Making Educational Games That Work in The Classroom

A New Approach for Integrating STEM Simulations

Alexander Repenning
University of Colorado Boulder
Department of Computer Science
Boulder CO. 80303
ralex@cs.colorado.edu

Ashok Basawapatna
University of Colorado Boulder
Department of Computer Science
Boulder CO. 80303
basawapa@colorado.edu

Michael Klymkowsky
University of Colorado Boulder
Department of MCDB
Boulder CO. 80303
michael.klymkowsky@colorado.edu

***Abstract*—The development of analytical skills is a central goal of the Next Generation Science Standards and foundational to subject mastery in STEM fields. Yet, significant barriers exist to students gaining such skills. Here we describe a new "gentle-slope" cyberlearning strategy that gradually introduces students to the authoring of scientific simulations via a Web-based modding approach called CyberMOD. Modding involves adding agents with predefined functionality to a simulation world to produce a unique combination whose behavior can then be visualized by running the simulation. This permits low barrier experimentation on the modded simulation, which is hoped to help students gain a deeper understanding of scientific phenomena that is the focus of the activity. Our research questions are: i) does this approach encourage students' interest in computational science and ii) does it enhance their analytical abilities, and iii) does it foster a deeper understanding of the processes modded? Here we take an in-depth look at what was created for the CyberMOD infrastructure and analyze the results of initial in-class studies as to the effectiveness of this strategy. The results support the premise that teachers can easily integrate CyberMOD into their in-class activities, that CyberMOD activities encourage creative student learning, and that the CyberMOD approach facilitates student understanding.**

Keywords—efficacy of educational games; modding; CyberMOD; STEM Education; Next Generation Science Standards; effect size;

I. Introduction

The need to increase the participation and performance of American students in STEM fields is well known. The crisis is real and quantifiable, with international Programme for International Student Assessment (PISA) studies pointing to the relatively poor math and science literacy of U.S. students. At the same time, it is becoming increasingly clear that students headed into STEM fields need to be computationally literate [1]. The recent President's Information Technology Advisory Committee Report (PITAC) report [2] recommends introducing computationally-based concepts (and skills) through simulation type activities, to encourage a stronger analytical science education pipeline from middle to graduate school. The use of simulations has slowly found its way into state science standards such as the Common Core Curriculum Standards (CCC), which are currently being implemented at a national level. Even more promising are the emerging Next Generation Science Standards (NGSS) [4], which call on simulations and computational thinking to be an an essential concept of STEM education [5, 6]. For example, the Structure, Function, and Information Processing section of the NGSS recommends that in grades 9-12 "students who demonstrate understanding can use modeling to explain the function of positive and negative feedback mechanisms in maintaining homeostasis that is essential for organisms [4]." Similarly, the CCC standards state that students should "recognize significant variables and the relationships among them [7]" through the use of simulations. Others agree [2, 3], but few K–12 programs that have attempted to integrate simulations have truly flourished – mainly due to lack of suitable curriculum, tools, teacher training, and time. With rapidly increasing interest and emerging curriculum support of simulation and modeling activities, the main question becomes one of how these activities can become more effectively introduced into K-12 education.

Over the years our research group have acquired experience in developing and using educational software in schools. These include teaching and authoring tools aimed at math, science, social studies, language arts, and computer science. Much of this experience involves students using the end-user programming tool AgentSheets, and its more recent 3D version AgentCubes, to create both games and simulations. For example, elementary school science students used AgentSheets software to create ecosystem models [8]; this involved having them create their own animals and placing them in a shared environment to create food webs. High School science students used Mr. Vetro [9], a collective simulation of a human being, to learn about physiology. In history classes, high school students used AgentSheets to create simulations of protest movements in American history and of the effects of the automobile on 20th century society [10]. Using Scalable Game Design, middle school students learned computational thinking through scaffolded game design and STEM modules. Evidence from these project points to increased motivation and enhanced participation of underrepresented communities in both computing education and STEM [11-12].

Activities related to designing and building simulations offer a unique opportunity to integrate STEM -center analytical thinking and computational tools into the classroom environment. Simulation design and authoring enables classes to meet upcoming science standards and effectively integrate STEM simulation activities into their classrooms. At present time, however, very few U.S. teachers are employing (or know how to employ) simulation or modeling activities. A study by the U.S. Department of Education, entitled Teachers Use Of Educational Technology In U.S. Public Schools, reported statistics for computer, math and science classes and found that only a quarter of all computer science/math/science classes tried to expose their students to modeling exercises more than once and half of these classes never tried a modeling exercise at all [13]. A recent report the National Academies [14] focusing on the topic of games and simulations, concludes:

> "There is moderate evidence that simulations motivate students' interest in science and science learning, and less evidence about whether they support other science learning goals.
>
> Evidence for the effectiveness of games for supporting science learning is emerging but is currently inconclusive. To date, the research base is very limited."

