Three-minute Constructionist Experiences

Ken Kahn, *toontalk@gmail.com*
Computing Services, Oxford University

Howard Noble, *howard.noble@oucs.ox.ac.uk*
Computing Services, Oxford University

Arthur Hjorth, *arthur.hjorth@u.northwestern.edu*
Learning Sciences, Center for Connected Learning, Northwestern University

Fábio Ferrentini Sampaio, *ffs@nce.ufrj.br*
Electronics Computing Center, Federal University of Rio de Janeiro

Abstract

Constructionist learning experiences typically take time: time to build and run programs – typically even more time to learn how to program. As authors of constructionist tools and learning designs we faced the challenge of giving visitors to the Royal Society Summer Science Exhibition an authentic constructionist experience when most visitors were expected to stay only a few minutes. Furthermore we needed to help the visitors learn something about the subject matter of the exhibit – the spread of viruses.

To meet these challenges we built and exhibited the *Epidemic Game Maker* (http://m.modelling4all.org/p/en/sse.html). Here we report on the design of the Epidemic Game Maker and our impressions of the more than one thousand visitors who created games during the exhibition.

Keywords

Agent-based modelling, Behaviour Composer, NetLogo, Epidemic Game Maker

The Epidemic Game Maker

We designed the Epidemic Game Maker (EGM) so that users with no computer programming knowledge, little computer experience, and the bare minimum knowledge about how infections spread can

- construct and run epidemic models in a minute or two
- construct and play increasingly complex epidemic games in five to fifteen minutes
- construct games that help them learn about the dynamics of epidemics and the trade-offs of different public health interventions
- can continue to play and share their games at home or school
- learn about computer modelling
- have fun

To address the first requirement we provide a ready-made model that consists of a population of students who travel daily from home to school and back. Initially one student is infected. When

an infected person is at the same location as a susceptible person then with specified odds the infection is transmitted. People recover from infections after a specified amount of time. This is based upon the standard susceptible-infected-recovered model of epidemics (Scherer & McLean 2002). The opening screen encourages them to run the model (as seen in Figure 1).

