Assignment 4 - LRL

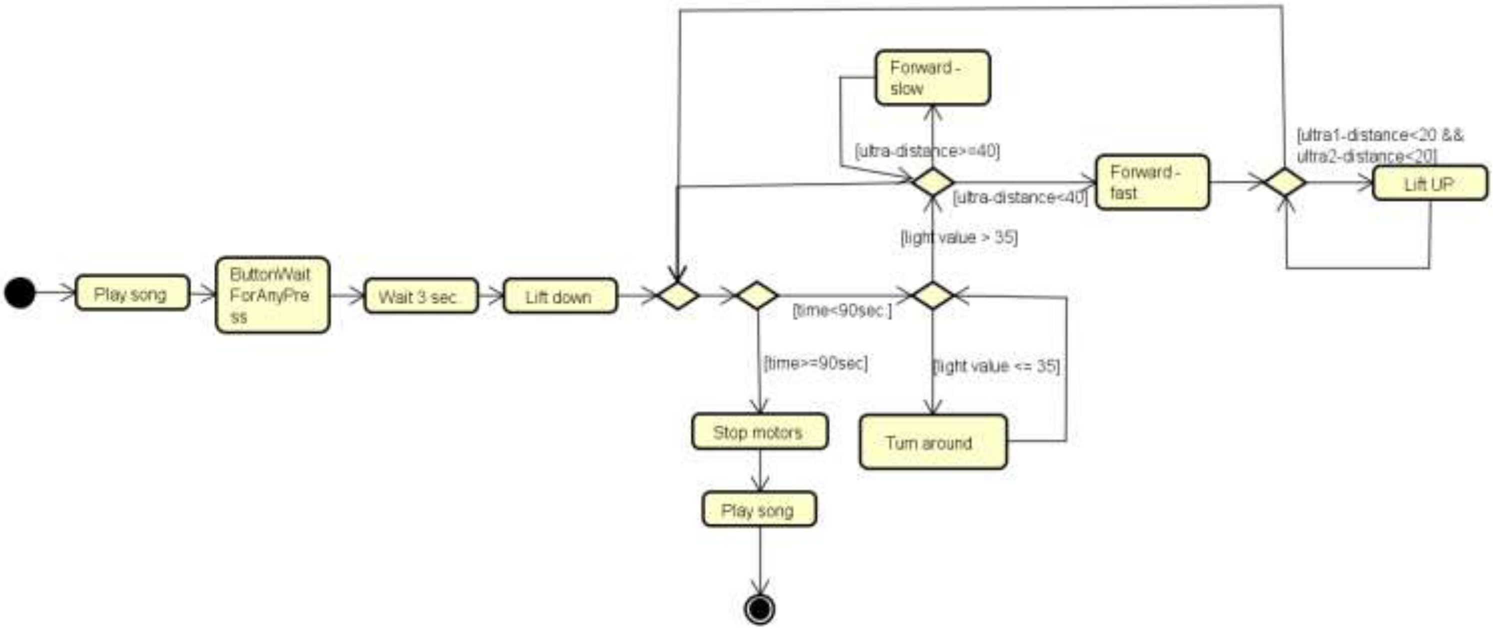
1. Description of how you solve the problem.

This task – assignment – was significant different that the previous ones. This time we had to prepare robot for the environment that is not exactly known. It was supposed to fight against other robot that we didn't know its tactics or construction.

First we had to build a proper robot that will be able to fight and stay on the arena. We created a shovel in front of the robot which aim was to push the opponents. There were also two Ultrasonic Sensors placed in front of the robot to detect the opponents. Third motor was installed that operated a lever which should try to pick up the other robot when it's really close. We used bigger tires than usually to enlarge the friction force between the robot and the arena's floor. Finally the Light Sensor was placed directed to the floor to detect the light values (to keep the track on the arena). It was placed in the really front of the robot so when it turns around it will not go out of the arena.

