

# Using Greenfoot and a Moon Scenario to Teach Java Programming in CS1

Randy J. Gallant and Qusay H. Mahmoud

Distributed Computing and Wireless & Telecommunications Technology

University of Guelph-Humber, Toronto, ON, Canada M9W 5L7

randy.gallant@humber.ca

qmahmoud@cis.uoguelph.ca

## Abstract

In this paper we describe a novel concept for teaching introductory Java programming to post-secondary students in their first year of higher education. The concept includes labs and a capstone project all linked together and all utilizing the Java-based Greenfoot programming environment. The concept is designed with two goals in mind: to improve the students experience in their first computer programming course by making it more entertaining; and to increase retention in the diploma or degree programs by peaking the student's interest early in their studies. This is accomplished through a Going to the Moon scenario we have designed and implemented into the Greenfoot programming environment.

## Categories and Subject Descriptors

K.3.2 [Computing Milieux]: Computer and Information Science Education – *computer science education, curriculum, human factors, literacy*.

## General Terms

Programming Environments, Simple Data Types, Algorithms, Design, Experimentation, Languages.

## Keywords

Programming for Fun, Greenfoot, Moon Scenario.

## 1. Introduction

Ever since the early days of the home-based personal computer, the technology and the use of the computer have exploded. There have been huge gains made in the area of hardware. Today we have technologies such as dual core and quad core processors, 64 bit processing and terabyte size storage. As fast as a new technology hits the market it will be replaced just as fast by something new and better. For this reason a lot of the market focus is on the exhilaratingly fast paced computer hardware side of the industry. This leaves the area of software design or programming hiding in the dim shadow of its counterpart.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACMSE'08, MARCH 28–29, 2008, AUBURN, ALABAMA, USA.

COPYRIGHT 2008 ACM 978-1-60558-105-7/08/03...\$5.00.

Students coming into the programming field for the first time are excited about the possibility of being the person to develop the next great action video game, or even work on the next version of an operating system or a Web browser. These same excited students soon realize that learning the art of programming is not all that thrilling; in fact, it could be quite tedious and boring. Just like any other discipline learning the basics is the hardest part. This is even made harder knowing that one must spend hours of time in front of a computer honing one's skills until the day comes where the new programmer can finally write a small useful application.

In this paper we present a technique for helping educators with their task of keeping the interest of the students by providing them with a new way to teach programming that utilizes a friendly, fun and responsive programming approach using the Greenfoot programming environment [4]. We provide a Greenfoot scenario, Going to the Moon, that involves a new set of Java-based experiments and final capstone project that we believe will provide a motivating framework for students. The rest of this paper is organized as follows. Section 2 compares and contrasts the related work. We will introduce you to the Greenfoot environment in section 3. The Going to the Moon scenario is presented in Section 4, and the proposed Lab-based approach is discussed in Section 5. Section 6 describes the method we used to evaluate the concept and our observations stemming from this preliminary evaluation. Section 7 discusses our future pilot usage of the project in real classroom environments. Our preliminary conclusions and future work are presented in Section 8. Finally, Sections 9 and 10 are acknowledgements and references.

## 2. Related Work

Since the beginning of the computer age, programming has been taught and learned. The first program most of us wrote using the Java Platform, Standard Edition [1] was a very simple "Hello World". There have been many programming languages come and go and many methods to learn each of them. This project is focused on a new way to teach Java programming to students in their first year of post-secondary education. Given this, the discussion of related work will be confined to that of the same language and at the same level.

Program developers at the University of Kent in the UK [5] in conjunction with Deakin University in Melbourne, Australia have developed a new kind of Java environment, BlueJ [6]. BlueJ is a Java development environment specifically designed to teach Java programming to secondary and first year post-secondary students. The goal of BlueJ was to develop an easy to use, interactive and visual environment that promotes learning and exploration into the world of Java. Many teaching professionals around the world

have started to use BlueJ in the classroom as a teaching aid. According to the BlueJ website currently 841 institutions around the world are officially stating that they are using BlueJ to teach Java.

