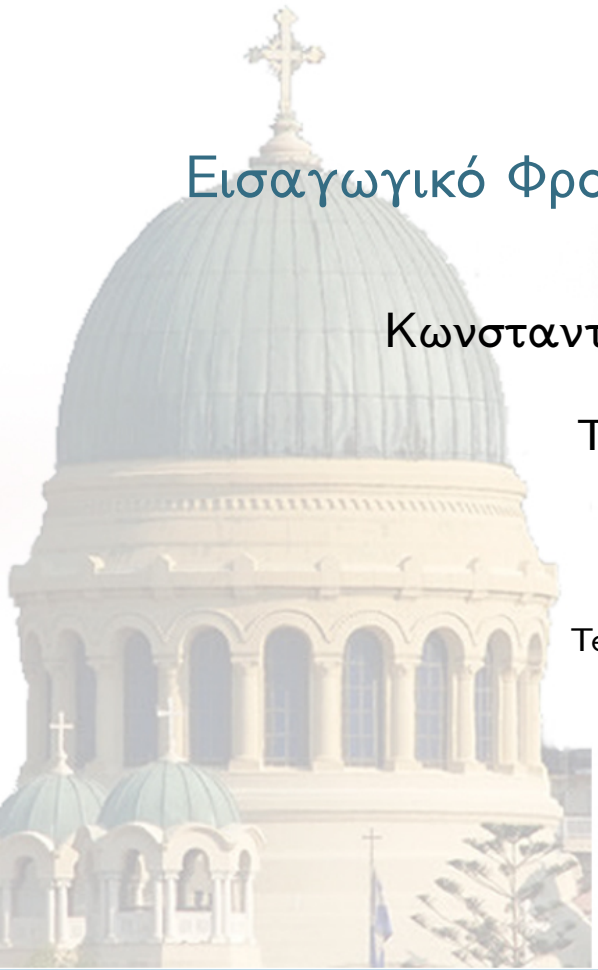


Τεχνολογίες Ευφυών Συστημάτων και Ρομποτική

Εισαγωγικό Φροντιστήριο: RobotDART και Forward Kinematics

Κωνσταντίνος Χατζηλυγερούδης - costashatz@upatras.gr

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

6 Μαρτίου 2023

Template made by **Παναγιώτης Παπαγιαννόπουλος**

Περιεχόμενα - Φροντιστήριο/Εργαστήριο

- **Ρομποτικοί Προσομοιωτές (Robotic Simulators)**
  - DART Simulator
  - robot_dart
- **Behavior Trees**
  - py_trees
  - BehaviorTree.CPP (συνοπτικά)
- **Βελτιστοποίηση Optimization**
  - ipopt
  - ifopt

Συγγράμματα

Προτεινόμενα:

- **Modern Robotics: Mechanics, Planning, and Control**, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press. ebook

Από Εύδοξο:

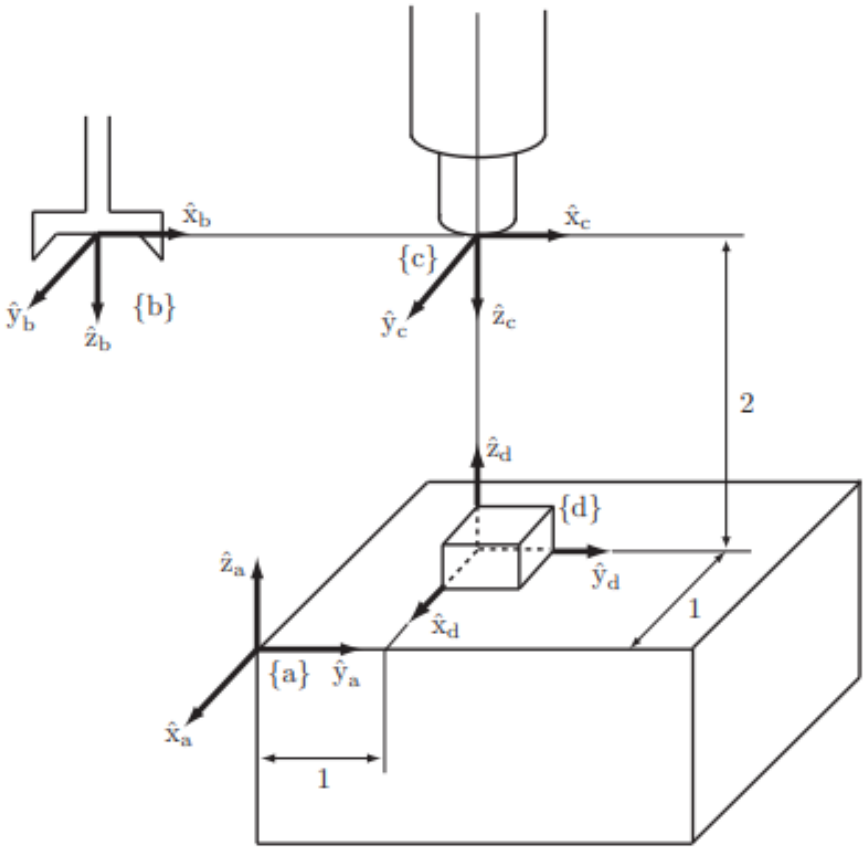
- **Εισαγωγή στη Ρομποτική**, *J. Craig*, 2020, Τζιόλα
- **Πιθανοτική Ρομποτική**, *S. Thrun, W. Burgard, D. Fox*, 2011, Κλειδάριθμος
- **Ρομποτική**, *Δ. Εμίρης, Δ. Κουλουριώτης*, 2020, Τζιόλα

