Educational Simulation Design to Transform Learning in Earth and Environmental Sciences

Michelle Zhu[a], Matthew Johnson[a], Aditya Dutta[a], Nicole Panorkou[b], Bharath Samanthula[a], Pankaj Lal[c] and Weitian Wang[a]
[a]Computer Science Department, [b]Mathematical Department, [c]Earth and Envriomental Studies Departmnet
Montclair State University
Montclair, NJ USA
zhumi, jonhsonm7, duttaa4, panorkoun, samanthulab, lalp, and wangw@montclair.edu

Abstract— **This Innovative Practice Full Paper presents several educational simulation designs to transform learning in the earth and environmental sciences. In recent years, K-12 education has seen a widespread pedagogical shift from traditional textbook-based learning to a student-centered interactive learning environment. Innovative teaching technologies including games and simulations are exploited to make learning fun and engaging for students at various grades. As technology continues to advances, stakeholders such as teachers, parents, and educational policymakers are motivated to know the most effective technology platform to engage students in active learning. In this paper, we discuss some useful simulation techniques from the perspectives of users' experience, system deployment and developers' point of view. In addition, we also present the design and implementation of some Earth and Environmental Science simulations that are developed using NetLogo and JavaScript. Some student assessment results are also provided to compare the learning effectiveness between the traditional textbooks and computer simulation-based lessons. Our results show that computer simulation can help enhance the student's content retention and graph interpretation skills.**

Keywords—simulation, computing, JavaScript, earth and environmental Science

I. Introduction

An education computer simulation follows a model of real process/system and allow students to test hypothesis by testing on "what-if" scenario. The most recent generation of learners have adapted to recreation that incorporates PC and console games, smartphone apps, and relatively sophisticated user interfaces. Although these have quickly become commonplace in everyday life, they are a relatively new addition to the classroom [1]. Regardless of their recent incorporation, there is an overwhelming body of evidence to support their use within the classroom. Researchers found that an "instructional simulation approach is superior to the conventional chalk-talk approach" when measuring secondary school students' achievement in basic science [2]. Research also show that the benefits of education simulations cannot be automatically gained. Many aspects of simulation design and presentation should be considered in addition to pedagogical strategies in order to engage students in effective exploration, formulation and testing of the hypothesis [3]. Consequentially, creating and maintaining accessible and intuitive interfaces, while engaging students and incorporating instructors remains a challenge for software developers as well as UX/UI researchers alike.

There is a lack of earth and environmental science simulation software accompanied with lesson plans, which integrate computational thinking with science aligned with the Next Generation Science Standards (NGSS)[4] and the Common Core State Standards for Mathematics (CCSSM) [5]. Although *PhET* [6] and many other sources make free online interactive science simulations, the integration of computational thinking practices are still not sufficient. In particular, the 3D web-based graphics techniques for education simulations need to be further explored to better engage students and facilitate easy software deployment.

Additionally, despite the United States spending 1.5 billion dollars every year on environmental studies research, these efforts fall short if not widely used among communities [7]. Thus, it is critical to develop well-maintained, open-source or low-cost educational simulations available to educators. Pragmatically, any educational software becomes burdensome if downloading or purchasing is required, by straining institutional or financial resources. Therefore, 3D web-based interfaces without installation requirement are desirable.

This study aims to answer the call for simple and effective educational simulations in the environmental sciences by integrating legacy Cascading Style Sheets (CSS) and HTML5 code with the JavaScript API. We also observe the classroom and perform quantitative data analysis to assess students' learning outcomes for both computational thinking and science.

We designed and developed several simulation software on earth and environmental science topics using JavaScript, CSS, and HTML5 with interactive parameters. Some developed simulations include the rock cycle, weather, day, night and seasons, and lunar phases and eclipse. These simulations are hosted via Amazon web services and can be accessed in a web browser with steerable parameters. Our simulations and accompanying lesson plans are aligned with the NGSS and CCSSM. Furthermore, Scratch block-based programming is integrated with various science topics and students learn basic programing construct such as condition and loop. In our experiments, the control group is taught with traditional textbooks and the treatment group is taught with our simulation-based lesson plans.

