



Ruby ile Drone Uçurmak
Onur Küçük – Özgür Yazılım A.Ş.
onur@ozguryazilim.com.tr

Envanter Robotu

- Stok takibi
- İnsanları teknoloji ile donatmak hatayı azaltmıyor
- Orta ölçekli bir depo 100 000 yuva
- Zaman, risk, masraf
- Hassasiyet yeterli değil, en az %10 hata payı

Ne Lazım ?

- Mevcut işleyişi geliştirmek
- Otonom
- Bütün depoyu yeni aygıtlarla donatmamak
- Ortam değişikliklerini takip etmek
- Robot insan etkileşimi
- Kısa sürede bol veri üretmek
- Bol veriyi verimli sunmak

Temel Araçlar

- GNU / Linux
- Özgür yazılım araçlar ve kitaplıklar
- Okumak, tasarlamak, kodlamak

Robot Operating System - ROS

- Robot uygulamaları geliştirmek için
 - Kitaplıklar
 - Geliştirici araçları
 - Sürücüler
 - Algoritmalar
- Farklı bileşenler farklı lisanslı, çekirdek BSD-3C
- Çeşitli programlama dilleri

ROS - İletişim

- Mesajlaşma altyapısı (Pub / Sub - Async)
- RPC (Request / Response - Sync)
- Mesaj kaydetme tekrarlama
- Global ayar depolama (anahtar – değer)