Διαμόρφωση Περιβάλλοντος

Για να παρακολουθήσετε επιτυχώς το μάθημα, θα πρέπει να:

- Έχετε ένα μηχάνημα (είτε physical είτε εικονικό) με διανομή Linux
  - Θα σας δώσουμε αναλυτικές οδηγίες για Ubuntu 20.04
  - Προτείνουμε επίσης ανεπιφύλακτα και Arch Linux
- Εξασκήσετε τις ικανότητές σας σε python (κυρίως) και C++
- Έχετε διάθεση για να γράψετε κώδικα, να διαβάσετε και να ψαχτείτε

Μητρώα Μετασχηματισμών - Άσκηση



- Να βρούμε τα μητρώα μετασχηματισμού T_{ad} και T_{cd}
- Να βρούμε το T_{ab}, εάν γνωρίζουμε επιπλέον ότι:

$$T_{bc} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Πηγή: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

Εγκατάσταση/Documentation robot_dart

- Ubuntu 20.04:

```
sh 0_install_packages.sh
sh 1_install_dart.sh
sh 2_install_magnum.sh
sh 3_install_robot_dart.sh
```

- Όλα θα εγκαταστηθούν στον φάκελο ‘‘/opt/**’’

Εγκατάσταση/Documentation robot_dart

- Ubuntu 20.04:

  ```
  sh 0_install_packages.sh
  sh 1_install_dart.sh
  sh 2_install_magnum.sh
  sh 3_install_robot_dart.sh
  ```

- Όλα θα εγκαταστηθούν στον φάκελο ‘‘/opt/**’’
- Σε κάθε νέο terminal:

  ```
  source 4_variables.sh
  ```

