. The Raspi believes it is outputting 16 by 9 1080p on all three boxes and all boot scripts are identical save for the name of the exectuable. We have discovered that manually changing the Raspi to output 1280x1024 makes the screen fill completely, but this change in resolution does not save between reboots. We have not figured out why.
- Braided Cable too Short
  - The braided monitor cable that comes with it is very short and gets unplugged easily.

Section 3: Assembly Instructions

Other materials you may need:
Screwdrivers (flathead and Phillips probably), M4 Allen Key, pliers, wire cutters, wire strippers, solder, soldering iron, heat shrink if you're fancy, electrical tape, extra 18awg wire, X-Acto knife or something else sharp, very small Raspberry Pi screws, SD card adapter, washers, pin headers.

Chassis Assembly

Step 1. Acquire all parts listed in the Bill of Materials (See Appendix section). Note that you may already have some of these parts, left over from the assembly of the initial 5 boxes. The CAD files for the wood panels are on the GitHub. It's recommended but not required to assemble the panels virtually first to get an idea of how the box is assembled.

Step 2. At least one of your team members must get red tag certification to use the laser cutter. You can start that process here.

Step 3. Laser cut, out of wood (or acrylic if you have it), the six flat panels that make up the chassis. You can find them here. Before laser cutting, double and triple check the orientations for each panel. The wood we used was rough on one side and smooth on the other, so we cut the pieces such that the smooth side was facing outwards on all panels. This will give the boxes a cleaner feel.

Step 4. Using the corner brackets and M4 screws and nuts, assemble the 4 large panels of the chassis. You can also mount the front plate, just don't attach the small one with holes for the buttons. Refer to Figures 3, 4, and 5 for details on how the chassis is meant to look fully assembled. Make sure that the side panel with the extra holes for the USB port (small rectangle) is on the left, and the back panel is mounted so that by looking at the two large holes, it can be seen that the slightly larger one is on the right. Make sure that the bottom plate is correctly oriented, with the four holes for the trackball being in front on the left, and the four holes for the Raspberry Pi bracket (these are slightly more spread out) being farther back on the right. Additionally, make sure that the side panels are correctly positioned, with the left side panel having an additional hole for the USB panel-mount hub.

Step 5. Install the rubber feet on each of the four bottom corners of the box.

Step 6. Locate the imperial toolbox of standoffs, spacers, and screws. This is what we used to install the trackball on the box. The trackball must be mounted as close to the top of the button plate as possible, so the trackball can stick up through its hole. We used a combination of standoffs to reach the desired height, which can be done in a number of ways. The shorter hex standoff in the picture below has a long screw threading all the way through it and securing to the long, round standoff.

Step 7. Locate your 5 buttons (1 large, 2 small green, 2 small white). For each button, unscrew the fastener and insert the button through one of the holes in the last wood panel yet to be mounted. It doesn't matter which side is which color, though the buttons on the same side should be the same color. Install the large button in the smaller of the two center holes. Secure the buttons onto the wood plate by putting the fasteners back on.

Step 8. Attach leads to all buttons. Each button has three pins: for the smaller buttons, don't use the middle pin. For the large button, don't connect the pin on the side. Try setting the button plate into the device. You might have to rotate some of the buttons so their leads are pointing into the device, not getting smashed against a wall.

Step 9. Install the top plate with all the buttons on it. Many of these screws will be difficult to access, but not impossible. All screws should be installed to ensure that the box is as sturdy as possible.

