Duckify

Week 3 review

Duckify

Week 3 review

GenAI

- **Milestones planned for this week 3 :**
  - Create full automated benchmark to evaluate and compare models, prompts, methods ❌
  - Solution that can generate a (future used model) luck with results that respect the constraint of the prompt without hallucinating using prompt engineering ❌
  - Explore resource-heavy solutions on Chacha and Disco, gather the results, and conduct a comparison ✅

GenAI

- https://miro.com/app/board/uXjVIpSPlA4=/?share_link_id=799167508538

MV-Adapter - Text2Texture

MV-Adapter - Text2Texture








MV-Adapter - Image2Texture

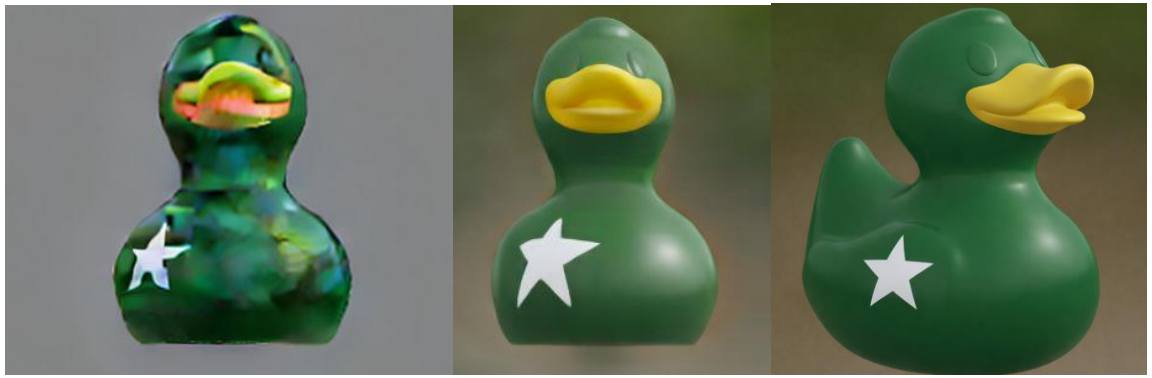
GenAI – Benchmark

- CLIP semantic match between prompt and image
  - Encoding both image and prompt then compare
  - Compute cosine similarity from -1 to 1

"Army Duck"	0.38
"Military green rubber duck, matte rubber toy material, yellow beak, white army star insignia on the side"	0.4299
"Yellow Duck"	0.27
"Red Car"	0.0410

GenAI – Benchmark Examples

- NIMA Aesthetic score (composition, lighting, visually appealing)
  - Trained on photos not 3d mesh
  - Score go from 0 to 10, the higher the better



4.0378 [NIMA SCORE] 4.6253 [NIMA SCORE] 5.2158[NIMA SCORE]

GenAI – Local Conclusion

- Focus on Disco instead of calypso (SLURM is working)
- Focus more on fine-tuning & prompt engineering of AI Solution

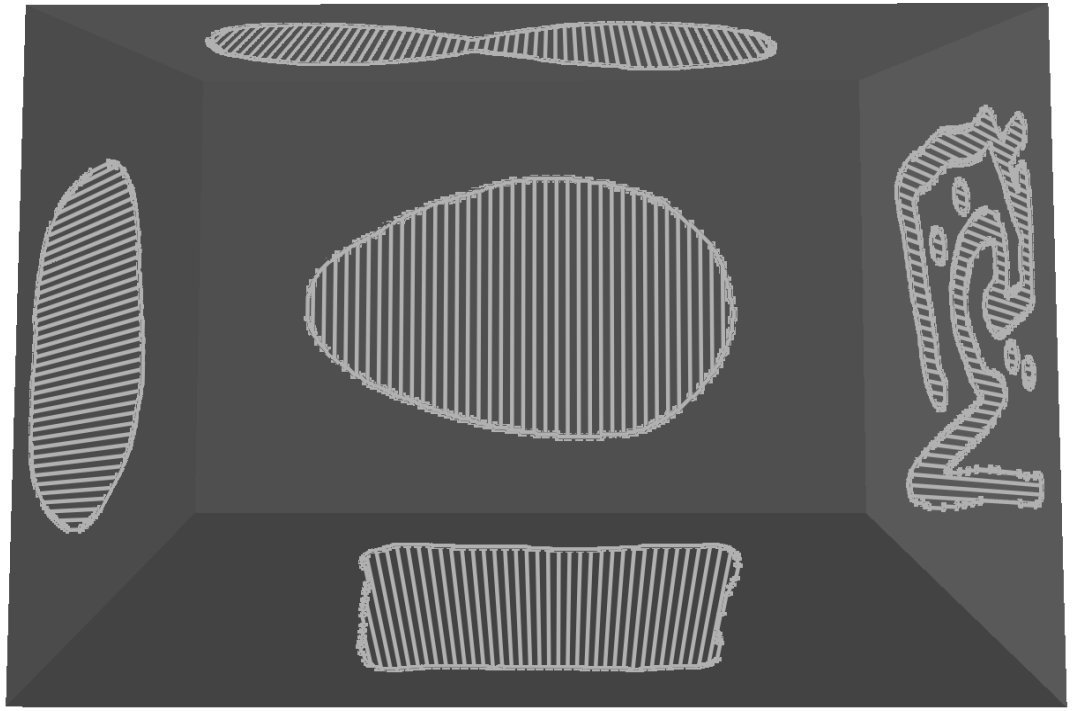
Tracing

Milestones planned for this week 3 :

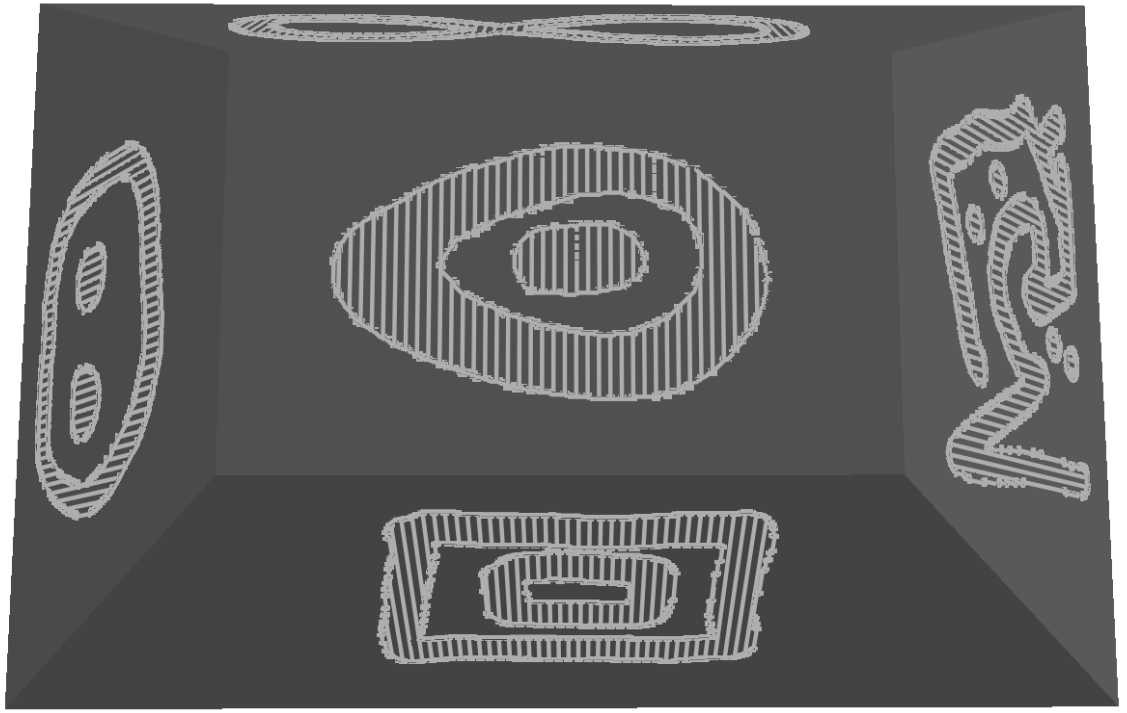
- Detection of multiple nested contours ☑
- Cross-faces traces ☑
- Palettization artefacts removal ☒
  - Tests and integration remaining
- Define performance evaluation criteria ☒
  - Time constraint

