1. Parameters:

num\_epochs = 100

num\_episodes = 20 # number of episodes over which success rate is computed

episode\_length = 500

optimization\_steps = 40

actor\_learning\_rate=1e-3

critic\_learning\_rate=1e-3

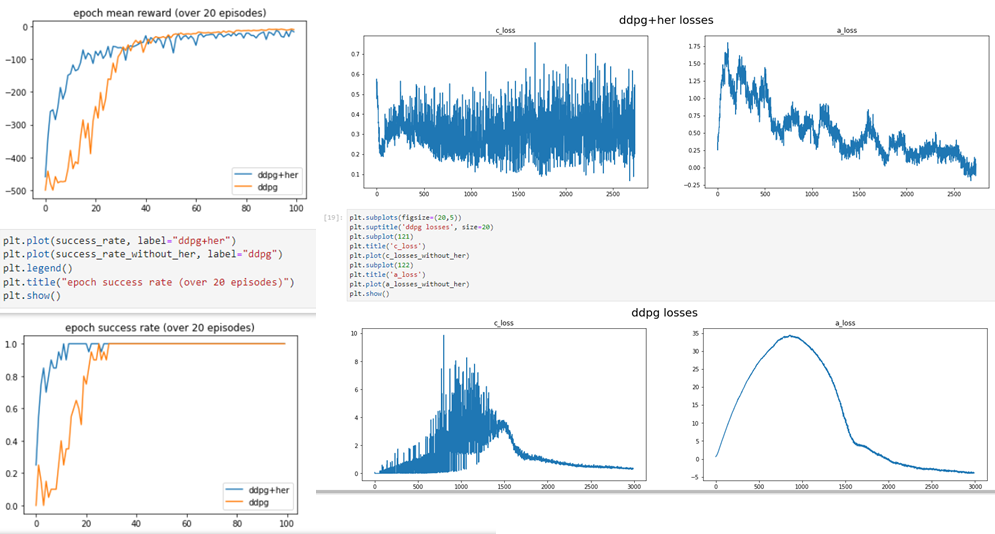
tau=0.1

Reward: -1 each step, +10 if touched detector