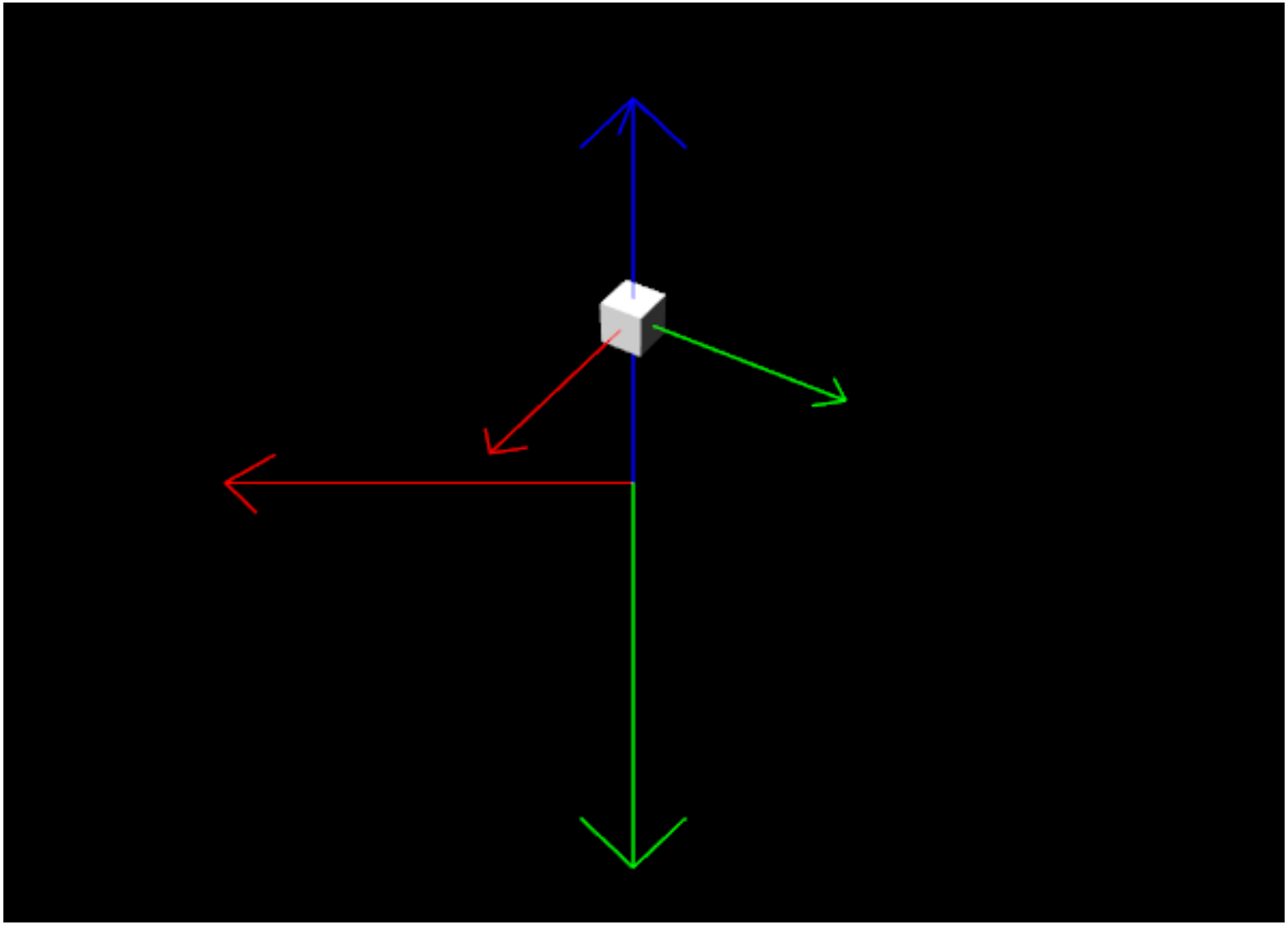
Εγκατάσταση/Documentation robot_dart

- Ubuntu 20.04:
  `sh 0_install_packages.sh`
  `sh 1_install_dart.sh`
  `sh 2_install_magnum.sh`
  `sh 3_install_robot_dart.sh`
- Όλα θα εγκαταστηθούν στον φάκελο ‘‘`/opt/**`’’
- Σε κάθε νέο terminal:
  `source 4_variables.sh`
- Εναλλακτικά προσθέτουμε το περιεχόμενου του αρχείου στο “∼/.bashrc” ή “∼/.zshrc”

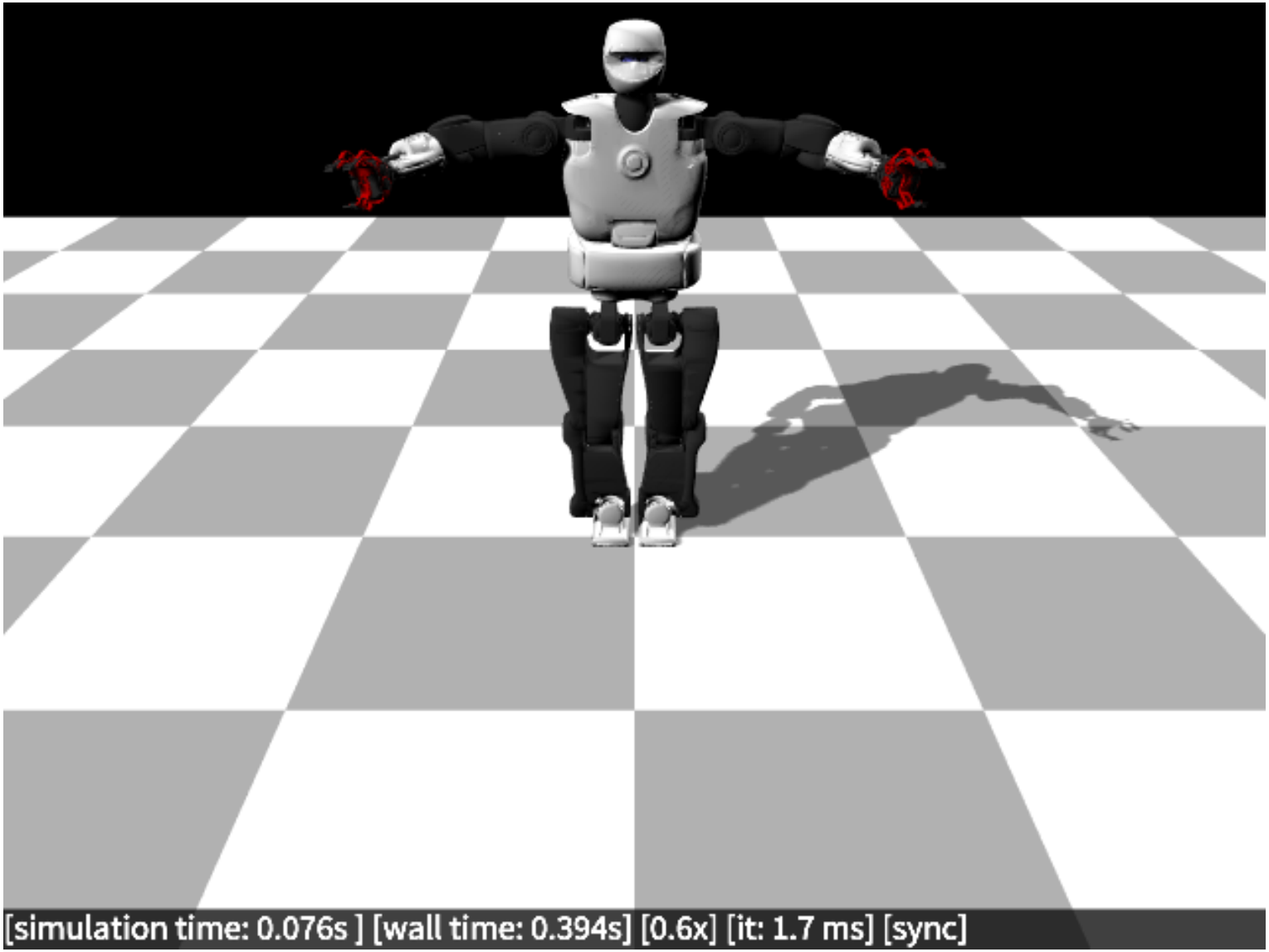
Εγκατάσταση/Documentation robot_dart

- Ubuntu 20.04:
  `sh 0_install_packages.sh`
  `sh 1_install_dart.sh`
  `sh 2_install_magnum.sh`
  `sh 3_install_robot_dart.sh`
- Όλα θα εγκαταστηθούν στον φάκελο ‘‘`/opt/**`’’
- Σε κάθε νέο terminal:
  `source 4_variables.sh`
- Εναλλακτικά προσθέτουμε το περιεχόμενου του αρχείου στο “∼/.bashrc” ή “∼/.zshrc”
- Documentation