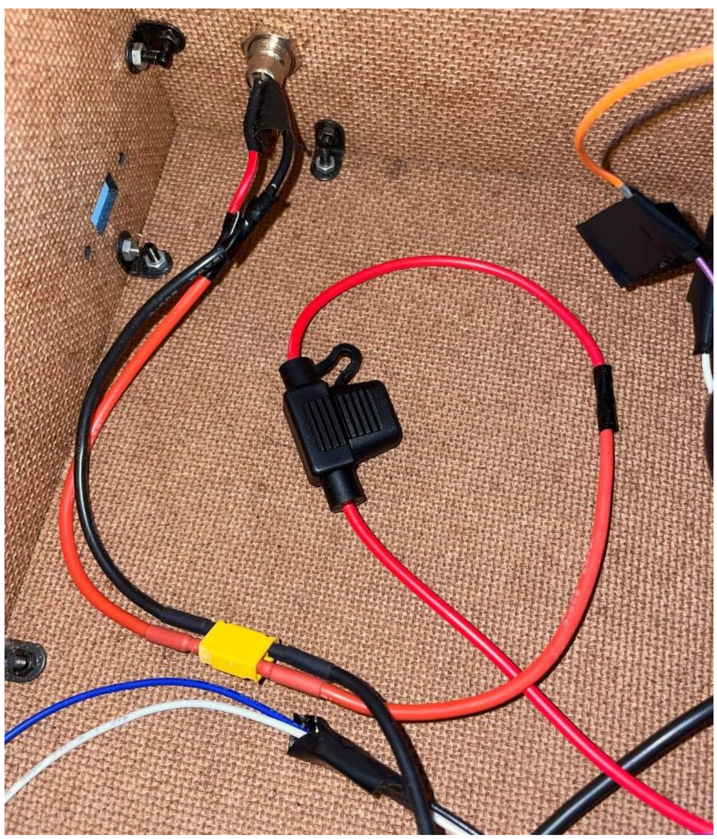
Wiring Setup

Step 1. Quick overview: The box gets 12V in from the wall, which then is split two ways. One way goes into the buck down transformer, to give the Raspberry Pi 5V, and the other way goes into the relay. It is still 12V coming out of the relay, at which point it feeds into the monitor.

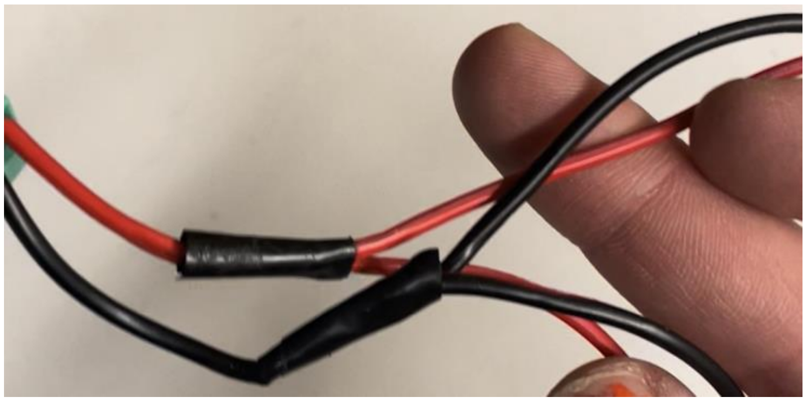
Step 2. Locate a panel-mount power jack and male-ended XT30 cable. Solder the ends of the XT30 cable to the power jack, with the positive cable connecting to the center pin and ground to the outer.