Two factors limit the implementation of modeling in schools. The first is the the lack of appropriate and accessible activities, such as tools that enable students to create the simulation from scratch [15]. The second, and perhaps more important major barrier, is finding the time needed (within what is often an already over-crowded schedule). A teacher can have students experiment on a pre-made simulation associated with the material, and while this is time-efficient, it is essentially passive. The analytic process involved in creating a simulation, including deciding on which parameters to include and which to ignore, is done for the students - the students simply "consume" the simulation. So while, in our experiences high school students can take over a week to learn an end-user programming tool and subsequently, create a simulation [16], this time (with the appropriate modeling tools) is necessary to allow students to experiment on and model the real world phenomenon being studied. That amount of time is often too prohibitive for teachers, especially in a class that does not take place in a computer lab. Given enough time and resources, however, end-user programming of simulations allows students to express ideas both analytically and computationally, and then to run a model, visualize the consequences and revise their model based its observed behavior -- a feedback loop common in professional practice, and likely to be necessary to consolidate computation skills.

What seems to be needed (or rather desirable) are better points of trade off between pre-made simulations and authoring simulations from scratch. Figure 1 depicts these alternatives in a consumer <=> creator continuum followed by a brief description of each activity.

- ***Animation***: Video narratives that users watch to gain insight into phenomena.
- ***Interactive Simulation***: pre-made simulations which allow users to alter parameters and run experiments.
- ***Collective Simulation***: Interactive simulations including social components involving role play and peer learning.

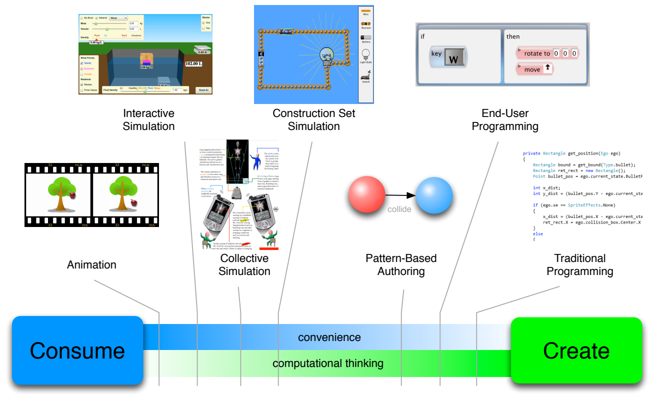

Fig. 1. The Consume Create Continuum

- ***Construction Set Simulation***: Users configure simulations employing palettes of pre-programmed components.
- ***Pattern-Based Authoring*** [17]: Users author behaviors of simulation components employing high level interaction pattern such as collision.
- ***End-User Programming***: Users employ high level visual and/or drag and drop programming languages to author simulations.
- ***Traditional Programming***: Users employ languages such as C++ to create simulations from scratch.

It should be noted that this is by no means an exhaustive list but serves as a good beginning point for a discussion of simulation use in the classroom environment. Starting from the consumer and going to the creator, we find activities that enable higher degrees of expressive freedom. Furthermore, with each item there exists limitations that begin to be overcome by traveling to activities located on the right of the diagram. For example, one way an animation is limited is that the user is presented with a narrative relating to a phenomenon, however the user cannot manipulate the animation in order to gain insight. Giving the user an interactive simulation instead eliminates this barrier and provides more freedom in exploring this phenomenon. Conversely, for activities to the right, there is always the the option of including activities to the left. For example, by using a programming language, like C++ or Java, the (expert) user has the tools needed to do everything offered through end-user programming all the way to creating a simulation animation that can be displayed to enhance student understanding. The novice user, however, is more likely to be discouraged.

"Consumer" activities have time efficient implementations in the classroom environment. Strategies towards the creator end of the spectrum enable a greater degree of freedom enabling the user to model a greater variety of real world phenomena. Regardless of which activity used on this spectrum, it must be effective in the in-class environment. Our research has identified some fundamental design principles that enable a simulation related activity to be effectively integrated in to the classroom environment:

1. ***Efficient use of class time***: For instance programming a simulation from scratch could take a long time without much disciplinary learning benefit (except perhaps in learning to program).

2. ***Increases student understanding***: There should be a measurable educational benefit relevant to the subject topic.
3. ***Fosters student creativity***: Support ownership and freedom to enable creative and engaging investigation.

Here we focus on the idea of a Construction Set Simulation, entitled CyberMOD, as a well working combination of these three design principles. The following sections will describe the CyberMOD approach, outline the methodology employed, and present results regarding the efficacy of the CyberMOD construction set simulations.

II. CyberMOD

We developed CyberMOD[1] to substantially reduce the barrier to entry for introducing scientific modeling into public schools; it provides a "gentle-slope" [19] introduction to authoring through modding. In general, modding refers to modifying a piece of software to perform functions not originally conceived or intended by the designer. The term modding is often used in the Open Source software movement and the computer game community to refer to creating new or altered content and sharing that content via the web. Modding can be a powerful educational approach [20] and a new way to express scientific principles by reusing existing models, modifying them, and gradually building on top of them.