Μητρώα Μετασχηματισμών/Συστήματα Αναφοράς



Σύνοψη Βιβλιοθήκης robot_dart



Βασική Χρήση robot_dart (1)

- Πρώτα πρέπει να δημιουργήσουμε έναν κόσμο:

python:

```
simu = rd.RobotDARTSimu(timestep)
```

cpp:

```
rd::RobotDARTSimu simu(timestep);
```

Βασική Χρήση robot_dart (1)

- Πρώτα πρέπει να δημιουργήσουμε έναν κόσμο:

python:

```
simu = rd.RobotDARTSimu(timestep)
```

cpp:

```
rd::RobotDARTSimu simu(timestep);
```

- Έπειτα να "φορτώσουμε" ένα ρομπότ (π.χ. arm.urdf) και να το προσθέσουμε στον κόσμο:

python:

```
robot = rd.Robot("arm.urdf")
simu.add_robot(robot)
```

cpp:

```
auto robot = std::make_shared<rd::Robot>("arm.urdf");
simu.add_robot(robot);
```

Βασική Χρήση robot_dart (1)

- Πρώτα πρέπει να δημιουργήσουμε έναν κόσμο:

python:

```
simu = rd.RobotDARTSimu(timestep)
```

cpp:

```
rd::RobotDARTSimu simu(timestep);
```

- Έπειτα να "φορτώσουμε" ένα ρομπότ (π.χ. arm.urdf) και να το προσθέσουμε στον κόσμο:

python:

```
robot = rd.Robot("arm.urdf")
simu.add_robot(robot)
```

cpp:

```
auto robot = std::make_shared<rd::Robot>("arm.urdf");
simu.add_robot(robot);
```

- Ας τρέξουμε την προσομοίωση:

python:

```
while True:
  if simu.step_world():
    break
```

cpp:

```
while(true) { if(simu.step_world()) break; }
```

Βασική Χρήση robot_dart (2)

- Ας προσθέσουμε γραφικά:

python:

```
graphics = rd.gui.Graphics()
simu.set_graphics(graphics)
```

cpp:

```
auto graphics = std::make_shared<rd::gui::magnum::Graphics>();

simu.set_graphics(graphics);
```

Βασική Χρήση robot_dart (2)

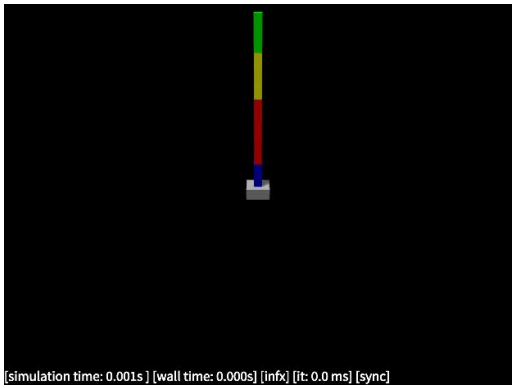
- Ας προσθέσουμε γραφικά:

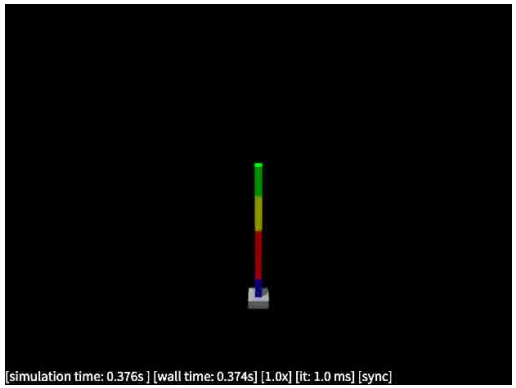
python:

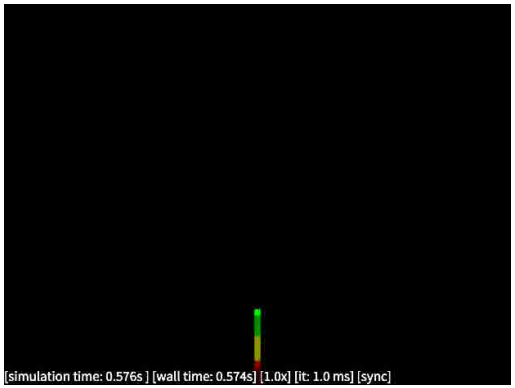
```
graphics = rd.gui.Graphics()
simu.set_graphics(graphics)
```

cpp:

```
auto graphics = std::make_shared<rd::gui::magnum::Graphics>();
simu.set_graphics(graphics);
```







Βασική Χρήση robot_dart (3)

- Πρέπει να "προσδέσουμε" το ρομπότ σε ένα σταθερό σημείο:

python:

```
robot.fix_to_world()
```

cpp:

```
robot->fix_to_world();
```

Βασική Χρήση robot_dart (3)