Many papers have been written on the subject of how to use BlueJ in the classroom. One such paper was written by Kouznetsova, S. in April 2007 [3], Using BlueJ and BlackJack to teach object-oriented design concepts in CS1. In this paper Kouznetsova describes a course in introductory Java that uses the BlueJ environment to develop and implement the card game BlackJack. Kouznetsova's lesson plan consists of 1 introductory assignment on the basics of BlueJ and 4 more assignments using BlueJ to implement a deck of cards and play a game of BlackJack. Kouznetsova's concept is to give students partial code from an assignment and have them finish the program to achieve the correct results.

From the same people that brought us BlueJ, Greenfoot is one of the latest development products hitting the circuit. The evolution of Greenfoot has sparked a new interest and excitement in classroom learning again. Greenfoot like BlueJ was developed to aid in the teaching of introductory Java programming to secondary and post-secondary students. It is based on a world environment with visual interactions between the world and the objects in it. Greenfoot was designed as a complete open source Java environment. New world scenarios are being developed everyday. Encouragement from the Greenfoot developers has manifested itself into programming competitions. One such competition was held in 2007 at the JavaOne [8] conference in San Francisco, CA where programmers were asked to create new and exciting Greenfoot world scenarios [9].

As the development of Greenfoot grows and the population using it grows as well, new ideas for use in the classroom will become plentiful. In following Kousnetsova's lesson plan, using something familiar and fun to learn Java with BlueJ, we hope Going to the Moon with Greenfoot will be even more challenging and yet just as fun. Our project will take the best of the ideas from what has come before it and apply them to a new world. The lesson plan, consisting of ten 2-hour lab sessions culminating into one final capstone project, has got enough structure to facilitate the learning of students with the freedom to let them explore the universe of Java. This ten-lesson plan should prove to be more valuable than just the four assignments proposed by Kousnetsova's in his approach. In the end students should have a well rounded base of Java knowledge to prepare them for further Java studies.

### 3. Greenfoot

Greenfoot is a programming environment supporting Java that has been developed by the same group who developed BlueJ. It is designed specifically to convey object-oriented concepts and principles in a clean, easily accessible approach. The Greenfoot environment makes the creation of graphics and interaction easy. Students can concentrate on the application logic, and engage and experiment with objects. Applications in Greenfoot are called scenarios, and the environment enables students to run these scenarios, make modifications and re-run the scenarios where they would be able to see the effect of the changes they have made right away. Figure 1 depicts the Wombat scenario that comes with the Greenfoot environment. The large grid area is called the "world", and the right side is the "class display" showing all Java classes involved in the Wombat scenario.

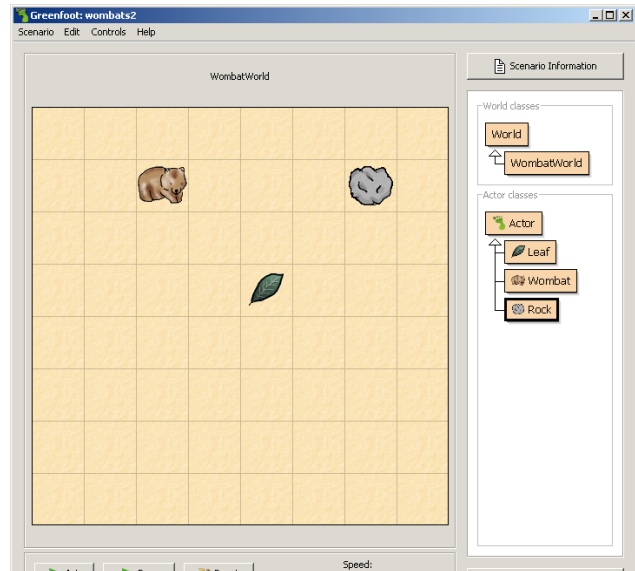


Figure 1: The Wombat scenario that comes with Greenfoot

Instead of running the whole scenario, Greenfoot enables students to invoke individual methods directly and that is a valuable feature for learners. This is demonstrated in Figure 2.