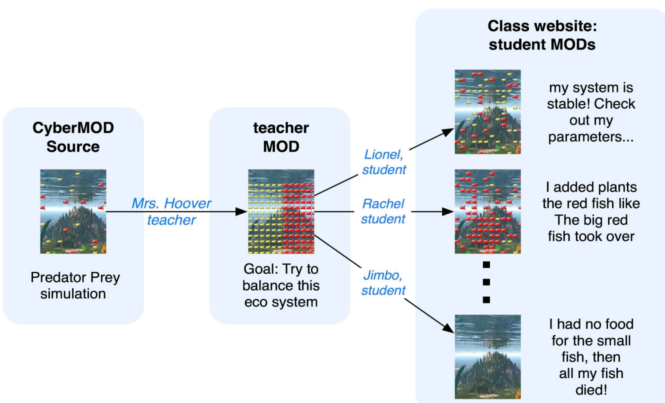

Fig. 2. Ms. Hoover mods an ecology simulation found in the CyberMOD Source repository, with predators feeding on prey, which in turn feed on plants. All species reproduce according to their own rules and rates. She sends the modded version to her students who run it and try to balance the system by modding design aspects of the simulation (changing parameters e.g. reproduction rates, starting populations). The resulting mods include student information and explanations and are added to the repository for Ms. Hoover and others to access.

By modding, users modify a pre-made simulation by adding different configurations of pre-existing agents with pre-determined behaviors. Figure 2 provides a modding example. CyberMOD allows the teacher to create or update a simulation source file to a classroom's purposes. Students use the CyberMOD infrastructure to clone their projects from the teacher's parent project. At this point, students add agents to the cloned simulation to explore a particular in-class concept. Each student's simulation becomes a unique interpretation of the teacher's parent project. Different agent configurations can have different properties associated with them and by exploring these configurations with their classmates, students can get a better understanding of tradeoffs associated with the system. Students can describe their particular modding choices using CyberMOD's social media infrastructure and even clone a fellow student's project to mod further. In this respect, CyberMOD obviates the need for students to author simulations from scratch by giving them the ability to customize simulations in novel ways.

CyberMOD includes web-based modding tools, scientific simulations, curriculum units that can be seamlessly integrated in high school Biology classes, and teacher training.

The ability to mod simulations is not meant to completely replace authoring from scratch. Authoring is necessary to create fundamentally new behaviors. The CyberMOD gentle-slope authoring approach lowers the threshold for adopting computational science tools and modeling in K–12 education, especially in time-limited science classrooms. The ability to not only use, but also design simulations through a gentle-slope interface can ultimately lead to an important tipping point between consuming and creating scientific models. When students have an opportunity to explore the structure of a simulation as designers, they are more apt to understand the key concepts of the simulation—as they are required to investigate how changes in probabilities and parameters affect the simulation [5]. Through the process of authoring, student inquiry and self-assessment are authentically introduced in the instructional process. The notion of externalization, product creation, and feedback can be highly motivational and is consistent with the constructivist learning theory [21] and the design of authentic learning environments [22]. While simulation authoring in computational science based science classes has been successful [10], most teachers and schools using this approach can probably be characterized as early adopters. To overcome barrier to adoption, CyberMOD makes authoring more accessible and motivational and supports a culture of modification.

III. Methodology

A team was assembled to create the construction set simulation to be used in two different high school biology classes. This team consisted of a Biologist, two high school biology teachers, and a Computer Scientist and a education researcher. This team decided to create two simulations relating to the idea of homeostasis and negative feedback: A House Thermoregulation Simulation and a Blood Sugar Regulation Simulation. The House Thermoregulation Simulation serves as an introduction to the fundamentals of regulatory systems, and the Blood Sugar Regulation Simulation allows students to apply these fundamentals to a commonly taught human homeostatic system.

[1] The CyberMOD simulation can be found at: http://scalablegamedesign.cs.colorado.edu/arcade/?q=search/content/Blood%20Glucose

Housing Thermoregulation Simulation

The Housing Thermoregulation simulation is a simple introduction to negative feedback systems. The following figure depicts the general feedback mechanisms involved in the Housing Thermoregulation system.

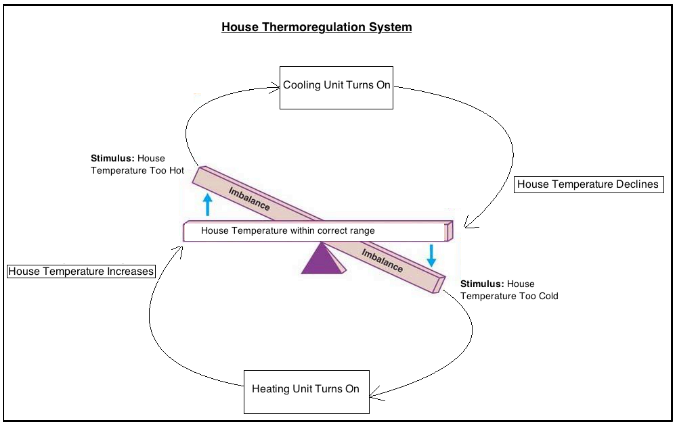

Fig. 3. The general negative feedback mechanism in the housing thermoregulation system.