- Πρέπει να "προσδέσουμε" το ρομπότ σε ένα σταθερό σημείο:
  python:
  ```
  robot.fix_to_world()
  ```
  cpp:
  ```
  robot->fix_to_world();
  ```
- Ας δώσουμε μία αρχική θέση στις αρθρώσεις του ρομπότ:
  python:
  ```
  robot.set_positions([0.2, 0.2, 0.2, 0.2])
  ```
  cpp:
  ```
  robot->set_positions(rd::make_vector({0.2, 0.2, 0.2, 0.2}));
  ```

Βασική Χρήση robot_dart (3)

- Πρέπει να "προσδέσουμε" το ρομπότ σε ένα σταθερό σημείο:

python:

```
robot.fix_to_world()
```

cpp:

```
robot->fix_to_world();
```

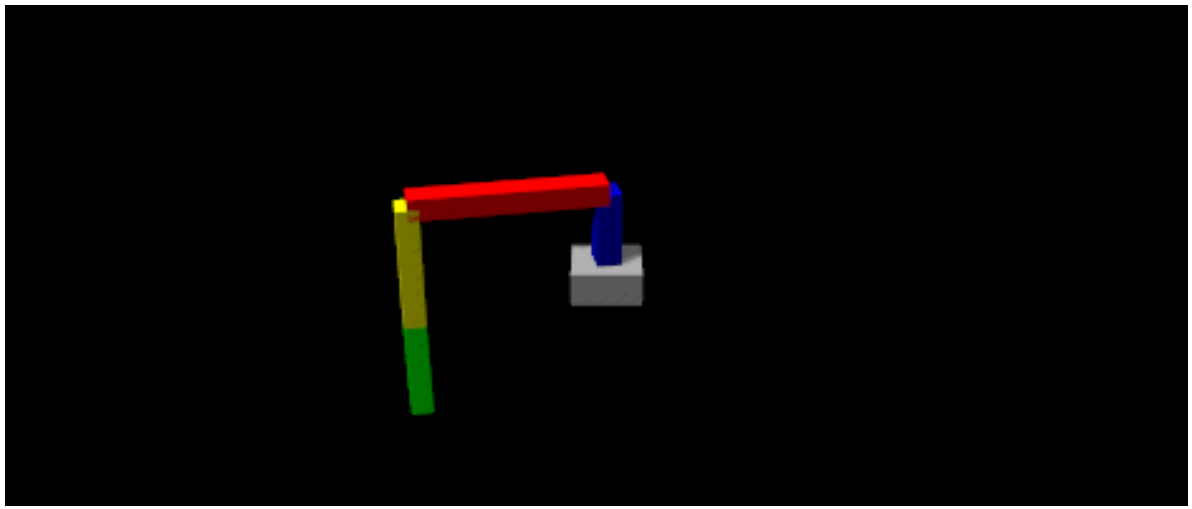
- Ας δώσουμε μία αρχική θέση στις αρθρώσεις του ρομπότ:

python:

```
robot.set_positions([0.2, 0.2, 0.2, 0.2])
```

cpp:

```
robot->set_positions(rd::make_vector({0.2, 0.2, 0.2, 0.2}));
```



Βασική Χρήση robot_dart (4)

- Ας αλλάξουμε "κινητήρες":

python:

```
robot.set_actuator_types("servo")
```

cpp:

```
robot->set_actuator_types("servo");
```

Βασική Χρήση robot_dart (4)

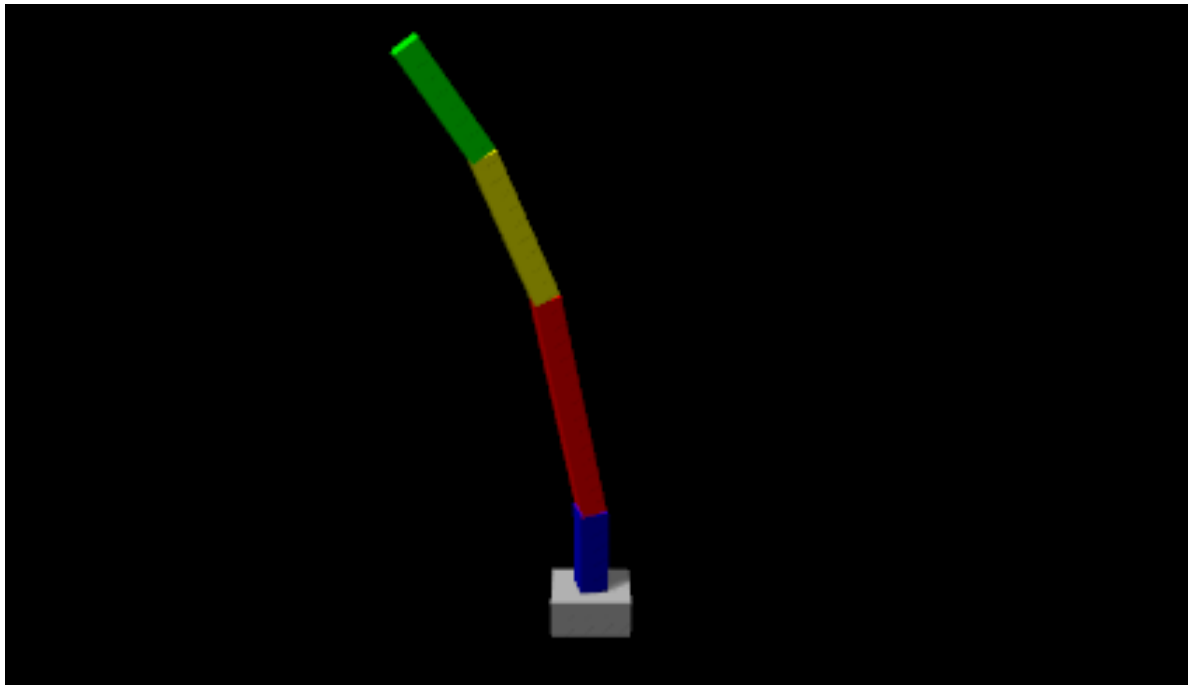
- Ας αλλάξουμε "κινητήρες":