Tracing

Detection of multiple nested contours



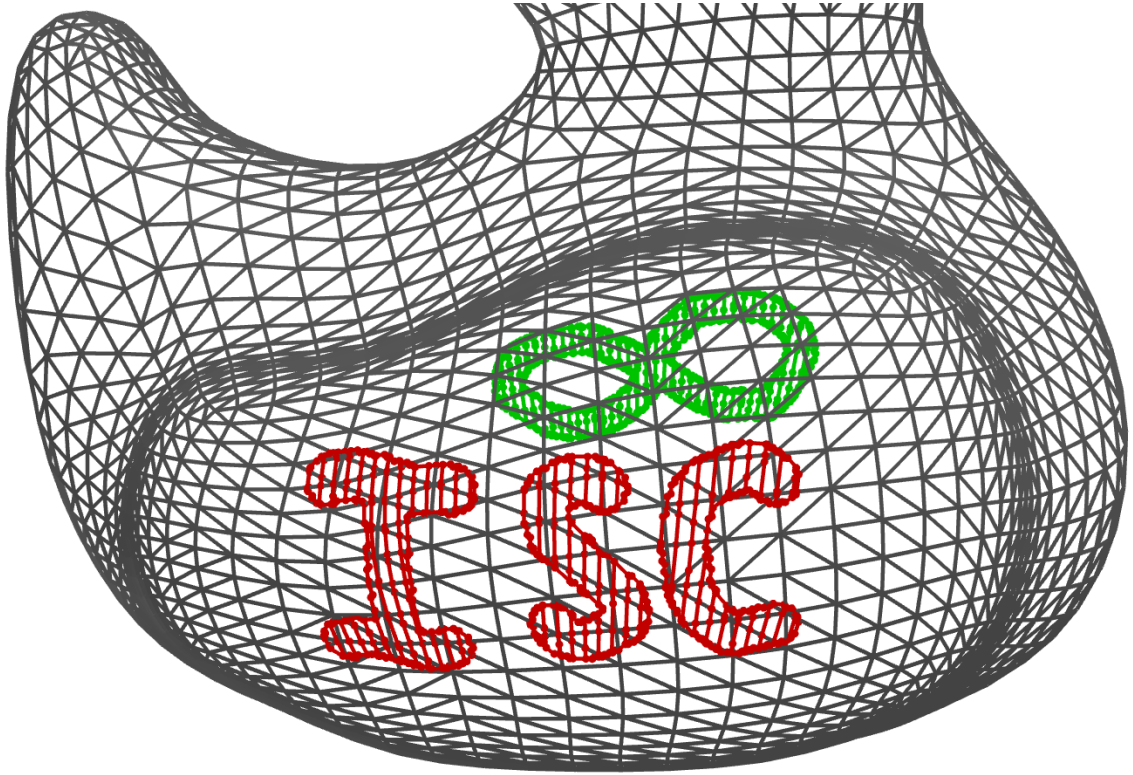
without the nested contours detection



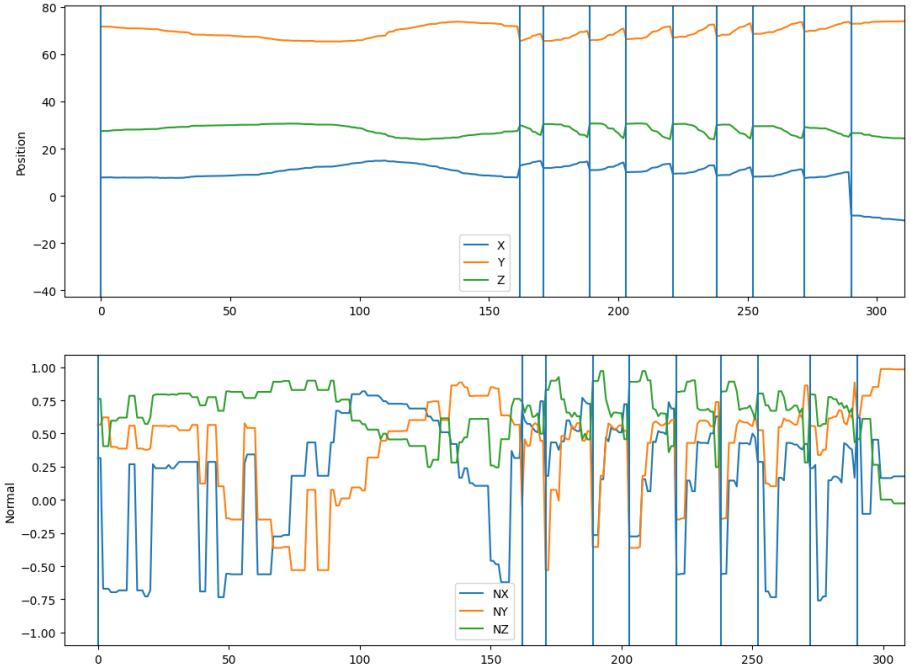
with the nested contours detection

Tracing

Cross-face traces



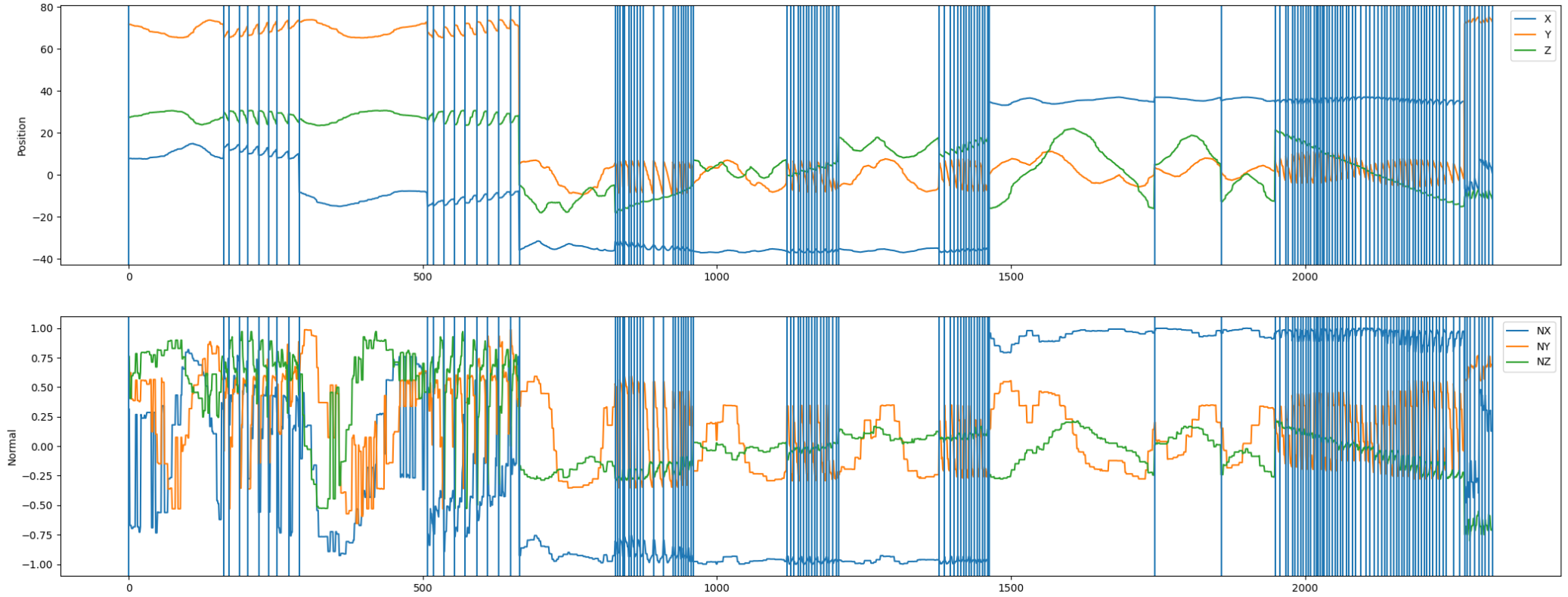
Cross-face paths on a duck