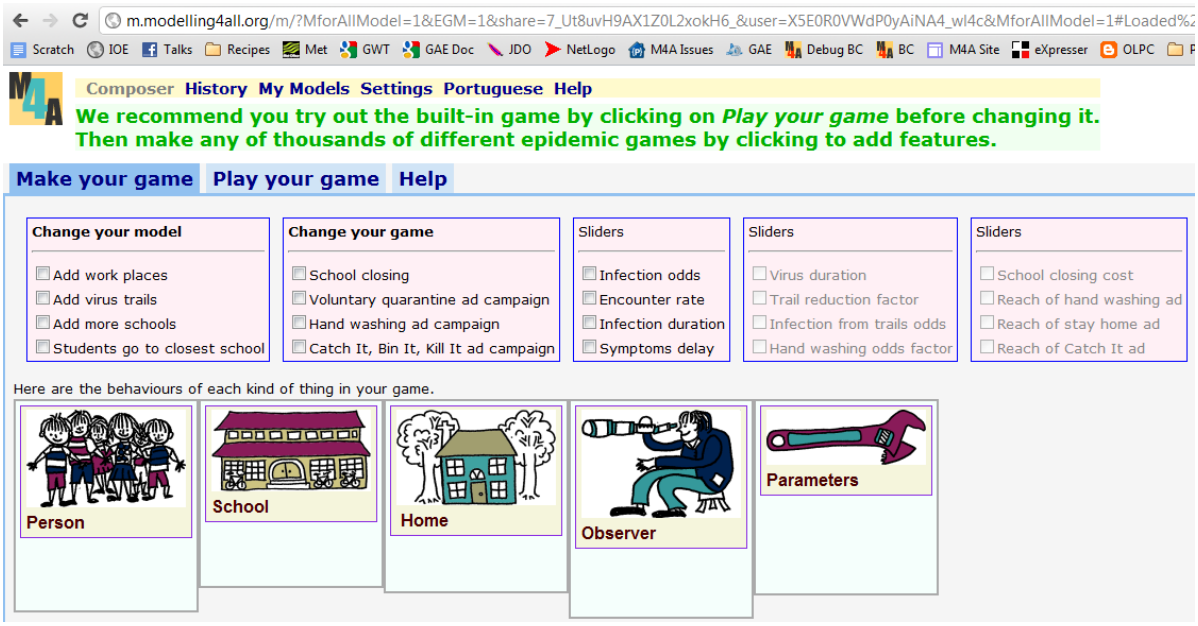

Figure 1 – Initial 'home page' of the Epidemic Game Maker

(the check boxes are explained below)

After clicking on the 'Play your game' tab, the Java applet for the game is loaded into the user's browser.

