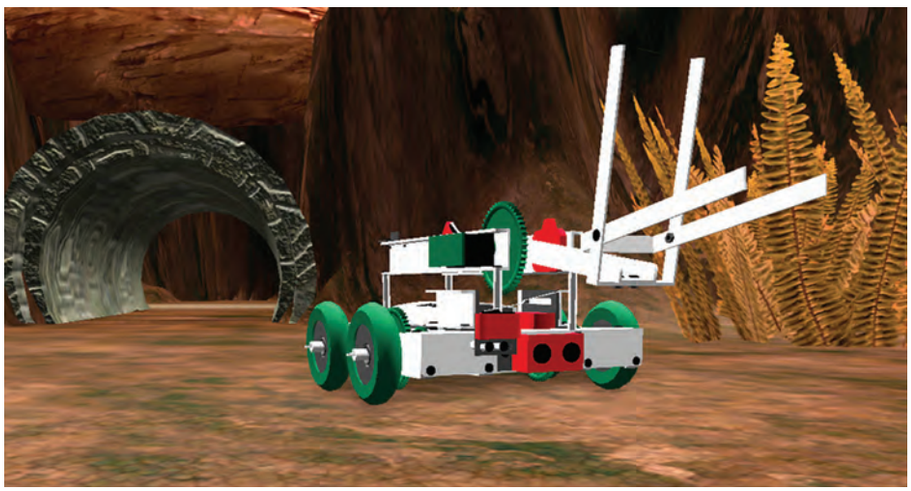
Grassroots Robotics

by Robin Shoop, Carnegie Mellon
Robotics Academy and FIRE Director

FIRE Unveils Robot Virtual World Games

Rescue a planet while programming your bot!

Computing and computer science (CS) have changed the world. Try to think of an industry that has not been impacted by CS... sports, entertainment, manufacturing, health care, automotive, energy – you name the industry and it has been affected by computer science in the past 10 years. According to the department of labor, by 2018 computing related occupations will account for nearly 60% of all new science, technology, engineering, and mathematics (STEM) jobs; STEM leads to innovation and innovation leads to wealth.



A VEX robot prepares to help terraform Planet H99.

While we cannot predict what the fastest growing technology will be in five years, we can confidently predict that it will involve computer science. Carnegie Mellon is creating a Computer Science Social Network (CS2N) with many engaging activities designed to enable students of all ages to learn how to think algorithmically; this article focuses on one of these projects - Robot Virtual Worlds.

THE VISION – ALL STUDENTS LEARN BASIC PROGRAMMING AND CAN THINK ALGORITHMICALLY

Lots of formal and informal education groups are using robots and robotic competitions to teach computer science and STEM, but unfortunately most robotic teams only have one or two main programmers, limiting the programming experience for the rest of the team. Robotics competitions and programs teach kids lots of great skill sets like project management, mechanical engineering, teamwork and problem solving, but how can we get more students involved with computer programming and control of their robots? Unfortunately, due to resources, that is not an option for most teams; there just are not enough robots to go around for every student.

One Solution – Build programmable digital robots (and make it fun to learn).

Carnegie Mellon's Computer Science Social Networking page.

An all terrain vehicle that is programmable in the virtual world. The interface allows the user to see debugging information, rotate the camera, and reset the world.

ROBOT TO THE RESCUE VIRTUAL WORLDS

We are trying to solve a resource problem, most schools don't have enough robots to go around and even when they do, most students don't have robots at home that they can practice programming with. So we start with the following questions: How can we provide every student with a robot that they can play with whenever they feel like it? And, can we design an engaging

"The Vision – All students learn basic programming and can think algorithmically"

SEE THE VIDEO!

Scan this code on your smartphone with a bar reader app or type in find.botmag.com/051102