Essentially, if the housing temperature increases past a given set point (due to changes in outside temperature) cooling units turn on, decreasing the house temperature; conversely if the house temperature drops, heating units turn on increasing the house temperature.

The following is a diagram of the CyberMOD Housing Thermoregulation simulation with a description of the agents involved (Fig. 4).

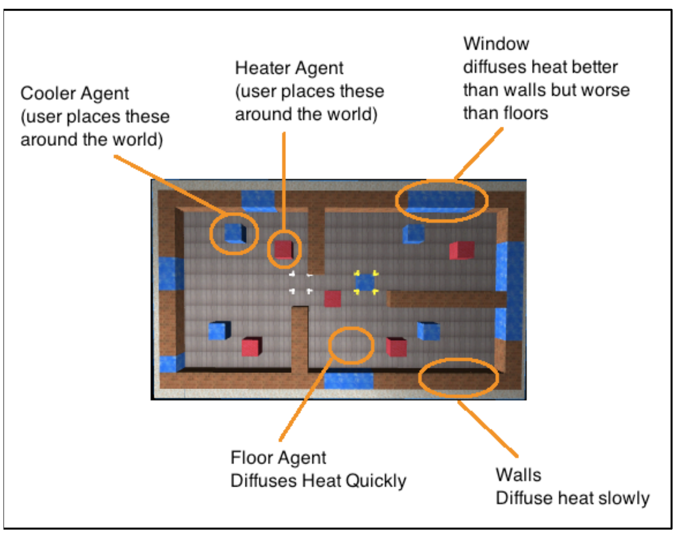

Fig. 4. The CyberMOD Thermoregulation Simulation

The user places Heater and Cooler agents around the level with the goal of keeping the temperature inside the house within a given temperature range. Each heating and cooling element has a defined set point within the simulation. After placing heaters and coolers, users can start a weather simulation which tests how their specific set of agents regulates the housing temperature in response to outside temperature changes. The following (Fig. 5) depicts a run of the Housing Thermoregulation simulation in CyberMOD.

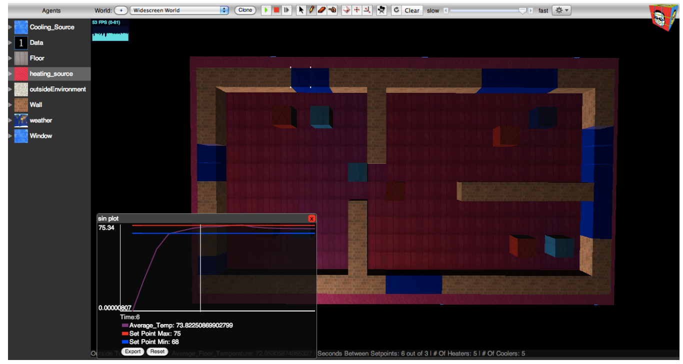

Fig. 5. CyberMOD Housing Thermoregulation System Run

We see some blue coolers and red heaters placed within a level. The Figure also depicts a graph of the current average interior housing temperature (in purple), the maximum set point (in red), and the minimum set point (in blue) for the system. The floor of the house ranges from blue in color, indicating a cold temperature point in the house, to red, indicating a hot temperature point in the house.. The number and placement of heaters and coolers determine how well the house temperature stays within the range of minimum and maximum set points given a change in outside temperature.

Blood Glucose Regulation Simulation

The Housing Thermoregulation simulation serves as an introduction to regulatory systems. The Blood Glucose Regulation simulation aims to have users apply the similar concepts to an example of physiological homeostasis. When the blood sugar level rises above a certain concentration the pancreas releases insulin which stimulates the body tissue and liver to absorb glucose from the blood stream, thereby lowering the level of sugar; conversely, when the blood sugar level drops below a certain concentration range the pancreas releases glucagon which stimulates the liver to start producing glucose, which is released into the bloodstream, which raise the presence of sugar in the bloodstream.

The following depicts a diagram of the CyberMOD Blood Sugar Regulation simulation with a description of the agents involved.

