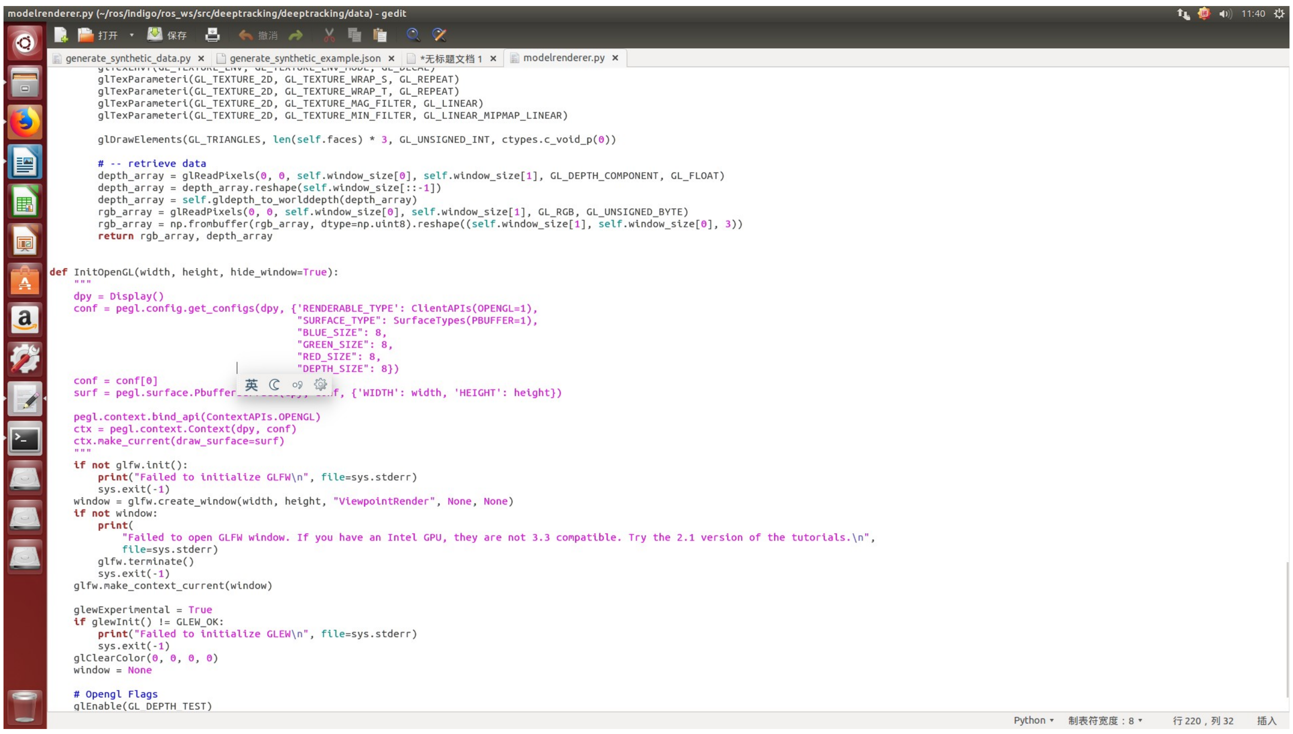
modelrenderer.py (~/ros/indigo/ros_ws/src/deeptracking/deeptracking/data) - gedit
11:40
打开
保存
撤消
generate_synthetic_data.py
generate_synthetic_example.json
*无标题文档 1
modelrenderer.py
Python
制表符宽度：8
行 220，列 32
插入

```python
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT)
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT)
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR)

        glDrawElements(GL_TRIANGLES, len(self.faces) * 3, GL_UNSIGNED_INT, ctypes.c_void_p(0))

        # -- retrieve data
        depth_array = glReadPixels(0, 0, self.window_size[0], self.window_size[1], GL_DEPTH_COMPONENT, GL_FLOAT)
        depth_array = depth_array.reshape(self.window_size[::-1])
        depth_array = self.gldepth_to_worlddepth(depth_array)
        rgb_array = glReadPixels(0, 0, self.window_size[0], self.window_size[1], GL_RGB, GL_UNSIGNED_BYTE)
        rgb_array = np.frombuffer(rgb_array, dtype=np.uint8).reshape((self.window_size[1], self.window_size[0], 3))
        return rgb_array, depth_array


def InitOpenGL(width, height, hide_window=True):
    """
    dpy = Display()
    conf = pegl.config.get_configs(dpy, {'RENDERABLE_TYPE': ClientAPIs(OPENGL=1),
                                         "SURFACE_TYPE": SurfaceTypes(PBUFFER=1),
                                         "BLUE_SIZE": 8,
                                         "GREEN_SIZE": 8,
                                         "RED_SIZE": 8,
                                         "DEPTH_SIZE": 8})
    conf = conf[0]
    surf = pegl.surface.Pbuffer[illegible]f, {'WIDTH': width, 'HEIGHT': height})

    pegl.context.bind_api(ContextAPIs.OPENGL)
    ctx = pegl.context.Context(dpy, conf)
    ctx.make_current(draw_surface=surf)
    """
    if not glfw.init():
        print("Failed to initialize GLFW\n", file=sys.stderr)
        sys.exit(-1)
    window = glfw.create_window(width, height, "ViewpointRender", None, None)
    if not window:
        print(
            "Failed to open GLFW window. If you have an Intel GPU, they are not 3.3 compatible. Try the 2.1 version of the tutorials.\n",
            file=sys.stderr)
        glfw.terminate()
        sys.exit(-1)
    glfw.make_context_current(window)

    glewExperimental = True
    if glewInit() != GLEW_OK:
        print("Failed to initialize GLEW\n", file=sys.stderr)
        sys.exit(-1)
    glClearColor(0, 0, 0, 0)
    window = None

    # Opengl Flags
    glEnable(GL_DEPTH_TEST)
```