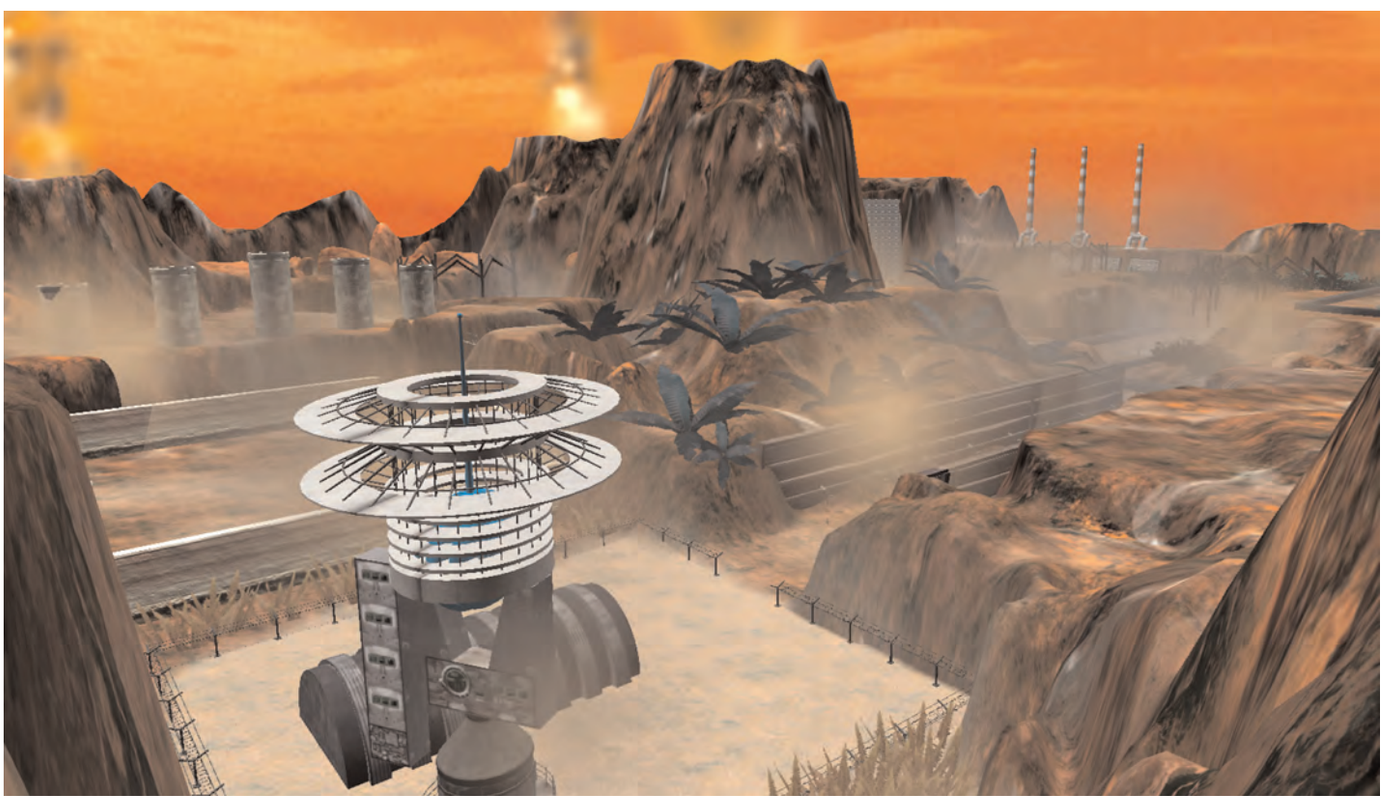
application that teaches programming and is fun? Those questions lead us to the development of our "Robot to the Rescue" series of Robot Virtual Worlds.

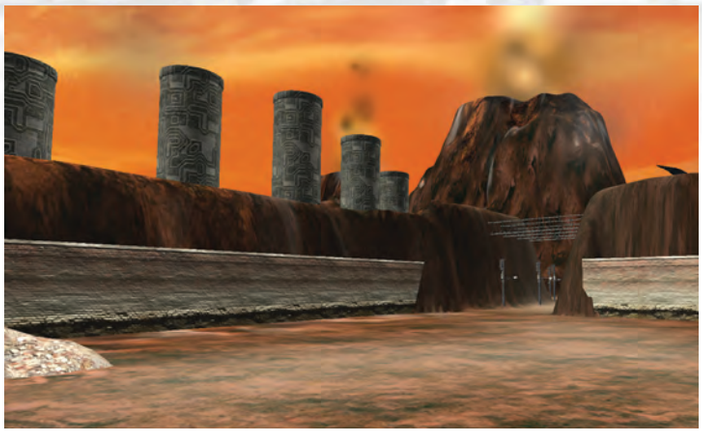
In the virtual worlds, we combine playing an online game, a compelling story, and scaffolded robotic missions where students learn to program as they complete each mission. Our first robot virtual world involves colonizing a planet, but we can imagine Robot to the Rescue themes around the weather, healthcare, economics, energy, and more. Our vision is to create virtual scenarios where kids are playing a story-based game and learning to program as they solve missions in the story.