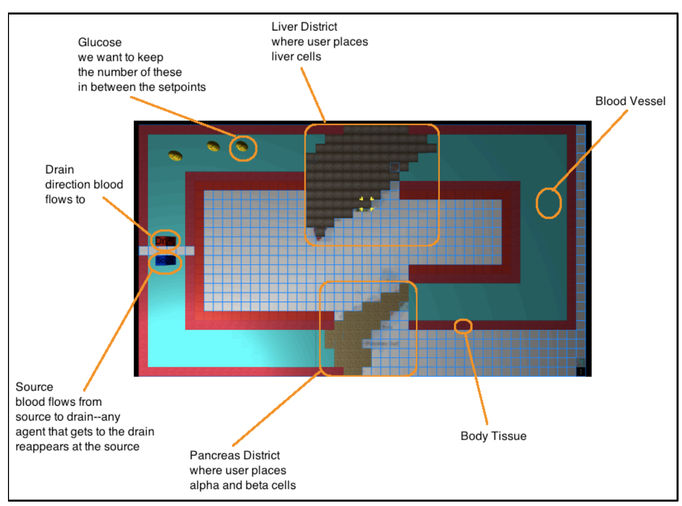

Figure 6: The CyberMOD Thermoregulation Simulation

The user places Pancreatic Alpha (glucagon-secreting) and Beta (insulin-secreting) Cell agents in the Pancreatic District in Figure 6 and Liver agents in the Liver District. Finally the user connects a Blood Vessel agent through the Pancreas and Liver District such that it runs adjacent to the Pancreas and Liver cells. The following diagram depicts the Blood Glucose Regulation simulation with Pancreatic Alpha and Beta cells and Liver cells placed within their respective regions and blood vessels connected through the Liver and Pancreas District.

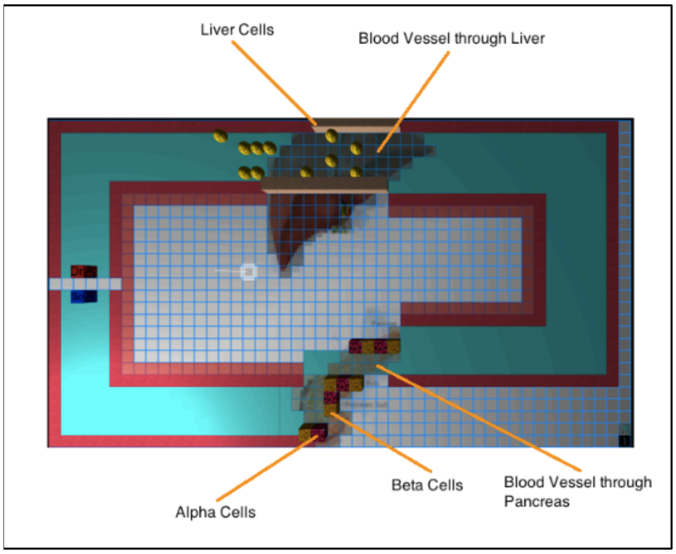

Fig. 7. The CyberMOD Thermoregulation Simulation with user-placed Liver and Pancreatic Cell agents

The following picture depicts a student running the Blood Glucose Regulation simulation in the CyberMOD environment.

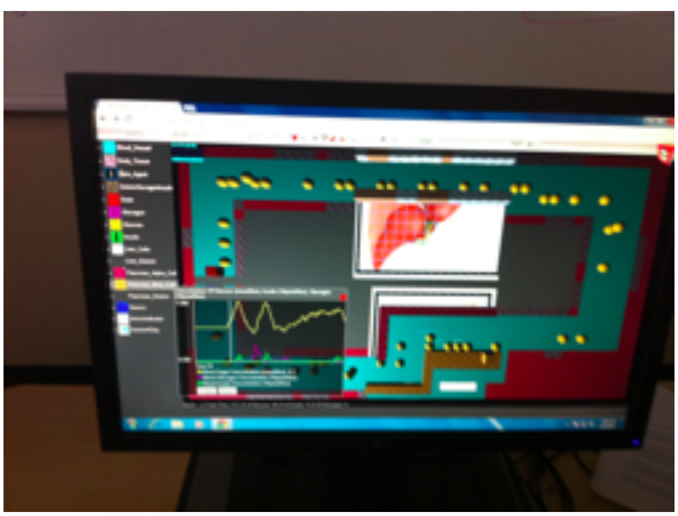

Fig. 8. A student in Kate H.'s N. High School class modding the Blood Glucose Regulation simulation.

A graph depicts the current blood glucose concentration (in yellow), the insulin concentration (in green), and the glucagon concentration (in purple). By noticing the spikes in the purple and green lines on the graph in Figure 9, we can begin to see when insulin and glucagon are released in relation to glucose concentration. Specifically the graph shows that as blood glucose concentration rises above the maximum set point, insulin is released and as blood glucose concentration drops below the minimum set point, glucagon is released. Students can actually watch Insulin and Glucagon agents interact with Body Tissue and Liver Cell agents during simulation runs. The following figure depicts a student mod of the Blood Glucose Regulation simulation with Glucagon agents present; Glucagon agents are represented as purple squares.

