**Summary:** This is a program that allows you to fight with the computer in a boxing match. The player starts the game by starting a new session. The difficulty is then chosen, with the last levels being exceedingly difficult. The user uses a smartphone connected to MATLAB to run the game and fight. Speed along the three axis, i.e x, y, and z will determne the damage done.

**Usage:** Run the game after connecting your phone to MATLAB using the MATLAB Mobile Application. After connecting your phone run the runME.m file and start a new game by choosing an opponent.

**Options:** The user may choose the opponent at the chosen difficulty. Harder opponents are both more likely to attack and deal more damage as well.

**Features:** The following additional features have been implemented.  
If you are able to hit the opponent, a brusied image of him threatening you is shown. If you miss the opponent mocks you and then tries to attack you.  
  
**Theory:** The damage done is based on the accelerometer sensor values passed on. I conducted two tests. In the first the mobile was placed on a desk and its values were recorded for about 10-12 seconds. These results were then averaged to determine the values being passed on regardless of any movement. To form appropriate thresholds the mobile was then placed in a hand with slight movements. These results were then again averaged. The results of these are commented in the attack.m functon file.

**Programming:**   
**Unusual Features:**   
Damage dealt is based purely on the movement of the smartphone. In addition to this an image of a phone is created prompting the user that they may attack since the accelrometer values are being registered. In addition to this all opponents have images placed on the GUI by using axis and adding images to the axis. The opponents name and countries are read using a CSV file. Editing the csv file will change the names in the game. Thus if at any time, we want tpo change opponent names, we can simply edit the CSV file.

**Structure:**   
The main program is started using runME. This runs the game. It plays a sound and then redirects the code to Menu.m

Menu.m: runs the main menu. It has two buttons. I.e quit game and new session. Quit game when pressed closes the menu. New game leads the user to opponent\_menu.m

Opponent\_menu.m: This shows the opponent menu. Images of individual boxerrs are shown and difficulty is displayed using static text boxes. The push button runs game() using the difficulty chosen by the user. i.e pressing the push button under the opponent with difficulty 5 would run game(5).

game(difficulty) function runs the game using the set difficulty. It shows the ring image and the boxer. Boxer name and country are shown using difficulty. So game(5) would get the data from CSV file for boxer number 5. A timer follows showing the match is about to begin. A sound plays after the timer. (LETS GET READY TO RUMBLE). When the match begins health bars are set up. A for loop is run for a large number of times (large enough for one of the players to lose). At the start attack() is run to determine whether the user attacked. The value is passed on to damage\_done. This will lower the computers health by the amount assigned to damage\_done. It also updates the health bar. Also an image of the brusied opponent is displayed in the case of a succesful attack. Otherwise the computer shows an image mocking the user. Computer movement is set to random. A random number is chosen between 0 and 15 with the chances of being hit equal to 12 minus the difficulty. Thus an opponent with a dfficulty of 3 would hit if the number is greater than 9. An opponent with 10 difficulty would hit if the number chosen is greater than 2.

attack() returns the damage done. A mobiledev object is created and the accelerometer is enabled. The data is gathered for about 1.5 seconds. When the accelrometer data is being collected a phone image is displayed. It is turned off after the data is collected. If the movement was large enough during any of the 1.5 seconds the maximum of those is taken. This happens for all three axis. If the movement is less then the thresholds set then no damage is dealt. At the end of the script the value assigned to damage\_done is given as the output argument.

win\_sound, lose\_sound and punch\_sound plays the appropraite sounds whenever theyre called.

attack\_anim and anim create an animation using the images. Pauses are given after which the visibilty is turned off and a new image is shown. This gives the effect of an animation.





