

Easter Release RoboDucks 3BHIF

Features

Features Overview

Since we have decided to switch from our framework to the hulk-framework, we concentrated on understanding the given code and becoming the code to run on the naos and on the simrobot.

Feature	Description
SimRobot	Software to simulate soccer games
Playable Nao	Hulk Code runs on the nao and is playable

Details

SimRobot

SimRobot is a software used for simulating a soccer game. Later on SimRobot will be a main tool for testing our own vision module. It is also quite convenient to test the general behavior of the robots on SimRobot because it is much easier to just start the software instead of a real nao.

SimRobot currently only works on linux. It might be possible to get it to run on windows but it is not known yet.

Before installing the Simrobot or compiling code for it you need to install the libraries which are mentioned in the [Hulks Documentation](#) on page 6.

To install SimRobot you first have to execute a shell script which is provided by our framework. This shell script can be found under the path:

```
./tools/SimRobot/build_simrobot
```

After this you need to Compile the code. How to compile the code is described under the section "compile the code". Then you have to start SimRobot under the path:

```
./tools/SimRobot/build/SimRobot
```

You will see a window which pops up. Under File/open you can open specific scenes. A scene is a file which contains the start information of a soccer game like the number of robots, the team numbers and so on.

Real Gameplay

To run the code on the real robot, the code must be compiled first (which is described below) then you have to use the following script:

```
./scripts/upload IP-ADDRESS
```

If that is finished you have to connect to the robot via ssh and then you have to go into the directory naoqi/bin. After that you have to execute the script which is in that specific directory.

Compile the code

For compiling the code for the Nao you need to get the toolchain files from one of our team members, because there are too many files for GitHub. If you want to compile Code for the SimRobot you don't need them.

When you like to compile the code you need to run the following scripts in this order.

```
./scripts/setup TARGET  
./scripts/compile -t TARGET -b BUILDTYPE
```

Target = nao/simrobot/replay

Buildtype = Debug/Develop/Release/RelWithDebInfo

Open Issues

- Finish soccer field

Since monday, the 15 April, we have a soccer field on which we can test our robots and gameplay. The only problem is, that the field is unfinished. Some lines are missing and one penalty area. This has to be fixed.

- test the light situation in the robo lab

In our robo lab, which is located in the cellar of our school, we have plenty of space but unfortunately the light sources are very limited. So we have to check if it is bright enough for the vision engine to work properly. If it is not bright enough we have to install some lights which will be provided from a team mate of us.

- understand tuhh sdk completely

In order to develop our own vision and motion modules we first have to understand how the tuhh and the different modules work and communicate between each other.

- improve shut down process

Currently we have the problem, that the robots lose their stiffness if we shut down tuhh. This behavior can lead to damage on the robots which we would like to avoid. In order to do this, we first have to understand how the threads which are running the different modules are started and closed. After we understand that, we will implement a behavior which moves the nao into a safe position before removing the stiffness.

Tests

Name of Test	Description	Status
run script compile	Run the script compile which is used to compile the code	passed
run script upload	Upload the code on the nao	passed
run script setup	Prepares the code to be compiled on a specific target	passed
run script build_simrobot	Builds the SimRobot software	passed
start the Simrobot	Start the SimRobot	passed
load Scenes into the Simrobot	load the scenes and look if they are playable	passed
create own Scenes for the Simrobot and load them	create scenes and look if they are playable	passed
simulate a game with the Simrobot and connect it with the Gamecontroller	Simulate a game with the simrobot and look if they are reacting to the gamecontroller	passed
listen to all gameconroller signals	test the naos if they are reacting to all gamecontroller Signals	passed
run the tuhhsdk on the Nao	try to get the code running on the nao	passed
let a nao walk behind a ball	start the tuhhsdk on a nao and lay a ball in front of him and see if he's walking towards the ball	passed
let two naos play against each other	start the tuhhsdk on 2 naos with different Teams and let them play for a few minutes	passed
let two naos play a whole game	let two naos play against each other a whole match inclusive Gamcontroller and Goals	open