Fig. 3 presents the code segment for users to interact with the controls and turn on/off the tilt angel of the earth in jQuery function. Fig. 4 shows the system diagram for the day and night

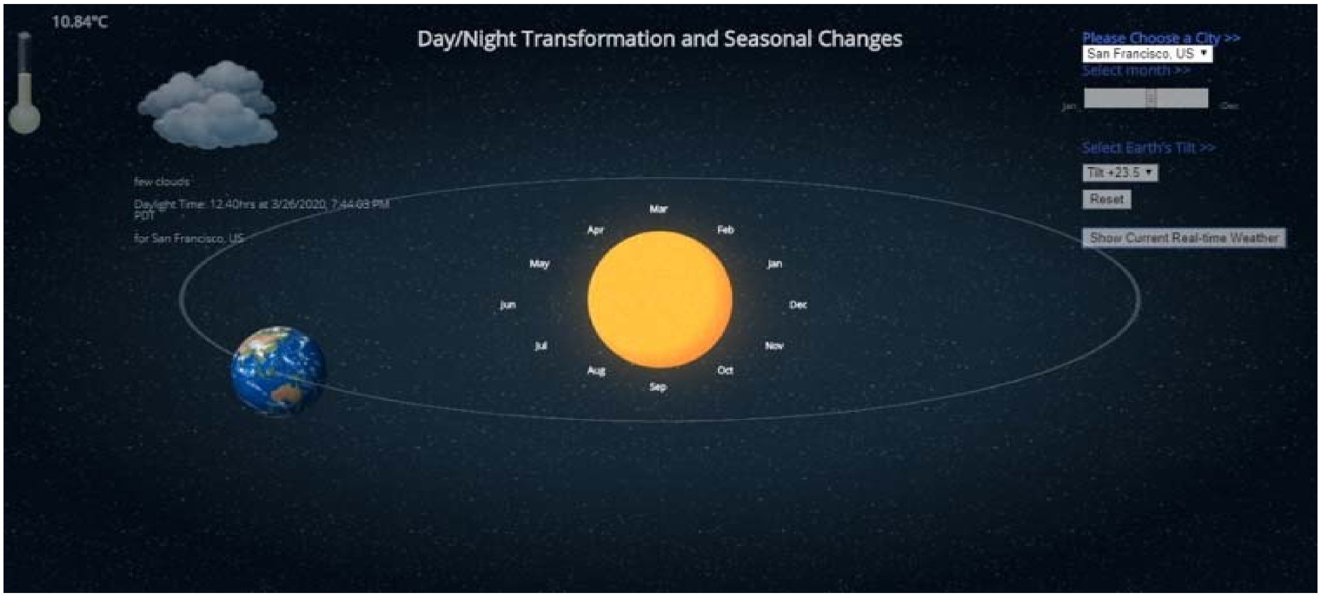

Fig. 1. Screenshot of the day, night and season interface.

II. System Design

The definition of Science and Engineering Practices (SEP) found on the NGSS website, is defined as *"The practices are what students DO to make sense of phenomena. They are both a set of skills and a set of knowledge to be internalized. The SEPs reflect the major practices that scientists and engineers use to investigate the world and design and build systems* [8]." Some earth and environmental science topics are either hard or not possible to replicate in lab. Thus, simulation with steerable parameters serve as an ideal tool for learning and teaching. Students learn from interacting with the simulation and examine questions embedded in the online lessons. Teachers motivate students' learning by connecting the lesson topic to students' daily experience and guide students throughout the lessons. A good simulation software should consider both the teachers' and students' perspectives. For example, if software is created too passively with respect to the instruction, students may not see any benefit [9]. Hence, the requirements of the simulation remain dynamic as the software will undergo rapid iterative cycles of implementation, testing and improvements.

Fig. 1 shows a screenshot of our day, night and season simulation. Users are able to choose one of the cities around the world and select the month of the year. The simulation will show the location of the city as a pin on the orbiting earth. The typical weather such as high, low temperatures and daylight hours for that month are shown. In addition, the real-time weather report for the chosen city can also be acquired by issuing an AJAX call to the openweathermap.org API to obtain the current weather. Fig. 2 illustrates the code segment of openweather API call. After retrieving the data, we needed to update the webpage dynamically. Every dynamic update in this simulation was handled through jQuery.