Drawing positions and normals

Tracing

Cross-face traces



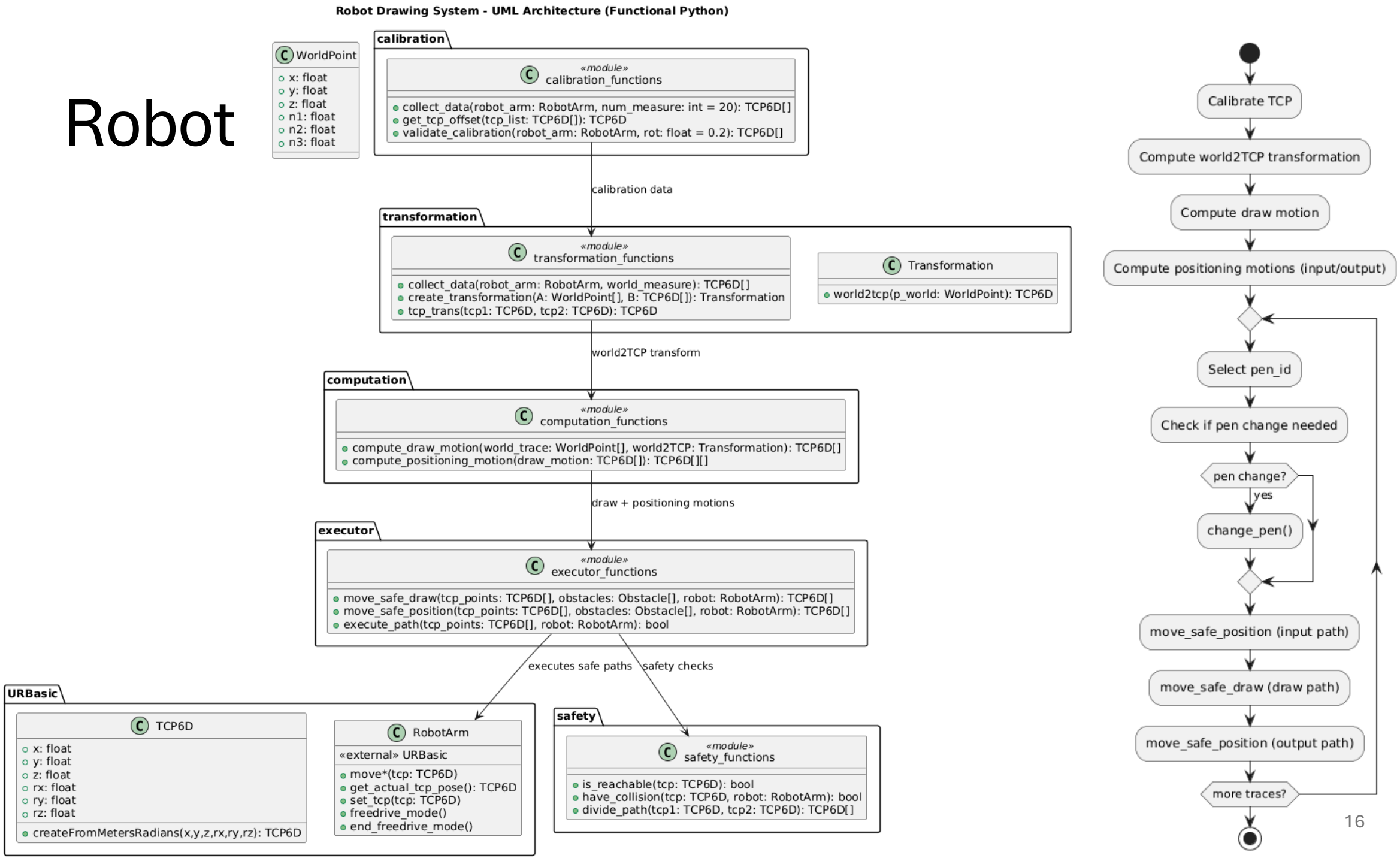
Robot

- **Milestones planned for week 3 :**
  - Conversion from world coordinate to TCP position (relative at the object)
  - Refine robot drawing pipeline (guideline)

  - Draw on a 3D surface (plane)