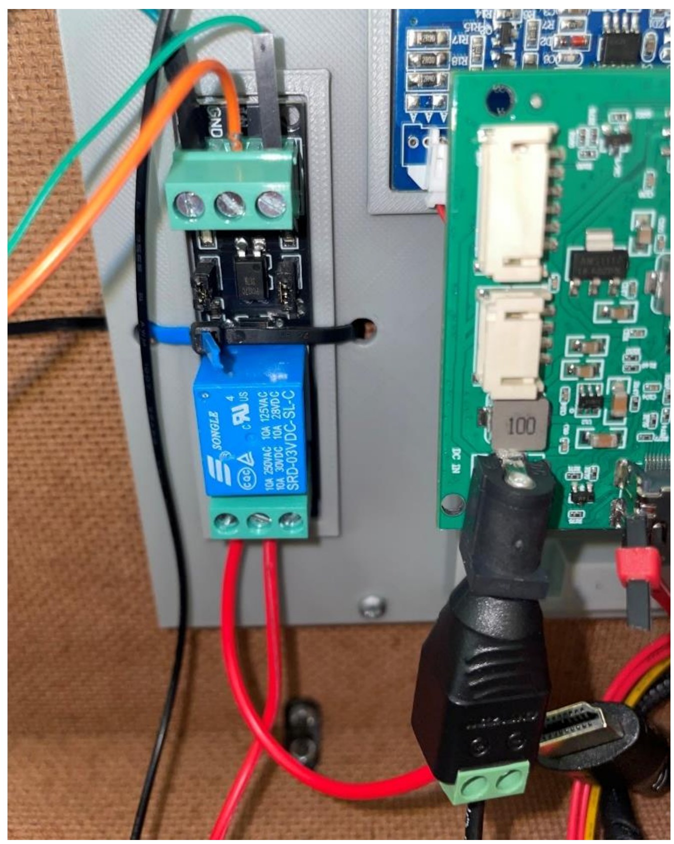
Step 3. Remove the fastener from the power jack and mount the power jack on the chassis - on the back plate, through the large hole on the bottom left. You'll have to feed the XT30 plug through the hole. You'll have to feed the XT30 through the fastener once or twice, and it probably won't fit. I had to shave the edges of it down a tiny bit for it to fit – nothing that would damage or expose any metal, though. It's either this or soldering the power jack after it's installed, which isn't a great idea either.

Step 4. Locate a female XT30 cable. Take its red end and solder the in-line fuse to it. This will prevent surges from impacting the box's circuitry.



Step 5. Take its ends and split the power and grounds. That is to say, the one XT30 red cable needs to connect to two more red cables, and black needs to do the same. For both red and black, one split cable should be extra 18awg wire, and the other should be the positive end of the buck down converter. Solder each split.



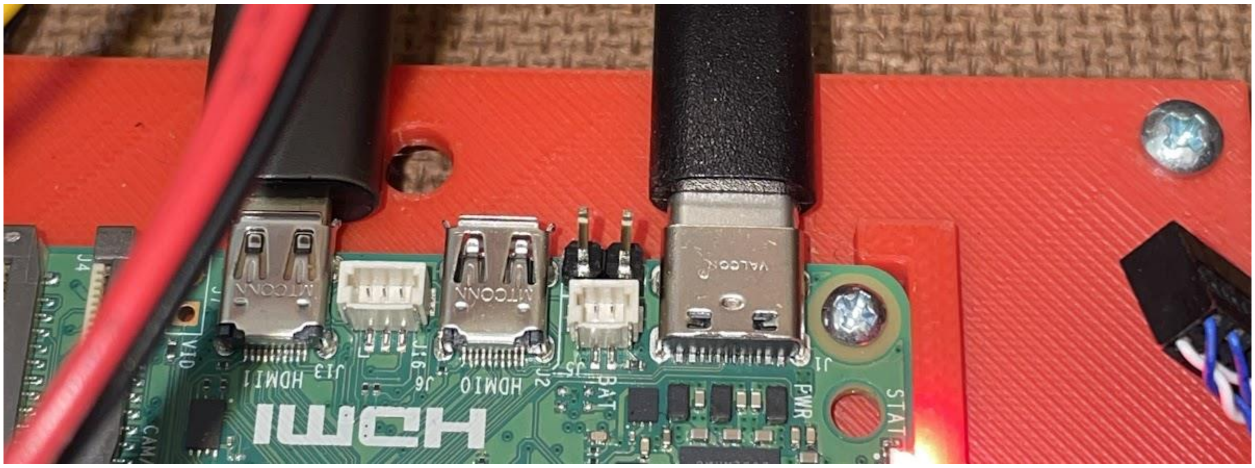


Step 6. Locate the relay. It will be used to control power to the monitor, so that when the Raspberry Pi is off, the monitor will be as well. On the side of the relay that has ports ON/COM/NC, strip and secure the end of the split red cable in the COM port.

