Second Programming Task

Catching Mice in a Smalltalk Game

Learn Smalltalk, and design and implement a game in Smalltalk. The game shall resemble that of the first programming task, but it need not be network-based, and only one cat (one player) tries to catch mice. It is recommended to use Squeak, a relatively up-to-date implementation of Smalltalk-80. All team members shall work on this task to get their own Smalltalk experiences.

Smalltalk programmers usually prefer to adapt existing code over writing new code from scratch. There is good support for graphical interfaces. It may be a good idea to start with the code of another game using appropriate graphical interfaces and change this code as needed for the game. Please try out this style of programming.

Please answer the following questions (in the Abgabegespräch):

- Do you prefer to program in a dynamic or static language? Why?

- How important is the run-time penalty of an interpreted language like Smalltalk?

- Is the concept of "personal computing" (each person has its own customized system) still adequate? Are there any good alternatives?

- How can team members cooperate when developing software together in Smalltalk?

- Which aspects of programming in Smalltalk do you like, which aspects don't you like?