

Better control of tests

Best Practices in Botball 2015

Mateusz Joniec ^{1,*}, Jakub Polański ², Bartłomiej Majkut ³, Wojciech Kojder ⁴.

¹ The Franciszek Leja State School in Grodzisko Górne, Poland

² The Franciszek Leja State School in Grodzisko Górne, Poland

³ The Franciszek Leja State School in Grodzisko Górne, Poland

⁴ The Franciszek Leja State School in Grodzisko Górne, Poland

* Corresponding author E-mail: mateusz@zsleji.pl

Abstract

During programming a robot it's often that program fail in unexpected moment. Sometimes it is hard to discover the cause of this situation. So we decided to create a library which helps in finding errors. It contains functions which are capturing different information during work of program. This data are writing to the hard disk and we may reproduce it at any time.

1. Introduction

We created a few functions helpful during programming. How can it be useful? Suppose that well-functioning program suddenly failed. If we controlled key moments in program, e.g. if we saved recording when robot was positioning to object. Now we can reproduce this data and easier detect error.

2. Figures

Main functions in our library:

a) `test([statement] , [condition] , [value])` – if we want to verify if value is correct in specific situation.

b) `test_catch_photo([statement] , [channels])` – saves photo from camera and information about position of objects (Picture 1)

c) `test_on_catching_record([statement] , [channels])` and `test_off_catching_record()` – saves series of photos and information about position of objects from camera in separated thread (one of these functions runs and another stops capturing information).

d) `test_depth_catch_photo([statement] , [x] , [y])` – saves photo from depth sensor and marks white circle in point (x ,y) (Picture 2).



Picture 1



Picture 2

3. Table

Table 1. Speeds of a few functions

Function	Frequency
camera_update()	8 Hz
depth_update()	14 Hz
test_catch_photo()	700 Hz
test..._catching_record()	9000 Hz
test_depth_catch_photo()	1200 Hz

The above table shows the speeds of a few main functions from our library. It clearly presents that it work definitely faster in comparison with procedures update_camera() and depth_update(). Thanks to that we do not have to be afraid of considerable decrease of program efficiency by their usage.