3. (opt) Make moves by dragging in 3D viewport

Monday, 18 September 2023 12:10

What does the user need to do?
Click on the puzzle, then drag the mouse while holding down the mouse button. The two locations of the mouse_down and mouse_up events form a user input vector. (2d location + 2d direction vectors)
Normalize the direction vector.

How to detect which move is meant to be made?
For all moves the clicked piece of the puzzle is a part of, calculate the rotation vector at the clicked surface point:

- *move normal*: ≈ rotation axis. The average of all rotation vectors of the points changed by the move. Rotation vector of a point A that moves to point B is: $(A - COM) \times (B - COM)$. Here, COM is the center of mass of all points changed by the move.
- *Surface normal*: normal of puzzle shape at clicked position (project 2d click position onto 3d puzzle shape). If puzzle is shown as points, this could be the vector $(P - COM)$. Here, COM is the cetner of mass of all points of the puzzle.
- *Rotation vector*: = surface_normal × move normal This vector indicates the direction in which the clicked point will move when making the current move.
- *Projected rotation vector*: project the rotation vector onto the viewport screen./ camera plane (vec3d -> vec2d). Then normalize the vector.

After getting these projected rotation vectors for all available moves, find the move whose *projected rotation vector* is closest to the *normalized input direction vector* (measure angle between the vectors)
=> sort by dot product, **1** = smallest angle, **-1** = largest angle