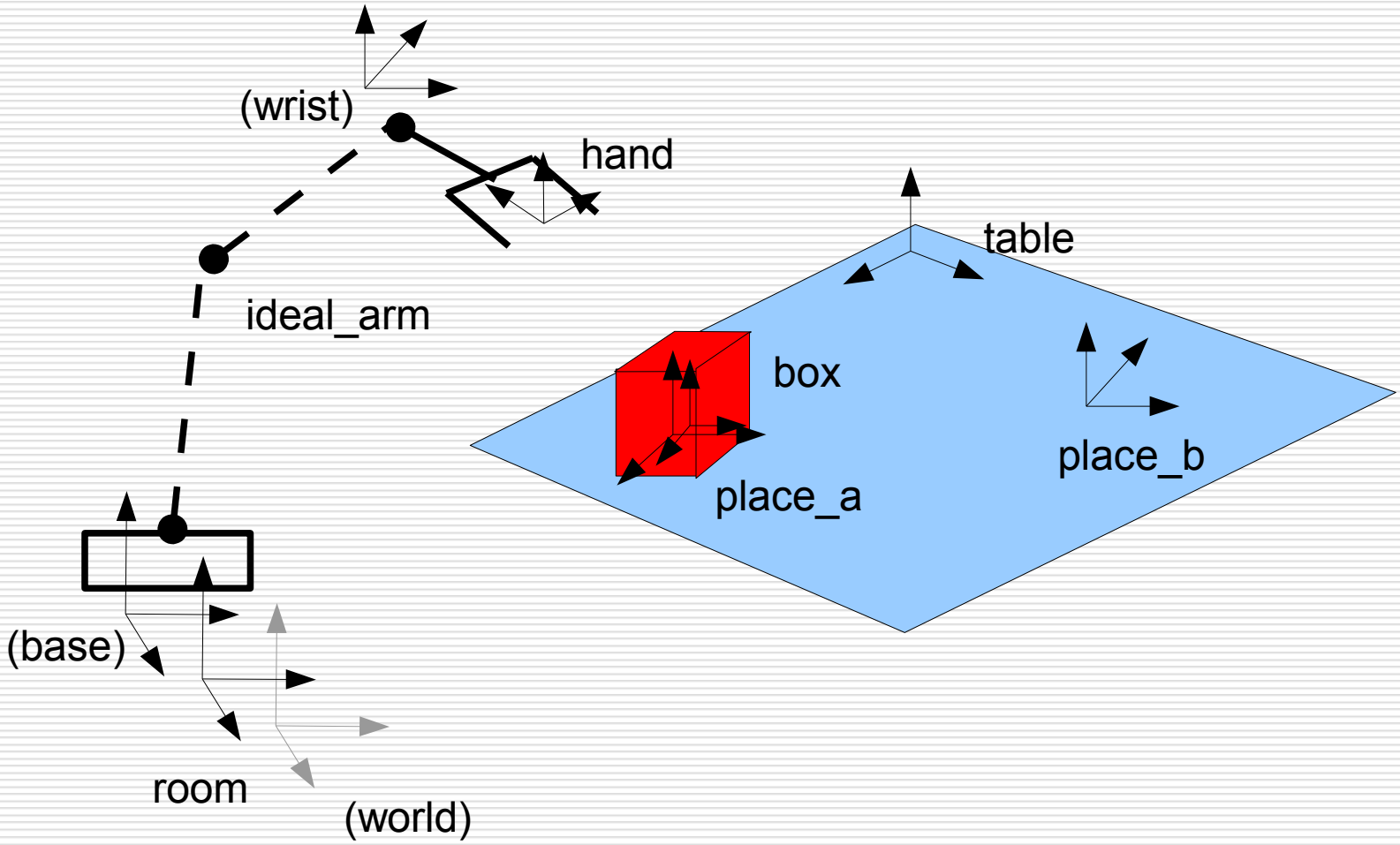
- 環境構築:例題8-1

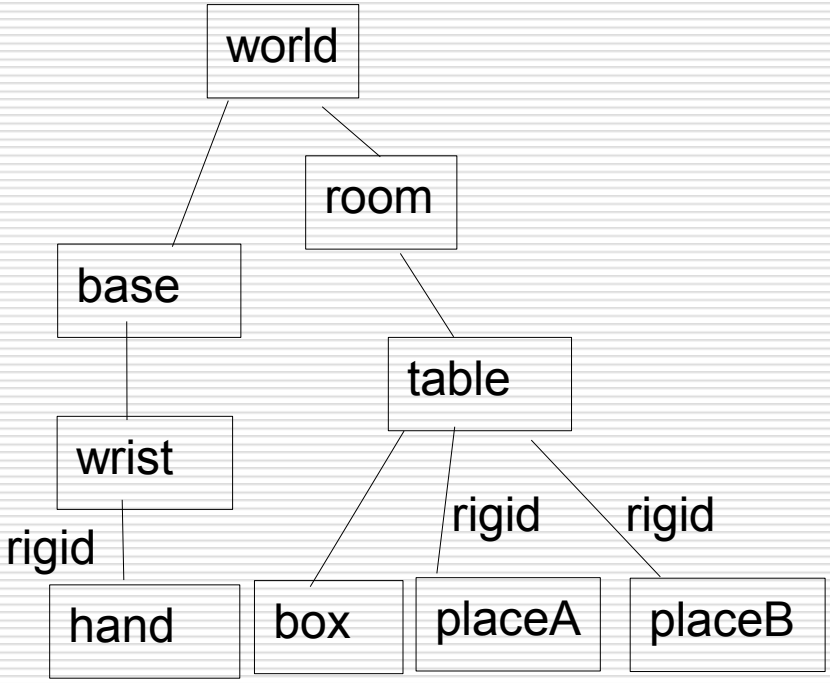
以下のロボット作業環境を生成せよ

- room: ワールドと一致した座標系
- table: room内、高さ0.5、z回りに30度
- place_a: tabel固定、x方向に0.1
- place_b: table固定、x方向に-0.1、z回りに -30度
- box: table上、place_aに一致
- hand: Hand()のインスタンス
- ideal_arm: ArmWithHand, handを持つ

- 例題8-1で構築される環境



- 例題8-1で構築される座標系連鎖



- 環境構築：例題8-1の解

env_model_coord.py

```
from object_model import *
room = CoordinateObject()
table = CoordinateObject()
table.affix(room,FRAME(xyzabc=[0,0,0.5,0,0,pi/6]))
place_a = CoordinateObject()
place_a.affix(table,FRAME(xyzabc=[0.1,0,0,0,0,0]), 'rigid')
place_b = CoordinateObject()
place_b.affix(table,FRAME(xyzabc=[-0.1,0,0,0,0,-pi/6]), 'rigid)
box = CoordinateObject()
box.affix(table,place_a.where(table))
hand = Hand()
ideal_arm = ArmWithHand(None, hand)
```

- 座標系連鎖の計算:例題8-2

- place_a、place_bのワールド座標系からの座標変換を求めよ。それを位置x、y、z、α、β、γで表示せよ。
