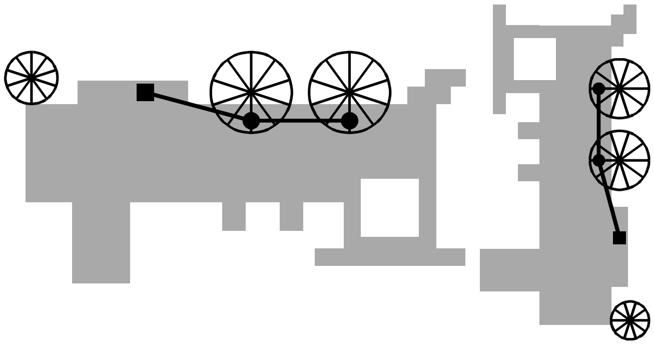
Workbook Physical Simulation
Interdisciplinary Exercises from the STEM Fields
www.physolator.com

Exercise Sheet 11

Geometric Transformations



In exercise sheet 7 a locomotive was drawn. This locomotive shall now be drawn at different sizes at different positions. In addition, the locomotive is to be rotated at different angles. This requires geometric transformations.

The Physolator provides two methods for applying geometric transformations to drawing commands: *beginTransformation* and *endTransformation*. If you have one or more drawing commands in the program code to which a geometric transformation is to be applied, place the *beginTransformation* command in front of these drawing commands and the *endTransformation* command behind them. The *beginTransformation* command contains a geometric transformation as a parameter. This transformation is applied to all drawing commands between the *beginTransformation* and *endTransformation* commands.

The following code shows an example. The *beginTransformation* command specifies that the transformation $(x, y) \mapsto (x, y+2)$ is to be applied to the following *drawLine* command. Therefore, the *drawLine* command does not produce a line from $(1,2)$ to $(3,4)$, but from $(1,4)$ to $(3,6)$.

```
beginTransformation((x, y) -> new Vector2D(x, y + 2));
drawLine(1, 2, 3, 4);
endTransformation();
```

Exercise 1

Change the program code of Exercise Sheet 7 so that the locomotive is no longer displayed in its original size, but enlarged by a factor of 2. Apply a centric stretching at the origin.

Exercise 2

Apply the following geometric illustrations to the locomotive in the same way as for the preceding exercise:

- axis mirroring on the x-axis

- axis mirroring on the y-axis
- a 45° counterclockwise rotation about the origin
- a translation by $(4,-3)$
- a point reflection through a center located a t $(6,2)$

Exercise 3

The following transformations are to be applied to the locomotive one after the other:

- First reflect the locomotive on the y-axis.
- Then scale the locomotive with a scaling factor of *scalingFactor.*
- Then rotate the locomotive counterclockwise through an angle of *rotationAngle* about the origin
- Finally translate the locomotive by an offset of (dx, dy) .

First, add the object attribute *stretchingFactor*, *rotationAngle*, *dx* and *dy* to your program code. Add appropriate annotations in the same way as in exercise 2 of exercise sheet 7. The user shall be able to change these variables at runtime.

Hint

The *beginTransformation-endTransformation* command pairs can be nested. A *beginTransformation-endTransformation* pair of commands causes the transformation to be applied to the drawing commands in between. In the drawing commands in between there can be other transformations. Be aware, that for nested transformations, the transformations are executed from the inside to the outside.

Exercise 4

Graphics or parts of graphics, that have once been programmed, can be flexibly positioned in the overall image using geometric transformations. In exercise sheet 7, The method *drawLocomotive* was implemented. In this exercise, this method shall be newly implemented. The methods *beginTransformation* and *endTransformation* are now to be used.

The method *drawLocomotive* previously had two parameters *x* and *y*, which were used to specify where the locomotive shall be drawn. These parameters shall now be omitted and the method *drawLocomotive* shall draw the locomotive at its original position. Remove these two parameters. If you want to draw the locomotive at a position other then the original position, you can do this by applying a translation.

Before the method *drawLocomotive* is programmed, an auxiliary method *drawWheel* shall first be programmed. The method *drawWheel* is to draw a wheel with a radius of 1. The center of the wheel shall be in the origin. The locomotive contains three wheels at different positions and with different radii and rotation angles *alpha* and *beta*. The method *drawWheel* shall be called three times in the method *drawLocomotive*. Transformations shall be used to bring the wheels into the correct positions. To bring the wheel to the desired size, a scaling shall be carried out. The desired angle of rotation is then to be determined by a rotation. By means of a translation, the wheel is finally brought to the desired position.