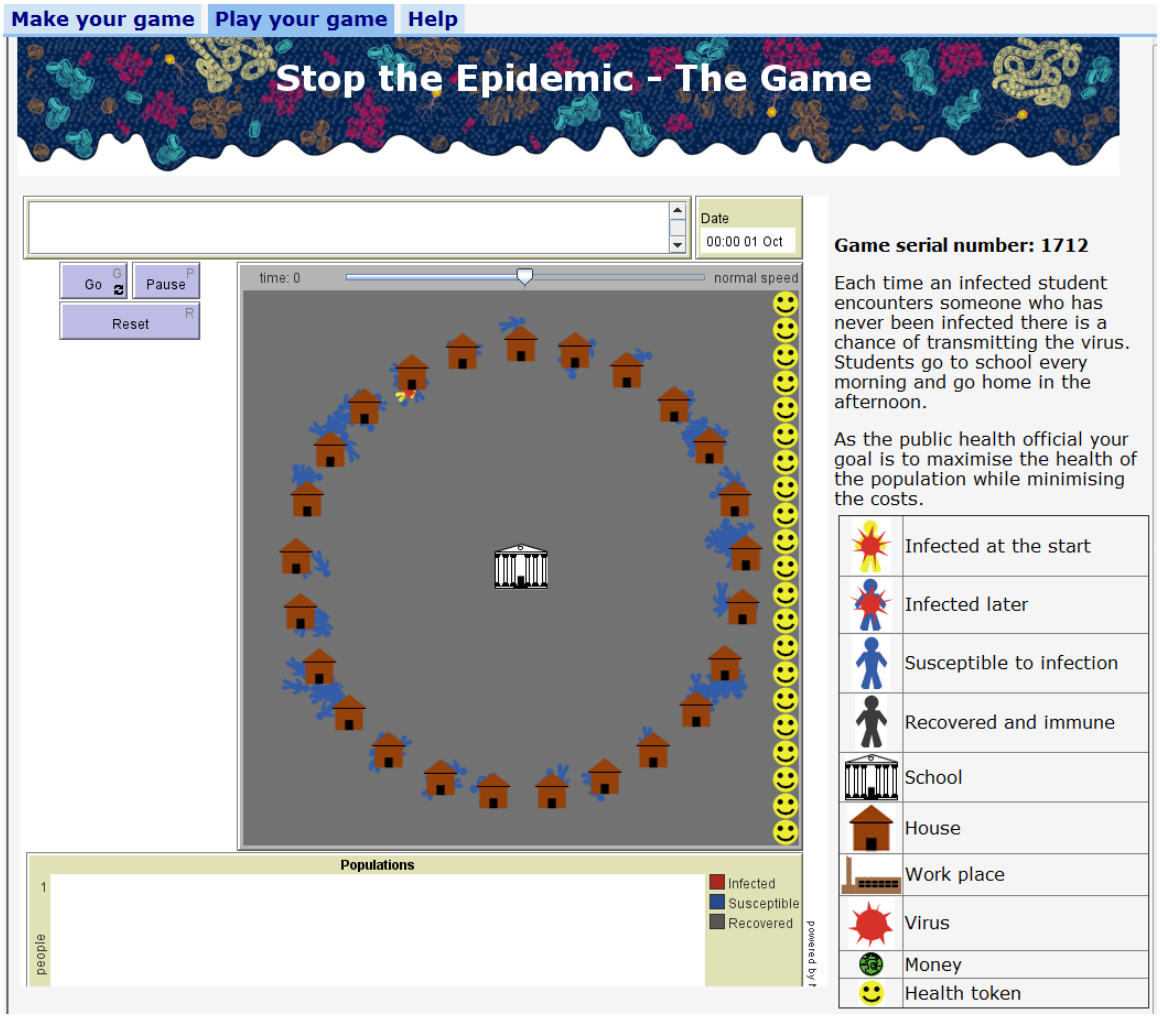

Figure 2 - Initial base game screen

There are two kinds of simulation games: (1) simulations where the player or players are among the individuals being simulated (e.g. controlling a fish that is part of a simulated school of fish) and (2) simulations where the player has global control (often called 'god games'). The EGM can be used to make pure models or models with game elements. With minimal settings the EGM makes a simple epidemic model that lacks game play elements. At the top of the interface there is an area for messages and a display of the simulated time. Buttons are available to run, pause, or reset the simulation. The world is portrayed as a circle of houses with a school in the centre. Students are at home and one is infected (in Figure 2 the infected student is at approximately 11 o'clock). The smiley faces act as a health gauge. At the bottom graphs of the infected, susceptible, and recovered populations are drawn.

The model can be run multiple times and due to the stochastic nature of the transmission of infections each run is different (as seen in Figure 3).

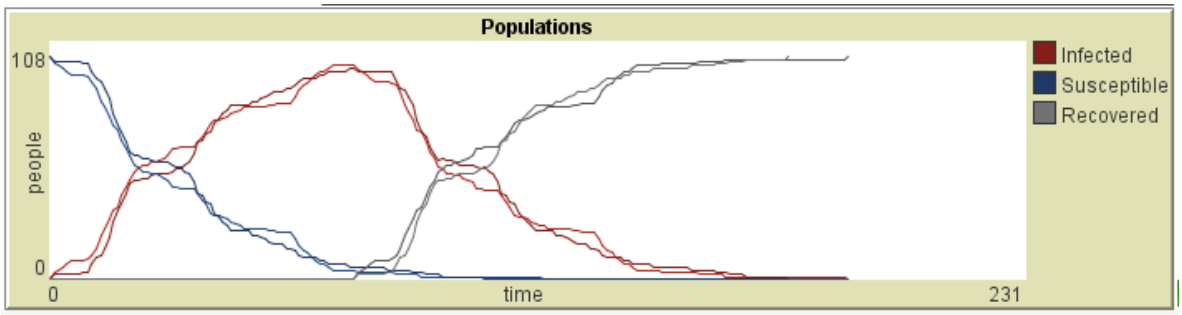

Figure 3 - A typical graph showing two runs

The constructionist aspect of the Epidemic Game Maker is supported by tick boxes that add or remove model and game enhancements as well as fine control over parameters. A user specifies their game design by ticking a subset of the twenty check boxes (as seen in Figure 1). If the user adds buttons or sliders to the game, then the consequences of these can be explored within the NetLogo applet.

The model changes supported are:

- **Add work places** – Introduces adults that regularly commute to work
- **Add virus trails** – Viruses that can infect people are left behind by infected people
- **Add more schools** – Adds three more schools
- **Students go to closest school** – Students go to the school closest to their home

Users can explore and discover interesting interactions between these enhancements. Adding more schools and requiring that students go to the local school limits the epidemic to a single school. Introducing adults then provides a means for an infection to jump from school to school (by a student infecting a parent at home who infects a co-worker from another neighbourhood who infects one of their children). This pattern is similar to how a disease can spread between countries via air travellers.

