ARDUINO UNO CODE:

```
#include <AFMotor.h>                    //Import library to control motor shield
#include <Servo.h>                      //Import library to control the servo

AF_DCMotor rightBack(1);                //Create an object to control each motor
AF_DCMotor rightFront(2);
AF_DCMotor leftFront(3);
AF_DCMotor leftBack(4);
Servo servoLook;                        //Create an object to control the servo

byte trig = 2;                          //Assign the ultrasonic sensor pins
byte echo = 13;
byte maxDist = 150;       //Maximum sensing distance (Objects further than this distance are
ignored)
byte stopDist = 50;               //Minimum distance from an object to stop in cm
float timeOut = 2*(maxDist+10)/100/340*1000000;       //Maximum time to wait for a return
signal

byte motorSpeed = 55;             //The maximum motor speed
int motorOffset = 10;             //Factor to account for one side being more powerful
int turnSpeed = 50;               //Amount to add to motor speed when turning


void setup()
{
 rightBack.setSpeed(motorSpeed);          //Set the motors to the motor speed
 rightFront.setSpeed(motorSpeed);
 leftFront.setSpeed(motorSpeed+motorOffset);
 leftBack.setSpeed(motorSpeed+motorOffset);
 rightBack.run(RELEASE);                  //Ensure all motors are stopped
 rightFront.run(RELEASE);
 leftFront.run(RELEASE);
 leftBack.run(RELEASE);
 servoLook.attach(10);                    //Assign the servo pin
 pinMode(trig,OUTPUT);                    //Assign ultrasonic sensor pin modes
 pinMode(echo,INPUT);
}

void loop()
{
 servoLook.write(90);                     //Set the servo to look straight ahead
 delay(750);
 int distance = getDistance();            //Check that there are no objects ahead
```

```
  if(distance >= stopDist)          //If there are no objects within the stopping distance, move
forward
  {
   moveForward();
  }
  while(distance >= stopDist)                  //Keep checking the object distance until it is
within the minimum stopping distance
  {
   distance = getDistance();
   delay(250);
  }
  stopMove();                                  //Stop the motors
  int turnDir = checkDirection();       //Check the left and right object distances and get the
turning instruction
  Serial.print(turnDir);
  switch (turnDir)                 //Turn left, turn around or turn right depending on the instruction
  {
   case 0:                                     //Turn left
     turnLeft (400);
     break;
   case 1:                                     //Turn around
     turnLeft (700);
     break;
   case 2:                                     //Turn right
     turnRight (400);
     break;
  }
}

void accelerate()                              //Function to accelerate the motors from 0 to full speed
{
  for (int i=0; i<motorSpeed; i++)               //Loop from 0 to full speed
  {
   rightBack.setSpeed(i);                      //Set the motors to the current loop speed
   rightFront.setSpeed(i);
   leftFront.setSpeed(i+motorOffset);
   leftBack.setSpeed(i+motorOffset);
   delay(10);
  }
}

void decelerate()                              //Function to decelerate the motors from full speed to zero
{
  for (int i=motorSpeed; i!=0; i--)              //Loop from full speed to 0
  {
   rightBack.setSpeed(i);                      //Set the motors to the current loop speed
```

```
    rightFront.setSpeed(i);
    leftFront.setSpeed(i+motorOffset);
    leftBack.setSpeed(i+motorOffset);
    delay(10);
  }
}

void moveForward()                    //Set all motors to run forward
{
  rightBack.run(FORWARD);
  rightFront.run(FORWARD);
  leftFront.run(FORWARD);
  leftBack.run(FORWARD);
}

void stopMove()                       //Set all motors to stop
{
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);
}

void turnLeft(int duration)           //Set motors to turn left for the specified duration then stop
{
  rightBack.setSpeed(motorSpeed+turnSpeed);          //Set the motors to the motor speed
  rightFront.setSpeed(motorSpeed+turnSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset+turnSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
  rightBack.run(FORWARD);
  rightFront.run(FORWARD);
  leftFront.run(BACKWARD);
  leftBack.run(BACKWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);                    //Set the motors to the motor speed
  rightFront.setSpeed(motorSpeed);
  leftFront.setSpeed(motorSpeed+motorOffset);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  rightFront.run(RELEASE);
  leftFront.run(RELEASE);
  leftBack.run(RELEASE);

}
```

```
void turnRight(int duration)        //Set motors to turn right for the specified duration then stop
{
 rightBack.setSpeed(motorSpeed+turnSpeed);              //Set the motors to the motor speed
 rightFront.setSpeed(motorSpeed+turnSpeed);
 leftFront.setSpeed(motorSpeed+motorOffset+turnSpeed);
 leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
 rightBack.run(BACKWARD);
 rightFront.run(BACKWARD);
 leftFront.run(FORWARD);
 leftBack.run(FORWARD);
 delay(duration);
 rightBack.setSpeed(motorSpeed);                        //Set the motors to the motor speed
 rightFront.setSpeed(motorSpeed);
 leftFront.setSpeed(motorSpeed+motorOffset);
 leftBack.setSpeed(motorSpeed+motorOffset);
 rightBack.run(RELEASE);
 rightFront.run(RELEASE);
 leftFront.run(RELEASE);
 leftBack.run(RELEASE);
}

int getDistance()                           //Measure the distance to an object
{
 unsigned long pulseTime;                   //Create a variable to store the pulse travel time
 int distance;                              //Create a variable to store the calculated distance
 digitalWrite(trig, HIGH);                  //Generate a 10 microsecond pulse
 delayMicroseconds(10);
 digitalWrite(trig, LOW);
 pulseTime = pulseIn(echo, HIGH, timeOut);        //Measure the time for the pulse to return
 distance = (float)pulseTime * 340 / 2 / 10000;    //Calculate the object distance based on the
pulse time
 return distance;
}

int checkDirection()                                //Check the left and right directions and
decide which way to turn
{
 int distances [2] = {0,0};                         //Left and right distances
 int turnDir = 1;                                   //Direction to turn, 0 left, 1 reverse, 2 right
 servoLook.write(180);                              //Turn servo to look left
 delay(500);
 distances [0] = getDistance();                     //Get the left object distance
 servoLook.write(0);                                //Turn servo to look right
 delay(1000);
 distances [1] = getDistance();                     //Get the right object distance
```

```
  if (distances[0]>=200 && distances[1]>=200)              //If both directions are clear, turn left
    turnDir = 0;
  else if (distances[0]<=stopDist && distances[1]<=stopDist)    //If both directions are
blocked, turn around
    turnDir = 1;
  else if (distances[0]>=distances[1])                     //If left has more space, turn left
    turnDir = 0;
  else if (distances[0]<distances[1])                      //If right has more space, turn right
    turnDir = 2;
  return turnDir;
}
```