Fig. 9. A student modded Blood Glucose Regulation simulation with purple Glucagon agents present (top right). In the background other students are creating their own mods of the simulation.

By placing agents in different configurations and amounts of agents in the simulation, students can see the differences between having, for example, less Pancreatic Beta cells or Liver Cells. Furthermore, students can simulate eating a high sugar meal, a medium sugar meal, and a low sugar meal by pressing various keyboard keys. Similarly, by hitting a key, users can also simulate giving an insulin injection. Finally, users can hit a key to simulate eating 8 different randomly selected meals over time, of various sugar content, to see how their system performs.

CyberMOD study

The CyberMOD study consisted of 5 classes in 2 different schools: N. High School and M. High School. Kate H., teacher at N. High School, ran the study with her block schedule General High School Biology Class. Kristin D., teacher at M. High School, ran the study in 3 AP Biology classes. Each study took between 2-3 hours of class time not including the pre– and post–tests, which took students around 20-30 minutes to complete. Data was taken from 71 students over the course of the study. Data collected from the study included the following:

- Answers to Pre/Post test questions
- Focus group interviews with students from both schools
- Informal observations of teachers and students, stories, comments and screenshots of simulations as students completed the units.
- RTOP (Reformed Teaching Observation Protocol) used to assess how the teachers taught the unit in-class

- Actual simulations created by students

In this paper we focus on a subset of the pre and post test questions, the actual simulations students created, and the results of the focus group interviews to see whether this brief two-day in-class intervention enhanced student understanding as well as fostered student creativity.

IV. Results

We present the results of the CyberMOD project in terms of the previously outlines design principles. As mentioned above the CyberMOD project took only two hours of class time, one hour for each construction set CyberMOD simulation. In our experience this is very little time for this kind of ambitious learning goal. However, all students were able to complete the modding tasks to create working models.

In terms of fostering students' creativity, we found evidence of creativity and ownership. For instance, Figure 10 depicts a students project featuring a complex arrangement of heaters and coolers keeping the temperature in the simulated house balanced.

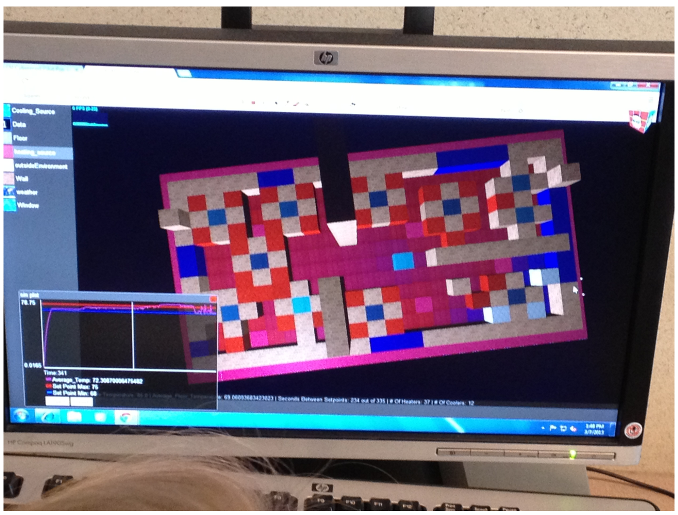

Fig. 10. This student created a very complex heater/cooler configuration keeping the house temperature in between the minimum and maximum set points (based on the plot to the left)

Students also used this creativity spontaneously to develop their own experiments. Figure 11 depicts a student engaging in an exploration of heat diffusion by deploying heaters on the outside and coolers on the inside of the house in order to observe temperature gradients.

Fig. 11. This student created an interesting heater/cooler arrangement with heaters on the perimeter and coolers on the inside.

Students even surprised us by coming up with unanticipated simulation configurations. For instance, in Figure 12 a student created a more direct blood vessel path from the pancreas to the liver yielding a more responsive blood sugar regulatory system.

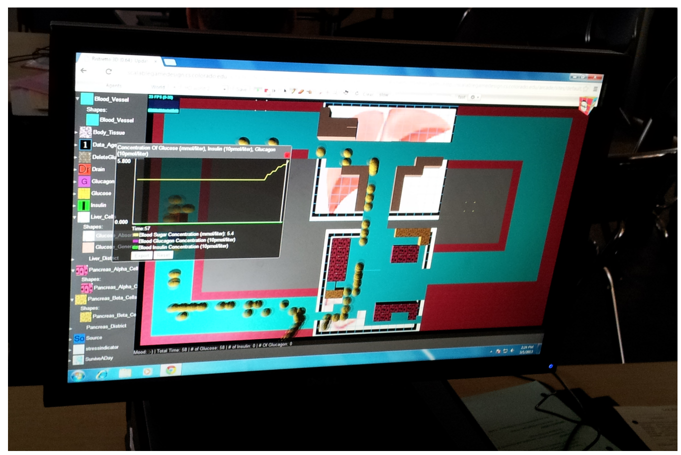

Fig. 12. This student figured out that by cutting across the level there would be a shorter distance between the pancreas and liver making glucagon and insulin more effective.