Step 7. Cut and strip both ends of a piece of red 18awg wire, about 3-4 inches long. Secure one end in the ON port of the relay, and the other end in the '+' port of the 12V power jack adapter.

Step 8. Strip and secure the end of the split black cable in the '-' port of the 12V power jack adapter. Connect the XT30 cables. There shouldn't be any exposed cable ends now.

Step 9. Locate the Raspberry Pi. Between the USB-C port and the first HDMI port, you should see two small holes. These represent two pins, which when shorted can send the board startup and shutdown signals. Take two pin headers and carefully solder them to these two holes so they can be used.

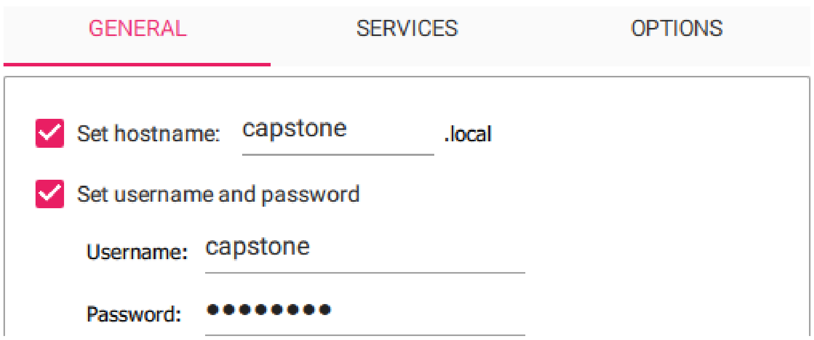


Step 10. Locate the panel mount power button and a set of button leads. Cut the button leads and keep the end with the white plug. Solder this end to the power button's red and black cables. It doesn't matter which solders to which.

Step 11. Install the power button on the chassis, in the large hole on the bottom right of the back plate. You'll have to feed the cable you just soldered through the hole.

PCB Configuration

Step 1. On a computer that can read/write to an SD card, download the Raspberry Pi Imager. This is how we will install the OS onto the Raspberry Pi. Plug the SD card into the computer, through an SD reader port, USB port, etc, and launch the Raspberry Pi Imager. For 'device', choose Raspberry Pi 5. For OS, choose 64-bit. You don't have to set any custom settings (we can do that later) but you can set the hostname, username, and password to 'capstone' if you'd like. Run the installation process, and once it's complete, insert the SD card into the SD port on the bottom of the Raspberry Pi.



Step 2. Locate the two 3D printed brackets. One will hold the Raspberry Pi, buck down converter, and trackball PCB. The other will hold the relay board and both monitor PCB. Note which is which – we'll start with the bracket that holds the Raspberry Pi.

Step 3. Using zip ties, secure the buck down converter and the trackball PCB to the 3D printed bracket. There are holes on the bracket that can be used to secure these, and depending on your zip tie length you may have to link multiple zip ties together. Ensure that components are oriented properly, fitting snugly in the bracket with any protruding parts fitting into their corresponding gaps.

