FLEX DANCE

Team A3: Caio Araujo, Spandan Sharma, Tushhar Saha
18-500 Capstone Design, Spring 2022
Electrical and Computer Engineering Department
Carnegie Mellon University

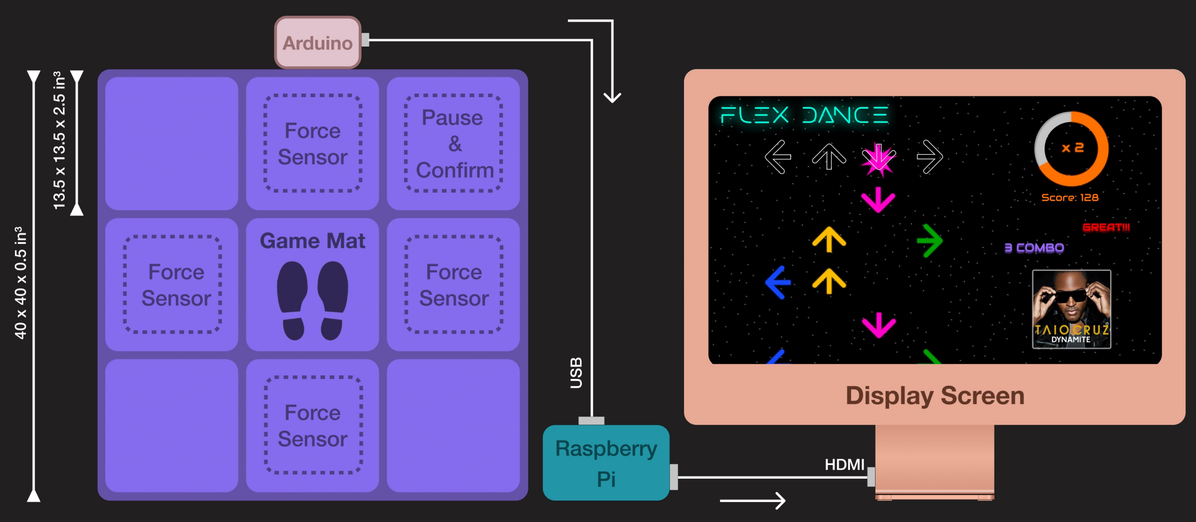
Electrical & Computer ENGINEERING
Carnegie Mellon

Product Pitch

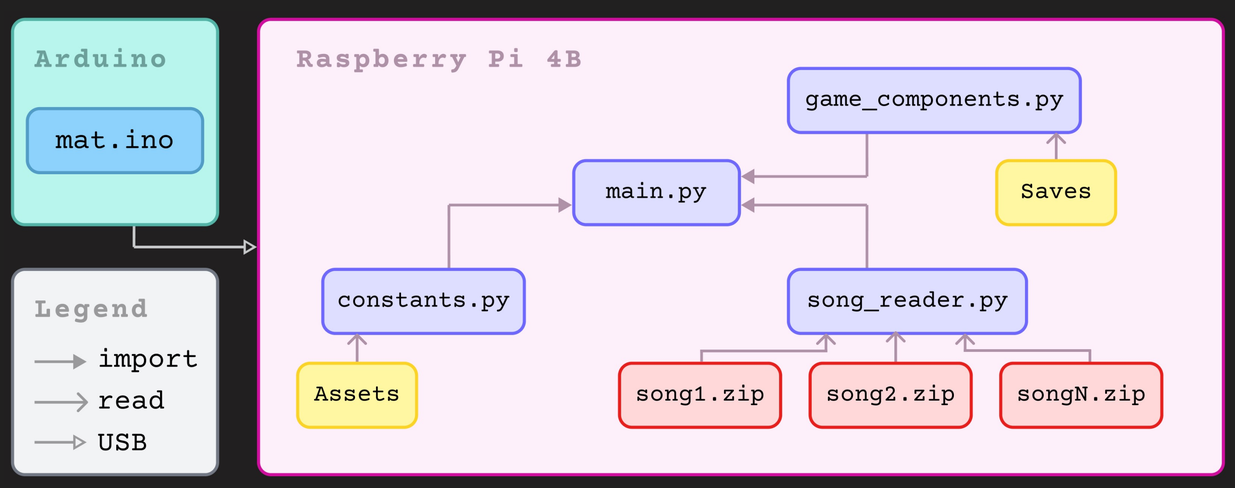
For many people, moving around during the day is their main physical activity. During the COVID-19 pandemic, however, this kind of activity was restricted. Flex Dance is an at-home exercising platform designed for casual exercisers to remedy this situation. It is meant to be beginner-friendly, easy to store, enjoyable, and affordable.

