Creating Engaging Science Projects with NetsBlox

Brian Broll, Hamid Zare and Akos Ledeczi
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

***Abstract*—NetsBlox is a visual blocks-based programming language that supports distributed programming. NetsBlox includes Remote Procedure Calls (RPC) that provide access to a set of online scientific data sources such as maps, weather, earthquakes, astronomy imagery, air pollution, etc. This demonstration shows how to use these RPCs to create engaging science projects.**

***Index Terms*—visual programming, distributed programming, computer science education, STEM**

I. Introduction

NetsBlox [1], [2] is a visual programming environment that extends Snap![3] with a few carefully selected abstractions that enables user to create distributed applications. Message passing enables communication between NetsBlox projects running on different computers and can be used, for example, to create multi-player games. Remote Procedure Call (RPC) is the highest level of distributed abstraction NetsBlox employs. Snap! supports the specification of Custom Blocks which are essentially functions. The NetsBlox server provides remote procedures (appearing as special custom blocks) accessible via RPC. RPCs can provide access to useful functionality running on the server and they are also very helpful in simplifying the implementation of multi-player games. For example, the server can manage the state of the game, and the client code can simply invoke an RPC to make the next move. The semantics of RPC are as expected: multiple input arguments, single output argument, pass-by-value and blocking call. These should be familiar to users of custom blocks.

The most common use of RPCs in NetsBlox is to provide access to various publicly available web services, such as Google Maps or the Sloan Digital Sky Survey (SDSS) [4]. Note that RPCs do much more than simply wrap web APIs. First, they provide much simpler and easier-to-use user interface. Second, they also store state information on the server. For instance, when the user requests a map image from Google, the NetsBlox server can carry our coordinate transformations from latitude, longitude to stage y and x, respectively, through additional RPCs without sending any further requests to Google. NetsBlox also caches maps for subsequent requests.

NetsBlox groups together related RPCs into services. For example, there is a Weather, an Earthquake, and an Air Pollution service, etc. There is a single block for invoking RPCs containing two pull down menus. The first specifies the service and once the user selects it, the second pull down menu is dynamically populated with the available RPCs.

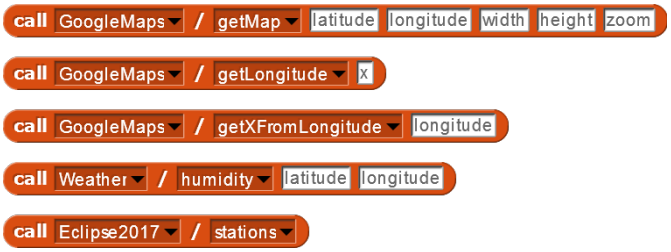

Fig. 1. RPC examples.

Upon selecting the RPC, the block updates again to show the required input arguments. See Figure 1 for examples.

The rest of the paper briefly summarizes a few example applications that we will demonstrate.

II. Interactive Weather App

The power of the NetsBlox RPCs combined with the Snap! event model is best illustrated by an interactive weather application that does not use loops or even if-statements. That is, it can be part of an introductory programming curriculum very early in the course.

The idea is is to display a Google map background with panning and zooming using the arrow and +/- keys and show a weather icon and the current temperature wherever the user clicks on the map. Optionally, the city name near the click can be displayed as well using the Geolocation service. The entire program consists of 8 event handlers with 2 to 5 blocks each.

III. Solar Eclipse Tracker

It is a well known fact that solar eclipses affect the temperature. We decided to verify this with a fun NetsBlox project during the Great American Eclipse of 2017. We created a new service called Eclipse2017 with a number of RPCs. The service was getting temperature data from 121 Weather Underground stations along the path of the total solar eclipse in 30-second resolution during the phenomenon. We wrote a NetsBlox app to visualize temperature changes during the eclipse. A cooling trend was indicated with a blue icon, while a warming trend was shown with red. As the shadow of the moon moved across the US, in front of the shadow, most of the stations indicated a cooling trend as a larger and larger area of the sun was covered by the moon, while behind the shadow, temperatures rose rapidly. Keep in mind that cloud cover and rain also have significant impact on temperature, so this blue/red wave was not uniform (that is why we were also showing weather conditions along the path). See Figure 2.