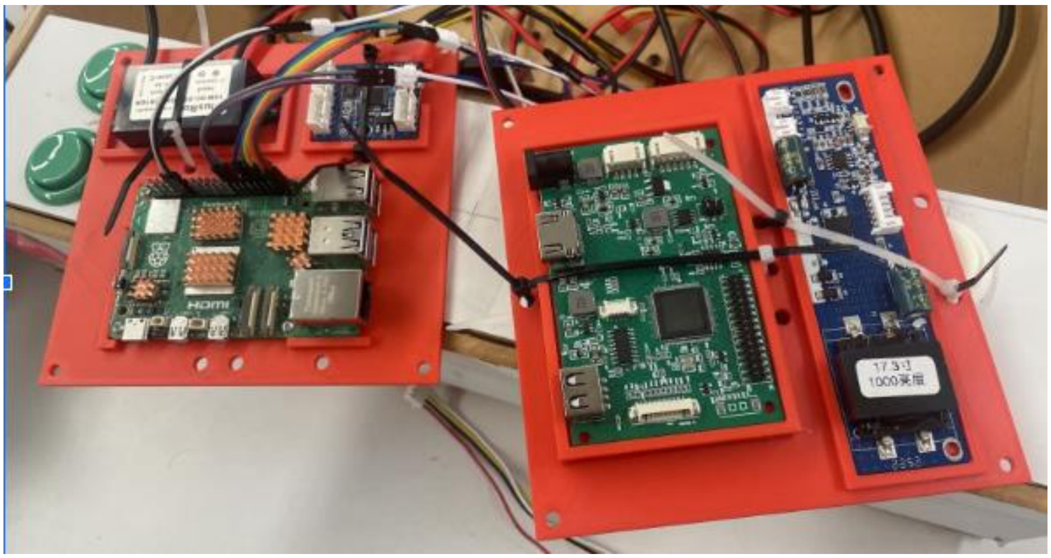
Step 4. Using zip ties or small screws, secure the Raspberry Pi to the bracket. Plug the USB-C cable coming out of the buck down converter into its power port. Plug the connector coming from the power button into the pin headers you soldered near the USB-C port.

Step 5. Mount the 3D printed bracket to the bottom plate of the chassis. You can use any kind of screws/spacers/washers that fit, as long as the bracket is sitting a few millimeters above the chassis.

Step 6. Locate the short Micro USB to USB cable. Use it to connect the trackball PCB to the Raspberry Pi. Any USB port on the Pi works. Connect the cable coming out of the trackball into the trackball PCB, in the port adjacent to the micro USB port.

Step 7. Locate the PCBs that came with the monitor. There will be four: we only need the largest two. The board with the buttons is useful for debugging, as it acts as a menu for the monitor, and it can access settings and change inputs. It plugs into the larger green PCB if needed.

Step 8. Secure the two monitor PCBs to the bracket. Locate and secure the relay board as well, with the ON/COM/NC side pointing downwards.



Step 9. Mount the 3D printed bracket to the back plate of the chassis. You can use any kind of screws/spacers/washers that fit, as long as the bracket is sitting a few millimeters above the chassis.

Step 10. Plug the 12V power jack adapter into the green monitor PCB. Connect the two monitor PCBs with the 5-wire cable.

Step 11. Locate the short HDMI cable and the micro HDMI to HDMI adapter. Connect the adapter to one of the HDMI ports on the Raspberry Pi. Plug one side of the HDMI cable into the adapter, and connect the other side to the HDMI port on the green monitor PCB. This will be a tight squeeze.

Step 12. Locate the panel mount USB jack and two M4 screws. Mount it to the left side panel of the chassis. Plug its other end into the Raspberry Pi. Plug the soldered power button cable into the pin headers you soldered earlier between the Raspberry Pi's USB-port and first HDMI port.

Software Setup

Step 1. Log in to Raspi and open a terminal. Type in “cd /home/capstone”. We will be building our file structure here

Step 2. Enter “git clone https://github.com/jasondfellows/CalPoly_Jigsaw_Hardware.git”. This will download a folder called “capstone” into your current directory

Step 3. Enter “cd CalPoly_Jigsaw_Hardware”

Step 4. Enter “cd builds” and download your individual build of the Jigsaw Puzzle Game. This can be done via USB splitter and thumbdrive or via Magic Wormhole (requires you to run “sudo apt install magic-wormhole"). Ensure the file has execute permissions (Can be added with “chmod +x [filename]”)

Step 5. Enter “cd ../testPrograms”

Step 6. Enter “nano bootScript.sh” and replace “[Install name]” with the name of the Jigsaw Puzzle Game file you just downloaded. Exit nano by pressing CTRL+X, save changes by pressing Y followed by Enter twice

Step 7. Type “crontab -e”. If you see a list of three options pop up, just hit enter.

Step 8. At the end of this file, type “sudo startx /home/capstone/CalPoly_Jigsaw_Hardware/testPrograms/bootScript.sh”. Exit nano the same way you did in step 6.