Our system consists of a rhythm game software installed in a Raspberry Pi (RPi) and a dance mat to connect to the RPi that controls the game. Our final product is engaging, works with only five different inputs, and fits into almost any home drawer, although it is also more expensive than we would have liked.

System Architecture



Our system comprises 3 main components. As shown in the diagram above, they are the Game Mat, Raspberry Pi and the Display Screen. The Game Mat is responsible for detecting inputs from the 5 force sensitive resistors (FSRs). The Arduino emulates a keyboard and triggers keypresses based on which FSR is pressed. This information is sent to the Raspberry Pi through a USB connection.



The Raspberry Pi contains everything needed to run the game including the game files, assets, an operating system and pygame. All the computations happen in this computer. Lastly, the relevant game screen and data is shown on the display screen through the HDMI cable.

Conclusions & Additional Information

http://course.ece.cmu.edu/~ece500/projects/s22-teama3/

Overall, we are very proud of what we accomplished in this project. Despite not meeting all our initial requirements, it gets reasonably close to all of them except for price. During our work, we learned the importance of communication amongst the team members. We all had our own views for Flex Dance, and it proved essential to communicate those views, so we knew what each one of us was expecting. If you would like to read more about our design process, scan the QR code to visit our blog!

System Description

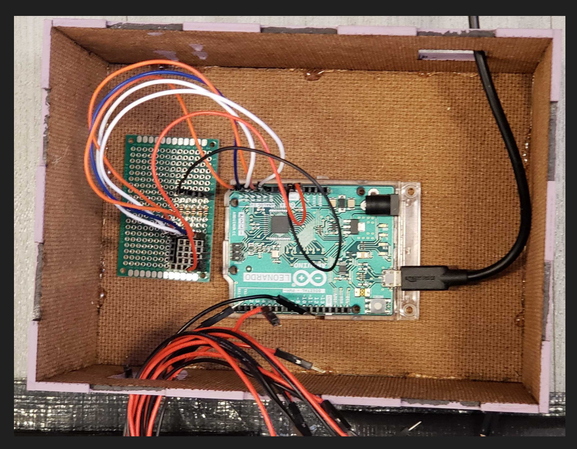


Figure 1: Inside of the Arduino Box (top)

Figure 2: Final Mat (left)

Component Interactions

Each button on the mat has 2 FSRs in parallel connected to the perfboard. The perfboard has the planned circuit soldered onto it and relays the relevant signals to the Arduino. The Arduino has code to calibrate the FSRs and emulate a keyboard so that the Raspberry Pi can read the signals.

Game Walk-through

The player uses the mat to choose a song to dance to. Each song has its own set of high scores to differentiate difficulty level. The game starts where the players matches the arrows at the correct time by pressing the buttons on the mat. To motivate them further, the game provides score multiplier on multiple correct sequences. If the player scores a high score, they input their name on the scoreboard.

System Evaluation

Requirement	Original Goal	Actual Measurement
Folded Size	<= 13in x 13in x 5.5in	**13.5in x 13.5in x 2.5in**
Unfolded Size	<= 39in x 39in	**40in x 40in**
Button Coverage	~360°	**~270°**
Latency	<= 0.1s	**0.008s**
Error Rate	<= 1%	**4%**
Price	<= $180.00	**Over $180**

Table 1: Testing Results

Coverage Evaluation

We selected different directions for stepping on the buttons and kept track of successfully detected presses. Our initial design for each button had one FSR folded into a Z shape. Its coverage was close to 360° but it quickly broke. We now have two parallel FSRs for each button, with a coverage of around 270°degrees, but they are much more reliable. For the error rate, we averaged all the missed steps we had during our coverage test.

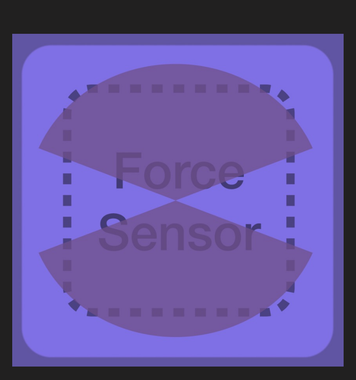


Figure 3: Coverage Area

Latency Test

We recorded a slow-motion video of someone playing the game. We then counted how many frames there were between applying force and seeing a change in the screen using video editing software. With this information we found the time between both events.

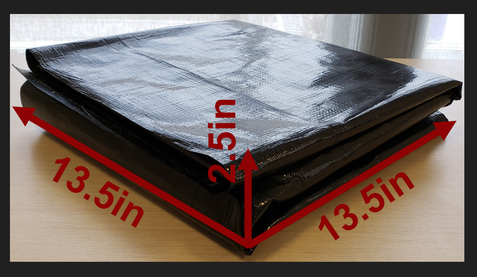


Figure 4: Mat Measurements