simulation. While not all use cases are represented as would be in testing, this diagram describes all the general case. The location of New York City, earth tilt as 23.5 degrees, and slider month as January are set by default. If the tilt is changed to zero degrees, earth does not display any seasons and the weather for each month remains the same. In this state, if the month slider is not used, the planet will continue to orbit the sun. Also, users may select a city and see the pin placed on the orbiting earth if he/she does not select a month. If a month is selected, then the earth only self-rotates at the corresponding month position around the sun. At this point, if the user clicks the "Display Real-Time Weather" button and the tilt is zero degrees, the page displays a pop-up window showing the message as *"There would be no seasons across the Earth. The weather would be like the middle of fall or spring throughout the entire year."* If the tilt is 23.5 degrees, the AJAX call the openweathermap.org API and the padded JavaScript Object Notation (JSON), which as a data interchange format will be returned with real-time weather information for the chosen city. The real-time weather is displayed in the upper left-hand corner of the screen as standardized icons that are commonly used in weather forecast. The day and night simulation has recently been finished and has not been implemented by teachers in the classroom.

When designing the interfaces, we largely consider Krumhansl et al. 's view that "Oceanographic and other Earth science data impose a high level of intrinsic cognitive load due to the number of interacting elements typically involved in the Earth systems [10].

As a result, it is critical that simulation interface designers take steps to reduce extraneous load, alleviate intrinsic cognitive load, and maximize germane cognitive load. Therefore, to reduce the amount of extraneous cognitive load on students, the parametric controls should be simplified to the minimum number necessary to achieve learning objectives while allowing for easy student manipulation.

```
$.ajax({
  url: 'https://api.openweathermap.org/data/2.5/weather? \
  appid=d0517bdbefd38429171c77e2ba6b00e4&units=imperial&q='
   + selectedCity,
  type: 'POST',
  dataType: 'jsonp',
  success: function (response) {
    var main = response['main'];
    var weather = response['weather'];
    var sys = response['sys'];
```

Fig. 2. AJAX call to openweathermap.org to get the current weather of the choosen city.

All our simulations are integrated, tested, and deployed at Amazon Web Services (AWS). Our simulations are accessible to the public at our project link: https://acmes.online/view_simulation_links

```
$('#tilt').change(function () {
  $('#earth .pos, #earth .planet, #earth.orbit').css("transform",
  "rotate(" + $("#tilt option:selected").val() + "deg)");
  if ($("#tilt option:selected").val() === '23.5'){
    setTimeout(function() {
    location.reload();}, 1);
  }
});
```