Step 9. To increase mouse sensitivity, enter “sudo nano /boot/firmware/cmdline.txt”. Then scroll to the end of the file using arrows or the End key, add a comma WITH NO SPACE, then type “usbhid.mousepoll=0”. Exit nano the same way you did in step 6.

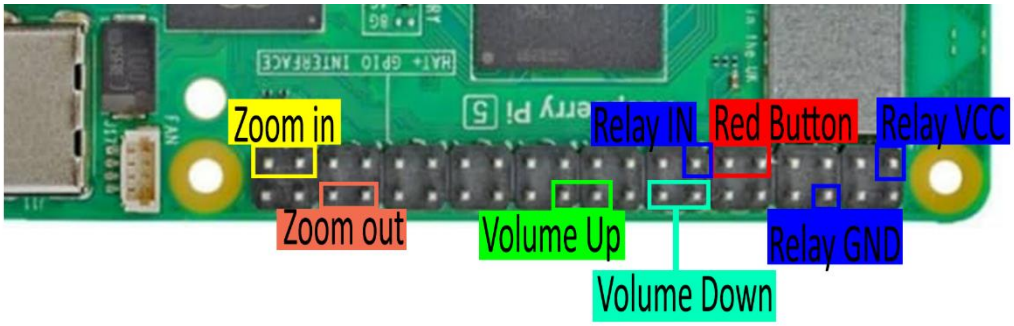
Step 10. Type “sudo reboot” and allow the device to reboot. This may take a few minutes. When it is done rebooting, the screen should display the jigsaw puzzle game.

Troubleshooting:

If you do not see the game, double check the name of the build (ls capstone/builds) and ensure it’s typed correctly in bootScript.sh (nano CalPoly_Jigsaw_Hardware/testPrograms/buildScript.sh). Also check the line in step 8 is typed correctly into crontab (crontab –e).