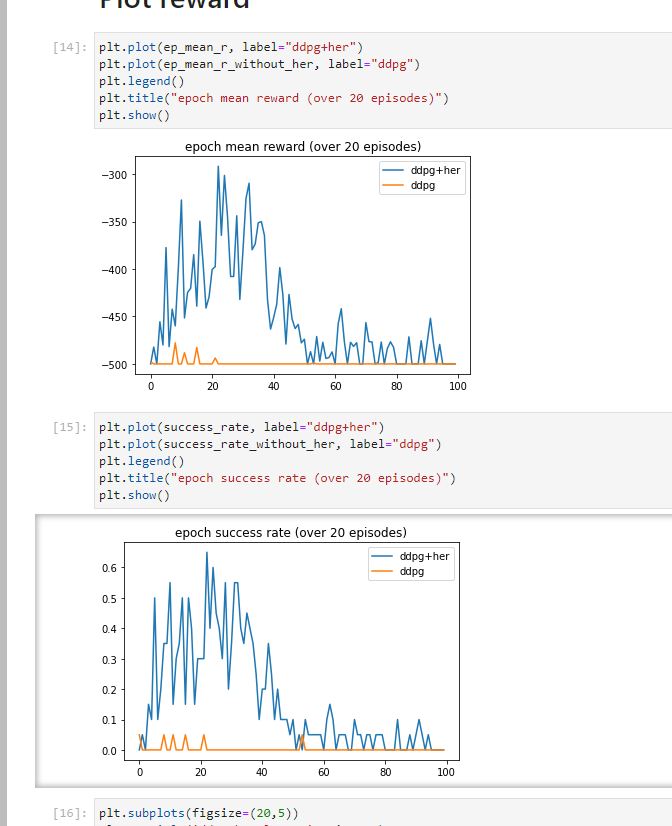
Saved models

HER: model/reward\_-1\_each\_step.ckpt

Without HER: model/reward\_-1\_each\_step\_without\_her.ckpt



1. Angles -120..120

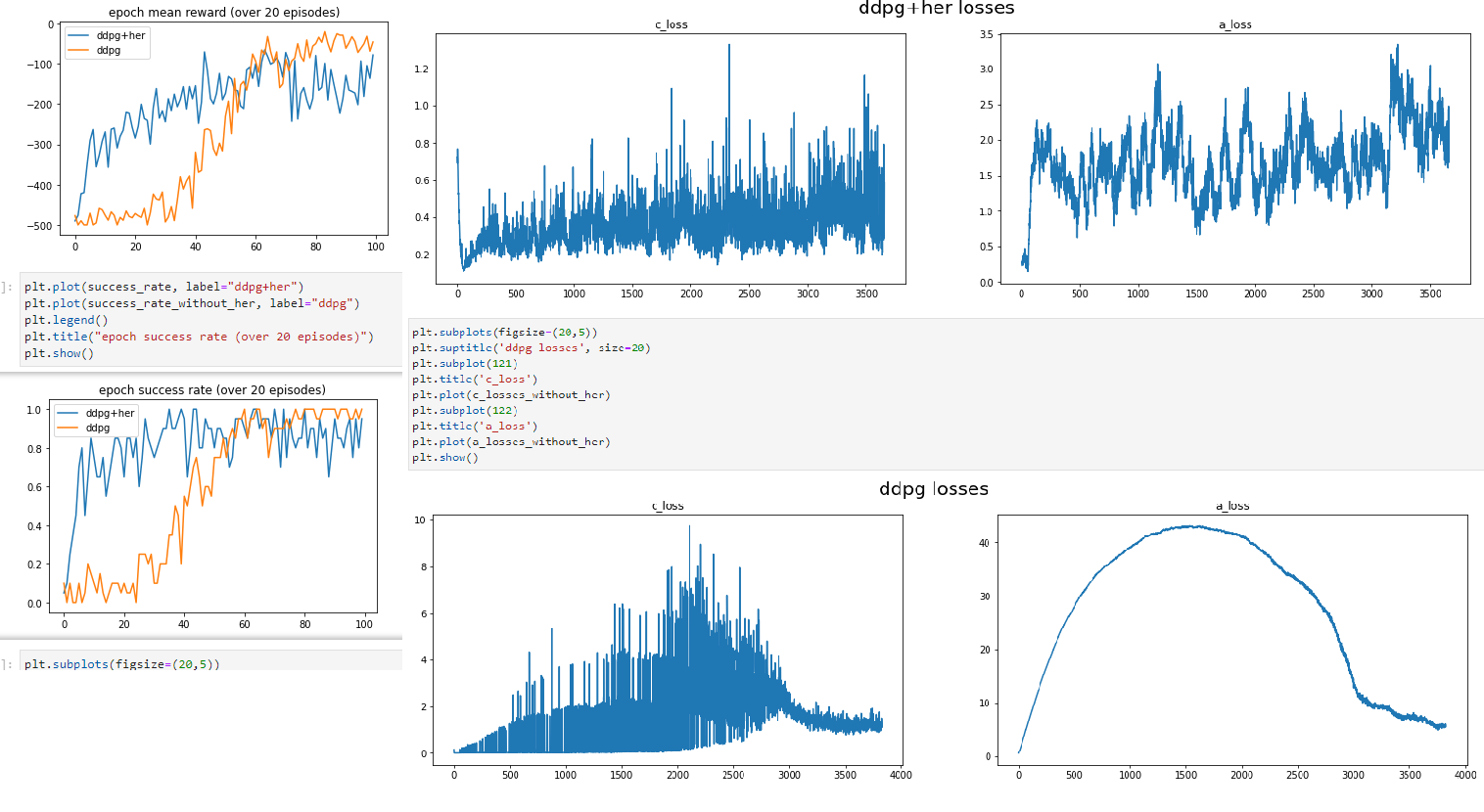


1. 3 actions – 2 angles for first hand, 1 – for second hand

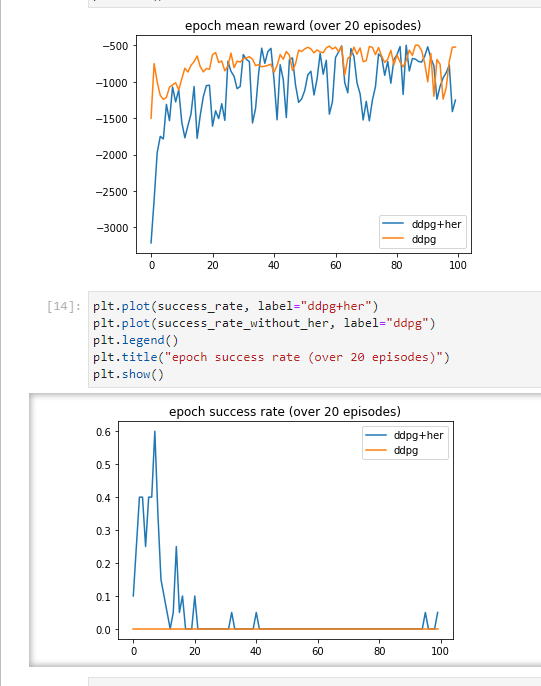
Models:

