FINAL REPORT ROBOTLAB

Laura Tigchelaar
Tjalling Haije

Pepper interactive games project

As the final assignment for the RobotLab Practical course, we have developed an application for the Pepper robot, where the user can play a set of interactive games with the robot.
In the following few chapters the requirements for the project and the functionality of the application will be discussed briefly, followed by a description of the two games and their implementation.

Project Requirements

For the final assignment the goal is to demonstrate the skills and knowledge learned during the previous assignments over the last weeks. These subjects include:
- Movement control over the robot, such as letting the robot assume (custom) postures or moving body parts separately.
- Sensor control over the robot, such as reacting to events fired by the touch sensors of the robot, using the microphones for speech recognition, and/or the camera for object recognition.
- Use of multithreading to ease projects requiring multiple modalities concurrently
- Using external sensors such as the Kinect for gesture recognition
- Combining all these factors in a working demo which is able to deal with the sometimes unexpected behaviour of the sensors, which is an inherent problem to working with robots.

Project Description

To conclude the assignments and work for this course, we decided to work with the robot Pepper. All our assignments were done using a Nao robot with the Naoqi library. However, as the Pepper robot is created by the same company, it has many commonalities with the Nao robot in its design, and aside from a few minor changes most of our previous code worked for Pepper.
The main advantage of Pepper is that the sensors and hardware are of higher quality, and furthermore is it also newer. Another difference is that it has wheels instead of legs, making it able to move more easily.

The goal of our project was to create an application which can be run on the Pepper robot and could be used as a fun demo of what a robot can do, for instance during open days of the university to enthuse children, young adults and other visitors about robots. As such we created a demo in which the user can play a set of games with Pepper. Right now these include a version of Marco Polo, and the I spy with my little eye game.

After starting the application Pepper will give the user the option to choose which game they want to play. Touching the head will initiate Marco Polo, touching the left hand will start I spy with my little eye, and touching the left hand will quit the program.

Marco Polo

The Marco Polo game is implemented as a slight variation on the Marco Polo hide-and-seek game. In this version Pepper has to find the user without vision, only based on the voice of the user.

After explaining the rules, Pepper will turn around facing the wall, and do a countdown while covering its eyes. When the countdown has reached zero, Pepper will turn around and start searching for the user by calling Marco, and listening for the user replying Polo. The Naoqi wordspotting proxy is used to recognize if the user said Polo, and if so uses the audio levels from the four microphones to pinpoint the audio direction. Pepper then rotates towards where it thinks it heard the sound, and tries to drive towards it. During the driving all sensors (lasers and ultrasonic sensors around base) are used to detect any obstacles. If any obstacles are detected, Pepper will try to maneuver around them to arrive near the user.
When the audio volume of the user speaking Polo is above a certain threshold, Pepper has found the user and the game has ended.

An alternative ending is when Pepper gets frustrated during the game. If the user doesn't reply with Polo when Pepper calls Marco, Pepper becomes frustrated. Likewise does the frustration of Pepper increase when it is unable to find the user after a long time of searching. If the frustration of Pepper exceeds a certain level, Pepper becomes irritated and the eyes sometimes flash red (is for a short duration as the running speech recognition also uses eye leds). When the frustration exceeds another threshold, Pepper becomes angry and quits the game, blaming it on bad sportsmanship of the user.

I Spy with my Little Eye

The I spy with my little eye game is also an adjusted version of the original game. In our version of the game, the user presents several coloured balls to Pepper by placing them somewhere in the room. After the user tells Pepper what colour ball to look for, Pepper will try and find it.

The game starts with the general rules of the game, so the user will know what to do. Then the user is instructed to tell Pepper which colour the intended ball is. Once Pepper has recognized one of the know colours, namely: blue, red, green, pink and yellow, the colour will be confirmed verbally and Peppers eyeleds will change into that colour. Now, the ball recognition process will start. While Pepper is analyzing what it sees, it looks around the room, making a 90 degree turn in a random direction every once in a while. Once Pepper has identified a ball in specified colour, it will stop moving, focus eye gaze on the ball, and ask the user if that would be the intended ball. The user now says either no or wrong, if another ball was intended, and Pepper will keep searching for the ball, or the user says yes or correct, if it was indeed the intended ball, and the game will be over.

MultithreadFramework.py

Main Thread

Setup

Setup threads & variables

Main

Which game do you want to play?

Listen for Touch events

ReactToTouch.py

Module for touch events

Catches TouchChanged events

Touched bodyparts

If hand touched

If head touched

Marco Polo Thread

Countdown

Do hands before eyes posture

Postures.py

Uses ALRobotPosture proxy to execute custom postures

Say Marco

Word recognized

If Polo heard

WordSpotting.py

ALSpeechRecognition proxy for wordspotting

Catches WordRecognized events

Locate the user and rotate

Drive towards user

Direction and energy sound

SoundLocalization.py

Module for soundlocalization

Catches SoundLocated events

I Spy with my Little eye thread

User picks ball

Recognized Ball Colour

Ball colour

Center on ball

Ball Detection Thread

Ball colour

Camera.py

Uses OpenCV to find balls

Locate ball

Ball located, Ball location

Ball located, Ball location