POLYTECHNIC OF CRETE

SCHOOL OF ENGINEERING

COURSE: INF 412 - AUTONOMOUS AGENTS

Assignment 1: The Webots Simulator

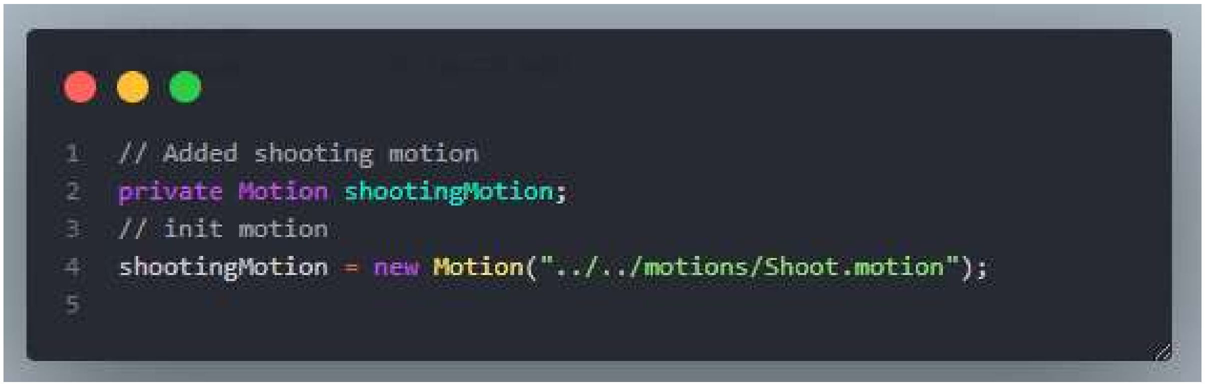
Name: Nikolaos Papoutsakis
Student ID: 2019030206

Introduction

In the context of this lab exercise we worked with the Webots simulation environment. We studied the code of the nao_team_1 folder and our goal was to optimize the controllers for the NAOs and add more functionality. The world in which this code is developed is a copy of the world in the Webots directory.

shooting motion

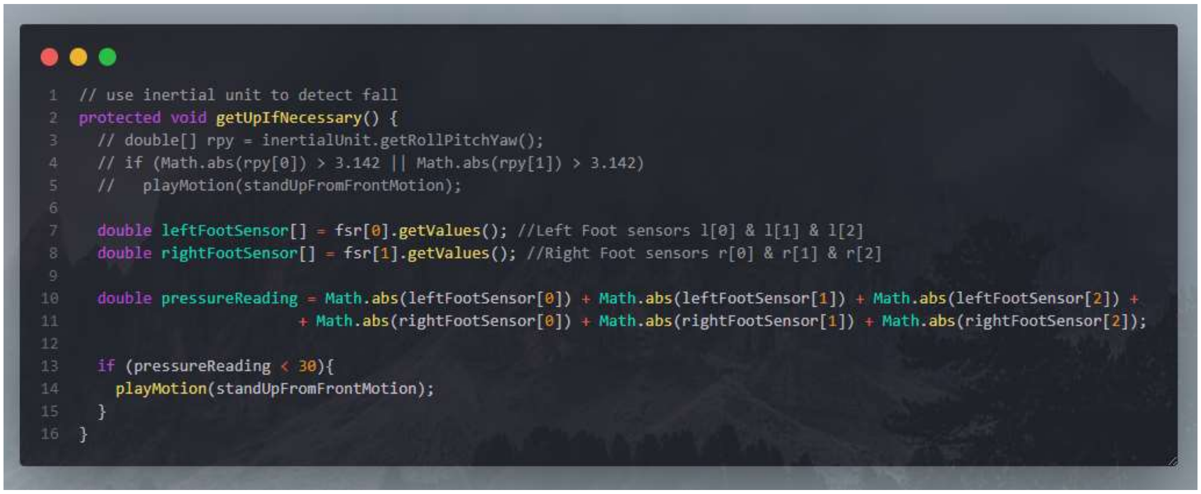
To implement the motion for the shot, we had to add the motion to the parameters of the FieldPlayer.java file and create a new motion. This was done as follows:



```
// Added shooting motion
private Motion shootingMotion;
// init motion
shootingMotion = new Motion("../../motions/Shoot.motion");