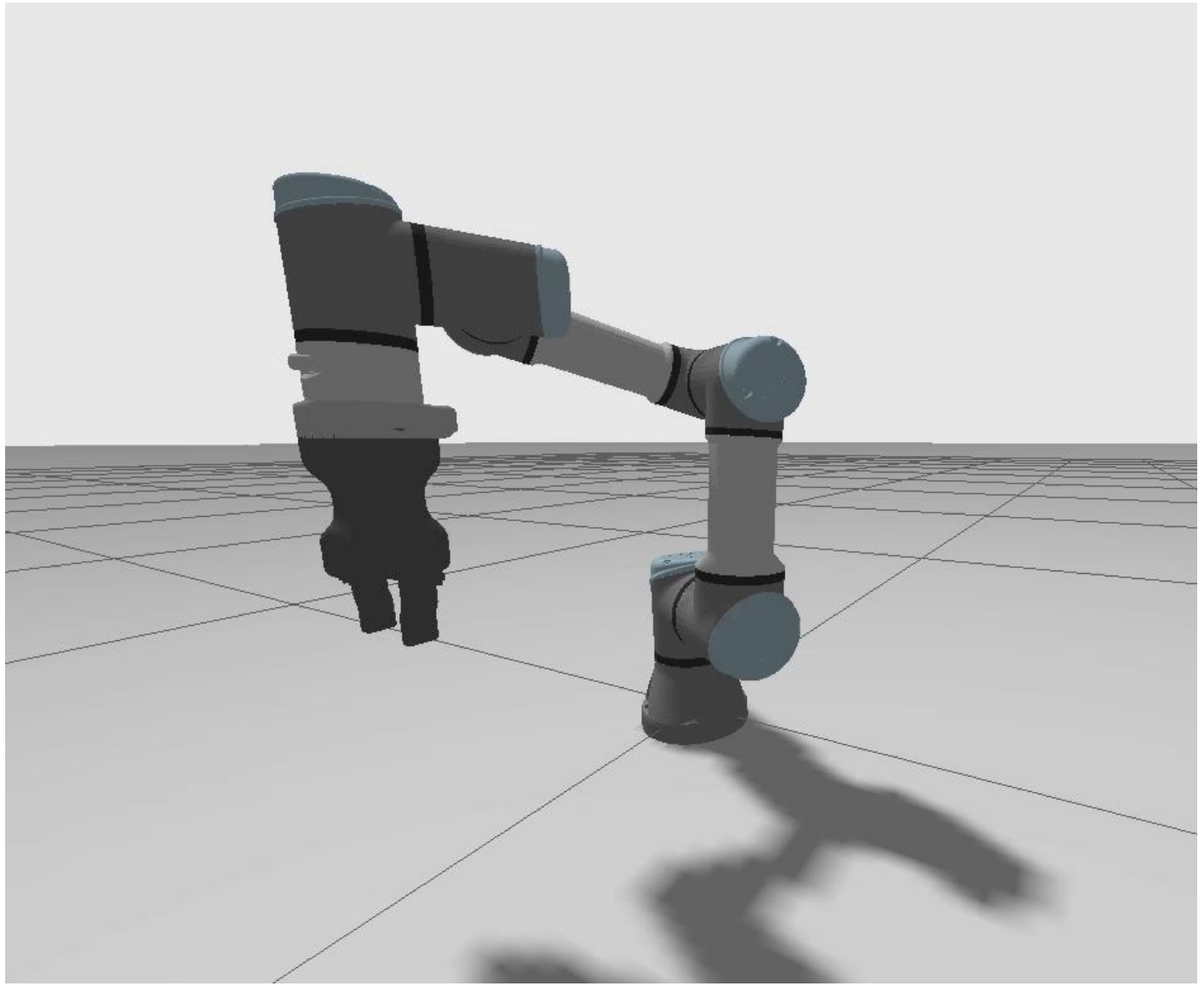
  - Can catch and manipulate tools (pens)
  - Draw on a 3D surface (duck)
  - Draw with 2 pens (change pens during drawing)