model/3actions\_not\_limited\_her.ckpt

model/3actions\_not\_limited\_without\_her.ckpt



1. -10 reward if out of angle range, 3 actions

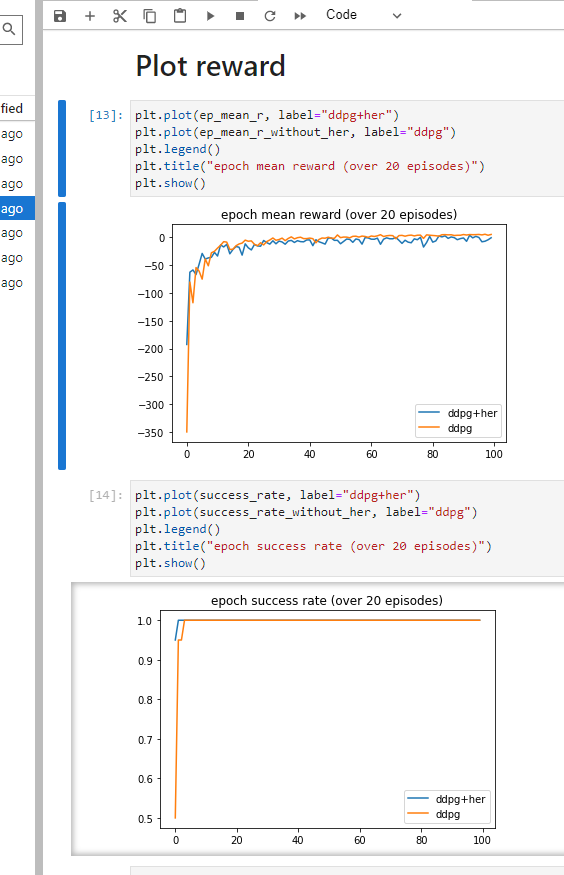


1. Robohand with 3 arms, each arm have 1 angle

action\_high = 1, action\_low = -1

model/old\_hand\_hands3\_step3\_withoun\_her.ckpt

model/old\_hand\_hands3\_step3\_her.ckpt



1. Роборука с тремя сочленениями, каждая имеет 1 угол. Повороты с помощью RotateSimple.

