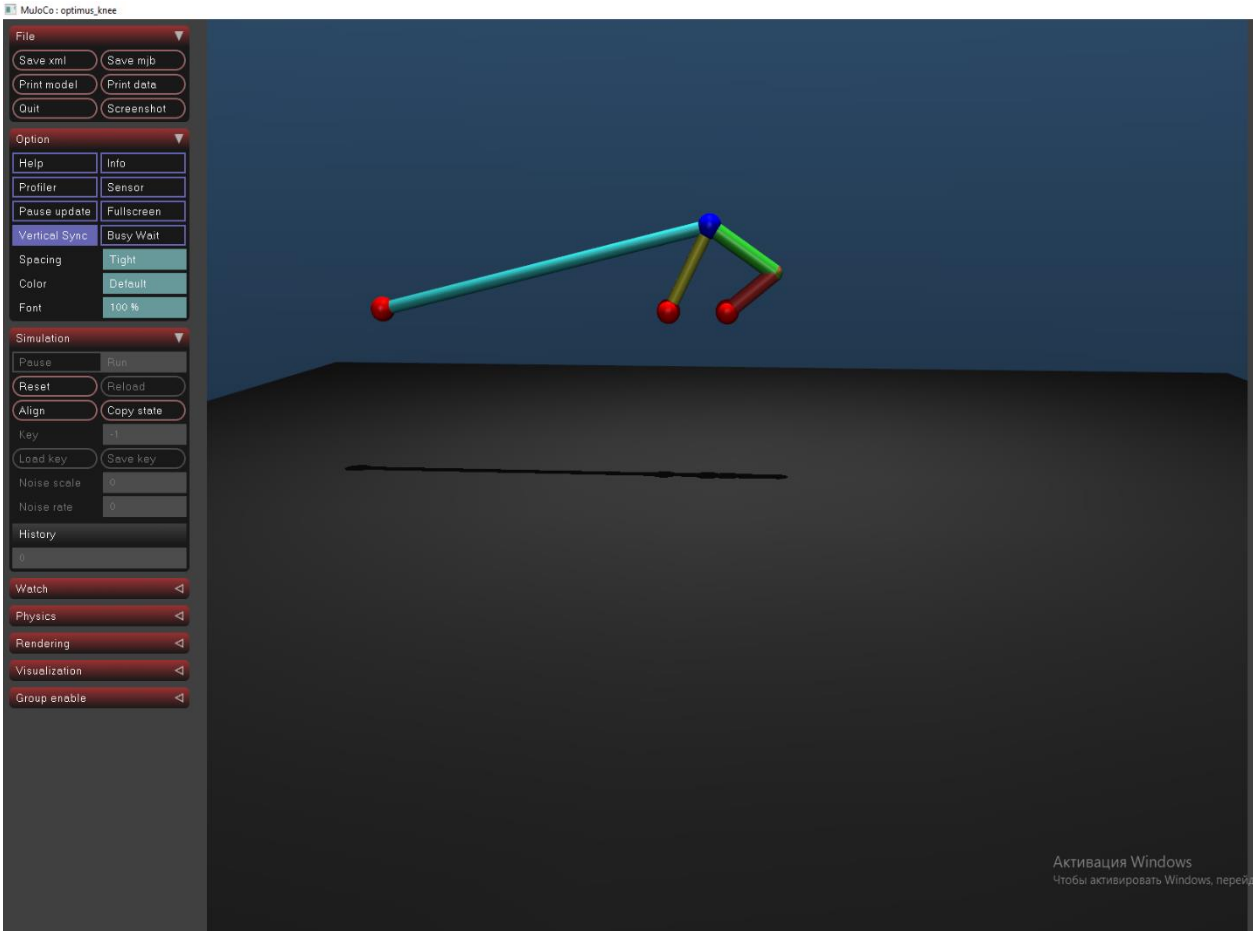
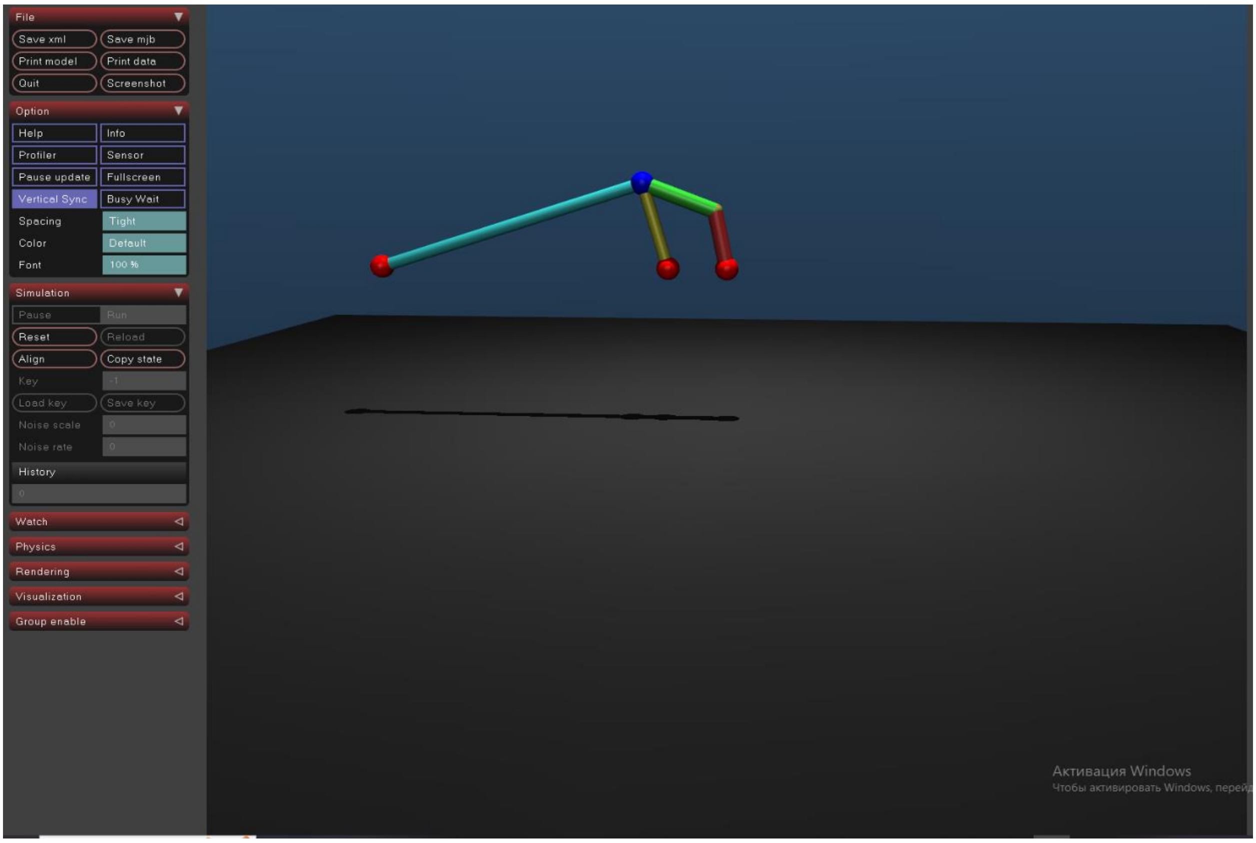


MuJoCo : optimus_knee

Spacing Tight
Color Default
Font 100 %
Key -1
Noise scale 0
Noise rate 0
History 0



Spacing Tight
Color Default
Font 100 %
Key -1
Noise scale 0
Noise rate 0
History 0

```
import mujoco
import mujoco.viewer
import numpy as np
import time
def create_optimus_knee_xml(l1, l2, l3, l4, l5, mass_per_meter=1.0):
    m1 = l1 * mass_per_meter
    m2 = l2 * mass_per_meter

    pos_o = "0 0 0.2"
    pos_c = f"{l4} 0 0.2"
    pos_d = f"{l4 + l5} 0 0.2"

    stiffness_L3 = 10000
    damping_L3 = 100
    kp_actuator = 1000

    xml = f"""
<mujoco model="optimus_knee">
  <compiler angle="radian" inertiafromgeom="true"/>



  <option integrator="RK4" timestep="0.0002" />

  <visual>
      <map znear="0.01" zfar="50"/>
      <rgba haze="0.15 0.25 0.35 1"/>
  </visual>

  <asset>
    <texture type="skybox" builtin="gradient" rgb1="0.3 0.5 0.7" rgb2="0 0 0" width="320" height="320"/>
  </asset>

  <worldbody>
  <light pos="0 0 2" dir="0 0 -1" diffuse="1 1 1"/>
  <geom name="ground" type="plane" size="1 1 0.1" rgba="0.2 0.2 0.2 1"/>
  <site name="site_O" pos="{pos_o}" size="0.015" rgba="1 0 0 1"/>
  <site name="site_C" pos="{pos_c}" size="0.015" rgba="1 0 0 1"/>
