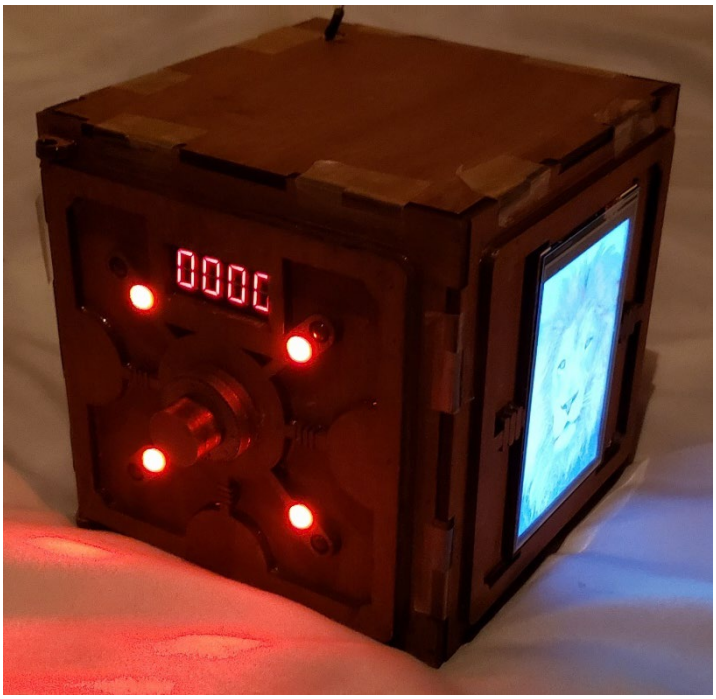
Arduino Based Puzzle Box

Description of Prototype

The project is a buzzle box that has three interlocking interconnected puzzles that must be solved to open the puzzle box and retrieve the prize inside.

The first puzzle is a knocking program that requires the user to input a specific sequence of knocks to solve the puzzle. An LED will light up for every correct knock. If a knock is incorrect, 4 green LEDs will flash for too fast and 4 red LEDs will flash for too slow. There is a total of 8 knocks in the sequence.



The second puzzle is a touchscreen LCD that contains a 3x3 sliding block puzzle. The image shows the solved puzzle on the right side of the box. When not solved, there will be 8 image tiles with an empty space for the ninth. The image tiles can be moved to the empty space thus emptying their previous spots. The goal is to organize the image so that all tiles are in the right spot. Once that is done, the image will complete itself and only one puzzle will be left.

The third puzzle is a four-digit guessing game where red LEDs will light up for every correct number in the wrong position. For example, if the code is 0123, inputting 1000 will light

up 2 red LEDs and 1032 will light up 4 red LEDs. Green LEDs will light up for every correct number in the correct position. For example, if the code is 0123, inputting 0100 will light up 2 green LEDs and 0124 will light up 3 green LEDs.

There is a fourth puzzle that features a 3x3 grid of LEDs and buttons on the LED screen where pushing one of the LED button combos will turn on all adjacent lights that are off and turn off all adjacent light that are on (including itself).

There is a fifth puzzle where the puzzle box is rotated a certain amount then a pushbutton is pressed. The lights on the front will light up green when the player has not moved far enough, red when the player moved too far, and red and green when the player succeeded. This continues for four iterations where the player has five attempts per iteration before the puzzle restarts.

Difficulties faced

There were a lot of problems that cropped up as the project continued. The biggest issue is that, by using the same size of the box as the original design while simultaneously adding a multitude of parts, my puzzle box ran out of room quite quickly. Another problem was the soldering. I soldered over 70 connections which was extremely difficult considering the fact that these connections would be placed side by side. The soldering was redone with crimping for a much cleaner and less error-prone circuit.

Furthermore, for reasons still unknown, I was unable to properly solder the middle LED of the 3x3 grid. The connection would always depend on how I positions the soldered parts and I would have to hold it in place with my finger for the connection to work, something impossible to maintain since I cant always hold that connection. It was also extremely difficult and time-consuming to glue in the push buttons such that they would not lock down the LEDs preventing them from pushing the button. I had to remove button, scrape off the glue, and reglue multiple times. I also had to flatten the LED wires at the LED's base then glue the whole base creating a flat surface to interact with the push button. Because of this, the pushbuttons were removed and the LEDS were instead adjusted using the touchscreen.

Another weird issue was that the rotary encoder seemed to create some sort of open circuit on the Arduino that could only be solved by connecting a wire to pin 2 (a pin that is not being used). Only pin 2 worked and other unused pins did not. Furthermore, the pin had to be closed manually by pinching the other end of the wire with two fingers to close the circuit. Connecting the wire to 5V or GND did not solve the issue and using fingers was the only solution I found. This issue was solved by properly resoldering all components, crimping, and heat shrinking.

Some other difficulties include needing to laser cut the same part multiple times because of issues with the dimensions and wasting time due to shipping time. In the end, a box was completed, but it is too small to properly fit the components which has caused some issues from poor wiring to unstable connections.

Parts list:

- TFT touchscreen LCD
- Arduino MEGA
- Many wires, solder, crimp
- Rotary Encoder
- 17 LEDs and 220 Ω resistors
- 9 Buttons and 1MΩ resistors
- Seven segment display
- 1/8. in MDF
- 2 9V batteries
- Glue
- Servo motor
- Accelerometer
- SD reader

Algorithms used

The only algorithms used were those that checked if a knock was correct, whether to 4-digit code was correct, whether a chosen tile can be moved to a chosen location, and other minor algorithms that helped the aforementioned algorithms become simpler. I did not use any of the well-known algorithms like bubble sort and all of my algorithms were self-made.

Pin List

Object	Name	Pin
Servo Motor	Servo Pin	23
Rotary Encoder	ButtonPin	17
Rotary Encoder	PinA	19
Rotary Encoder	PinB	18
LCD	LCD Pin	9
TFT	LCD_RD	A0
TFT	LCD_WR	A1
TFT	LCD_CD	A2
TFT	LCD_CS	A3
TFT	LCD_RESET	A4
Touchscreen	YP	A2
Touchscreen	XM	A3
Touchscreen	YM	8
Touchscreen	XP	9
Touchscreen	SD_CS	53
Piezo	Sensor	A8
LED	correctNumLEDs	27
LED	correctNumLEDs	29
LED	correctNumLEDs	37
LED	correctNumLEDs	35
LED	correctPlaceLEDs	25

LED	correctPlaceLEDs	31
LED	correctPlaceLEDs	39
LED	correctPlaceLEDs	33
LED	{1,1}	14
LED	{1,2}	41
LED	{1,3}	10
LED	{2,1}	15
LED	{2,2}	43
LED	{2,3}	11
LED	{3,1}	16
LED	{3,2}	45
LED	{3,3}	12
SevSeg	digitPins	22
SevSeg	digitPins	34
SevSeg	digitPins	38
SevSeg	digitPins	44
SevSeg	segmentPins	26
SevSeg	segmentPins	42
SevSeg	segmentPins	36
SevSeg	segmentPins	28
SevSeg	segmentPins	24
SevSeg	segmentPins	30
SevSeg	segmentPins	40

SevSeg	segmentPins	32