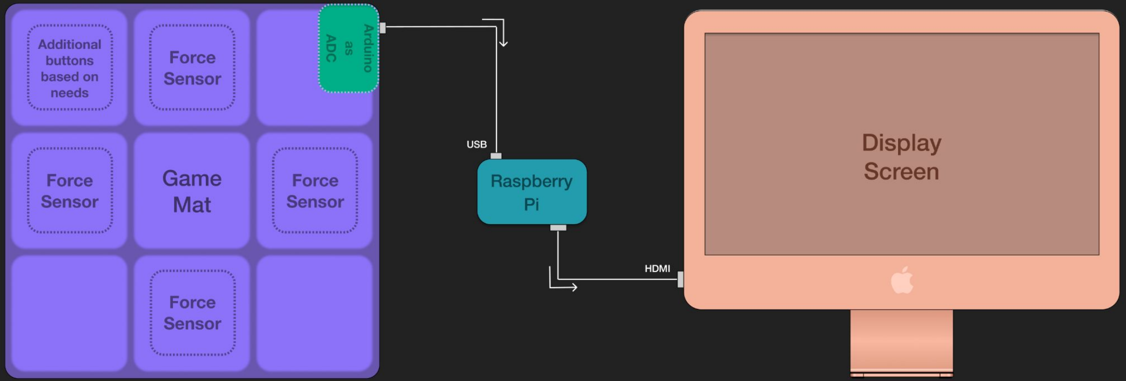
Team A3 - Flex Dance

Spandan Sharma, Caio Araujo, Tushhar Saha

Contents:

- Use Case
- Use-Case Requirements
- Technical Challenges
- Solution Approach
- Testing, Verification and Metrics
- Tasks and Division of Labor
- Schedule

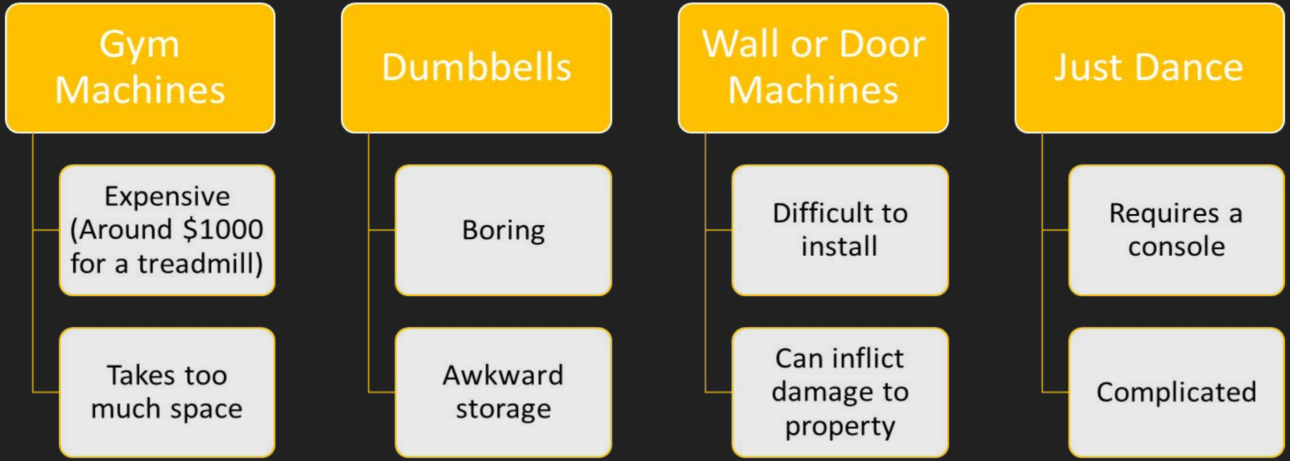


FLEX DANCE

Problems and Existing Options

- Forced to stay at home in the pandemic
  - Not enough exercise
  - Lack of enjoyment

- Alternate Options?