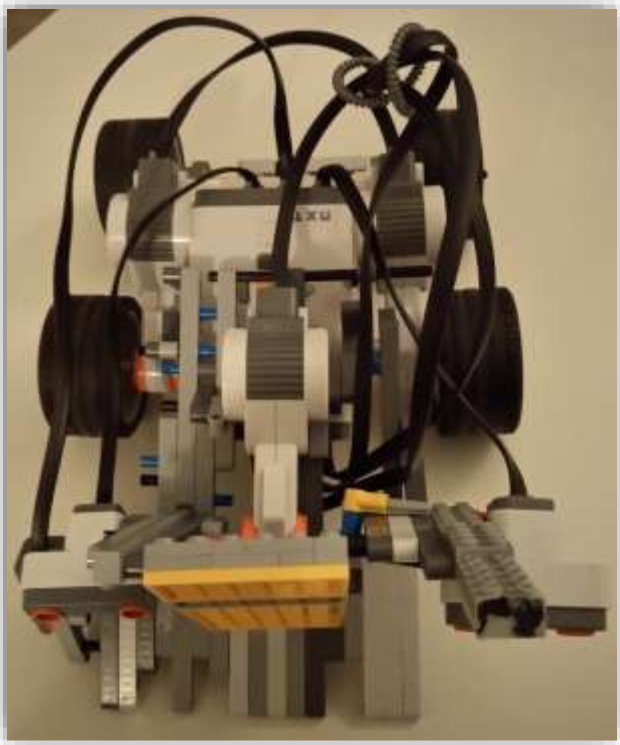
After building our Lego Sumo Fighter named "Andrzej" we had to train it (program it…) to be able to beat its opponents. The first think that was done was to program it so as when it detects the black border of the arena it turns around and continues to attack the other robot. Unfortunately because of big tires it had to be done by going backwards instead of turning around in the same place (one wheel forward, one backwards). The fight was supposed to be maximum 90 seconds long that's why we put everything in a while loop that last this amount of time. As it was required before the fight there was song played and the robot waited to start by pressing any button. There were some if statements implemented. First of all when the robot is on the white surface it uses Ultrasonic Sensor to detect the opponent. When the distance is less than 40 centimeters it goes forward with the full power (100), and when it gets even closer to the opponent – 20 centimeters – it uses the lever and tries to lift up the other robot. Also while turning around when there is an opponent detected by Ultrasonic Sensor – it stops turning around and attacks the opponent. When there is no opponent detected the robot drives with power of 75.

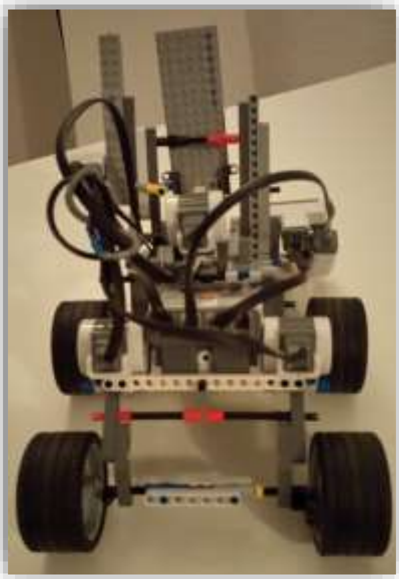
2. Activity diagram.

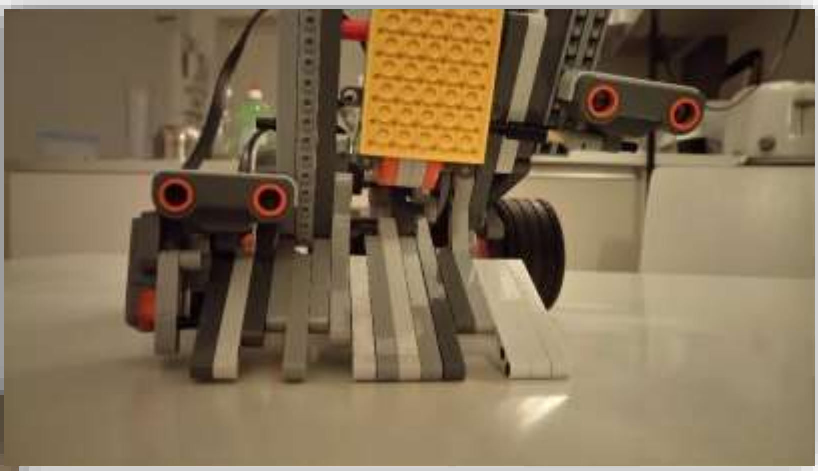


3. How the robot was constructed.

We decided to use big tires this time to enlarge friction force between robot and the floor to make it more difficult for others to push our robot out of the arena. We used four wheels like this and the two front ones are plugged to motor ports. At the front of the robot we built big shovel that's aim is to push the opponents in an effective way. It is directed to the floor with some angle that should make it easier to push the robot. Between there is a lever that can try to uplift the opponent when it is very closed.







Three sensors were used. Two of them are Ultrasonic Sensors placed at the front of the robot. There was no place for it in the middle, so we decided to place two on the sides. Both should detect if there is an opponent and if one of them does it is enough for the robot to accelerate. The third sensor is the Light Sensor that detects the field in the arena – whether it is inside (white area) or it touched the border (black area). It was placed right on the front to detect it as soon as possible.

