

is used to satisfy the first requirement. It should be called every time a user changes a solution. This makes framework to track the current solution, and later provide it for a module when it loads.

3.3. Saving and loading solutions

After a virtual laboratory loads and opens, the framework calls the `loadSolution()` method for two times. The first time it loads the best solution, this makes module know the best result and properly display it. The second time the framework loads the solution that was the last saved with the `autosave()` method. Note that a proper behavior of a module is already guaranteed if it properly implemented all `KioProblem` interface methods, so one does not need to implement startup logic separately.

Each time the `autoSaveSolution()` method is called, the framework gets the current solution with the `solution()` method and stores it. The same method is called to store the best solution when the framework detects that a new best solution is obtained. It happens in the `submitResult()` method. And the same method is called if a user manually saves his or her solution by means of a button located to the right side of the screen (see Figure 2). The right gray side is general for all the modules; it is drawn and supported by the framework. So, all Flash modules have the same right side as in the Figure 2. There is also a load button at the right side, it calls the `loadSolution()` method. The same method is called when a “load record” button is clicked; it is possible because the framework stores the best solution. And the same method is called after the contest during the solutions check.

So, just a few methods can handle all the saving and loading logic. There are other generalizations that may be done

within the framework. First, we do not currently have a clear button at the right side, this buttons are implemented by programmers. And there is an idea to move information about problems evaluations to the right side, the framework has enough information to properly display it. We did not do it earlier because in Flash this information was always a part of the laboratory's design, on figure 2 it is located at the bottom.

4. Conclusion

Virtual laboratories or software modules for the CTE contest require a lot of programming, but they worth it because they are used not only in the contest, but also in the usual classwork. There are a number of requirements on the behavior of laboratories about loading, saving, and working with solutions. But the framework used to implement them makes it very easy to satisfy all the requirements by implementing a small interface and using just about 3-5 api methods.

5. References

- [1] Pozdniakov S., Posov I., Pukhov A., Tsvetkova I., “Science Popularization by Organizing Training Activities Within the Electronic Game Laboratories”, *International Journal of Digital Literacy and Digital Competence (IJDLDC)*, vol 3: 2 Issues, 17-31, 2012
- [2] <http://ipo.spb.ru/kio>: The official CTE contest site
- [3] Papert S., *Mindstorms: Children, computers, and powerful ideas*, New York: Basic Books, 1980
- [4] Joseph O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987