Gym Machines
- Expensive (Around $1000 for a treadmill)
- Takes too much space

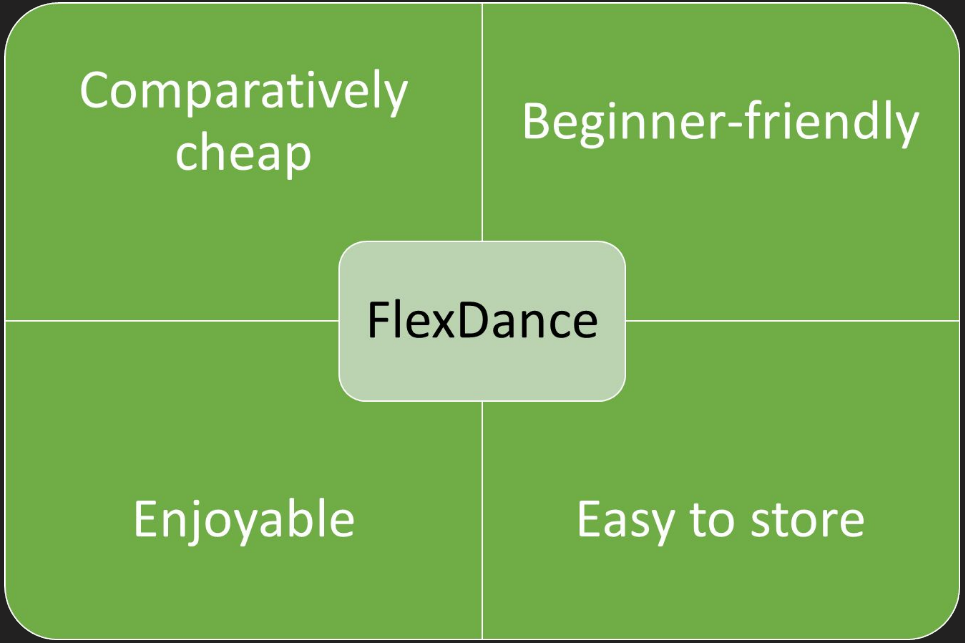
Dumbbells
- Boring
- Awkward storage

Wall or Door Machines
- Difficult to install
- Can inflict damage to property

Just Dance
- Requires a console
- Complicated

FLEX DANCE

Use Case



FlexDance

Comparatively cheap

Beginner-friendly

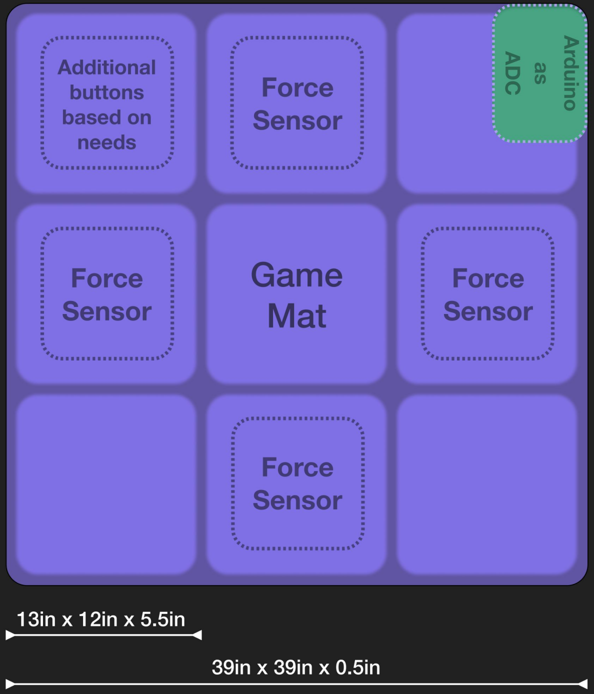
Enjoyable

Easy to store

Areas covered: Software Systems and Circuits

Use Case Requirements

Requirement	Why?
Selling cost < $200	Affordable; current alternatives start from $300
Storage size: Folded size < 13in x 12in x 5.5in Unfolded size < 39i x 39in	Easy to store; Size of average drawer; Living room space
Easy installation: connect to display through HDMI cable	Accessibility: game should be able to be installed by a child or older people
Force detection threshold ~10lbs	Resting foot weight



Use Case Requirements

Requirement	Why?
Error rate < 1%	See calculation on the side
Latency of signal between Arduino to Raspberry Pi < 100ms	Humans perceive images in 1/10th of a second
Arrow button 360° coverage	User can press the buttons in any feet orientation
Linear scoring scale	Users expect to receive points even if they don't time their step perfectly

Perfect player should ace 1 in 4 games (25%). If a 3 minute song track has an average of 200 arrows (a little more than 1 arrow/second), we get:

- ➔ Probability of scoring every arrow correctly = 0.25
- ➔ $(1\text{-error rate})^{200} = 0.25$
- ➔ Error rate ~= 1%

Use Case Requirements

Requirement	Why?
Easy to start the game: Game screen should be 3 clicks away	Beginner-friendly interface
Differentiate between pressing and holding buttons	We will need pressing for navigation between screens and holding for choosing letters on the mat
Stimulating while respecting visual weight: follow 60-30-10 rule	Keep user engaged while not overwhelming or confusing them