action\_high = 1, action\_low = -1

model/angles\_info\_her.ckpt

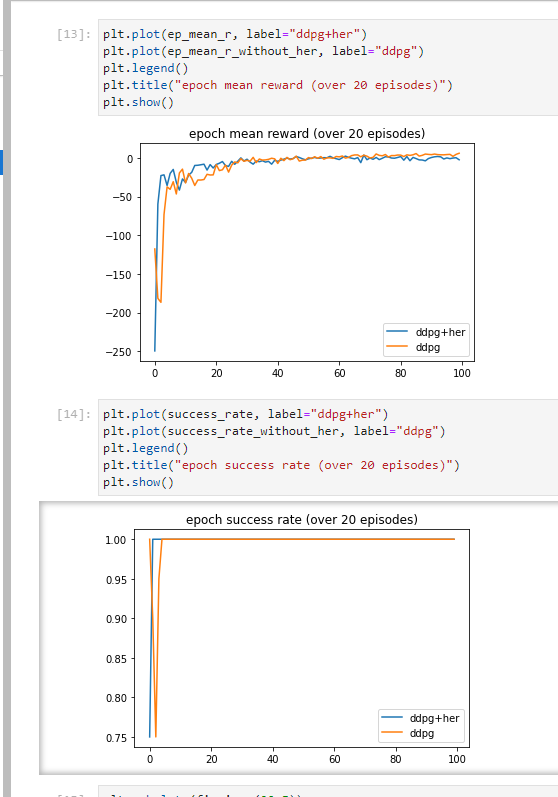
model/angles\_info\_without\_her.ckpt



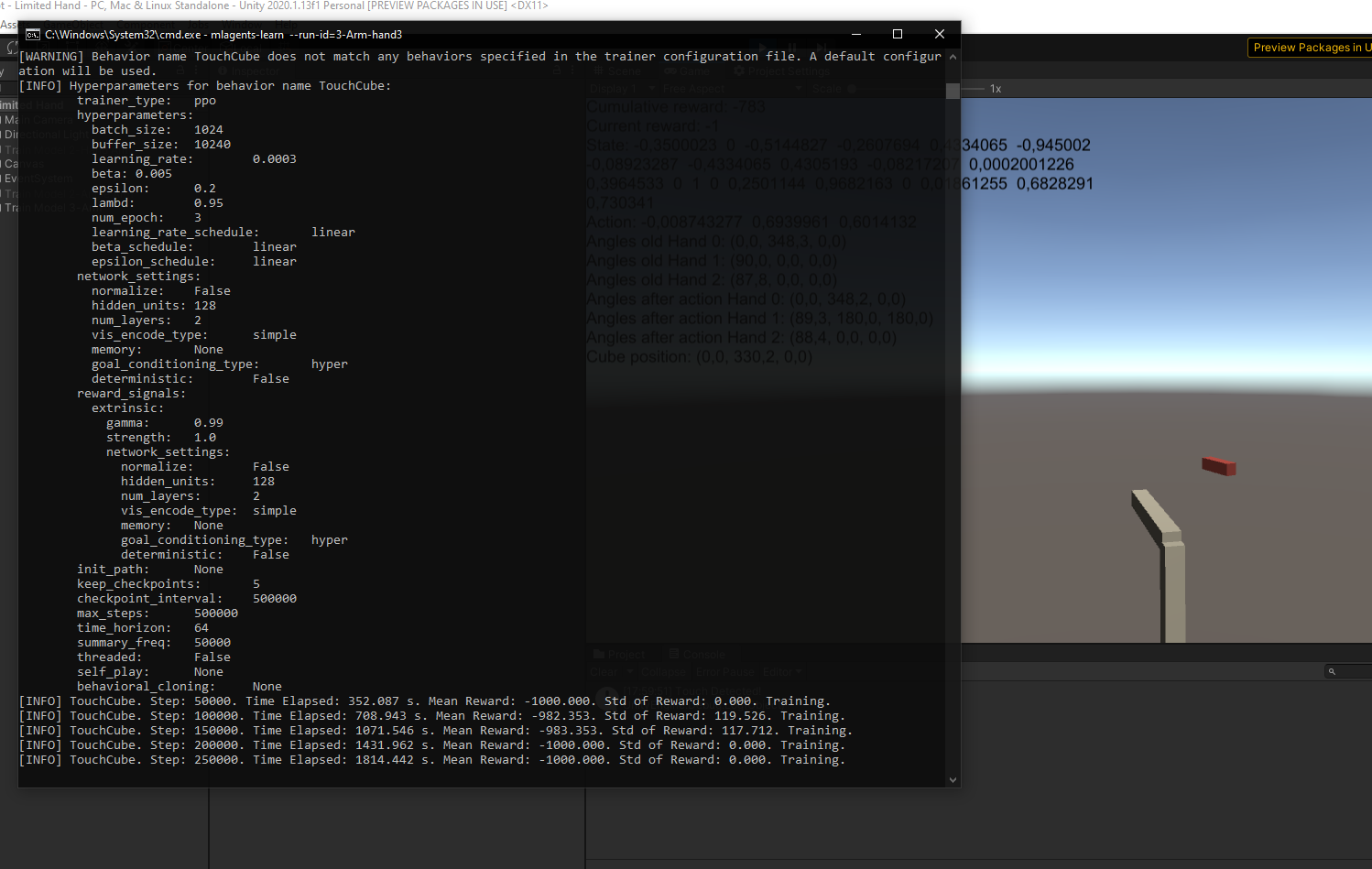
action\_high = 3, action\_low = -3



action\_high = 5, action\_low = -5, RotateAround



Обучение в юнити с RotateSimple и state из 21 элемента (state size = 3 (cubePosition) + 3 (cubePosition - endPosition) + 3 (endPosition) + 3 (jointPoint.position) + 3 \* 3 (robotPart.Rotation for 3 joints) = 21)

