```
double getSensor(String sonars, int curr_x, int curr_y) {
	int match = 0;
	for (int y = 1; y < mundo.height - 1; y++) {
		for (int x = 1; x < mundo.width - 1; x++) {
			if (x == curr_x && y == curr_y) {
				if (mundo.grid[x][y - 1] == Integer.parseInt(Character.toString(sonars.charAt(0)))) {
					++match;
				}
				if (mundo.grid[x][y + 1] == Integer.parseInt(Character.toString(sonars.charAt(1)))) {
					++match;
				}
				if (mundo.grid[x + 1][y] == Integer.parseInt(Character.toString(sonars.charAt(2)))) {
					++match;
				}
				if (mundo.grid[x - 1][y] == Integer.parseInt(Character.toString(sonars.charAt(3)))) {
					++match;
				}
			break;
			}
		}
	}

	double result = Math.pow(sensorAccuracy, match) * Math.pow((1 - sensorAccuracy), (4 - match));
	return result;
}

double getTransition(int action, int curr) {

	if (action == curr) {
		// Pm
		return moveProb;
	}
	return (1 - moveProb) / 4;
}

void updateProbabilities(int action, String sonars) {
	// your code

	double sum = 0.0;
	double[][] newProbs = new double[mundo.height][mundo.width];
	for (int y = 0; y < mundo.height; y++) {
		for (int x = 0; x < mundo.width; x++) {
			newProbs[x][y] = 0.0;
		}
	}

	// Transition model
	for (int y = 1; y < mundo.height - 1; y++) {
		for (int x = 1; x < mundo.width - 1; x++) {

			if (probs[x][y] != 0.0) {
				newProbs[x][y] += getTransition(action, STAY) * probs[x][y];
			}
			if (probs[x][y-1] != 0.0) {
				newProbs[x][y - 1] += getTransition(action, NORTH) * probs[x][y];
			}
			if (probs[x][y+1] != 0.0) {
				newProbs[x][y + 1] += getTransition(action, SOUTH) * probs[x][y];
			}
			if (probs[x-1][y] != 0.0) {
				newProbs[x - 1][y] += getTransition(action, WEST) * probs[x][y];
			}
			if (probs[x+1][y] != 0.0) {
				newProbs[x + 1][y] += getTransition(action, EAST) * probs[x][y];
			}
```

```
            }
        }

        // Sensor Model
        for (int y = 1; y < mundo.height - 1; y++) {
            for (int x = 1; x < mundo.width - 1; x++) {
                        newProbs[x][y] *= getSensor(sonars, x, y);
                        sum += newProbs[x][y];
            }
        }

        double test = 0.0;
        for (int y = 0; y < mundo.height; y++) {
            for (int x = 0; x < mundo.width; x++) {
                newProbs[x][y] = newProbs[x][y] / sum;
                test += newProbs[x][y];
            }
        }

        for (int y = 0; y < mundo.height; y++) {
            for (int x = 0; x < mundo.width; x++) {
                probs[x][y] = newProbs[x][y];
            }
        }

        myMaps.updateProbs(probs); // call this function after updating your probabilities so that the
                                   //    new probabilities will show up in the probability map on the GUI
}
```

I think that better understanding how transition model works would help me make progress on this lab faster. Other than that, I think everything was clear.