Users can also add buttons to introduce game elements that enable players to try to stop the epidemic. The supported enhancements are

- **School closing** – When schools are closed students stay home. Keeping schools closed has societal costs and they will be automatically re-opened if funds run out.
- **Voluntary quarantine ad campaign** – Every button click reaches a specified proportion of the population to stay home once they are aware that they are infected.
- **Hand washing ad campaign** - Every button click reaches a specified proportion of the population causing them to wash their hands frequently thereby reducing the odds of acquiring the infection from viruses left behind.
- **Catch It, Bin It, Kill It ad campaign** - Every button click reaches a specified proportion of the population to use tissues to reduce the trail of viruses they leave behind.

Each run of an ad reduces the remaining budget. The latter two only make sense if '**Add virus trails**' has been added to the model. All of these enhancements can spark good discussions. Does school closing really cause children to stay home? What if the model was enhanced with shopping centres where the students might go when school is closed? A '**Hand washing ad campaign'** encourages people to act in their own best self-interest to reduce their odds becoming ill while the '**Catch It, Bin It, Kill It ad campaign**' encourages people to stop harming others.

The effects of all of these enhancements depend upon the values of model parameters. Users can choose to add sliders to explore the consequences of different values. The sliders they can add are

- Infection odds
- Encounter rate
- Infection duration
- Symptoms delay
- Virus duration
- Trail reduction factor
- Infection from trails odds
- Hand washing odds factor
- School closing cost
- Reach of hand washing ad
- Reach of stay home ad
- Reach of Catch It, Bin It, Kill It ad

To enable visitors to continue to play and enhance their game the EGM generates a four-digit serial number for each game. We gave each visitor a nicely printed card where they could write down their serial number. The card included the project URL (modelling4all.org) where they can enter their serial number to restore all of their settings.

The Epidemic Game Maker has a check box to switch to *Advanced Mode* where the underlying behaviours of the model become visible and editable (as seen in Figure 4).. The full features of the Behaviour Composer including the ability to add any NetLogo code then become available for those interested in delving deeper than what a short visit to an exhibition can support.