Robot

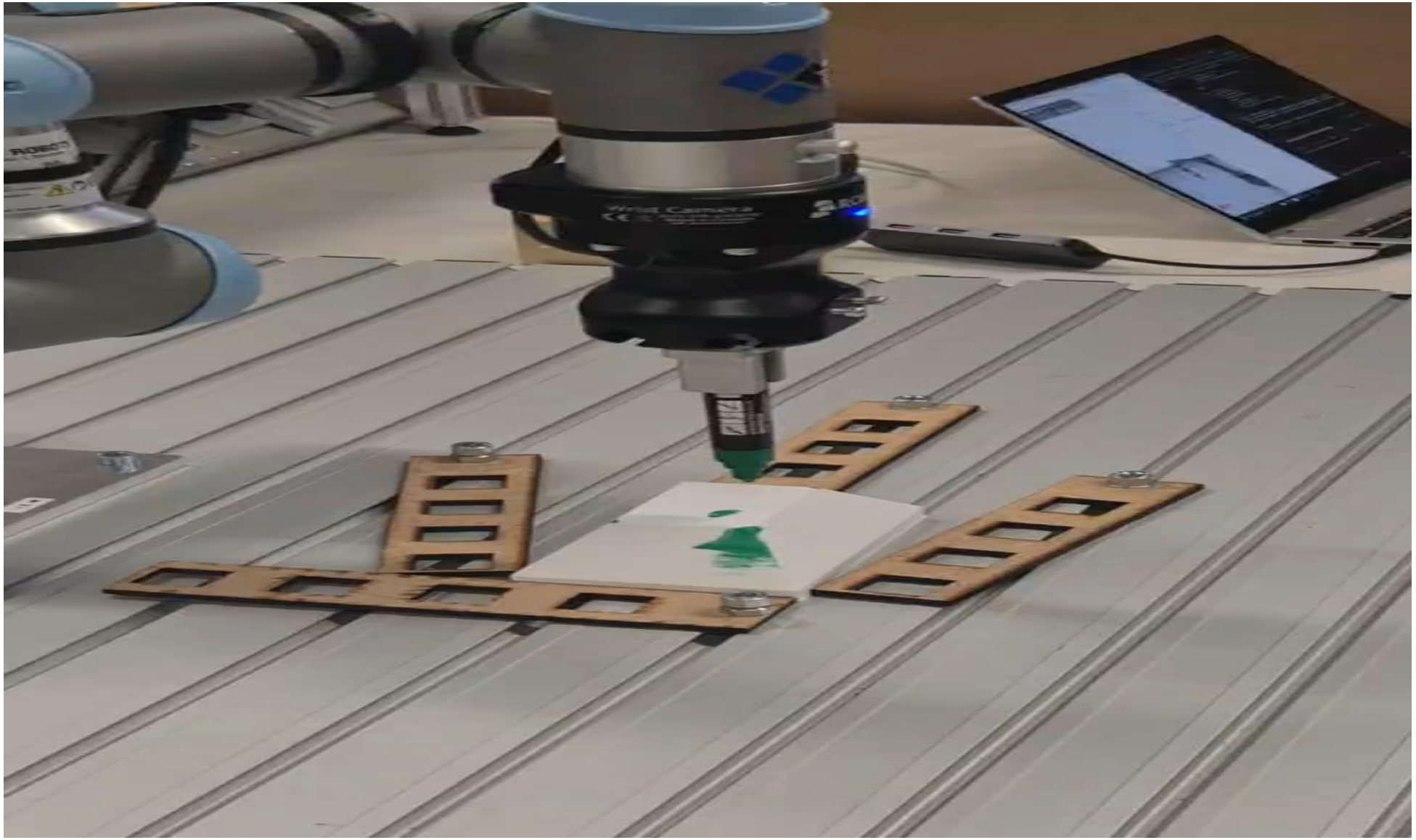

Robot Drawing System - UML Architecture (Functional Python)
WorldPoint
x: float
y: float
z: float
n1: float
n2: float
n3: float
calibration
«module»
calibration_functions
collect_data(robot_arm: RobotArm, num_measure: int = 20): TCP6D[]
get_tcp_offset(tcp_list: TCP6D[]): TCP6D
validate_calibration(robot_arm: RobotArm, rot: float = 0.2): TCP6D[]
calibration data
transformation
«module»
transformation_functions
collect_data(robot_arm: RobotArm, world_measure): TCP6D[]
create_transformation(A: WorldPoint[], B: TCP6D[]): Transformation
tcp_trans(tcp1: TCP6D, tcp2: TCP6D): TCP6D
Transformation
world2tcp(p_world: WorldPoint): TCP6D
world2TCP transform
computation
«module»
computation_functions
compute_draw_motion(world_trace: WorldPoint[], world2TCP: Transformation): TCP6D[]
compute_positioning_motion(draw_motion: TCP6D[]): TCP6D[][]
draw + positioning motions
executor
«module»
executor_functions
move_safe_draw(tcp_points: TCP6D[], obstacles: Obstacle[], robot: RobotArm): TCP6D[]
move_safe_position(tcp_points: TCP6D[], obstacles: Obstacle[], robot: RobotArm): TCP6D[]
execute_path(tcp_points: TCP6D[], robot: RobotArm): bool
executes safe paths
safety checks
URBasic
TCP6D
x: float
y: float
z: float
rx: float
ry: float
rz: float
createFromMetersRadians(x,y,z,rx,ry,rz): TCP6D
RobotArm
«external» URBasic
move*(tcp: TCP6D)
get_actual_tcp_pose(): TCP6D
set_tcp(tcp: TCP6D)
freedrive_mode()
end_freedrive_mode()
safety
«module»
safety_functions
is_reachable(tcp: TCP6D): bool
have_collision(tcp: TCP6D, robot: RobotArm): bool
divide_path(tcp1: TCP6D, tcp2: TCP6D): TCP6D[]

Calibrate TCP
Compute world2TCP transformation
Compute draw motion
Compute positioning motions (input/output)
Select pen_id
Check if pen change needed
pen change?
yes
change_pen()
move_safe_position (input path)
move_safe_draw (draw path)
move_safe_position (output path)
more traces?