- place_bからみたplace_aへの座標変換(place_a wrt place_b)を求めよ。

- tableをz回りに90度回転させよ。
- place_a、place_bのワールド座標系からの座標変換を求めよ。それを位置x、y、z、α、β、γで表示せよ。
- place_bからみたplace_aへの座標変換を求めよ。

- 例題8-2の解(1)

```
>>> place_a.where().xyzabc()
[0.08660254037844387, 0.04999999999999996, 0.5, 0.0, 0.0,
0.5235987755982988]
>>> place_b.where().xyzabc()
[-0.08660254037844387, -0.04999999999999996, 0.5, 0.0,
0.0, 0.0]
>>> place_a.where(place_b).xyzabc()
[0.17320508075688776, 0.09999999999999992, 0.0, 0.0, 0.0,
0.5235987755982988]
```

- 例題8-2の解(2)

```
>>> table.where().xyzabc()
[0.0, 0.0, 0.5, 0.0, 0.0, 0.5235987755982988]
>>> table.rel
>>> table.rel_trans
f:(m:[[0.8660254037844387, -0.4999999999999994, 0.0],
[0.4999999999999994, 0.8660254037844387, 0.0], [0.0, 0.0, 1.0]],v:[0.0, 0.0,
0.5])
>>> R=MATRIX(c=pi/2)
>>> table.set_trans(FRAME(mat=table.rel_trans.mat*R,vec=table.rel_trans.vec))
>>> table.where().xyzabc()
[0.0, 0.0, 0.5, 0.0, 0.0, 2.094395102393195]
```

