Micromouse. Od zera do sera

Micromouse. From scratch to cheese

Embedded meetup, Gdańsk 7.05.2024

About competition

➤ <u>About competition</u>

- About me, why MicroMouse?
- My approach to project development
- Post-mortem thoughts and questions
- Demonstration and Q&A

Micromouse competition

- Started around 50 years ago
- Fully autonomous mobile robot
- Size of the maze is 16x16 cells
- Each cell is 18cm wide
- Starting position is one corner
- Target position is in the middle
- Maze is unknown

Robot might do multiple runs.

Final score = (time of first run)*0,1 + (time of fastest run)



Photo credit: author's own resources

What it means

- Multi-disciplinary problems
 - mechanics
 - electronics
 - software
- Complex tasks to solve on different application level in the same time
 - mapping
 - motion planning
 - motion profiling
 - wall sensing
 - other

High entry-level



About me, why MicroMouse?

- ✤ <u>About competition</u>
 - About me, why MicroMouse?
- My approach to project development
- Post-mortem thoughts and questions
- Demonstration and Q&A

About me

Engineer at heart. Willing to construct autonomous robots that will change the world... for better



Why micromouse?

seems challenging, but achievable

dunno

learn!

Project timeline



My approach to project development

- ✤ <u>About competition</u>
- About me, why MicroMouse?
 - My approach to project development
- Post-mortem thoughts and questions
- <u>Demonstration and Q&A</u>

TDD

Kitchen conversation ucgosu course <1min test cycle

Test Driven Development (TDD)



Photo credit: https://www.kaizenko.com/what-is-test-driven-development-tdd/

Scale of the project (LOC)





Written lines of code are deployed to target

Continuous Integration

Rule: Everything done up to now must work all the time

Continuous integration (not: continuous delivery)



photo credit: https://www.davefarley.net/?p=314

O localhost:8080/job/MM_pipeline/

🔮 Jenkins

Dashboard > MM_pipeline >

Pi	peline	MM_	pip	elin

</> Changes

E Status

- D **Build Now**
- 6 Configure

0 Rename

?

0

× #294

× #293

× #292

✓ #291

× #290

#289

#288

× #287

#286

#285

-

睮 Delete Pipeline

Q Full Stage View

Pipeline Syntax

Build History

Q Filter builds...

Stage View

Regression Unit Tests failure e

Q Search (CTRL+K)

Declarative: git init build with UT build sim test with sim Checkout SCM submodules Average stage times: 7s 5s 85 7s 13s (Average full run time: ~57s) trend ~ Feb 21 20:20 1 #293 Feb 21 7s 5s 35 13ms 15**ms** Feb 21, 2024, 8:20 PM × 20:15 failed failed Failed Feb 21, 2024, 8:15 PM Feb 21 Feb 21, 2024, 8:10 PM 7s 5s 46ms 12ms 45 20:10 Feb 21, 2024, 7:34 PM Failed failed failed Feb 19, 2024, 8:31 PM Feb 21 7s 7s 10s 10s 15s Feb 18, 2024, 10:32 AM 19:34 Feb 18, 2024, 10:28 AM #290 Feb 18, 2024, 10:21 AM Feb 19 7s 5s 17s 10s 9s Feb 18, 2024, 10:19 AM 20:31 Failed Feb 18, 2024, 10:18 AM

14

Top most priority: fix it now.

Stage View



O localhost:8080/job/MM_pipeline/

🔮 Jenkins

Dashboard > MM_pipeline >

- 🖹 Status
- </>
 Changes
- D **Build Now**
- 3 Configure
- 勔 Delete Pipeline

Stage View

#294

#293 Feb 21

Feb 21

20:20

20:15

Feb 21

20:10

Feb 21

19:34

20:31

#290 Feb 19

trend v

1

- Q Full Stage View
- 0 Rename
- Pipeline Syntax
- \bigcirc **Build History**
- Q Filter builds...
- ✓ #294 Feb 21, 2024, 8:20 PM
- × #293 Feb 21, 2024, 8:15 PM
- × #292 Feb 21, 2024, 8:10 PM
- #291 Feb 21, 2024, 7:34 PM
- × #290 Feb 19, 2024, 8:31 PM #289
- Feb 18, 2024, 10:32 AM ✓ #288 Feb 18, 2024, 10:28 AM
- × #287 Feb 18, 2024, 10:21 AM