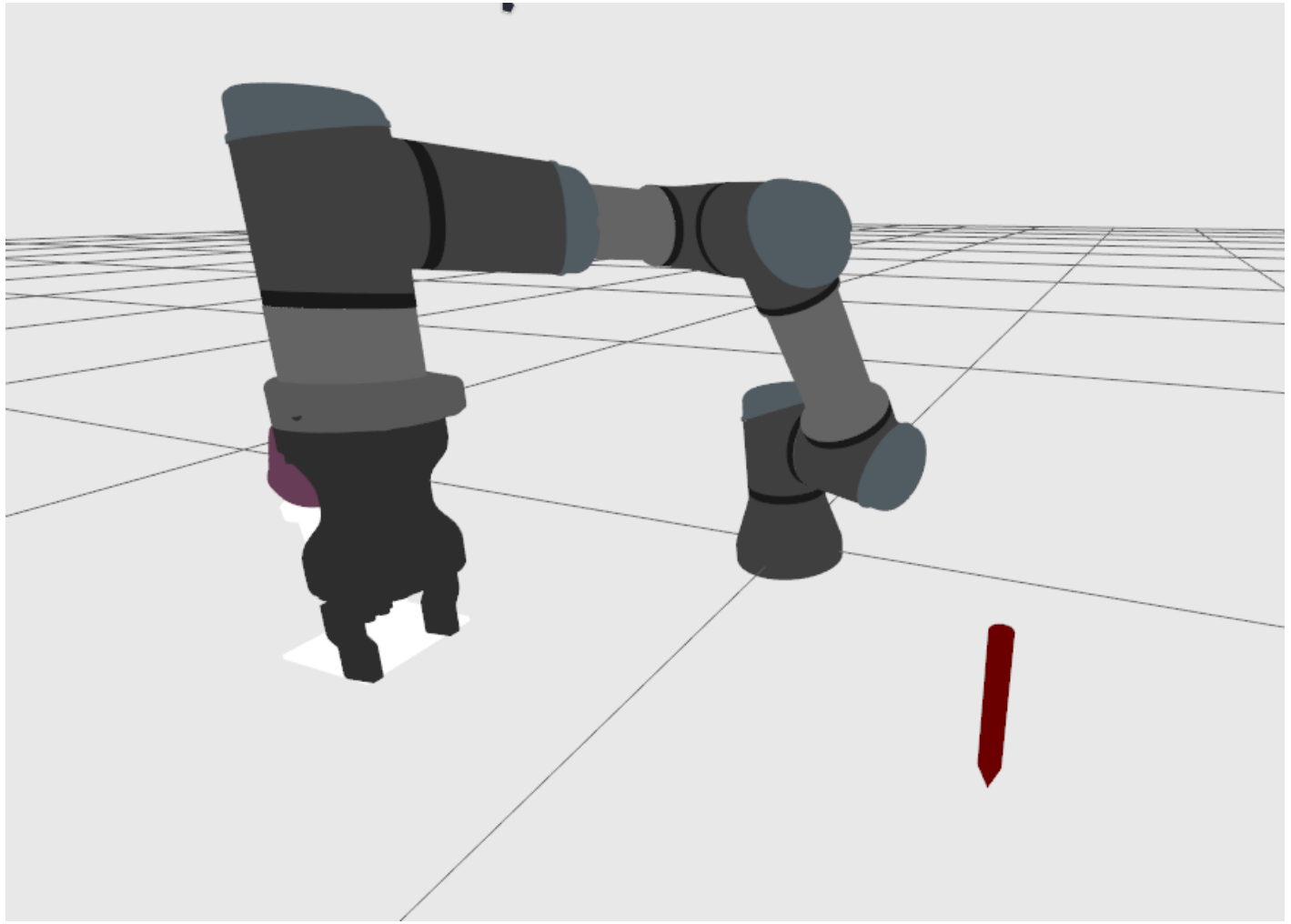
Robot



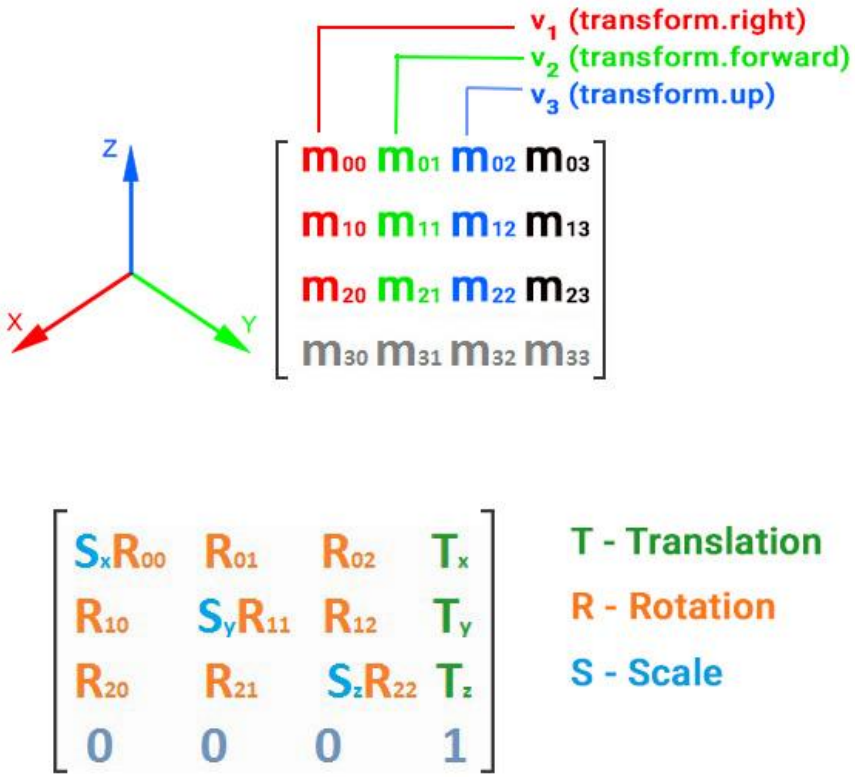
Robot

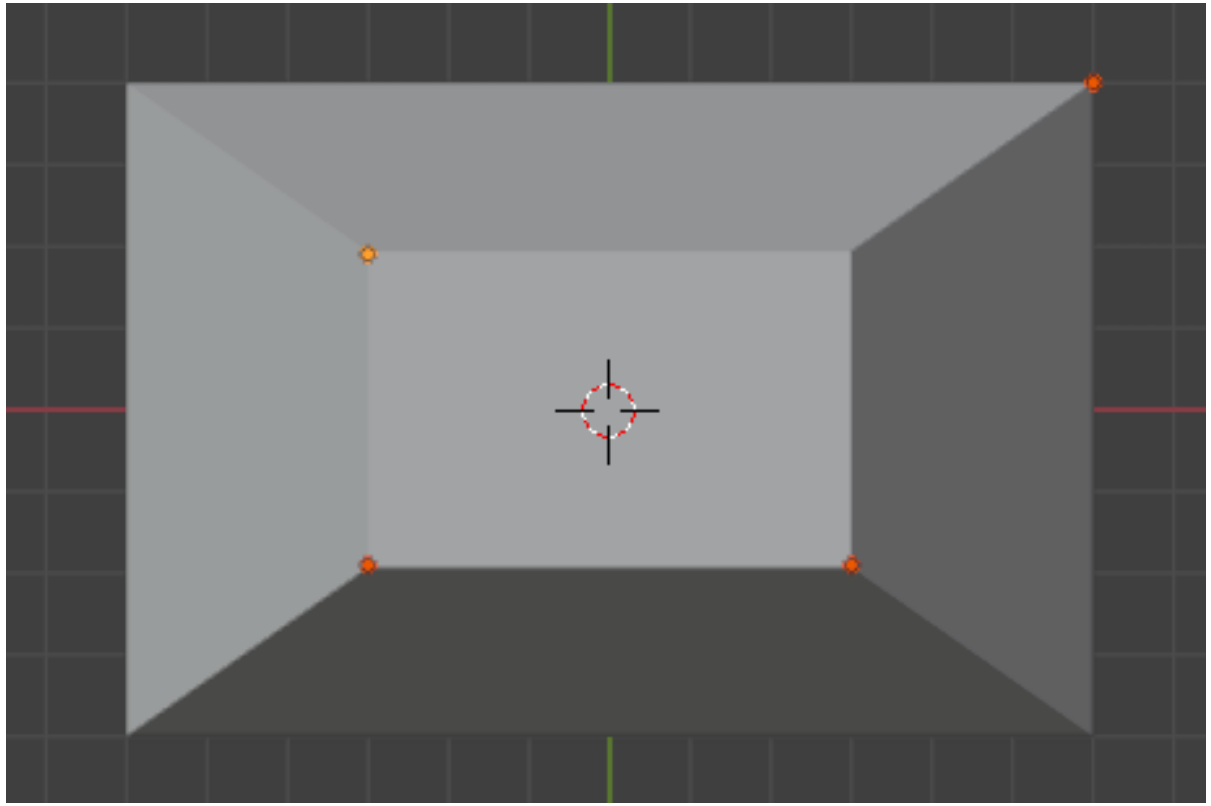


Robot - Get Pen



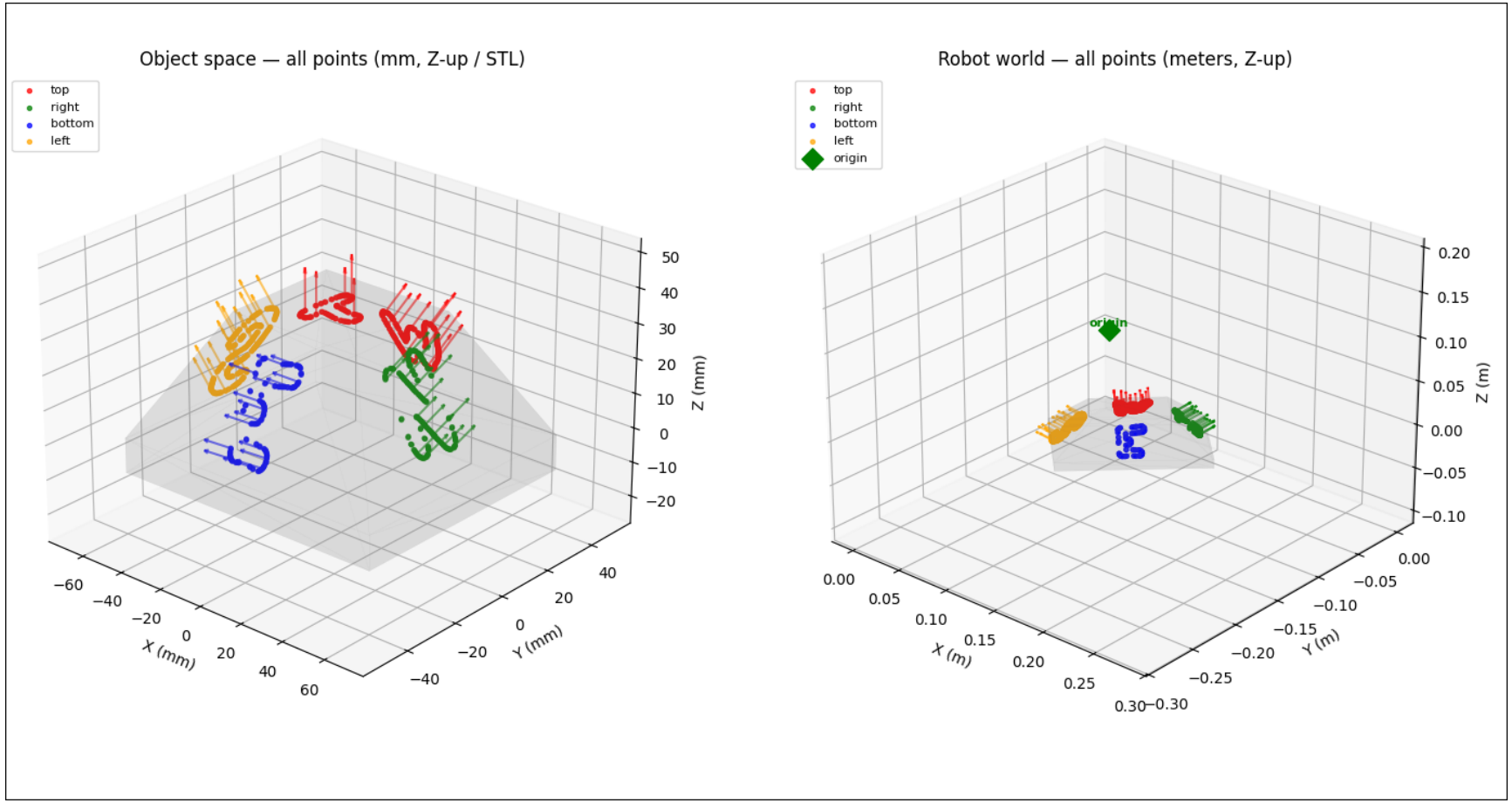
Robot



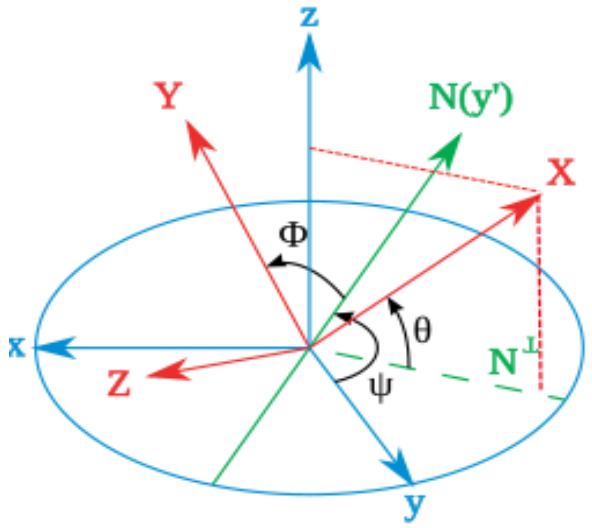


https://forum.unigine.com/topic/6451-matrix-transformation-confuses-me/

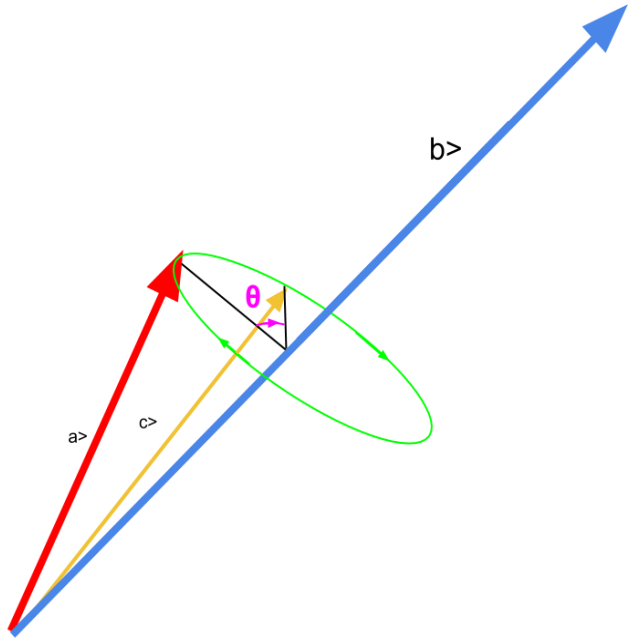
Robot


Object space — all points (mm, Z-up / STL)
top
right
bottom
left
X (mm)
Y (mm)
Z (mm)
Robot world — all points (meters, Z-up)
top
right
bottom
left
origin
X (m)
Y (m)
Z (m)

Robot



https://en.wikipedia.org/wiki/Euler_angles



https://math.stackexchange.com/questions/511370/how-to-rotate-one-vector-about-another

Robot

- Issues:
  - Math understanding
  - Time with robot
- Afternoon: Validation with real robot