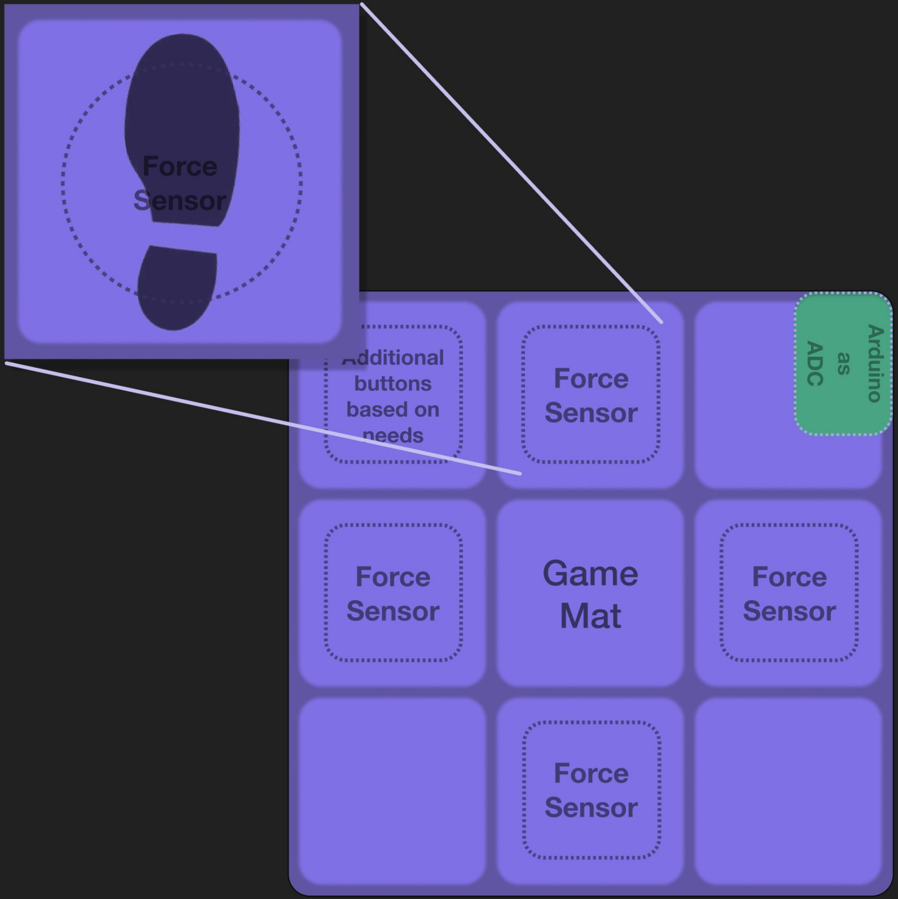
Technical Challenges

- Reducing latency between sensors and scores
  - Lenient with scoring
  - Slow songs
- Calibrating pressure sensors
  - Detecting deliberate vs accidental movement
  - Manipulate threshold to avoid accidental triggers
- Selling cost
  - Monitor components’ cost

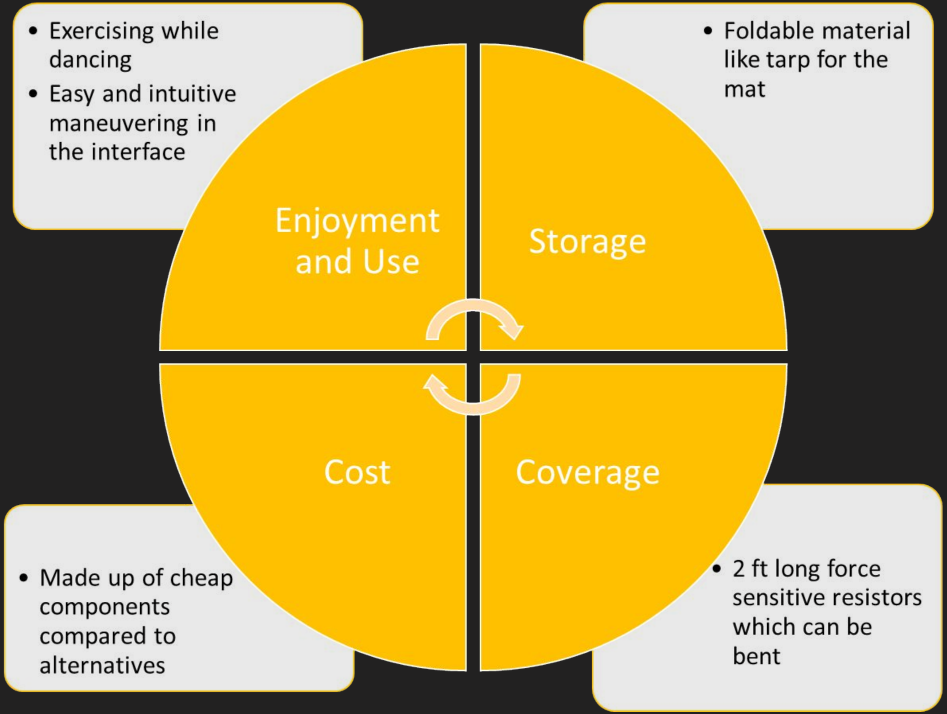
FLEX DANCE

Technical Challenges

- Detecting feet in any orientation
    - Use force sensors in a circle
- Creating a design interface with 3 clicks
    - Examine pre-existing interfaces
- Finding the right mat material
    - Experiment with different fabrics