Students also enjoyed the "ownership" aspect of the modding exercise. When asked what students may or may not have found interesting about this unit, a male student responded.

"I enjoyed the part where you got to build and change different aspects of the system, how you got to build your pancreas and build your liver and it made it more your own creation rather than going in and having it all made for you, it made you think more about what you were learning"

Similarly here are two quotes from other students that express excitement at the idea that you could control it yourself.

"I think it was cool because it was interactive and you were doing it yourself"

"It was sort of like teaching yourself almost, that was really cool"

In general, students seemed to enjoy learning through CyberMOD compared to other more traditional classroom methods. One student said the following during the focus group when asked if they had a better understanding of how the liver and pancreas work after completing the modding exercise.

"I think the simulation helped with visualizing it and being made to mess around with it as opposed to watching a lecture"

V.Evidence of Student Understanding

As mentioned above, we analyzed two question sections of the pre and post test. In general, students performed significantly better on the post test as a whole, with the CyberMOD project resulting in a large effect size of 1.166. Our findings closely matched effect sizes of simulation based learning discussed in Learning Science Through Computer Games And Simulations [14].

The pre and post test for the first assessment (questions 1-3), Graph Interpretation-Labeling, provides students with the following graph and asks students to label each line on the graph using the included word bank (to the right).

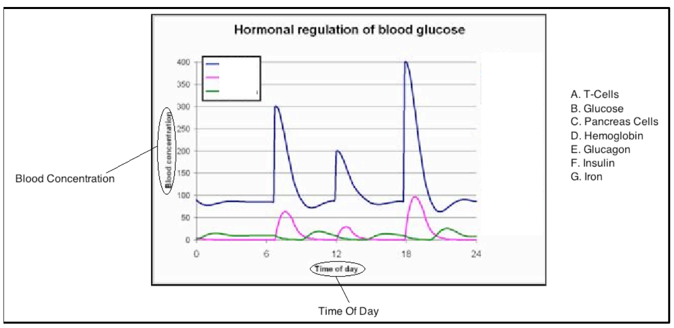

Fig. 13. Graph Provided For Questions 1-3 on the pretest

The correct answer to this question requires that the student understand how to interpret the graph, that is, that the blue line represents glucose concentration, the pink line represents insulin concentration (spikes when blood sugar is too high) and the green line is represents glucagon concentration (spikes when blood sugar is too low). To answer this question correctly students had to realize the relationship between the concentrations of glucose, insulin, and glucagon in the blood stream (a total of three distinct ideas, each assigned a point). Students improved markedly on the post test as compared to the pretest. On average, after the CyberMOD unit, students scored about 1 point higher (1.29 on pre to 2.25 on post) with a large effect size (0.881).

Similar to assessment section 3, assessment section 4 asked students to label a diagram of the blood glucose regulation cycle; and students picked answers using a word bank provided. To answer this question correctly students had to realize that 1) when the blood glucose level becomes too high the pancreas releases insulin stimulating body tissue uptake of glucose and liver tissue conversion of glucose to glycogen; conversely 2) when the blood glucose level gets too low, the pancreas releases glucagon stimulating liver cells to release glucose by breaking down previously stored glycogen. Out of 6 questions, the average student score was 1.93 on the pre test and 4.155 on the post test. Furthermore, the effect size was 1.242 which indicates a very large effect. This data strongly indicates that after completing the CyberMOD Blood Sugar Regulation simulation, students had a better understanding of the body mechanisms involved in blood glucose homeostasis.

VI. CONCLUSIONS

With over 100 diverse students (ranging from special education students to advanced placement students) at two schools participating in two CyberMOD activities, we collected qualitative (classroom observations, interviews and open ended survey questions) and quantitative (pre/post assessment Likert scale question) data from 71 students to assess efficacy. Here are three highlights of this project:

- ***Education***: Strong evidence of student learning: In a relatively short period of time (2 class periods), students employing modding are able to develop a sophisticated understanding of the scientific phenomena they are studying. Three out of five pre/post knowledge assessment sections indicated a large learning effect size. The overall effect size betwen the pre and post test (n=71) was 1.166 (Cohen effect size measure: $\geq$ 0.5 considered strong), and was statistically significant
- ***Usability***: CyberMOD is accessible and supports intuitive student in-class use: Students are able to quickly modify simulations and explore different features of a given STEM model using the CyberMOD infrastructure. All our students turned in a modded simulation. Moreover, the open-ended nature of the simulations resulted in high degrees of student engagement and exceptionally creative designs that were not anticipated by either the teachers or the activity designers.
- ***Teaching Practice***: Teachers can easily integrate CyberMOD into their in-class activities: Teachers are able to intuitively create and teach full curriculum units centered around simulation modding using the CyberMOD infrastructure

Our observations strongly upports the conclusion that CyberMOD achieves the three design fundamentals outlined above and provides a feasible classroom simulation authoring related activity that enhances student understanding and creativity in learning. Future research should look into how the CyberMOD activity enables students to better understand a variety of topics in biology and other sciences and look at ways students can further author simulations beyond the construction set domain.