  <site name="site_D" pos="{pos_d}" size="0.015" rgba="1 0 0 1"/>
  <body name="L1_body" pos="{pos_o}">
    <inertial pos="0 0 {l1/2}" mass="{m1}" diaginertia="0.01 0.01 0.001" />
    <joint name="J_O" type="hinge" axis="0 1 0" limited="true" range="-2.5 2.5"/>
    <geom name="G_L1" type="capsule" fromto="0 0 0 0 0 {l1}" size="0.01" rgba="0.8 0.2 0.2 1"/>
    <body name="L2_body" pos="0 0 {l1}">
      <inertial pos="0 0 {l2/2}" mass="{m2}" diaginertia="0.01 0.01 0.001" />
      <joint name="J_A" type="hinge" axis="0 1 0" limited="true" range="-2.5 2.5"/>
      <geom name="G_L2" type="capsule" fromto="0 0 0 0 0 {l2}" size="0.01" rgba="0.2 0.8 0.2 1"/>
      <site name="site_B" pos="0 0 {l2}" size="0.015" rgba="0 0 1 1"/>
    </body>
  </body>
</worldbody>

<tendon>
  <spatial name="tendon_BC" width="0.008" rgba="0.8 0.8 0.2 1"
           stiffness="{stiffness_L3}" damping="{damping_L3}" springlength="{l3}">
    <site site="site_B"/>
    <site site="site_C"/>
  </spatial>
  <spatial name="tendon_BD" width="0.008" rgba="0.2 0.8 0.8 1">
    <site site="site_B"/>
    <site site="site_D"/>
  </spatial>
</tendon>

<actuator>
  <position name="actuator_BD" tendon="tendon_BD" kp="{kp_actuator}" />
</actuator>
```

```
</mujoco>
    """
    return xml