Fig. 3. User interaction with control buttons

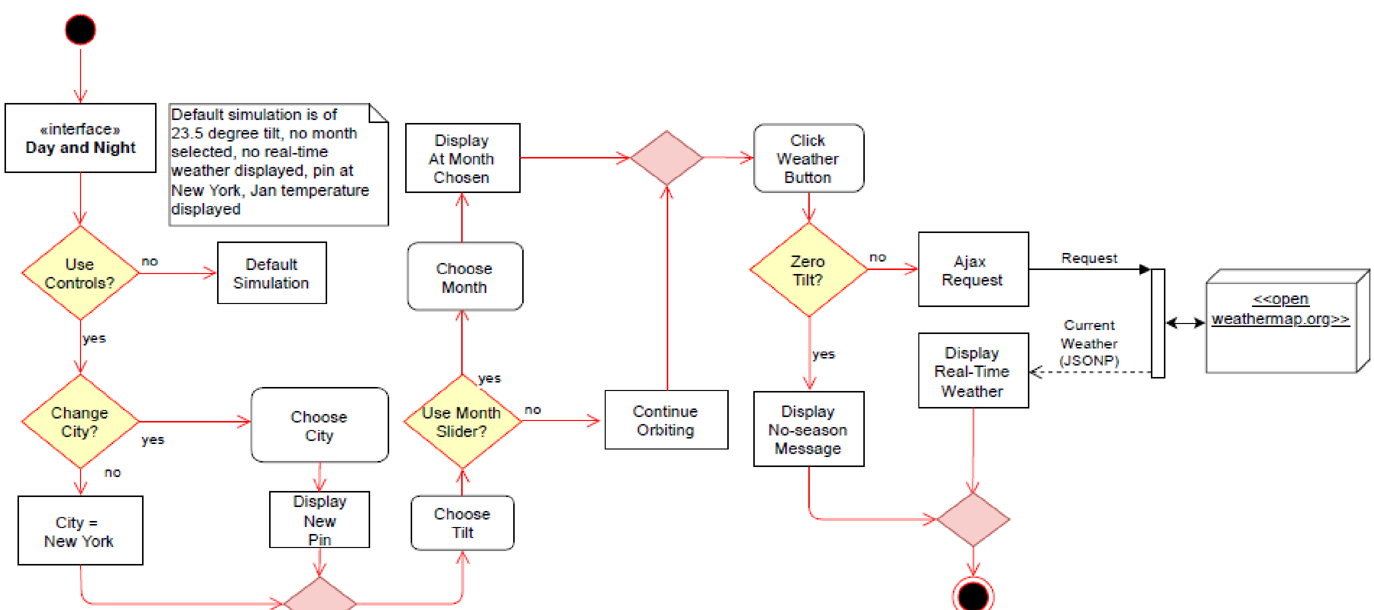

Fig. 4. The flowchart of the day night and season simulation.

III. Experimental Results

A. Student Performance Assessment

One of the science modules conducted with a control and treatment group was the water cycle which was implemented in 2D NetLogo [11] and Scratch. Two 6th grade classes participated in this study as control and treatment groups. The control group used standard textbook, and the treatment group used our simulation-based lessons. Pre and post-test of the same assessment was given to analyze the performance of the two groups. Students also learned Scratch coding to create an animation to illustrate a lifecycle of an imaginary water molecule.

Fig. 5 captured a screenshot of our water cycle simulation, which illustrated concepts of evaporation, condensation, precipitation, infiltration and runoff processes in a dynamic way. Students can choose the humidity and various temperatures to control the water cycle patterns. We are currently at the stage of converting most of our simulation programs, which were coded in NetLogo to JavaScript due to the web compatibility of JavaScript. Fig. 6 showed a Scratch programming project for students to draw the path of an imaginary water molecule. Students need to apply programming concept to water cycle knowledge and it serves as a good example of integrating Computer Science with Science [12].

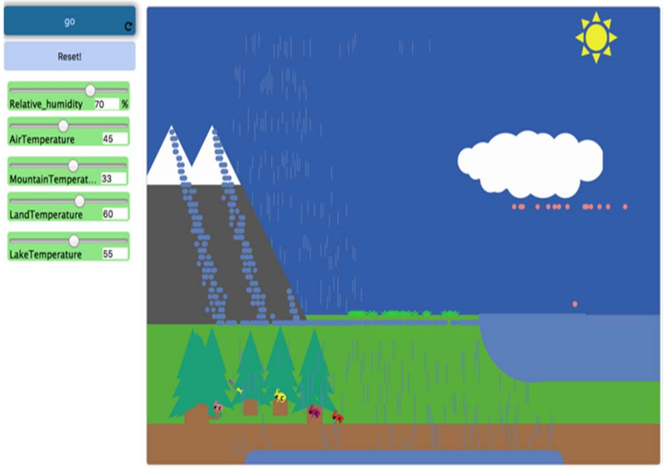

Fig. 5 A screenshot of our water cycle simulation devleped using NetLogo.

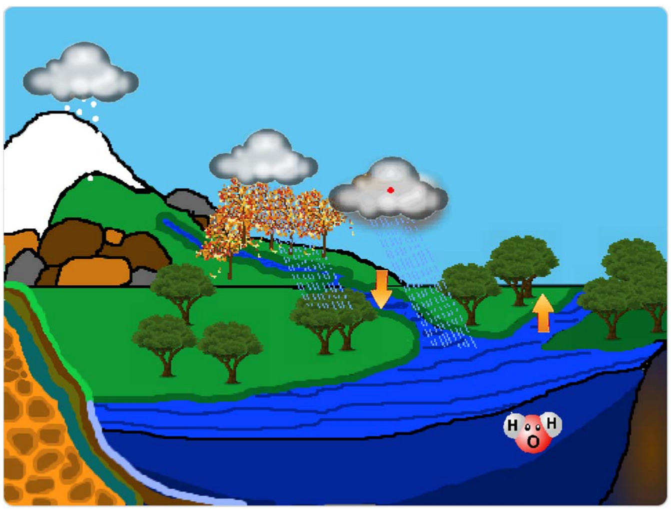

Fig. 6 Scratch programming project for water cycle.

Both the control and treatment groups took the pre and post assessment tests. The student size of the control group was n = 18, and n = 21 for the treatment group.