python:

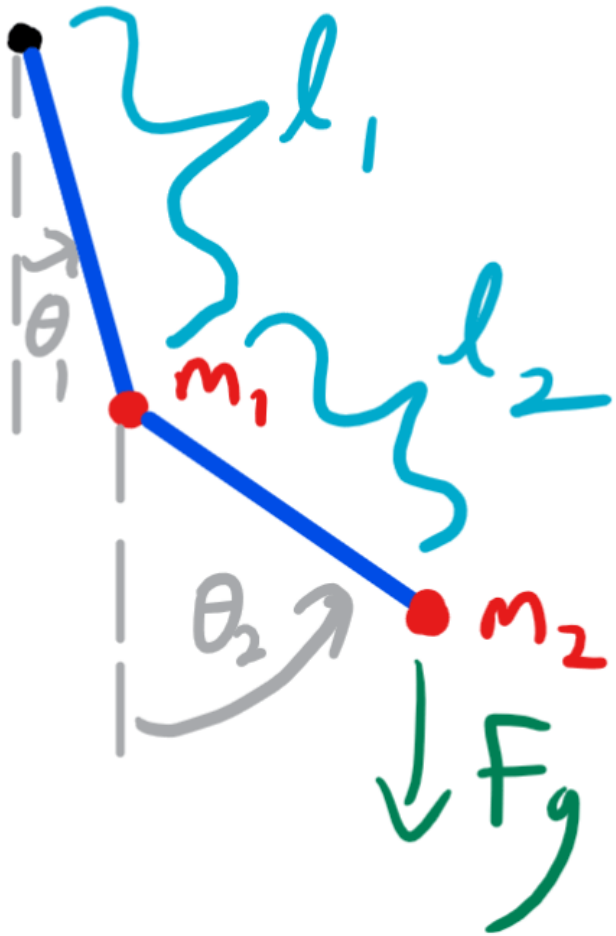
```
robot.set_actuator_types("servo")
```

cpp:

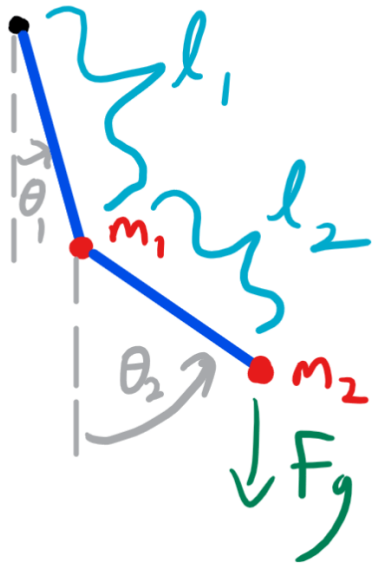
```
robot->set_actuator_types("servo");
```



Διπλό Εκρεμμές (Double Pendulum)