4. The code

```
import lejos.nxt.*;

public class As1a
{
   UltrasonicSensor ultra;
   UltrasonicSensor ultra2;
   LightSensor light;
   TouchSensor touch;
   MotorPort A;
   MotorPort B;
   MotorPort C;
   int time = (int) (System.currentTimeMillis() + 90000);

   public As1a()
   {
      ultra2 = new UltrasonicSensor(SensorPort.S1);
      ultra = new UltrasonicSensor(SensorPort.S3);
      light = new LightSensor(SensorPort.S2);
      A = MotorPort.A;
      B = MotorPort.B;
      C = MotorPort.C;
   }

   public static void main(String[] args) throws InterruptedException
   {
      As1a fight = new As1a();

      fight.playSong();
      Button.waitForAnyPress();
      Thread.sleep(3000);
      fight.liftDown();
      fight.letsFight();
      fight.stopMotors();
      fight.playSong();
   }

   public void calibrate()
   {
      while (!Button.ENTER.isDown())
      {
         LCD.drawString("Press LEFT to calibrate LOW and RIGHT to calibrate 
HIGH", 0, 0);
         LCD.drawInt(light.getLightValue(), 0, 3);
         if (Button.LEFT.isDown())
         {
            light.calibrateLow();
            LCD.drawString("Calibrated LOW", 0, 0);
         }
         if (Button.RIGHT.isDown())
         {
            light.calibrateHigh();
            LCD.drawString("Calibrated HIGH", 0, 1);
         }
      }
   }

   public void playSong()
   {
      Sound.playTone(262, 3644);
      Sound.pause(1280);
      Sound.playTone(392, 1642);
      Sound.pause(740);
      Sound.playTone(349, 166);
      Sound.pause(250);
      Sound.playTone(330, 166);
      Sound.pause(250);
      Sound.playTone(294, 166);
      Sound.pause(250);
      Sound.playTone(523, 3642);
      Sound.pause(1250);
      Sound.playTone(392, 1100);
      Sound.pause(1150);
      Sound.playTone(349, 166);
      Sound.pause(250);
      Sound.playTone(330, 166);
      Sound.pause(250);
      Sound.playTone(294, 166);
      Sound.pause(250);
      Sound.playTone(523, 3642);
      Sound.pause(1250);
      Sound.playTone(392, 1400);
      Sound.pause(1250);
      Sound.playTone(349, 266);
      Sound.pause(250);
      Sound.playTone(330, 266);
      Sound.pause(250);
      Sound.playTone(349, 266);
      Sound.pause(250);
      Sound.playTone(294, 1642);
      Sound.pause(1250);
   }

   public void liftDown()
   {
      C.resetTachoCount();
      while (C.getTachoCount() < 90)
      {
         C.controlMotor(60, 1);
      }
      C.controlMotor(60, 3);
      C.resetTachoCount();
   }

   public void liftUp()
   {
      while (C.getTachoCount() > -90)
      {
         C.controlMotor(100, 2);
      }
      C.controlMotor(100, 3);
   }

   public void forwardFast()
   {
      A.controlMotor(100, 1);
      B.controlMotor(100, 1);
   }

   public void forwardSlow()
```

```
   {
      A.controlMotor(75, 1);
      B.controlMotor(75, 1);
   }

   public void turn()
   {
      A.resetTachoCount();
      while (A.getTachoCount() > -100)
      {
         A.controlMotor(100, 2);
         B.controlMotor(100, 2);
      }
      A.resetTachoCount();
      while (A.getTachoCount() > -840)
      {
         if (ultra.getDistance() > 25 || ultra2.getDistance() > 25)
         {
            A.controlMotor(75, 2);
            B.controlMotor(75, 3);
         }
         else if ((ultra.getDistance() <= 25 && ultra2.getDistance() <=
25))
         {
            forwardFast();
            break;
         }
      }
   }

   public void stopMotors()
   {
      A.controlMotor(100, 3);
      B.controlMotor(100, 3);
      C.controlMotor(100, 3);
   }

   public void letsFight()
   {
      while (System.currentTimeMillis() < time)
      {
         if (light.getLightValue() > 35)
         {
            if (ultra.getDistance() < 40 || ultra2.getDistance() < 40)
            {
               forwardFast();
               if (ultra.getDistance() < 20 && ultra2.getDistance() < 20)
               {
                  liftUp();
               }
            }
            else
            {
               forwardSlow();
            }
         }
         else
         {
            turn();
         }
      }
```

```
        }
}
```