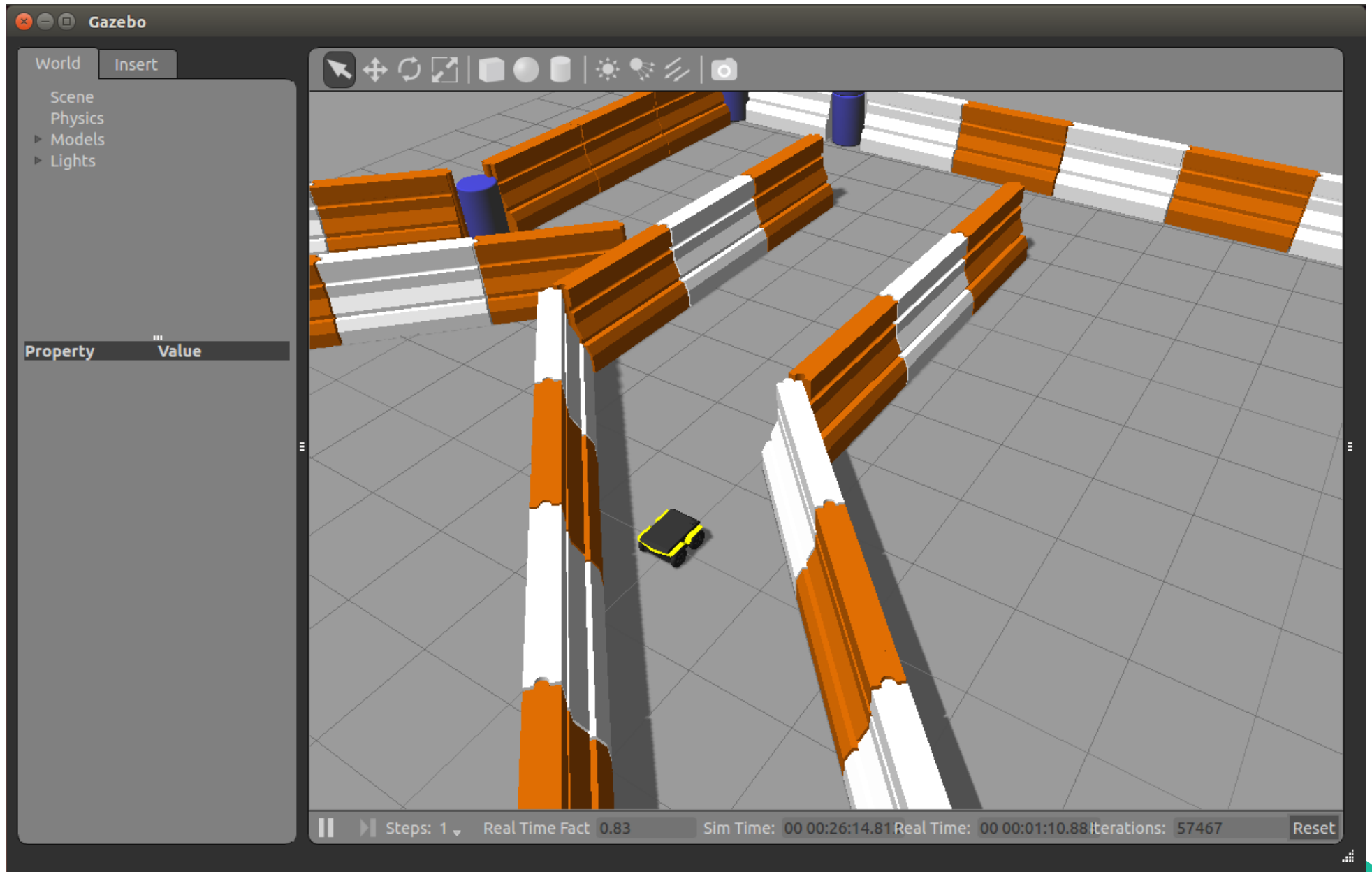
ROS – Robot Araçları

- Standart mesaj tanımları
- Geometri koordinat sistemi
- Unified Robot Description Format (URDF)
- Etkisizleştirilebilir RPC
- İzleme tanı koyma
- ...

The screenshot displays the RVIZ (Robot Visualization) interface. The central 3D view shows a white PR2 robot in a simulated environment with a point cloud background. The interface includes several panels:

- Displays:** A list of visual elements on the left side, including:
 - 01. PointCloud2 (Pol...)**: A point cloud visualization.
 - 02. Marker (Marker)**: A visualization of markers.
 - 03. RobotModel (Rob...)**: The 3D model of the PR2 robot.
 - 04. LaserScan (Laser...)**: Laser scan data.
 - 05. TF (TF)**: Transformation frames.
 - 06. PointCloud2 (Po...)**: Another point cloud visualization.
- Tool Properties:** A panel on the right showing properties for the 2D Nav Goal and 2D Pose Estimate tools.
- Views:** A panel on the right showing the current view type (Orbit) and a Zero button.
- Selection:** A panel on the right for selecting objects in the scene.
- Time:** A panel at the bottom showing Wall Time (1378418073.08), Wall Elapsed (247.36), ROS Time (1378418073.08), and ROS Elapsed (247.36).

Gazebo



Diğer Araçlar

- PX4
- Mavros
- OpenCV
- RTABMAP (SPLAM)
- SLAM (Graph, Fast ...)
- ...

Geliştirme

- Yapılacak iş için uygun araç ve dil seçilmeli
- En hızlı != en iyi yönetilebilir
- Ruby, Python, C, C++, ASM vs.

Ruby

- 1995 - Yukihiro "Matz" Matsumoto
- Geliştirici mutluluğu odaklı
- POLA (en az hayret ettirme prensibi)
- OOP hemen her şey obje
- Metaprogramlama
- DSL
- ~~Sadece Web için!~~

Ruby



```
def adult?  
  age >= 18  
end
```



```
def adult?  
  return age >= 18  
end
```



```
def adult?  
  return age >= 18 ? true : false  
end
```



```
def isAdult()  
  if age >= 18  
    return true  
  elsif age < 18  
    return false  
  end  
end
```

Ruby – Hiz

- <https://benchmarksgame-team.pages.debian.net/benchmarksgame/faster/ruby.html>
- 2018-05-10
- Ruby 2.5.0p0 (2017-12-25 revision 61468) [x86_64-linux]
- Python 3.6.3

Ruby - Hız

binary-trees

source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	50.97	367,192	1083	140.99	65% 62% 92% 59%
<u>Python 3</u>	93.55	280,624	589	337.74	92% 89% 87% 93%

spectral-norm

source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	125.65	52,368	835	484.43	96% 97% 97% 96%
<u>Python 3</u>	180.97	15,876	443	720.51	100% 100% 100% 100%

n-body

source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	641.41	8,848	1137	641.26	8% 93% 0% 0%
<u>Python 3</u>	838.39	10,324	1196	838.20	95% 1% 5% 0%

regex-redux

source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	13.14	314,092	751	7.85	90% 41% 41% 47%
<u>Python 3</u>	15.22	447,324	512	27.44	25% 33% 32% 91%

Ruby - Hız

<u>pidigits</u>					
source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	2.97	306,652	485	2.96	1% 0% 3% 100%
<u>Python 3</u>	3.43	12,716	386	3.43	100% 1% 1% 0%

<u>fannkuch-redux</u>					
source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	650.83	33,224	1432	2,526.31	99% 99% 94% 98%
<u>Python 3</u>	565.97	15,528	950	2,172.63	95% 94% 95% 100%

<u>k-nucleotide</u>					
source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	95.75	381,692	858	334.94	80% 82% 97% 95%
<u>Python 3</u>	77.65	182,700	1967	302.86	97% 99% 97% 98%

