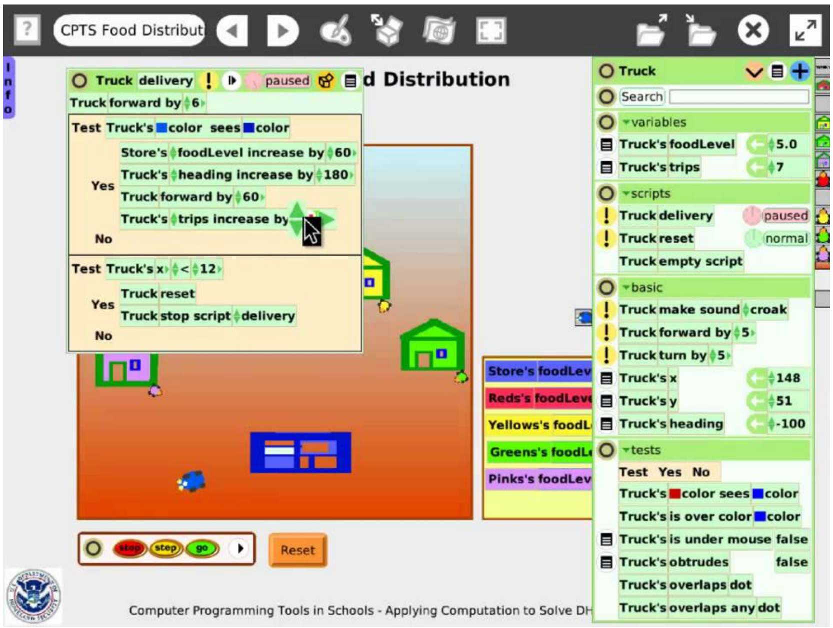
Design and Implementation of a Live Programming Tool Set for Heterogeneous Simulations in Squeak/Smalltalk

Master Project Proposal, Summer Term 2019
Software Architecture Group, Prof. Dr. Robert Hirschfeld

In this project, we will design and implement a tool set to enable a live programming workflow for working on simulations with heterogeneous objects in an educational setting. To support a large variety of object behaviors, the tool set will be based on the entity component system (ECS) architectural style.

In the course of this project, we will review the problem domain of programming-based learning environments, such as Scratch, Etoys, Kedama, and StarLogo, and the design space of programming tools for heterogeneous simulations. Based on these insights, we will design, implement, and evaluate the tool set in the Squeak/Smalltalk live programming environment.

Learning through Live Programming in Simulations



Interactive simulations can be used in teaching to allow learners to validate and reconsider assumptions they have about, for example, a physics model. Even further, when learners can develop and program the model themselves, they might discover conflicts between the behavior of the simulation and their initial mental model, consequently evolving their model further [1]. This idea of learning-by-programming simulations has been implemented in systems such as Scratch, Etoys, Kedama, and StarLogo.

To keep the feedback loop between changing the simulation and seeing the effects of changes as short as possible many environments include live programming capabilities so that learners can change the simulation code while it is running [2].

Further, the existing environments focus mostly on models with either a low number of heterogeneous objects (for example Etoys) or a large number of homogeneous objects (for example Kedama). As a result, the tools and programming mechanisms are optimized for either one of these two cases. However, besides these types of simulations there are also simulations for models which require a large number of heterogeneous objects with changing dynamic behavior. An example for such a system is a city transport simulation which involves inhabitants, numerous vehicles of various types, transportation infrastructure, and physical terrain. The behavior of all these objects additionally may depend on a number of factors such as the time of day or weather conditions.

Background: The Entity Component System (ECS) Pattern

Simulations with large numbers of heterogeneous objects with dynamically changing behavior can be built with the entity component system (ECS) architecture pattern. ECS is an architecture designed to scale the development of simulations with regard to the multitude and dynamicity of the combinations of behaviors of simulation objects. ECS splits up a simulation object in three parts: the mere identity (entity), data for each aspect of the object (component), and simulation logic working on multiple entities (system). A system works on a group of entities which contain a set of components [3]. The ECS architecture is well suited for interactive simulations for learning as it explicitly features heterogenous and dynamically changing components. Further, its performance characteristics allow for a large number of objects.

Challenge: Designing and Implementing Live Programming Tools for ECS

ECS as an architectural style assumes control over the object life-cycle and control flow of the simulation. Existing live programming tools are not aware of ECS and do not support users in an interactive workflow when working on a heterogeneous simulation. To design a suitable tool set for programming heterogeneous simulations with large numbers of objects, we will:

- Review the problem domain of existing environments for programming simulations interactively in education
- Review the design space of existing interactive simulation programming environments and ECS tool sets
- Design and implement a tool set for educational live programming of heterogeneous simulations in Squeak/Smalltalk
- Document prototypical application scenarios of the tool set and discuss the design with respect to the determined requirements space

Contact

Prof. Dr. Robert Hirschfeld, Patrick Rein, Stefan Ramson
Software Architecture Group, Potsdam
http://www.hpi.uni-potsdam.de/swa, hirschfeld@hpi.uni-potsdam.de

References

[1]: Allen-Conn, B.J., Rose, Kim. “Powerful ideas in the classroom: using squeak to enhance math and science learning”. (2004). *Viewpoints Research Institute, Inc*. (2004). isbn-10: 0974313106.

[2]: Patrick Rein, Stefan Ramson, Jens Lincke, Robert Hirschfeld, and Tobias Pape. “Exploratory and Live, Programming and Coding: A Literature Study Comparing Perspectives on Liveness”. In: *The Art, Science, and Engineering of Programming* 3.1 (2018). doi: 10.22152/programming-journal.org/2019/3/1.

[3]: Patrick Rein, Stefan Ramson, Jens Lincke, Tim Felgentreff, and Robert Hirschfeld. “Group-Based Behavior Adaptation Mechanisms in Object-Oriented Systems”. In: *IEEE Software, SWSI: Context Variability Modeling* 34.6 (2017), pp. 78–82. doi: 10.1109/MS.2017.4121224.