TERRAFORMING PLANET H99 IN 2050

The first Robot to the Rescue game is set on Planet H99. The year is 2050 and the Global Federation of World Evolution has a collaborative research project named H99; short for Habitat 99. The goal of H99 is to place a human colony on the planet by 2099. The planet is being terraformed by a team of humans and robots with the majority of the work being done by the robots. (Terraforming is the process of deliberately modifying that atmosphere of a planet to make it habitable by humans.) The student's job is to program and manage the robots!

By working in a digital game we are able to integrate strategies used by other online games designed to make the games intriguing. Think of the Planet H99 Robot to the Rescue game like SimCity. Imagine a game where once the student learns some basic programming skills, the world will dynamically change based on the decisions the student makes. In Planet H99 the student is responsible for all com-





ponents of the colony including mining, farming, transportation, fuel extraction, and terraforming. Students will receive points as they complete missions and move to the next level where they will have access to better sensors, new robots, and other parts of the world as they become more skilled.

ROBOTC & LABVIEW

Initially, students will be able to program their VEX, LEGO, and H99 planet vehicles using the ROBOTC programming software; ROBOTC is currently used in thousands of schools to program VEX, LEGO, and Arduino robot controllers. The virtual world is developed so that students can use the exact same code on their virtual world robot as they do on VEX, LEGO or Arduino robots. A student can draw from ROBOTC's extensive library of sample programs to solve the problem, or they can begin programming from scratch.

Our development team is also working with National Instruments (NI), an industry leader in graphical system design, to develop versions of the Robot to the Rescue series for National Instruments LabVIEW, a graphical programming language. The company also provides free resources such as NI's LabVIEW Web UI Builder software that enables students to create web-based user interfaces that run inside a web browser. The development team will work with NI to enable students to program UI front panels like the one shown below that can monitor and control data remotely via a browser, www.ni.com/uibuilder.

For instance, imagine Planet H99's

greenhouse being thermostatically controlled based on algorithms built inside the game that control the planet's temperature. We see opportunities for students to learn to earn points in the game as they build the front panels that will enable them to easily access and interpret data from the virtual world. The great thing about this solution is that students are learning to program using multiple programming languages, and they are just playing a game. Programming challenges are incorporated into the mission and students start with working code that they are required to modify; they learn as they play.

SCAFFOLDED LEARNING ENVIRONMENT

The Tech Demonstration version of the Robot Virtual World Software will support encoders, touch sensors, sonar, and a compass sensor as well as timers and power levels. Students will have access to all of our ROBOTC online training materials which can be found at the Robotics Academy's website but our strategy for teaching programming will be to embed the training into each mission.

For example, initially students need to repair fueling stations that are found on the planet. The student is given a link to a fly-over (imagine Google Maps) where they see the path that the robot needs to take, they will also be given a map with technical specifications on it that include distances and speed, and finally they will be given commented working code that they can use to begin to program their robot. In this example, students only need to move their robot to the fuel station and point a transmis-

```
39
40  Movement Functions
41
42    setMotor(armMotor);          // motor port: A,     speed: 75
43    forward();                   // speed: 75
44    backward();                  // speed: -75
45    swingTurn();                 // direction: right,  speed: 75
46    pointTurn();                 // direction: right,  speed: 75
47    stop();                      // time: 0.5s
48
```

sion feed at the fuel stations receiver to update the fuel stations firmware. Once students finish this easy level of the challenge they are given another level where their robot has more capabilities and the programming challenge becomes more difficult.

NATURAL LANGUAGE LIKE FUNCTIONALITY

In order to make this environment user friendly for younger students we have begun to integrate commands into ROBOTC like: forward, backward, point_turn(left), point_turn(right), wait, stop, and a variety of simplified sensor commands. In order to access the natural language simplified code a student will select Robot, Platform Type and then Natural Language. Once this selection is made, they will be able to program their robots using a simplified version of ROBOTC.