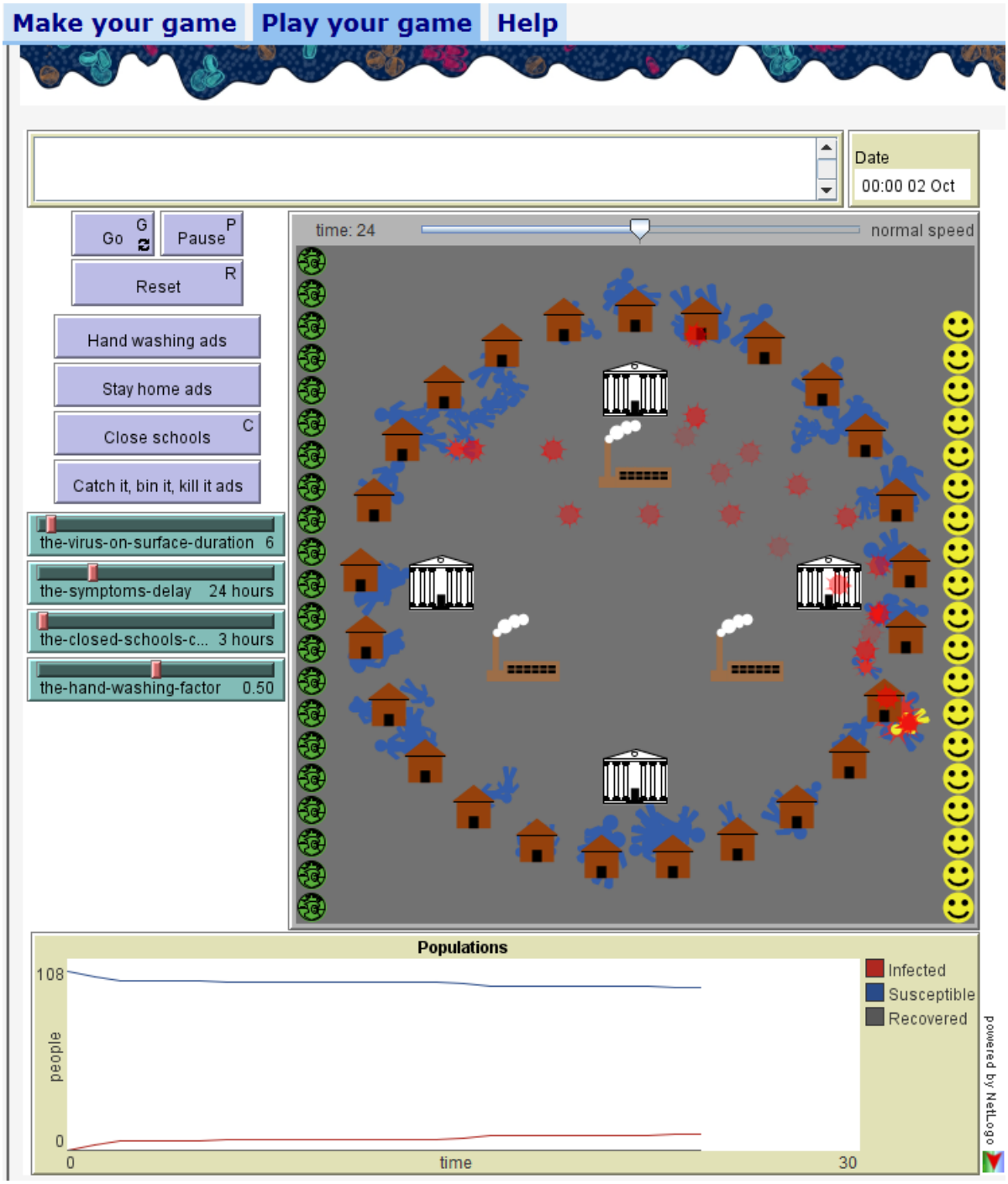

Figure 4 – A game with all enhancements and four sliders

The Royal Society Summer Science Exhibition

The Royal Society holds the United Kingdom's most prestigious science exhibition every summer. We participated in the 2010 event (http://seefurtherfestival.org/) which was expanded to celebrate the 350th anniversary of the Royal Society. Fifty thousand people visited the exhibition over ten days. This included almost two thousand registered students and 240 teachers. 1750 VIPs (including the Queen of England) came to a closed showing. The EGM was part of the exhibit on 'Emerging infections: viruses that come in from the wild'

(http://seefurtherfestival.org/exhibition/view/emerging-infections-viruses-come-wild).

The main aim of the Royal Society event was to provide the general public with an opportunity to speak to real scientists about their field of research. We were asked to focus primarily on 12-16 year olds but to expect people of all ages. The EGM allowed us to discuss both the factors that determine the spread of a virus and how computer models might contribute to our understanding.

Experiences of a thousand visitors ranging from young children to fellows of the Royal Society

During the exhibit over 1100 games were created using the EGM by approximately a thousand visitors (many games were made by small groups). In the two weeks following the exhibition 200 games were made by on-line visitors. It was noticeable that people varied in terms of how willing they were to experiment with the model as opposed to discuss epidemics with members of the Oxford team hosting the stand. As the exhibition continued we arrived at ways to structure conversation so that:

- Those who were quick to centre their attention on experimenting with the EGM could be encouraged to generate ideas for improving the model
- Those who preferred to discuss epidemics orally would turn to the EGM to test the assumptions they held in their head.

In both cases the art of conversation was to arrive at a point with each visitor gained a healthy level of scepticism about computer modelling i.e. they should neither treat the model as a black box for predicting the future, nor just as simply a toy that did not match the *common sense_model they held in their head*.

In this way we tried to give each visitor the intuition that:

- Computer models of epidemics are useful because they augment our ability to think about systems that are too complex to think about in our heads alone.
- All models of epidemics are built by making very important assumptions about how people will behave in a given situation, and many assumptions they might think are important could have been left out for the sake of simplicity.