- Security of the motion, collision avoidance, path planner
  1. From scratch
     - Implement ourself some basic code
  2. Extern library
     - Search and implement/adapt to another code
  3. Movit2
     - Heavy and complex, but linked to URBasic

Robot

- **Milestones planned for week 4 :**
  - Can catch and manipulate tools (pens)
  - Draw on a 3D surface (duck)
  - Draw with 2 pens (change pens during drawing)
  - Investigate and implement a safety solution

Impression 3D

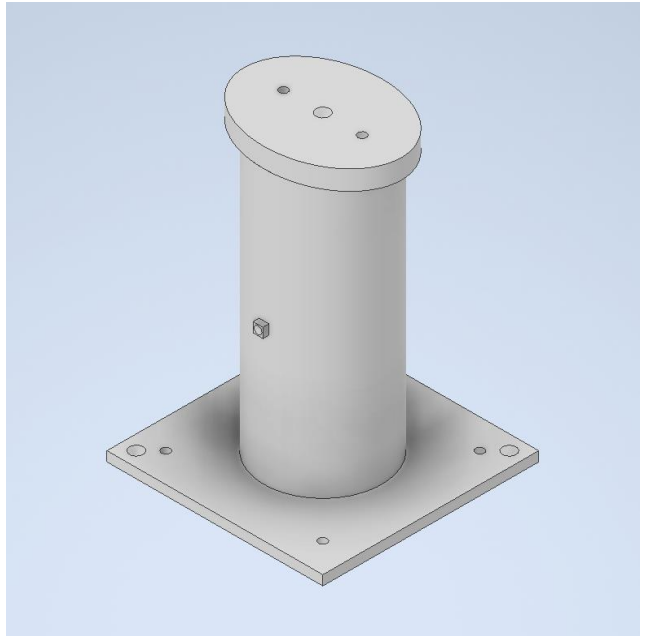
Milestones planned for this week 3 :

- Validate all the non-confirm pieces : ☑
  - Duck support
    - Expert (Ms. Richard)
  - Duck
    - CEO
  - Wooden base
    - Expert (Mr. Darbellay)

Impression 3D

Milestones planned for this week 3 :