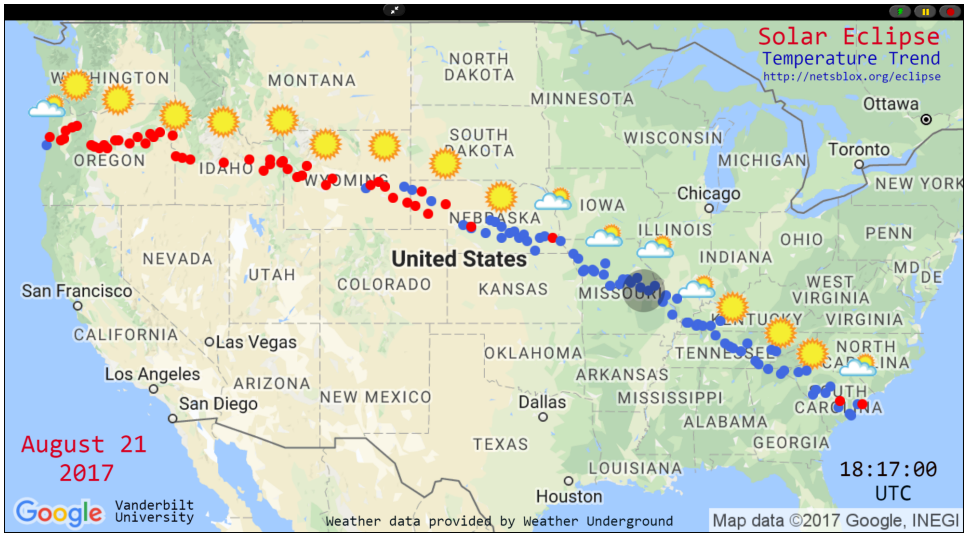

Fig. 2. Eclipse temperature trends

The demonstration will replay the eclipse using archived data. Once the animation is over, the user may click on any of the weather stations to display a plot of the temperature at that location during the solar eclipse. See Figure 3. The plot is generated using the RPCs of the chart service. This is another useful tool for visualizing any kind of numerical data stored as NetsBlox/Snap! lists.

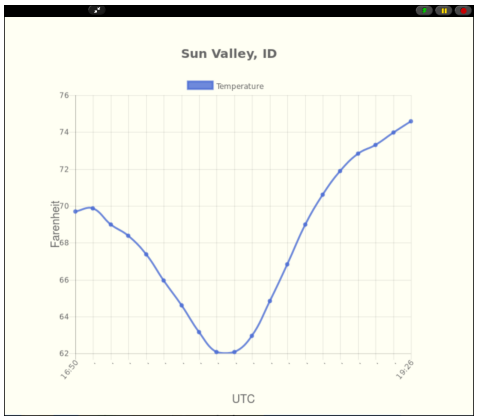

Fig. 3. Temperature trend in Sun Valley during the solar eclipse.

IV. Internet of Things

ThingSpeak [5] is an open Internet of Things (IoT) platform supported by MathWorks, the creators of MATLAB. Anybody can deploy their sensor on the internet by registering it with ThingSpeak and pushing the data to the platform. Consequently, there are thousands of sensors around the world that are freely accessible through ThingSpeak. We have created a ThingSpeak service for NetsBlox through which users can search for sensor data by keyword or location and then download, process and visualize the data.

During this demonstration we will showcase this capability by searching for and displaying sensors on an interactive map and plotting available data using the chart service.

V. Conclusion

As the examples above illustrate, NetsBlox supports the creation of innovative and engaging science projects. Many of these would fit into the curriculum of introductory computer science classes. In addition to the projects presented in this abstract, many other applications will be highlighted including one displaying historical earthquake events anywhere on earth using USGS data and an interactive map of the sky relying on the Sloan Digital Sky Survey (SDSS) [4]. In addition to scientific data, projects geared towards students interested in the humanities are also supported. The Movie Database and the collection of the British Museum are also available through NetsBlox RPCs making it possible to bring art into the computer science curriculum.

Acknowledgment

Funding from the Trans-institutional Programs (TIPs) of Vanderbilt University made possible to start the development of the tool. This material is also based in part upon work supported by the National Science Foundation under Grant Numbers CNS-1644848 and DRL-1640199. 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.

References

[1] B. Broll, A. Lédeczi, P. Volgyesi, J. Sallai, M. Maroti, A. Carrillo, S. L. Weeden-Wright, C. Vanags, J. D. Swartz, and M. Lu, "A visual programming environment for learning distributed programming," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 2017, pp. 81–86.

[2] "NetsBlox," https://netsblox.org, cited 2017 July 25.

[3] "Snap!: a visual, drag-and-drop programming language," http://snap.berkeley.edu/snapsource/snap.html, cited 2016 March 16.

[4] D. G. Y. et al., "The sloan digital sky survey: Technical summary," *The Astronomical Journal*, vol. 120, no. 3, p. 1579, 2000. [Online]. Available: http://stacks.iop.org/1538-3881/120/i=3/a=1579

[5] "ThingSpeak: open IoT platform," https://thingspeak.com/g, cited 2017 September 21.