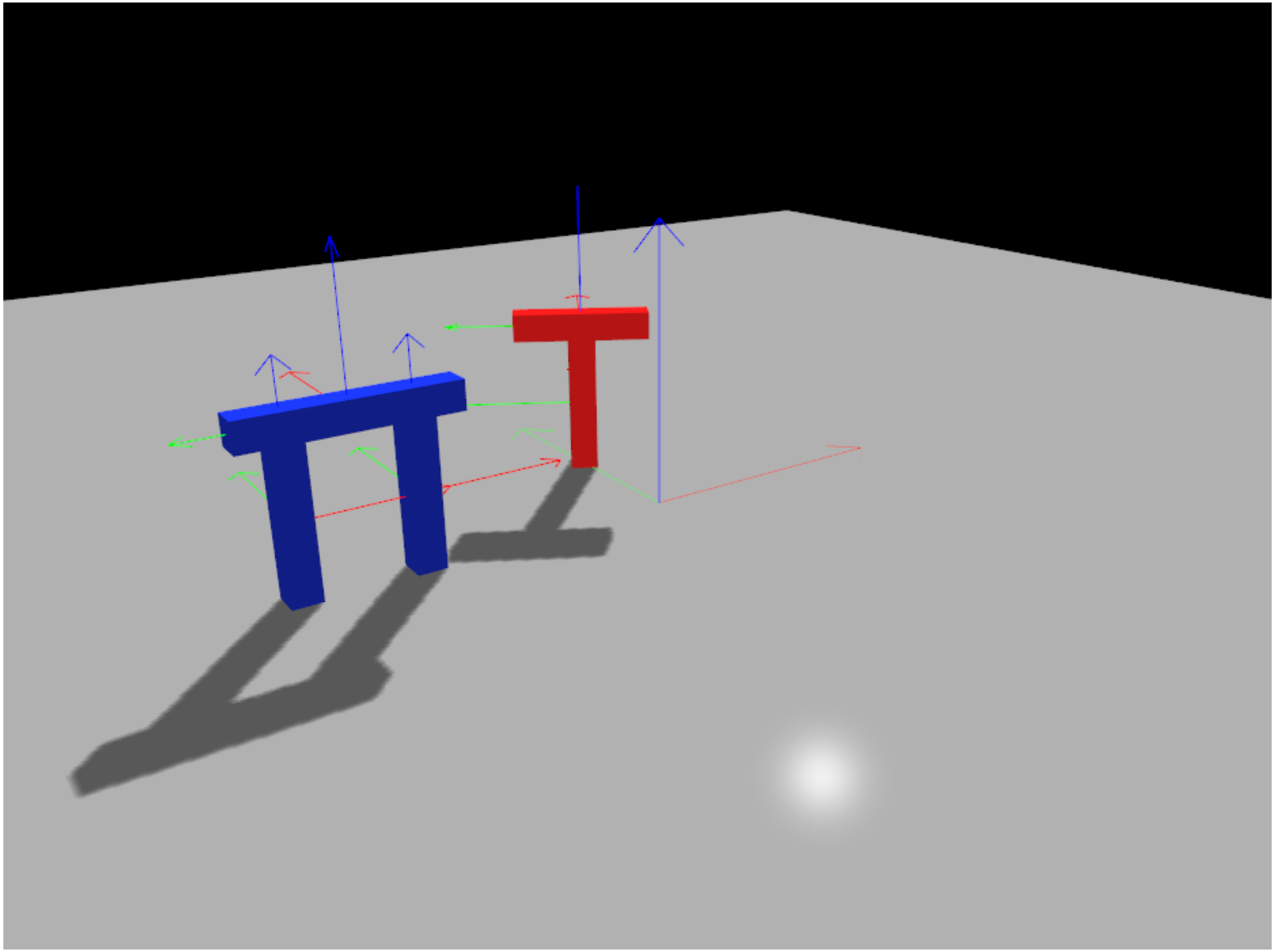


Διπλό Εκρεμμές - Forward Kinematics

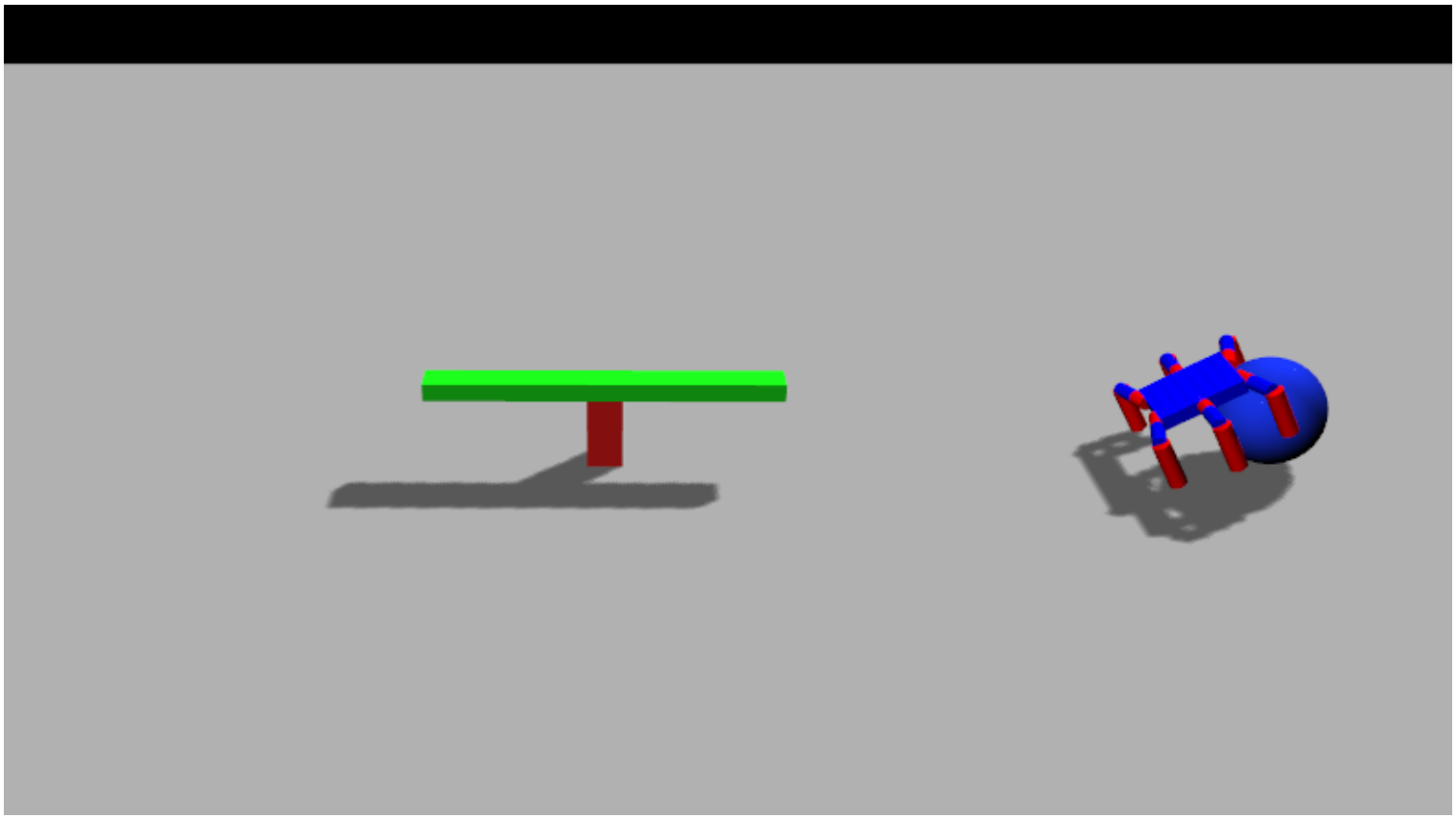


$$x_1 = l_1 \sin\theta_1$$
$$z_1 = -l_1 \cos\theta_1$$
$$x_2 = x_1 + l_2 \sin\theta_2$$