def run_simulation():
    L1 = 0.075
    L2 = 0.0975
    L3 = 0.1125
    L4 = 0.075
    L5 = 0.375
    mass_density = 1.0

    xml_string = create_optimus_knee_xml(L1, L2, L3, L4, L5, mass_density)

    with open("optimus_knee.xml", "w") as f:
        f.write(xml_string)

    try:
        model = mujoco.MjModel.from_xml_string(xml_string)
        data = mujoco.MjData(model)
    except Exception as e:
        print("Ошибка при загрузке модели XML:", e)
        return

    with mujoco.viewer.launch_passive(model, data) as viewer:
        print("\nСимуляция запущена. Управление приводом начато.")

        actuator = model.actuator("actuator_BD")

        base_length = 0.40
        amplitude = 0.04
        frequency = 0.5

        while viewer.is_running():
            step_start = time.time()
            sim_time = data.time
            target_length = base_length + amplitude * np.sin(2 * np.pi * frequency * sim_time)
            data.ctrl[actuator.id] = target_length
            mujoco.mj_step(model, data)
            viewer.sync()
            time_until_next_step = model.opt.timestep - (time.time() - step_start)
            if time_until_next_step > 0:
                time.sleep(time_until_next_step)

if __name__ == "__main__":
    run_simulation()
```