VII. ACKNOWLEDGMENT

Part of this work is supported by the National Institute of Health under Grant Numbers 1R43 OD012081-01, 1R43 RR022008-0, and 1R43 RR022008- 02. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the National Institute of Health. Alexander Repenning is the founder of AgentSheets Inc.

VIII. REFERENCES

[1] Organization for Economic Co-Operation and Development (OECD), Programme for International Student Assessment (PISA), PISA 2006 Science Competencies for Tomorrow's World, 2006.

[2] President's Information Technology Advisory Committee (PITAC), Report to the President: Computational Science: Ensuring America's Competitiveness, June 2005.

[3] Organization for Economic Co-Operation and Development (OECD), Programme for International Student Assessment (PISA), PISA 2003 Technical Report, 2003.

[4] Next Generation Science Standards. Achieve, Inc. on behalf of the twenty-six states and partners that collaborated on the NGSS, Washington, DC, 2013.

[5] A. Repenning, D. Webb, and A. Ioannidou. Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the 41st ACM technical symposium on Computer science education, pages 265–269. ACM, 2010.

[6] J. M. Wing, Computational Thinking, Communications of the ACM, 49, (3) pp. 33-35, March 2006.

[7] Common Core State Standards. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington, DC, 2010.

[8] Repenning, A. and Ambach, J. The Agentsheets Behavior Exchange: Supporting Social Behavior Processing. In Proceedings of the CHI 97, Conference on Human Factors in Computing Systems, Extended Abstracts (Atlanta, Georgia, 1997). ACM Press, 26-27.

[9] A. Ioannidou, A. Repenning, D. Webb, D. Keyser, L. Luhn, and C. Daetwyler. Mr. vetro: A collective simulation for teaching health science. International Journal of Computer-Supported Collaborative Learning, 5(2):141–166, 2010.

[10] Ioannidou, A., Repenning, A., Lewis, C., et al. Making Constructionism Work in the Classroom. International Journal of Computers for Mathematical Learning, 8, 1 2003), 63-108.

[11] A. Repenning and A. Ioannidou. Broadening participation through scalable game design. ACM SIGCSE Bulletin, 40(1):305–309, 2008.

[12] Peckham, J., Stephenson, P. D. and Harlow, L. L. Broadening Participation in Computing: Issues and Challenges. In Proceedings of the 12th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSEí '07) (Dundee, Scotland,, June 23-27, 2007, 2007)

[13] L. Gray, N. Thomas, and L. Lewis. Teachers' use of educational technology in u.s. public schools: 2009. first look. nces 2010-040. National Center for Education Statistics, May 2010.

[14] N. R. C. C. on Science Learning. Learning Science Through Computer Games And Simulations. The National Academies Press, Washington, DC, 2011.

[15] A. R. Basawapatna, A. Repenning, and C. H. Lewis. The simulation creation toolkit: an initial exploration into making programming accessible while preserving computational thinking. In Proceeding of the 44th ACM technical symposium on Computer science education, pages 501–506. ACM, 2013.

[16] Basawapatna, A., Koh, K. H., Repenning, A., Using Scalable Game Design To Teach Computer Science From Middle School to Graduate School, ITiCSE '10: Annual Conference on Innovation and Technology in Computer Science Education, Ankara, Turkey, June 26-30, 2010.

[17] Basawapatna, A., Repenning, A., and Lewis, C., 2013. The simulation creation toolkit: an initial exploration into making programming accessible while preserving computational thinking. In Proc. of SIGCSE '13. ACM, Denver, Colorado.

[18] National Science Foundation Task Force on Cyberlearning, "Fostering Learning in the Networked World: The Cyberlearning Opportunity and Challenge. A 21st Century Agenda for the National Science Foundation," National Science Foundation.

[19] A. I. Mørch, "Three Levels of End-User Tailoring: Customization, Integration, and Extension," in Computers and Design in Context, M. Kyng and L. Mathiassen, Eds. Cambridge, MA: The MIT Press, pp. 51-76, 1997.

[20] M. S. El-Nasr and B. K. Smith, "Learning through game modding," Computers in Entertainment, 4 (1), p. 7, 2006.

[21] R. Yager, "Constructivism and Learning Science," in Learning Science in the Schools: Research Reforming Practice, S.M. Glynn & R. Duit, Eds., Mahway, NJ: Lawrence Earlbaum Associates, 1995.

[22] F. Newmann, Marks, H. M., Gamoran. A., Authentic pedagogy and student performance, American Journal of Education, 104, (4) pp. 280-312, 1996.