First, set up an Eclipse environment with the required Henshin plug-ins. An Eclipse including all required Eclipse Modeling Tools can be found at http://www.eclipse.org/downloads (see also the lecture homepage). An installation instruction for the transformation tool EMF-Henshin presented in the lecture can be found at http://www.eclipse.org/henshin/install.php.

Familiarize yourself once again with the basic operation of the tool by working through, and understanding, the tutorial Bank Accounts step by step (http://www.eclipse.org/modeling/emft/henshin/examples.php?example=bank).

**Task 5.1:** Use of graph transformation engines

The topic of this exercise is a simple variant of the game “PacMan”. The characters of the game, PacMan and Ghost, move on defined fields. For all fields their adjacent fields are statically defined and cannot change during the game. The game characters PacMan and Spirit can move from one field to the next field via the neighbourhood relationships. In addition, there can be marbles, which can be collected by PacMan, on fields. PacMan itself can be eaten by Ghosts. A representation of the concepts described above is shown in figure 1. In the terminology of model-based development, this is also referred to as the metamodel.

![Diagram](image_url)

Figure 1: Static PacMan concepts: Type graph (meta model)

Use the Eclipse Modeling Framework (EMF) and the graph transformation system EMF-Henshin to simulate the PacMan game by applying graph transformation rules. The transformation system shall include the following rules:

- **move PacMan** (movePM): PacMan is moved from its current field to an adjacent field.
- **collect marble** (collect): When applying this rule, PacMan is moved from its current field to an adjacent field. However, there is an additional marble on the field that is removed from the field during rule application. Furthermore, the number of marbles collected by PacMan is incremented by one.
- **move ghost** (moveGhost): Like PacMan, a ghost can be moved from its current field to an adjacent field.
- **kill PacMan** (kill): If a ghost is moved to an adjacent field on which PacMan is located then PacMan is eaten (killed).

**Hints** (s. also http://pi.informatik.uni-siegen.de/Mitarbeiter/dreuling/teaching/ss17/st2/):

1. Start with the translation of the type graph shown in Fig. 1 into EMF-Ecore. Pay attention in particular to the technical condition that a container class is defined for each class of an Ecore model.
2. Draft the transformation rules with the graphical editor (Henshin Diagram) presented in the lecture. Do not forget to import the previously defined type graph.
3. Create one or more dynamic instances of your Ecore model, as you have done several times in the EMF-related exercises. You can then use these instances as test data for the application of the transformation rules.

**Task 5.2:** Graph transformationen in practice

Imagine you have to develop an editor for UML class diagrams. The editor should offer various...
editing commands for changing models, from setting simple local properties of model elements to complex object-oriented refactorings. The editing commands are to be implemented using Henshin transformation rules. Implement two such editing operations for UML2 class models as examples:

1. adding a binary association
2. deleting a binary association.

As a metamodel, use the EMF-based implementation of the UML2 metamodel (www.eclipse.org/uml2/). The UML2 plug-ins are already included in the EMF variant of Eclipse.

Test your rules using UML2 test models created by you.