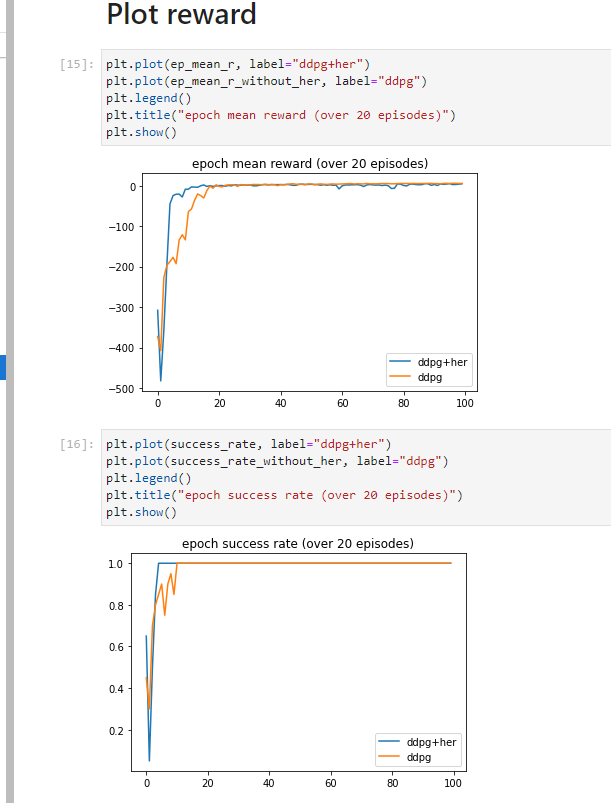
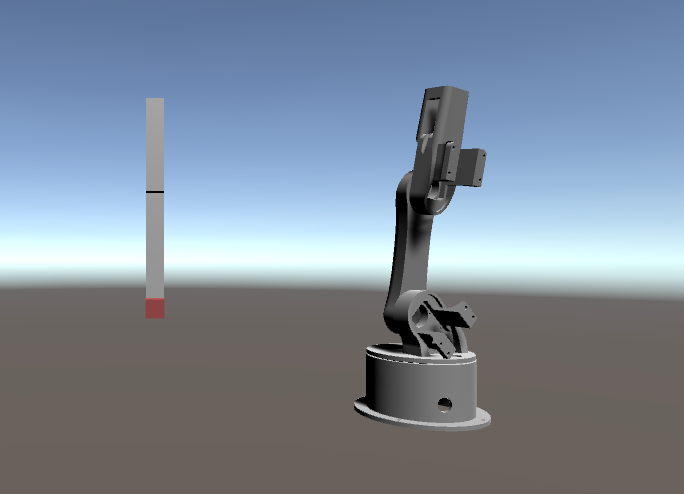
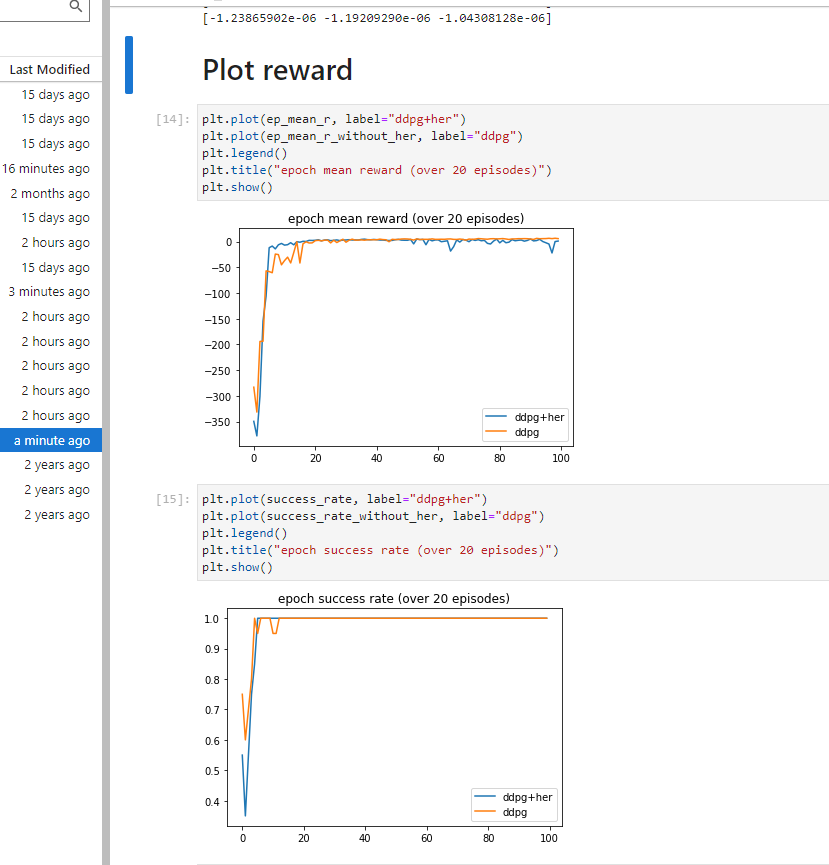