```

Get up if necessary:

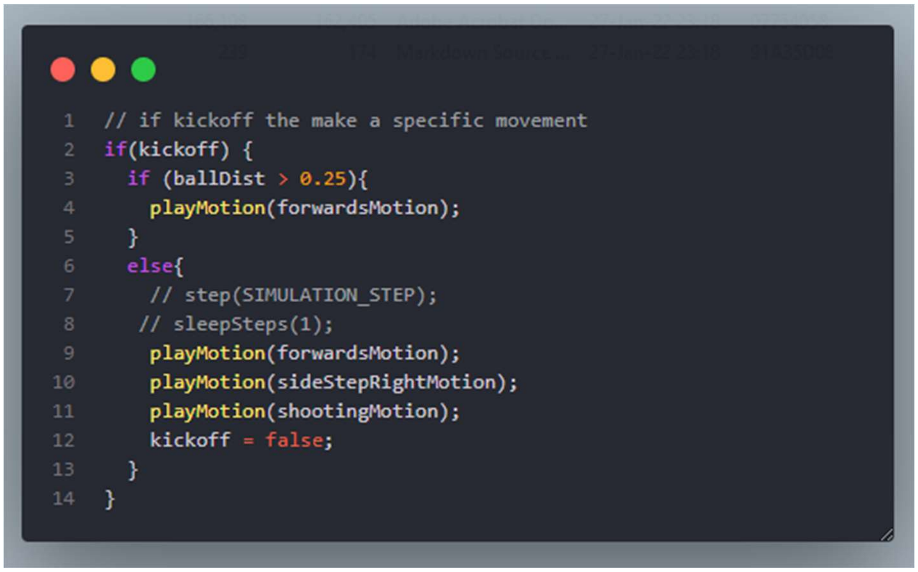
For the implementation of this function, we used the touch sensors with which we were able to get information about the 'pressures' of each leg of the robot. Thus, when the robot was lying on the ground, we were able to set a small threshold in order to define when the standUpFromFront.motion would be executed.



```
// use inertial unit to detect fall
protected void getUpIfNecessary() {
  // double[] rpy = inertialUnit.getRollPitchYaw();
  // if (Math.abs(rpy[0]) > 3.142 || Math.abs(rpy[1]) > 3.142)
  //   playMotion(standUpFromFrontMotion);

  double leftFootSensor[] = fsr[0].getValues(); //Left Foot sensors l[0] & l[1] & l[2]
  double rightFootSensor[] = fsr[1].getValues(); //Right Foot sensors r[0] & r[1] & r[2]

  double pressureReading = Math.abs(leftFootSensor[0]) + Math.abs(leftFootSensor[1]) + Math.abs(leftFootSensor[2]) +
                         + Math.abs(rightFootSensor[0]) + Math.abs(rightFootSensor[1]) + Math.abs(rightFootSensor[2]);

  if (pressureReading < 30){
    playMotion(standUpFromFrontMotion);
  }
}
```

Run:

In the run() method we added additional logic so that the robot can approach the ball and shoot at the goal. This is achieved by using variables ballDir, ballDist and goalDir. A small screenshot of this logic is shown below:



```
// if kickoff the make a specific movement
if(kickoff) {
  if (ballDist > 0.25){
    playMotion(forwardsMotion);
  }
  else{
    // step(SIMULATION_STEP);
   // sleepSteps(1);
    playMotion(forwardsMotion);
    playMotion(sideStepRightMotion);
    playMotion(shootingMotion);
    kickoff = false;
  }
}
```