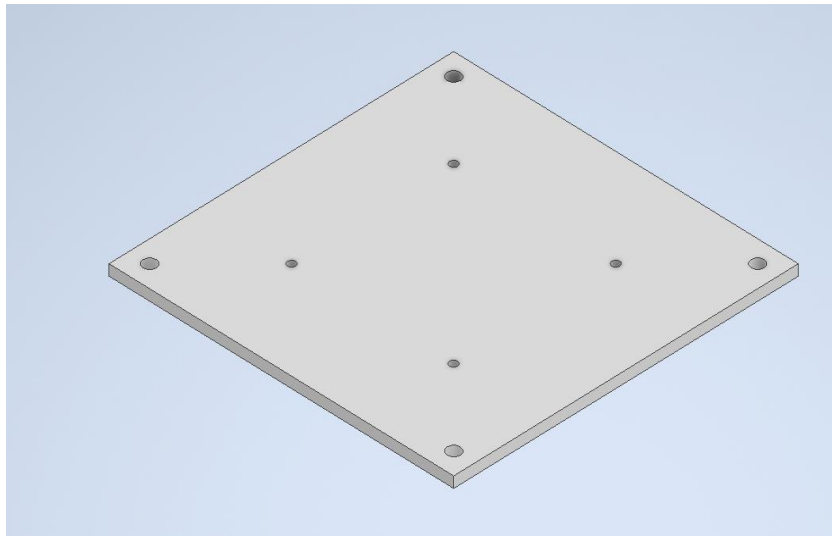
- Have a duck support printed ☑

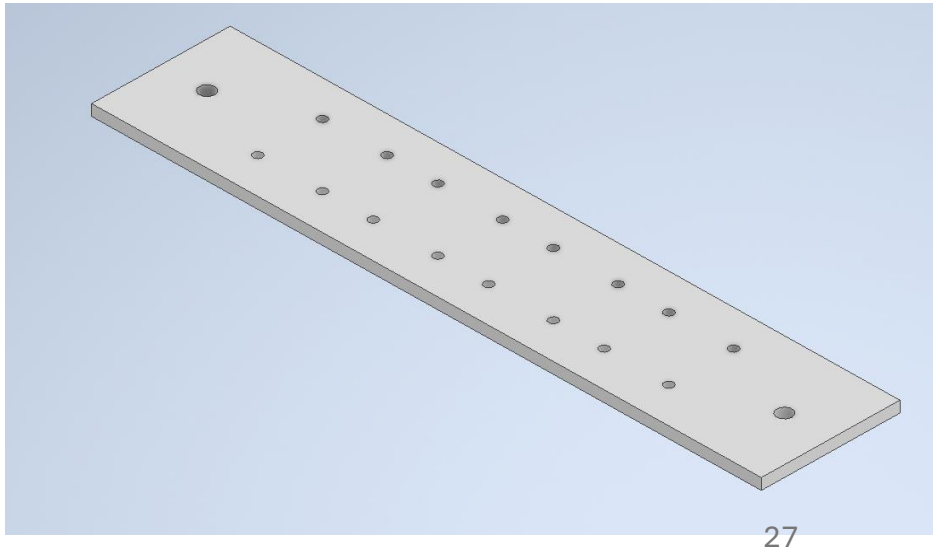


Impression 3D

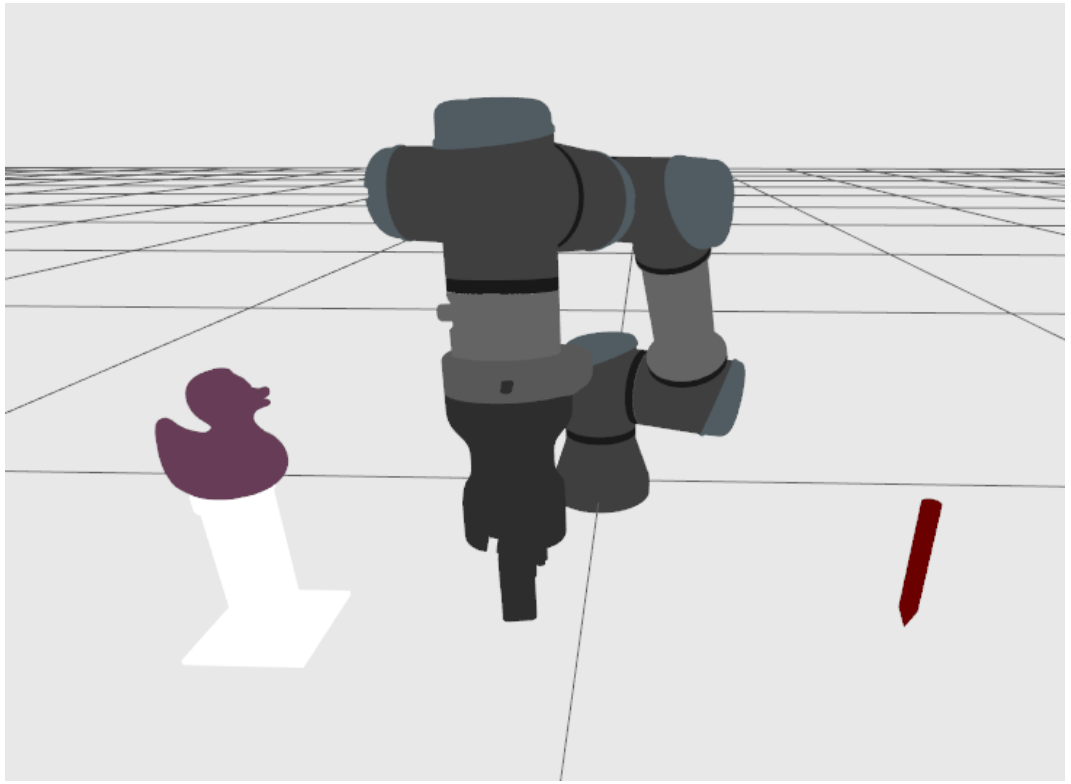
Milestones planned for this week 3 :

- Have the wooden piece used as support to arrange the position ☑ of the rest (pen support, duck support) done
  - Model done
  - Cut





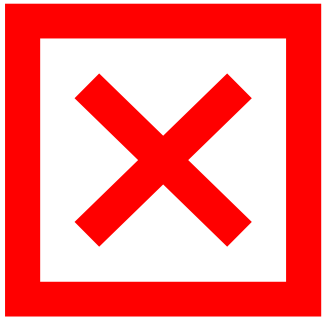
Simulation / Reality



Week 3 main milestone

- Main milestone was :
  - **MVP : ***"Using the complete pipeline, draw basic shapes (points, line) on a duck installed on the support using only one color"*

- **Blockers**
  - Implementation paradigm changes in robot team
  - Time constraint



Next week

- *Using the complete pipeline, draw basic shapes (points, line) on a duck, installed on the support using only one color*
  - *GenAI Blocker*
  - *Coordinate validation*

- Using the complete pipeline, draw locally generated shape contours with selected GenAI solution (Image2Texture), on a duck, installed on a stable support using multiple colors while changing colors automatically.