For example, to program your robot to move forward a certain number of encoder counts the student would write the following code:

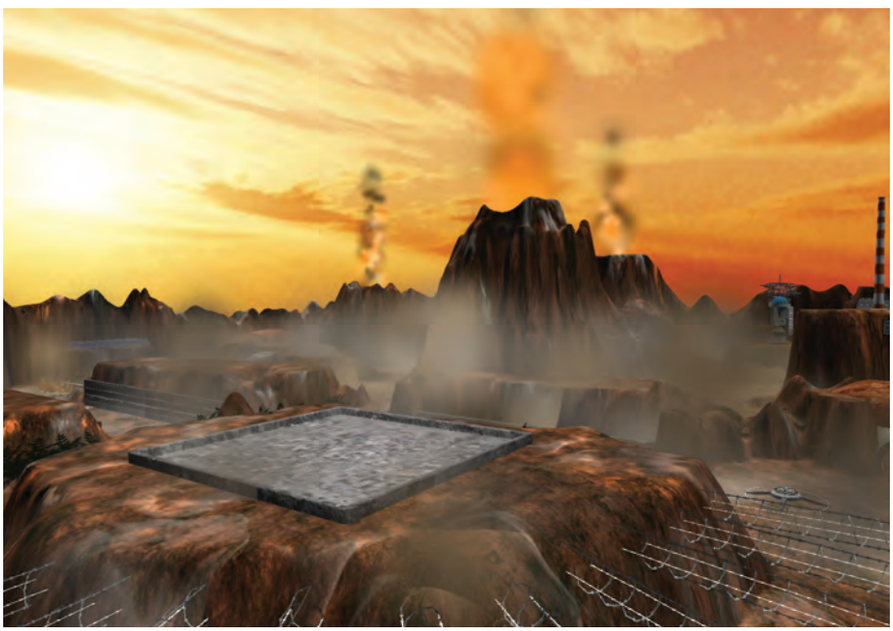
```
Forward(powerlevel);
Until_rotations(rotations, motor);
Stop(motor);
```

OTHER COOL FEATURES

There are several features about the project that students will find cool.

The virtual worlds have a full physics engine integrated into the game and the robots movements will be affected by the physics engine.

It is cool to program VEX and LEGO robots in the virtual world, but we believe that students will find it more engaging to program all terrain vehicles as well as elements in the world like updating the fuel station or monitoring the green house, mining systems, and space flying vehicles. All of the programming can be accomplished using ROBOTC or LabVIEW.



In the future we envision opportunities to link our virtual worlds to web based data. Imagine a weather station in your hometown controlling the weather on the planet!

```
15
16  Wait Until Functions
17    untilTouch();                //                   sensor port: S1
18    untilRelease();              //                   sensor port: S1
19    untilBump();                 //                   sensor port: S1
20    untilSonarGreaterThan();     // distance: 30cm,   sensor port: S4
21    untilSonarLessThan();        // distance: 30cm,   sensor port: S4
22    untilButtonPress();          // button: orange square
23    untilLight();                // threshold: 30,    sensor port: S3
24    untilDark();                 // threshold: 30,    sensor port: S3
25    untilSound();                // threshold: 50,    sensor port: S2
26    untilRotations();            // rotations: 1
27    untilDegrees();              // degrees: 360
28    wait(15.375);                // = 0:15:375
29    wait(195.726);               // = 3:15:726
30    waitInMilliseconds(61573);   // = 1:01:573
31
32
```

The Robot to the Rescue series will eventually integrate many features popular with online gaming like: status boards, league ladders, friending, as well as new world features.

Students will eventually have the opportunity to build their own robots and robot elements and integrate them into the virtual worlds.

For more information as well as access to a free download of the Planet H99 Virtual World software please go to: www.cs2n.org/h99. Stay tuned for further updates on Robot Virtual World Games in future issues of *Robot*! ◎

Links

Carnegie Mellon Robotics Academy, www.education.rec.ri.cmu.edu, (412) 681-7160

Furthering Innovation through Robotics Innovation (FIRE), http://fire.cmu.edu, (412) 681-7160

National Instruments Web UI Builder, www.ni.com/uibuilder/default.htm

Planet H99 Virtual World, www.cs2n.org/h99

ROBOTC, www.robotc.net

RobotEvents.com, www.robotevents.com

VEX Robotics Design System, www.vexrobotics.com, (903) 453-0800

For more information, please see our source guide on page 89.