The improvement for both control and treatment group appeared to be normal in the histogram and this assumption allowed to proceed with the t test. The two-sample t test as shown in Fig. 7, the null hypothesis was the average improvement of treatment group was the same as the average improvement of the control group; The alternative hypothesis was the average improvement of treatment group was significantly higher than that of the control group. We rejected the null hypothesis since the p-value was much lower than the significance level of 0.05 and concluded that the average improvement of the treatment group was significantly higher than that of the control group. It demonstrated that our simulation-based lessons were more effective for students to achieve the learning outcomes.

```
> t.test(TreatmentImp,ControlImp,alternative='greater')

	Welch Two Sample t-test

data:  TreatmentImp and ControlImp
t = 3.9836, df = 36.183, p-value = 0.0001572
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 3.997137      Inf
sample estimates:
mean of x mean of y
 9.047619  2.111111
```

Fig. 7 Two sample t-test results for score improvements for both control and treatment groups.

Graph interpretation skills: We were also interested in students' ability to answer graph interpretation related questions. Two of the eleven questions were related to graph interpretation between the relationship of multiple variables. Fig. 8 showed the graph of temperature and humidity in 10 days of Hunter mountain in New York. Students were asked to predict the most-likely precipitation/rain date.

Paired-t tests were conducted for both control and treatment groups as shown in Fig. 9. The paired t tests compared the difference between pre and post test results on two graph related questions. The null hypothesis was that there is no significant difference between post and pre-tests. The alternative hypothesis was that the improvement is significantly higher than zero. For the control group, we can see that we failed to reject the null hypothesis with a p-value of 0.3839 higher than 0.05. For treatment group, we rejected the null hypothesis and concluded that the difference between post and pre is significantly higher than zero. The results confirmed that students acquired significantly better graph interpretation skills by using the simulation-based lessons.

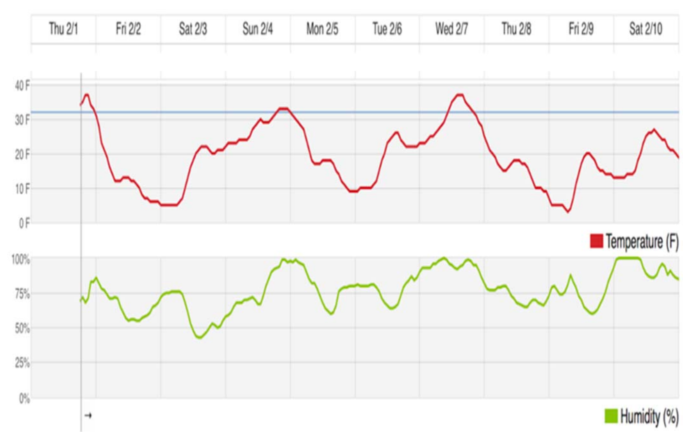

Fig. 8 The graph used to answer questions of the water cycle.

```
> t.test(Control$`Post 10-11`,Control$`Pre 10-11`,paired=T,alternative='greater')

        Paired t-test

data:  Control$`Post 10-11` and Control$`Pre 10-11`
t = 0.29991, df = 17, p-value = 0.3839
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -0.8000657        Inf
sample estimates:
mean of the differences
              0.1666667

> t.test(Treatment$`Post 1-9`,Treatment$`Pre 1-9`,paired=T,alternative='greater')

        Paired t-test

data:  Treatment$`Post 1-9` and Treatment$`Pre 1-9`
t = 7.9592, df = 20, p-value = 6.307e-08
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 5.222029      Inf
sample estimates:
mean of the differences
               6.666667
```

Fig. 9 Paired sample t-test for graph interpretation skills: top for control group and bottom for treatment group.

IV. Conclusion and Future Work

Our observation and statistical analysis results show that 1) A simple and intuitive interface can stimulate students' interests and increase their satisfaction level. 2) The easy deployment of the simulation software is important. Most schools do not allow software installation. A web-hosted simulation with low delay will remove the technical barrier of adoption for schools. 3) Teacher's active role is still indispensable in order to achieve the student's learning goals and equity in computer education. Experienced teachers usually introduce the topic based on students' prior experience and bring some questions that students can connect to their daily lives. Thus, Professional Development (PD) training on pedagogical content knowledge is important for teachers to learn how to teach particular content in particular ways to enhance student understanding. 4) Our treatment group improves significantly more than the control group in terms of the difference between pre and post-assessment scores. The improvements in graph interpretation skills are higher for the treatment group than the control group.