$$z_2 = z_1 - l_2 \cos\theta_2$$

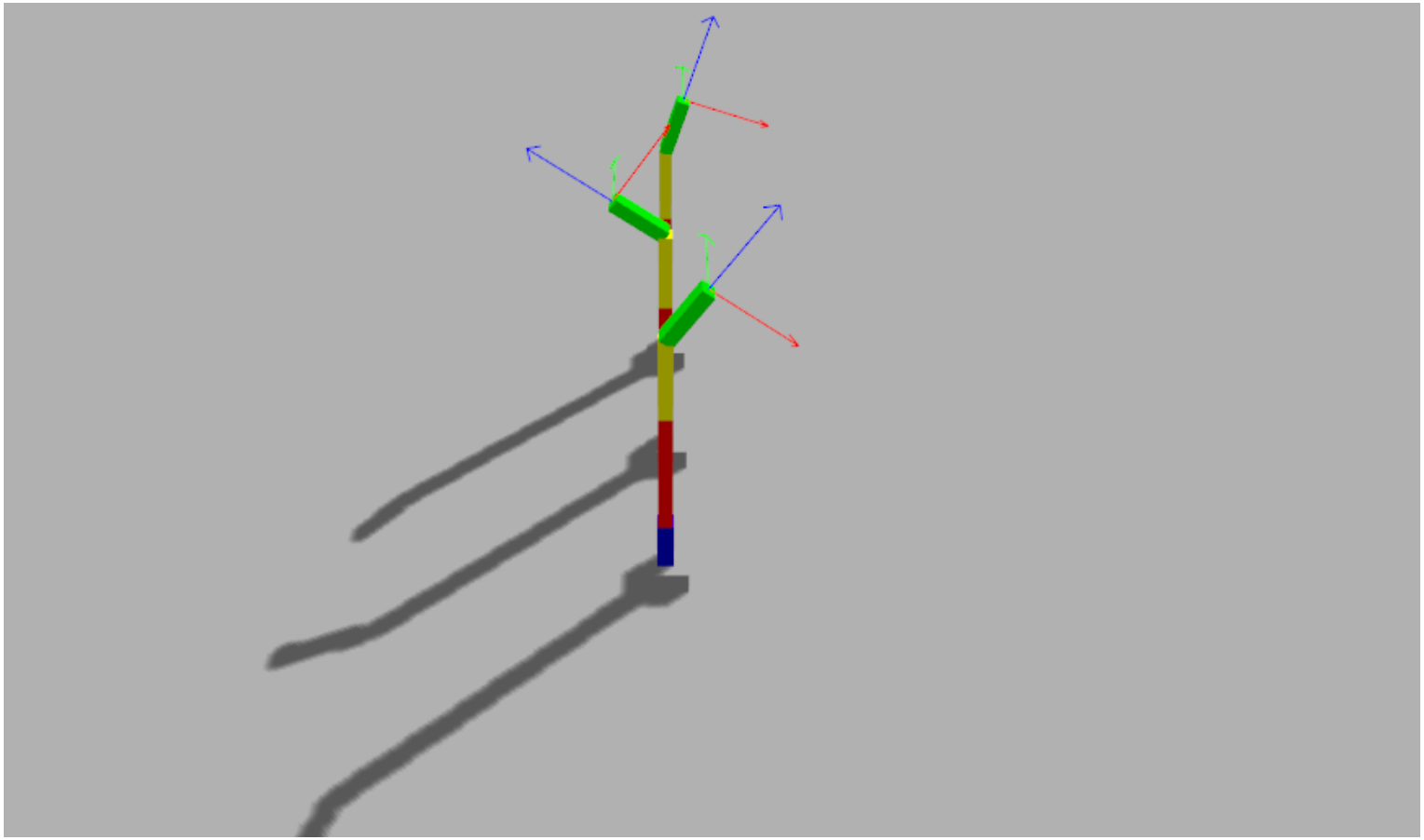
Παραδείγματα robot_dart (1)



Παραδείγματα robot_dart (2)



Παραδείγματα robot_dart (3)



Παραδείγματα robot_dart (4)