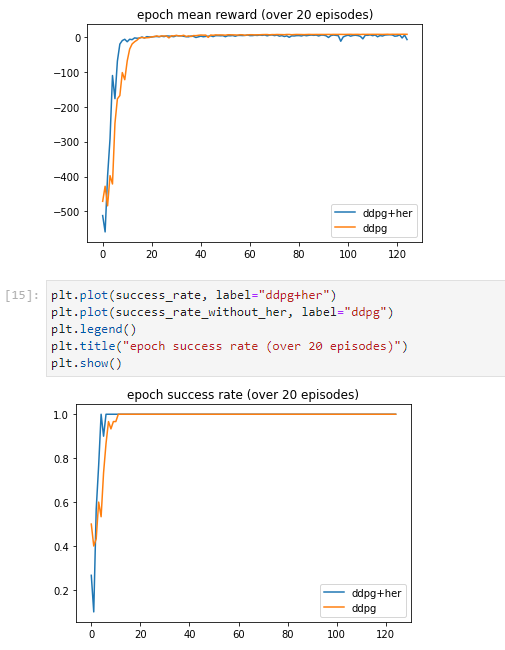
1. State size = 21, RotateSimple  
     
   state size = 3 (cubePosition) + 3 (cubePosition - endPosition) + 3 (endPosition) +  
    3 (jointPoint.position) + 3 \* 3 (robotPart.Rotation for 3 joints) = 21  
     
   model/her\_fixed.ckpt  
   model/without\_her\_fixed.ckpt  
   

1. state size = 2 (cubePosition) + 3 (cubePosition - endPosition) + 3 (endPosition) +

3 (robotPart.RotationFloat for 3 joints) = 11  
Изображение выглядит как текст

Автоматически созданное описание  


1. Заменены функции RotateAround и RotateSimple (transform.localRotation = angle) на Rotate.  
   model/her\_rotate.ckpt  
   model/without\_her\_rotate.ckpt  
     
   state size = 12: 3 (cubePosition) + 3 (cubePosition - endPosition) + 3 (endPosition) +  
    3 (robotPart. RotationFloat for 3 joints)  
     
   
2.   
   model/her\_rotate\_scaled.ckpt  
   model/without\_her\_rotate\_scaled.ckpt  
     
   
3. Target is circle  
   model/her\_sphere\_target.ckpt  
   model/without\_her\_sphere\_target.ckpt

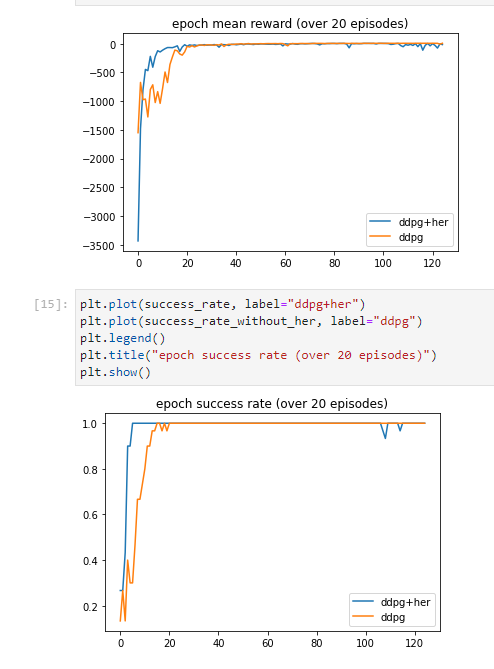
  
Проблема: иногда тянется к цели задом, тк нет ограничения в виде коллайдера стола и это быстрее для достижения цели. Проблема исправлена в пункте 12.

Изображение выглядит как текст

Автоматически созданное описание

1. Модель, учитывающая стол. Дает -5 награды за каждый шаг, в который какая-либо часть руки касается стола (стол чуть ниже самой нижней точки руки). Обучать надо дольше, поэтому параметры следующие:  
   num\_epochs = 125

num\_episodes = 30

episode\_length = 600   
  
model/without\_her\_rigid\_table.ckpt  
model/ her\_rigid\_table.ckpt  
  


Переобучить на разные Y у цели. Переобучить для другого размера сочленений. Переобучить на [-10:10] или [-20, 20]. Обучить для настоящей модели робота с увеличенным масштабом.

От -20 до 20 плохо (model/without\_her\_range20.ckpt), тк сервы не двигаются, если угол выходит за границы.