In terms of simulation technology, it became clear that CSS animations alone would not be enough to continue to produce engaging interactive simulations. The main issue being that the CSS implemented in the simulations depended on concurrent keyframes rather than data bound to the DOM (Document Object Model). Over the course of the simulation development and maintenance, CSS became incrementally difficult to satisfy new teacher and student requirements in a timely manner without moving to a different platform. To streamline simulation controls and animations, we will be standardizing the integration of the Data Driven Documents API, namely D3.js. D3.js, short for Data-Driven Document is a highly dynamic data visualization JavaScript library. D3.js utilizes Scalable Vector Graphics (SVG), HTML5 and CSS standards. Thus, integration with legacy CSS code can be seamless. In addition, we will consider the constantly evolving landscape of human computer interaction. For example, would three-dimensional simulations benefit students through increased engagement? Should we consider using three.js, a cross-browser JavaScript library and API to create 3D graphics in a web browser? What kind of user interface design and visual outputs would attract particularly female and underrepresented minorities? Those are all questions we plan to think about for our future simulation development.

It is our interest to adopt "situated learning" for students to design and build robots to understand science concepts in our future work. *Situated learning* engages trainees in a process of communities practice, such participation is at first legitimately peripheral but would increases gradually in engagement and complexity [13]. We believe that hands-on robots project imbued with Artificial Intelligence (AI) will attract more students to computing and STEM fields. Furthermore, the United States will expand opportunities for all Americans to gain the skills needed to participate in an with AI-ready workforce, including those from diverse backgrounds who are historically underrepresented in STEM fields" [14].

Acknowledgment

This research is supported through a STEM+Computing grant from the Division of Research on Learning of the National Science Foundation (#1742125)

References

[1] Adams, W.K., Reid, S., LeMaster, R., McKagan, S., Perkins, K., Dubson, M. & Wieman, C.E. (2008). A Study of Educational Simulations Part II – Interface Design. Journal of Interactive Learning Research. 19 (4), pp. 551-577. Waynesville, NC: Association for the Advancement of Computing in Education (AACE).

[2] Nwafor, C. and Okechukwu, A. (2016). Effects of Instructional Simulation on Secondary schools students's achievement in Basic Science. *Journal of the Science Teachers Association of Nigeria (JSTAN). 51.*

[3] 16 Thomas, R.C. and Milligan, C.D. (2004). Putting Teachers in the Loop: Tools for Creating and Customising Simulations. *Journal of Interactive Media in Education (Designing and Developing for the Disciplines* Special Issue), 2004 (15). ISSN:1365-893X http://www-jime.open.ac.uk/2004/15

[4] Next Generation Science Standards: https://www.nextgenscience.org/

[5] Common Core Math Standards: http://www.corestandards.org/Math/

[6] PhET Simulation: https://phet.colorado.edu/

[7] Bostrom, A., Morgan, M. G., Fischhoff, B., and Read, D. (1994). What do people know about global climate change? *Mental models. Risk Analysis*, 14(6), 959-970.

[8] National Science Teaching Association: Science and Engineering Practices. Retrieved in March 2020, from https://ngss.nsta.org/PracticesFull.aspx

[9] Willingham, D. (2017). Ask the Cognitive Scientist: Do Manipulatives Help Students Learn? *American Educator*, Vol 41., No.3.

[10] Krumhansl, R., Busey, A., Krumhansl, K., Foster, J., and Peach, C. (2013). Visualizing Oceans of Data: *Educational interface design*. *OCEANS* - San Diego, Oceans - San Diego, 20.

[11] Wilensky, U. (1999). NetLogo. (http://ccl.northwestern.edu/netlogo/). Center for Connected Learning and Computer-Based Modeling, Northwestern University, IL.

[12] Zhu, M. *et al*., (2018) "Integrating interactive computer simulations into K-12 earth and environmental science," *2018 IEEE Integrated STEM Education Conference (ISEC)*, Princeton, NJ, pp. 220-223.

[13] Lave J. and Wenger, E. (1991) Situated learning: Legitimate peripheral participation. Cambridge university press, 1991.

[14] US AI Initiative Annual Report, *https://www.whitehouse.gov/wp-content/uploads/2020/02/American-AI-Initiative-One-Year-Annual-Report.pdf*, 2020.