<u>fasta</u>					
source	secs	mem	gz	cpu	cpu load
<u>Ruby</u>	74.75	107,448	1069	74.60	1% 6% 100% 2%
<u>Python 3</u>	59.47	15,996	1947	138.97	55% 55% 63% 66%

Ruby C API

```
#include <stdio.h>
#include <sys/sysinfo.h>
#include "ruby.h"

VALUE uptime_in_minutes(VALUE self) {
    struct sysinfo myinfo;
    sysinfo(&myinfo);
    return rb_float_new(myinfo.uptime / 60);
}

void Init_myinfo() {
    VALUE Myinfo = rb_define_class("Myinfo", rb_cObject);
    rb_define_method(Myinfo, "uptime", uptime_in_minutes, 0);
}
```

```
>> require 'myinfo'
=> true
>> puts format('Uptime: %s', Myinfo.new.uptime)
Uptime: 245.0
```

Ruby - Inline

```
require 'inline'

class InlineTest
  inline do |builder|
    builder.include '<math.h>'
    builder.c '
      int inline_pow(int a, int n) {
        return pow(a, n);
      }'
    builder.c '
      long inline_factorial(int max) {
        int i=max, result=1;
        while (i >= 2) { result *= i--; }
        return result;
      }'
    builder.c '
      int inline_fibonacci(int n) {
        int a = 1, b = 1, c, i;
        if (n == 0) {
          return 0;
        }
        for (i = 3; i <= n; i++) {
          c = a + b;
          a = b;
          b = c;
        }
        return b;
      }'
  end
end

puts InlineTest.new.inline_factorial(5)
puts InlineTest.new.inline_fibonacci(9)
puts InlineTest.new.inline_pow(2, 10)
```

Ruby on Rails



ROS Ruby Pub / Sub

```
## Publisher
require 'ros'

node = ROS::Node.new('talker')
publisher = node.advertise('/chatter', Std_msgs::String)
msg = Std_msgs::String.new

while node.ok?
  msg.data = "Hello, rosruby: #{Time.now}"
  publisher.publish(msg)
  node.loginfo(msg.data)
  sleep(1.0)
end

## Subscriber
require 'ros'

node = ROS::Node.new('listener')

node.subscribe('/chatter', Std_msgs::String) do |msg|
  puts "message: #{msg.data}"
end

node.spin
```

ROS Ruby Service / Client

```
## ROS Service
require 'ros'

node = ROS::Node.new('sample_service_server')

node.advertise_service('/add_transform_map', TransformQuery::Type) do |req, res|
  res.mapped = req.x + req.y + req.yaw * PI
  node.loginfo("Mapping result = #{res.mapped}")
  true
end

node.spin

## ROS Client
require 'ros'

node = ROS::Node.new('sample_service_client')

if node.wait_for_service('/add_transform_map', 1)
  service = node.service('/add_transform_map', TransformQuery::Type)

  req = TransformQuery::Request.new
  res = TransformQuery::Response.new

  if service.call(req, res)
    pp res.mapped
  end
end
```

OpenCV

```
require 'opencv'  
include OpenCV  
  
image_source_file = 'sample_with_face.jpg'  
image_target_file = 'sample_output.jpg'  
data = './data/haarcascades/haarcascade_frontalface_alt.xml'  
detector = CvHaarClassifierCascade::load(data)  
image = CvMat.load(image_source_file)  
  
detector.detect_objects(image).each do |region|  
  color = CvColor::Blue  
  image.rectangle! region.top_left, region.bottom_right, :color => color  
end  
  
image.save_image(image_target_file)  
  
window = GUI::Window.new('Face detection')  
window.show(image)  
GUI::wait_key
```

D-Bus

- **IPC / RPC**
 - Session
 - System
- **Araçlar**
 - systemd
 - NetworkManager
 - Packagekit
 - Polkit
 - ...

D-Bus

```
require 'dbus'
sysbus = DBus.system_bus

upower_service = sysbus['org.freedesktop.UPower']
upower_object = upower_service['/org/freedesktop/UPower']
upower_object.introspect

upower_interface = upower_object['org.freedesktop.UPower']
on_battery = upower_interface['OnBattery']

if on_battery
  puts 'Pil kullanılıyor'
else
  puts 'Pil kullanımda değil'
end
```


Graphviz

```
require 'graphviz'

g = Graphviz::Graph.new

foo = g.add_node('Foo')
foo.add_node('Bar')

foo.attributes[:shape] = 'box3d'
foo.attributes[:color] = 'red'

puts g.to_dot
Graphviz::output(g, :path => 'test.pdf')
```

Sorular ?