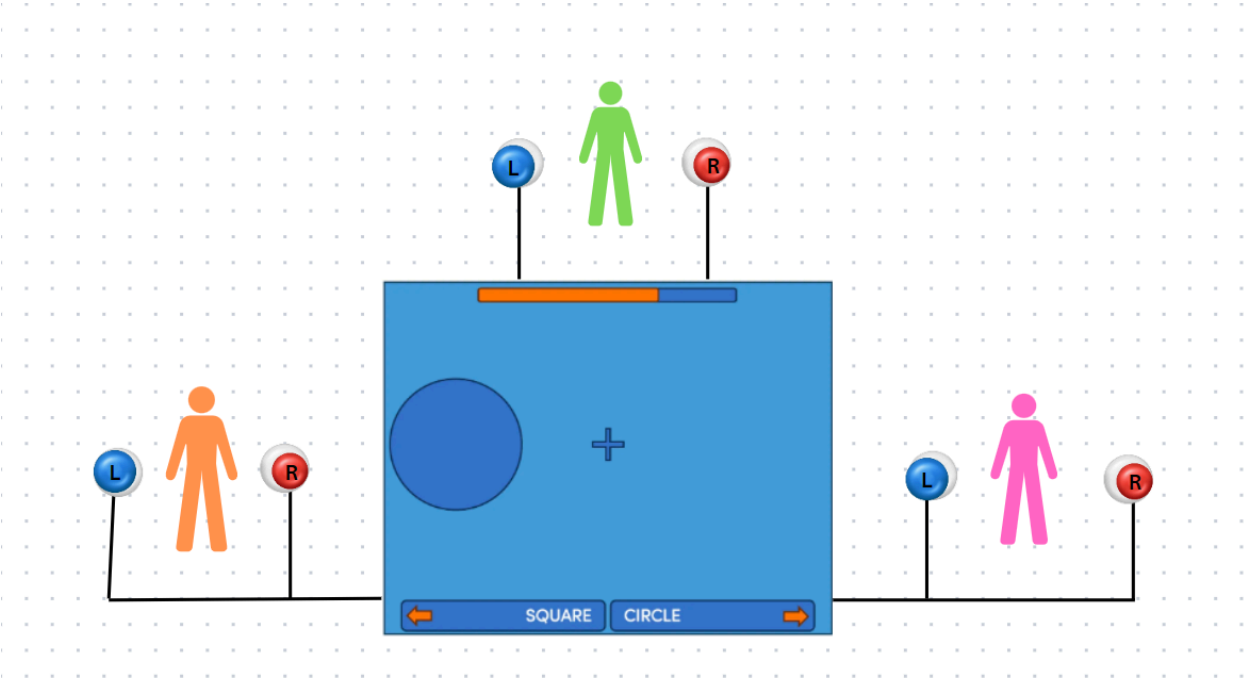
Project Title: ShapeShift

Teammates: Zhiquan Cao, Anar Bhagat, Abhimanyu Suthar, Nikhil Khandelwal

Abstract:

We are developing a multiplayer, real-time interactive game. The concept is based on a quantitative recruiting assessment exam used in real-world businesses, which measures players' reaction times in an entertaining and participatory way.

The visual representation is this:



Game Rules:

Circles and squares will briefly appear on the LCD screen in succession. Your task is to hit the **right arrow key** for **circles** and the **left arrow key** for **squares.**

The circles are not always shown on the right, and the squares not always on the left. These can show up at any side. Your task is to not get confused and keep your accuracy high. Purely focus on the shape, not on which side they appear!

After several rounds, the rank of the players will be displayed on the screen. There will also be some shining LEDs to reward the winner!

The real test lies in finding the optimal balance between speed and accuracy. Avoid rushing; begin at a moderate pace and gradually increase your speed once you're comfortable with the mechanics.

This game takes **2.5 minutes.**

Components & Plans:

The hardware components are as follows:

- Durable buttons for players.
- Start-game button.
- Several LED diodes as count down signals.
- Arduino UNO as the microcontroller.
- A LCD screen to display the interface of the game, as well as the rank and reaction time for each player.

To implement the game, several technologies are expected to be used:

- Configuration wtih LCD screen:
    - Display the squares and circles randomly on either left or right side of the screen with appropriate delay in between.
    - Providing UI/UX from this LCD screen.
- Classes and structs of players modules to encapsulate the implementation of the game rules.
- Run-time optimization for the code since the game focuses on players' reaction time.
- Real-time USART communication may be required if more than four gamers are playing.
- Use of the internal clock in Arduino:
    - This ensures a fair synchronized grading mechanism for multiple players.
- Ranking algorithm:
    - We will use prioriety_queues in c++ to implement such a grading system.