Feb 16, 2024, 7:50 PM

- Feb 18, 2024, 10:19 AM
- #285 Feb 18, 2024, 10:18 AM

⊘ <u>#284</u>

Pipeline MM_pipeline

Average stage times:

(Average full run time: ~57s)

UT repaired

Declarative:

Checkout SCM

7s

7s

7s

7s

7s

7s

git init

submodules

5s

55

5s

5s

7s

5s

build with UT

8s

95

3s

4s

10s

10s

failed

failed

build sim

7s

10s

13ms

46ms

10s

9s

failed

Failed

-

test with sim

13s

17s

15ms

12ms

15s

17s

Failed

Failed

failed

Q Search (CTRL+K)

16

No hardware!

... minimize it as much as possible



Photo credit: author's own resources



Physical modelling

First order system - step response on straight line



Split code into units

the only way to go fast is to go well



Photo by Karolina Grabowska from Pexels: <u>https://www.pexels.com/photo/photograph-of-wooden-building-blo</u> <u>cks-near-a-kid-7269687/</u>

Code architecture





Physical modelling



Post-mortem thoughts and questions

- ✤ <u>About competition</u>
- About me, why MicroMouse?
- My approach to project development
 - Post-mortem thoughts and questions
- Demonstration and Q&A

Unit tests – Holy Grail or necessary evil?

For me: the only way to produce a high quality code is by creating it in TDD framework.

Case study – Unit Tests are not always right



Simulation decision retrospective



Focus



TO DO 1	IN PROGRESS 1	DONE 8 🗸
schedule time for MM MISCELLANEOUS	take part in competition 🖌 SHOW-OFF MM-8 Om	perform full run from starting point to center when maze known FULL RUN ON HW
+ Create issue	+ Create issue	□ MM-62 ✓ 2d 🕑

Photo credit: author's own resources

Photo credit: https://pixabay.com/illustrations/rubiks-cube-cube-rubik-puzzle-toy-3347244/

SRP – single responsibility principle

A function (or module) should do exactly one thing - and it should do it properly. Applies not only for code **Robert C. Martin Series**

Clean Architecture

A Craftsman's Guide to Software Structure and Design

> Robert C. Martin With contributions by James Grenning and Simon Brown

> > Foreword by Kevlin Henney Afterword by Jason Gorman

*

Photo credit: Amazon

6

```
You, 1 second ago | 1 author (You)
     #pragma once You, 3 years ago • PID implement
     #include "ret codes.h"
 3
     #define PID CONTROLLER STRUCT SIZE BYTES 64
 5
     struct pid controller;
     typedef struct pid controller* pid controller t;
 9
     ret code t pidInit(
10
       pid controller t pid,
11
         double P,
12
         double I,
13
         double D);
14
15
     ret code t pidStep(
16
         pid controller t pid,
17
         double time,
18
         double error,
         double *output);
19
```

Another principle from uncle Bob

More or less: do not mess up with logic layers.

High level function should not interfere with low level stuff. It can call lower layer functions and pass some objects though.

If it works – don't touch it!

First implementation changed "just because"



over one week lost

Due to stupid decision which was at this time arbitrary

When to switch from sim to hw?

Infrared (IR) distance sensors - case study

Simulation is fine, but one must try the code on HW

Know your limits -> And simulate it







IR distance sensors have different measurement when angle changes

When to switch from sim to hw?

Switch fully?

Or maybe partially?

Maybe develop hw and sim simultaneously?

Keep environment and tools "portable"

VS

Make environment dependent on specific machine/OS

Development environment v1



Development environment v2



Development environment v3



Keep environment and tools "portable"

VS

Make environment dependent on specific machine/OS

Compiler is your friend

- It's your first "static code analysis" tool
- Case study (warnings are important!): "abs" vs "fabs"

Recap

- TDD write high quality code
- Cl server be sure it's always working
- SRP modularize the code
- Learn from others and books

- Simulation vs HW
- Dockerization of development tools

- Enable HW earlier to know its limitations
- Don't change approach without good justification

Demonstration and Q&A

- ✤ About competition
- About me, why MicroMouse?
- My approach to project development
- Post-mortem thoughts and questions
 - Demonstration and Q&A

DEMO