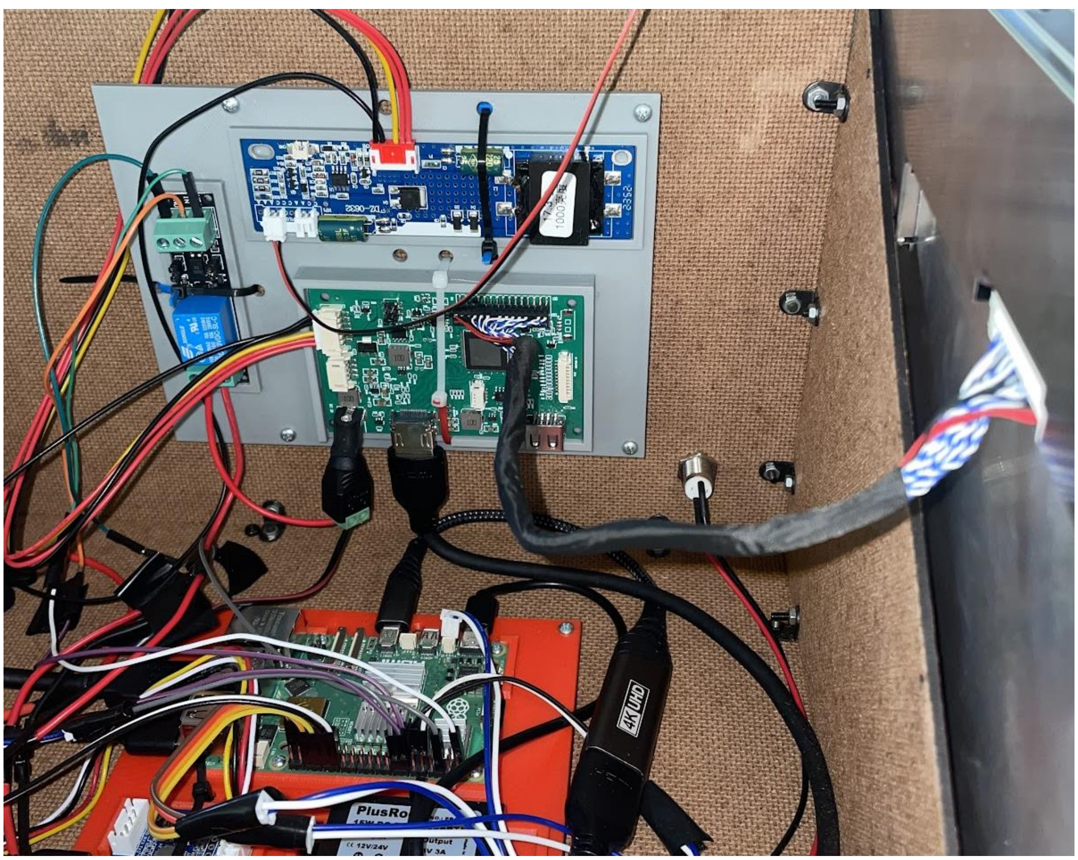
Finishing Touches

Step 1. Connect the leads to the GPIO pins according to the diagram below. If the leads are too short, you may use additional Male to Female pin leads as extensions. Ensure all additional leads are functional and place tape across the connection to strengthen it.



Step 2. Connect the three leads from the GPIO pins to the VCC/IN/GND side of the relay.

Step 3. Locate the monitor. There should be two cables with it: one very thin red/black pair and one thick braided cable. The first plugs into the blue monitor PCB, and the second into the green. You can see them connected in the picture below. Note that the braided cable has a white dot on one side of its large, flat connector. This dot faces downwards.



Step 4. Before mounting the monitor, try plugging in the device using the power adapter and test if it works.Note that the braided monitor cable is very short. You may want to provide additional security for it, or buy a longer one. As it is now, it’s a good idea to physically hold it in place as you align the monitor.

Step 5. From the playtest, note which way is up for the monitor. Push the monitor into the top of the box. It will be a tight fit, which will make it easier to line up the four holes on the sides. Put an M4 screw into each of these holes and tighten.

Section 4: Appendix

Bill of Materials

Name	Quantity	Source	Price
Large Button	1	Adafruit	5.95
Small Button (Color 1)	2	Sparkfun	7
Small Button (Color 2)	2	Sparkfun	7
12V 5A Power Supply	1	Amazon	12
Button Leads (10pk)	1	Adafruit	4.95
Raspberry Pi 5	1	Sparkfun	60
Micro HDMI to HDMI Adapter	1	Amazon	5.99
1ft HDMI Cable	1	Amazon	8.99
SD Card (32GB)	1	Amazon	20
Speaker	1	Amazon	14
Trackball	1	Thunderstick	40
Trackball PCB USB Adapter	1	Amazon	20
RPi 5 Heatsink Kit	1	Amazon	10
12V to 5V Buck Converter with USB-C Output	1	Amazon	10
17.3" Monitor and Controller Board	1	eBay	158
XT30 Connectors	1	Amazon	12.99
Rubber Bumpers (4pk)	1	Amazon	8
Short Zip Ties	1	Amazon	6.5
Panel Mount Power Jack	1	Amazon	7
Panel Mount Power Button	1	Amazon	10
Corner Brackets (40pk)	1	Amazon	9
Mounting Brackets (3D Printed), type A	1	Ask Seng for file	
Mounting Brackets (3D Printed), type B	1	Ask Seng for file	
Imperial Standoffs/Spacers/Screws		Ask Seng for toolbox	
M4 10mm Screws (60 required)	1	Amazon	9
USB Panel Mount Jack	1	Amazon	8
M4 Nuts (54 required)	1	Amazon	5
12V Power Jack adapter (Male)	1	Amazon	8

Relay Switch	1	Amazon	10
Wood for Chassis Panels	3	Ask Seng	
Short Micro USB to USB Cable	1	Amazon	8
In-line Fuse	1	Amazon	5

References

Raspberry Pi 5: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf

LCD screen: https://www.vslcd.com/product/5317881818010915.html