Обучить для движущейся цели.

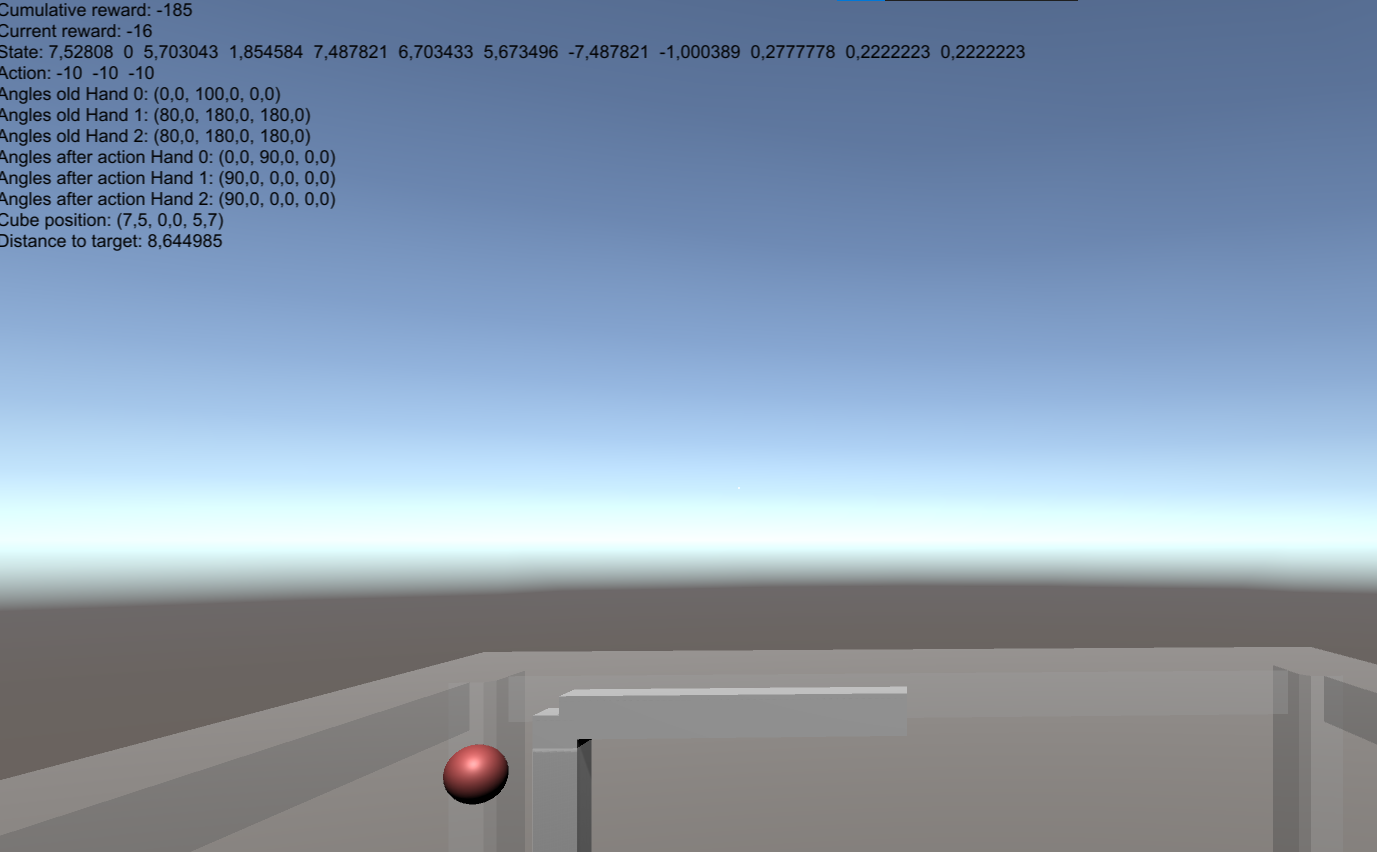
Попробовать увеличить размер руки в тестовом билде для проверки алгоритма.  
  