- 例題8-2の解(3)

```
>>> place_a.where().xyzabc()
[-0.04999999999999989, 0.08660254037844387, 0.5, 0.0, 0.0,
2.094395102393195]
>>> place_b.where().xyzabc()
[0.04999999999999989, -0.08660254037844387, 0.5, 0.0, 0.0,
1.5707963267948966]
>>> place_a.where(place_b).xyzabc()
[0.17320508075688776, 0.09999999999999992, 0.0, 0.0, 0.0,
0.5235987755982988]
>>>
```

- 作業例(座標管理なし)1:例題8-3

以下のロボット作業を座標系管理(affix, unfix)を使わずに実行せよ。

- (まず上0.2の座標変換up200を生成する)
- handをboxの上0.2に移動する
- handを0.1開く
- handをboxに移動する
- handを0に閉じる

- 例題8-3の解

```
>>> hand.where()
f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],v:[0.0, 0.0, 0.0])
>>> up200=FRAME(vec=VECTOR(0,0,0.2))
>>> ideal_arm.move(hand,up200,box)
>>> box.where()
f:(m:[[0.866025403784438 71, -0.4999999999999994, 0.0],
[0.4999999999999994, 0.866025403784438 71, 0.0], [0.0, 0.0, 1.0]],
v:[0.0866025403784438 79, 0.04999999999999996, 0.5])
>>> hand.where()
f:(m:[[0.866025403784438 71, -0.4999999999999994, 0.0],
[0.4999999999999994, 0.866025403784438 71, 0.0], [0.0, 0.0, 1.0]],
v:[0.0866025403784438 79, 0.04999999999999996, 0.6999999999999996])
>>> hand.open(0.1)
open
>>> ideal_arm.move(hand,box)
>>> hand.close(0)
close
```

- 作業例(座標管理なし)2:例題8-4

以下のロボット作業の続きを実行せよ

- boxをplace_bに持っていく。
- handを0.1開く
- handをplace_bの上0.2に移動する

- 例題8-4の解

```
>>> ideal_arm.move(box,place_b)
error: cannot move unfixed object
False
>>> ideal_arm.move(hand, place_b)
>>> ideal_arm.hand.open(0.1)
open
>>> ideal_arm.move(hand,up200,place_b)
>>> place_b
<object_model.CoordinateObject instance at 0x0155B1E8>
>>> place_b.where()
f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],
v:[-0.0866025403784438 79, -0.04999999999999996, 0.5])
>>> hand.where()
f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],
v:[-0.0866025403784438 79, -0.04999999999999996, 0.6999999999999996])
>>> box.where()
f:(m:[[0.866025403784438 71, -0.4999999999999994, 0.0],
[0.4999999999999994, 0.866025403784438 71, 0.0], [0.0, 0.0, 1.0]],
v:[0.0866025403784438 79, 0.04999999999999996, 0.5])
```

- 作業例(座標管理あり):例題8-5

以下のロボット作業をaffix, unfixを用いて座標系を管理しながら実行せよ。

- handをboxの上0.2に移動する
- handを0.1開く
- handをboxに移動する
- handを0に閉じる
- boxをplace_bに持っていく。
- handを0.1開く
- handをplace_bの上0.2に移動する

- 例題8-5の解

```
>>> ideal_arm.move(hand,up200,box)
>>> ideal_arm.hand.open(0.1)
open
>>> ideal_arm.move(hand,box)
>>> ideal_arm.hand.close(0)
close
>>> box.unfix()
>>> box.affix(ideal_arm.hand)
>>> ideal_arm.move(box,place_b)
>>> ideal_arm.hand.open(0.1)
open
>>> box.unfix()
>>> box.affix(table)
>>> place_b.where()
f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],
v:[-0.0866025403784438 79, -0.04999999999999996, 0.5])
>>> box.where()
f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],
v:[-0.0866025403784438 79, -0.04999999999999996, 0.5])
>>>
```