Solution Approach



Enjoyment and Use

- Exercising while dancing
- Easy and intuitive maneuvering in the interface

Storage

- Foldable material like tarp for the mat

Cost

- Made up of cheap components compared to alternatives

Coverage

- 2 ft long force sensitive resistors which can be bent

Foldable Dance Dance Revolution (DDR) Kit

○ Raspberry Pi

○ Force Sensitive Resistors

○ Arduino as an ADC

○ Pygame

FLEX DANCE

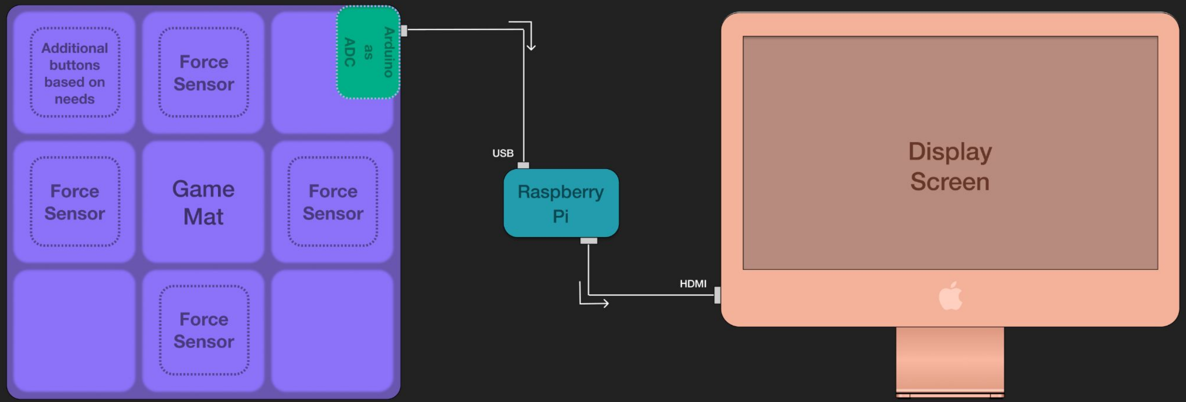
Testing, Verification, and Metrics - Mat

Requirement	How to measure	Goal
Folded size	Measuring tape	<= 13 in x 12 in x 5.5 in
Unfolded size	Measuring tape	<= 39 in x 39 in
Minimum force detected	Arduino serial monitor and force gauge	~ 10 lbs
Arrow button coverage	Step on the mat in different orientations	Circular shape

Testing, Verification, and Metrics - Software

Requirement	How to measure	Goal
Latency	Measure time between sending and receiving signal using python	<= 0.1 s
Error rate	Step on the mat and register successful detections	<= 1%
Linear scoring scale	Have a few people play the game and state if they get frustrated	<= 25% people frustrated
Cost	Sum up components’ cost	<= $200

How far we will go:



Game Software

(Game screen, highscore board, music selection)

Game Hardware

(Dance Mat)

Relevant Music Files

(Encode 2 song tracks)

	Week 1 (Feb 7-13)	Week 2 (Feb 14-20)	Week 3 (Feb 21-27)	Week 4 (Feb 28-Mar 6)	Week 5 (Mar 7-13)	Week 6 (Mar 14-20)	Week 7 (Mar 21-27)	Week 8 (Mar 28-Apr 3)	Week 9 (Apr 4-10)	Week 10 (Apr 11-17)	Week 11 (Apr 18-24)	Week 12 (Apr 25-May 1)
Deadlines	Peer Review (Feb 9)	Design Review Slides (Feb 20)	Design Review Peer (Feb 23)	Design Review Report (Mar 2)					Interim Demo (Apr 4 or 6)		Final Presentation Slides (Apr 24)	Peer Review (Apr 27)
Read sensors from Arduino		🟥										
Test sensors - 1		🟥										
Read Arduino/USB from RPi			🟫									
Test sensors - 2			🟥				🟥					
Construct physical mat					🟫	🟫						
Calibrate sensors							🟥			Buffer (Debugging)		
Learn Pygame	🟫											
Find good OS	🟨											
Design game interface	🟦	🟦										
Plan game structure		🟨										
Research & plan custom track file format		🟩										
Implement game structure using arrow keys for menu screen			🟦	🟦	🟦	🟦						
Implement game structure using arrow keys for game screen			🟨	🟨	🟨	🟨						
Make a song playable with arrow keys							🟦					
Integrate game and mat							🟫	🟫				

Legend: 🟥 Tushhar 🟨 Caio 🟦 Spandan