**Aug-Hit**

**-Aniket Bajpai**

**-Manish Singh**

**-Amal George M**

**-Shaurya Shankar**

**Task**

* The robot will be given an feed of the arena which needs to be processed to find the position of the ball and play the game according to the rules.
* The robot needs to hit the ball in such a way that it breaks the bricks either directly or by collision with the walls.
* The robot will move in the real arena. A marker will be provided by organisers to stick on the bot. This marker will be detected by organisers webcam to position our bot in the virtual feed being given. This marker may be rotated by the player to rotate the paddle in the virtual arena.

**Arena**

* The Arena will be virtual. The robot will be allowed to move in the x-axis(horizontal)
* The ball will be virtual, and the robot will have to process the feed given to the participants which will then be used to determine the location of the ball. Using this location of the ball, the bot will have to move in the real arena so as to intercept the coming ball. This will be reflected in the virtual arena by our game.
* The marker provided to stick on the bot will be Red in Colour. This will be used by organisers to detect the position of the bot in the virtual arena. The marker may be rotated by the player to rotate the paddle in the virtual arena to aid the hitting of bricks.
* Bricks of different colours are present in the arena, and they will all fetch different points(15,20 or 25).(Moving brick in round fetches 50 points)

**Materials**

* Soldering Iron
* Laptop
* (USB Webcam)
* MATLAB
* 1 Microcontroller (Atmega 32)
* 1 Programmer (ISP)
* 4 Wheels
* Metal Chassis
* 2 DC motors
* 1 Servo motor
* 1 Development Board
* Motor Driver (L293D)
* Relimates

**Reason for choice of components and summary:**

The video feed of the game screen will be provided to us, which will be fed into our laptop. We are using MATLAB for image processing. We have chosen MATLAB, as Matlab allows fast prototyping, for instance in order to test a method, compare the accuracy of different techniques, estimate which implementation runs faster etc. It also allows faster debugging. MATLAB was chosen over Open CV because time or memory efficiency is not the theme of this task. The data obtained from image processing will be fed into the microcontroller, in which code will already be fed. We have currently decided on ATMega microcontroller, as one of the group members already had one, and it would be very cheap anyway. To connect the pins of the microcontroller with various components, we will need a breadboard (already present with one of the members), relimate connecting wires and a soldering iron. Relimate connectors are being used as they provide for fast, strong and neat soldering. For developing, testing and debugging the code, we are using the ISP programming software. The microcontroller will feed its output into the L293D motor driver, which will apply the corresponding voltage across the motor for movement of the robot in x direction. A single motor per wheel is used to prevent unwanted rotation effects, and because high speed is not required anyway (judging from the video). To support the movement of the entire robot smoothly, 2 back wheels and 2 front wheels will be used. For rotation of paddle, we will use a servo motor, as it will provide angular precision (It will rotate only upto a certain angle and then stop and wait for next signal unlike a regular motor which keeps on rotating). This will be controlled by another signal generated by the microcontroller , and will control the direction of the ball after bouncing from paddle. The robot design is kept simple, as it involves minimum movement. It consists of a metal rectangular chassis (dimensions will be decided after components and their arrangement is finalized, and will easily satisfy the constraints of the problem). Below the chassis will be the wheels(2) in the rear and castors(2) in the front. In the body will be the development board and the motors. There will be a separate protruding arm for the servomotor which will be connected to marker.