Some nice examples that we feel demonstrate a positive outcome from using the EGM:

- Many visitors were very quick to point out that in the middle of London it is very important to model how people move between home, work and school i.e. whether they walked, cycled, took the bus, tube or boat.
- Particularly teenage girls were quick to point out that any public health intervention that focussed on improved personal hygiene (washing hands) would be most effective if targeted at boys.
- Most teenagers who engaged with the model were adamant that a stay-at-home policy would never work – teenagers would instead congregate at someone's house or a park. (They correctly pointed out that this would likely negate any gains from closing the school).
- Some young children (6-10 year-olds) built a series of games and demonstrated a clear understanding of the model and the interventions. Younger children enjoyed playing with the game – a few played for more than ten minutes.
- Many people questioned the assumptions built into the model pertaining to the cost of closing schools and workplaces to the economy.
- A few learners asked to see the underlying code. One "improved" the model by adding one

of the generic Behaviour Composer micro-behaviours "wander randomly" to a school. This caused the school to move around on the screen, and all its pupils to chase around after the school. While this is not directly related to learning about epidemiology, we felt it was an interesting example of a learner opening up the black box and seeing the model as something malleable that he could do with whatever he pleased.

Of course we would have loved to have spent more time with people to support them in building their own assumptions and ideas into an enhanced version of the EGM. In the few brief minutes we had with visitors we hope they were left with at least a latent desire to do so one day.

Theoretical underpinnings

We wanted the EGM to work as an object to think *with* for the learners when engaging with new knowledge of epidemics. It was important to us that the EGM would facilitate thinking and conversation about epidemics that learners could relate meaningfully to their own lived lives.

Epidemiological modelling is largely taught using variations of the SIR (Susceptible-Infected-Recovered) model, using sets of deterministic differential equations that predict the relationship between changes in the three categories of people. Because of the use of differential equations, the conventional SIR model requires a high level of knowledge of calculus, and is typically not taught until the undergraduate level. However, using Agent Based Modelling as a restructuration (Wilensky & Papert, 2010), we were able to meaningfully engage young learners in discussing and thinking about this complex issue.

(Wilensky & Resnick, 2006) argue that Agent Based Modelling can facilitate an 'embodied modelling approach' by asking learners to think like the agents they are modelling. By doing so they are able to break down the behaviours of agents into bits of code and generate computational theories about the relationships between them. We took this idea as a starting point for the design of the EGM. By focusing on the lives of our learners, the main question that we hoped to engage our learners in was, "Given this situation or policy or intervention, is this what *you* would do?" As we demonstrated above in the examples of learning with the EGM, those were exactly the questions that people asked of the EGM in order to both make sense of the model, and to offer informed critiques of it. For instance, in the example of the stay-at-home policy not being effective, the main critique offered by learners was that they would in fact not stay at home, but go and hang out with their friends. This critique was drawn from learners' own lives, and illustrated to us that learners were able to engage with this complex issue in a deep and critical manner.

Finally, we wanted to engage learners in thinking about the relationships between their own and their family's behaviours; the properties of the virus; the policies enacted; and the aggregate outcome of the epidemic model. (Wilensky & Resnick, 1999) argue for an 'emergent view' of levels in complex systems in which aggregate outcomes appear at 'levels' that emerge out of the complex interactions between agents. By allowing learners to add or remove complexities (number of schools, workplaces, virus trails, etc.) our hope was that learners would experience these emergent levels and see that what happens at the aggregate level in the model is not a hard coded, black box phenomenon, but simply the result of the complexities that learners added to the model.

Building other 'Game Makers' in the Behaviour Composer

The Epidemic Game Maker was implemented by building it on top of the Behaviour Composer. This approach provides a smooth integration but requires a deep understanding of the

implementation of the Behaviour Composer. Subsequently we have enhanced the Behaviour Composer so that an 'end user' could build something like the EGM without touching the source code of the Behaviour Composer.