Учитывается стол + уменьшено второе сочленение: model/without\_her\_rigidtable\_scaled.ckpt

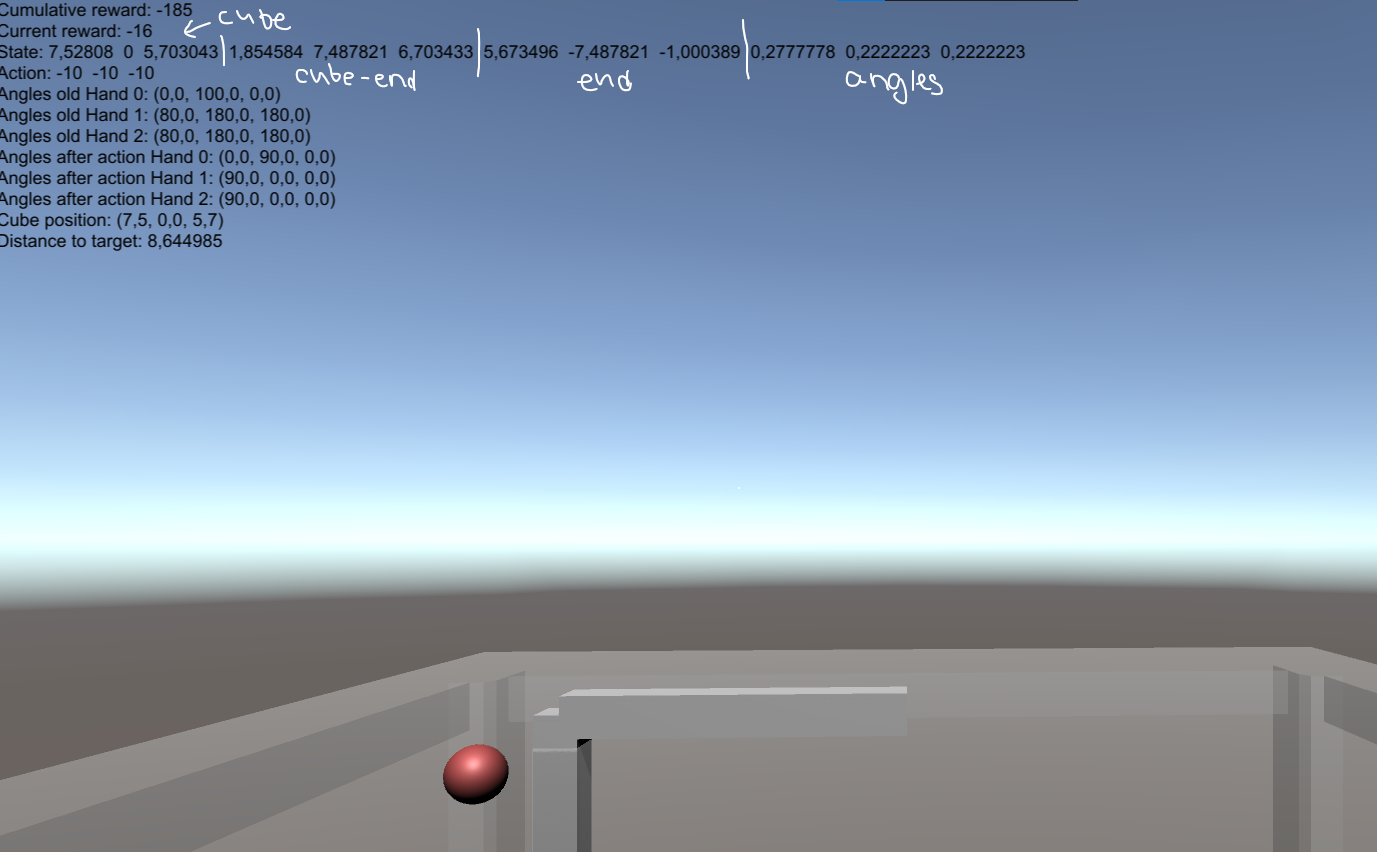
Проверить вид нормализованного угла в Unity – совпадает ли с angle/360? – да

Нужно нормализовать к 1!!  
Изображение выглядит как текст

Автоматически созданное описание

State для руки, model/without\_her\_rigidtable\_scaled.ckpt, полученной умножением масштаба на 20





Сделать калибровку системы координат: камера видит одну систему, надо перевести её (смещением и поворотом) в локальную систему координат для роборуки (камера будет считывать какой-нибудь qr код на руке, чтобы понять её положение и поворот) Добавить на роборуку метку (стрелочку) для определения её позиции и поворота

1. Выставить руку в (0,0,0) – значения углов.
2. Распознать расположение и ориентацию осей роборуки по метке (стрелочка).
3. Распознать положение кубика.
4. Перевести положение кубика из системы координат камеры в систему координат роборуки (смещение и поворот).
5. Подавать на вход нейросети новые координаты кубика.