To create a different game maker (or a model maker) one starts by creating the base game or model in the Behaviour Composer. From the 'share' tab one can obtain a URL that will load the model. One then authors and hosts an ordinary HTML web page. The model URL needs to be enhanced to include '&tab=…' where the URL to the web page is added. To add check boxes the page should contain the following text for each check box:

```
Begin Replay Session Events Check Box:
ID: a name for the check box
Label: the label associated with the check box
Session ID: a session ID described below
Do message: a message displayed when checked
Undo message: a message displayed when unchecked
Title: the title displayed when the mouse hovers over the check box
End Replay Session Events Check Box
```

Figure 5 – Mark up code needed to generate a session replay check box

When this page is loaded into the Behaviour Composer all this text is removed and replaced by a check box. The session ID of a check box is obtained by loading the base model into the Behaviour Composer and then making changes to the model. The system provides a session ID that can be used to recreate the changes performed. By copying and pasting the ID into the check box form the check box when ticked will replay all the changes and unticking it will remove the changes. The rest of the web page can provide background and instructions.

Conclusions

Although our experience with the EGM lasted for just some minutes with each participant we think that the game, together with the Behaviour Composer environment, open many possibilities to teachers and students to explore models and simulations in educational settings integrating different disciplines such as biology, mathematics and social science.

The Logo family of languages, including NetLogo and Scratch (Resnick et. al., 2009), aspire to have a low threshold so users can begin to build without a large upfront investment to acquire the prerequisite knowledge. However, in the context of an exhibit the threshold needs to be a few seconds of instruction. For turtle programming there have been near-zero threshold implementations called 'Instant Logo' or 'single-key Logo' since the mid-1970s (Goldenberg, 1974; Solomon & Papert, 1975). In the same period Radia Perlman pioneered special hardware interfaces for pre-school children to program turtles (Morgado et. al., 2006).

Agent-based modelling is significantly more complex than turtle programming. NetLogo, for example, extends turtle programming with agents, agent sets, links, patches, and much more. The author of a game maker such the EGM will need much of this richness. The challenge is how to (temporarily) hide this complexity from users. The approach presented here is to provide a simple or partial model and to provide a user interface that automates the addition of several major model enhancements. Unlike 'Instant Logo' we are not providing the user with generic primitives upon which to build but instead high-level domain-specific components. We are attempting to support 'middle-up' programming rather than 'bottom up'. The low level code (i.e., NetLogo) is

available and accessible. Users are encouraged, but not required, to deal with it.

Program and source code availability

The EGM is freely available from http://m.modelling4all.org/p/en/sse.html. The Behaviour Composer is available from http://m.modelling4all.org. The source code is available at http://code.google.com/p/modelling4all/source/checkout.

References

Goldenberg, P. (1974) FASTR – A Simple Turtle Runner. Logo Working Paper No. 30, MIT AI Lab, Massachusetts Institute of Technology, Cambridge, MA, USA.

Morgado, L., Cruz, M., & Kahn, K. (2006). Radia Perlman – A pioneer of young children computer programming. *Current Developments in Technology-Assisted Education*: 1903–1908.

Kahn, K. & Noble, H., (2010). The Modelling4All Project — A web-based modelling tool embedded in Web 2.0. J. Clayson & I. Kallas (Eds.), *Proceedings of the Constructionism 2010 Conference*. Paris, France.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, November 2009.

Scherer A. & McLean A., (2002) Mathematical models of vaccination, *British Medical Bulletin* 2002;62 187-199.

Solomon, C. & Papert, S., (1975) Teach: A Step Toward More Interactive Programming. Logo Working Paper No. 43, MIT AI Lab, Massachusetts Institute of Technology, Cambridge, MA, USA.

Wilensky, U & Rand,W. (in press). *An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo*. Cambridge, MA: MIT Press.

Wilensky, U. & S. Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. J. Clayson & I. Kallas (Eds.), *Proceedings of the Constructionism 2010 Conference*. Paris, France

Wilensky, U. & Reisman, K. (2006) Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and Instruction* 24, no. 2 (2006): 171–209.

Wilensky, U. & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology* 8, no. 1 (1999): 3–19.