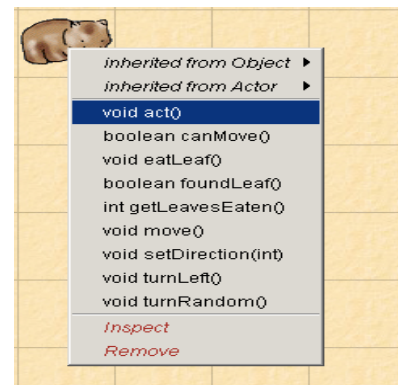


Figure 2: Directly invoking individual methods

We have designed and developed a fully functional new scenario called Going to the Moon along with ten labs and a capstone project that can be used to effectively teach Java in CS1.

### 4. Programming for Fun

The focus of this project is to provide educators with a better way to teach the fundamentals of computer programming using the Java programming language and the Greenfoot programming environment. The project will be broken up into two main pieces: the final capstone project and the laboratory experiments used to help them learn the concepts for the project. During the time that students spend in the lab they will be learning the basics of Greenfoot as well as the basics of Java programming. The basic skills that the student will develop will all come together in a final programming project for the student to complete near the end of the semester. The secret plan behind this project is to give students an opportunity to learn some basic skills without them realizing that they are learning. To do this the concepts and layout must be as entertaining as they are educational.

Being educators for years, we have realized that students like to be challenged, but only if it is fun for them. They like to think and use their brain, but it must be on something interesting to them. It is for this reason that we have chosen the idea of **Going to the Moon**.

## 4.1 Going to the Moon

The capstone project that will be implemented near the end of the semester is based around a computer version of a space vessel traveling to the Moon and returning to Earth again. Using all of the skills developed over the semester the students will be given some starting code for a scenario in Greenfoot and asked to finish the program. Along the way the space vessel will encounter asteroids and space junk which the students' program will have to identify and maneuver around. Fuel is also an issue that the students have to take into account. When they leave Earth they will not have enough fuel to get to the Moon so a refueling stop at a space station will have to be factored into the trip. The space vessel will be equipped with missiles to destroy asteroids and space junk if there is not enough fuel to negotiate around. Student must keep track of their arms and only use them as a last resort. With all of the moving around in open space the student's program must be able to keep track of where the vessel is in space at all times and constantly maneuver itself back on to the path to the Moon's landing point. Figure 3 demonstrates a screen shot of the Going to the Moon full scenario.



Figure 3: Going to the moon full scenario

Obviously the goal of the program is to make it to the Moon and then back home for the party and happy ending. This will not always be the case. Unlike the Apollo 13 mission there will be no second chance for these space travelers. The space vessel that the students will be using in their program will destroy itself if their program makes a tragic error. Some of the situations that will cause the destruction of the space vessel are as follows: Hitting an asteroid or some space junk, running out of fuel, missing the landing spot on the Moon or Earth, destroying the Moon or Earth with missile weapons or crashing into the Moon or the Earth during landing maneuvers. All of these scenarios must be avoided or the mission will be a failure.

## 5. Laboratory Experiments

As in almost any practical based learning situation lab experiments play a big role in the learning of the topics of the course. In this case there is no difference. The semester hours will be broken up to include 10 different lab periods. A different set of experiments and learning activities will take place in each lab period. The culmination of the 10 lab sessions will be the programming knowledge and skills to tackle the capstone project near the end of the semester. The weekly lab session will be designed in a way that the student will develop various skills that will be utilized in the project, without giving the solution directly to them. In other words the concepts they learn will translate to the project but not the exact code. Therefore, the students must still have to think about how to use the learned concepts to implement the project solution. Every lab has a starting scenario with particular goals in mind. For example, Lab 4 teaches the students how to detect space objects in their path and navigate around them. Figure 4 shows a screen shot of the starting scenario for Lab 4.

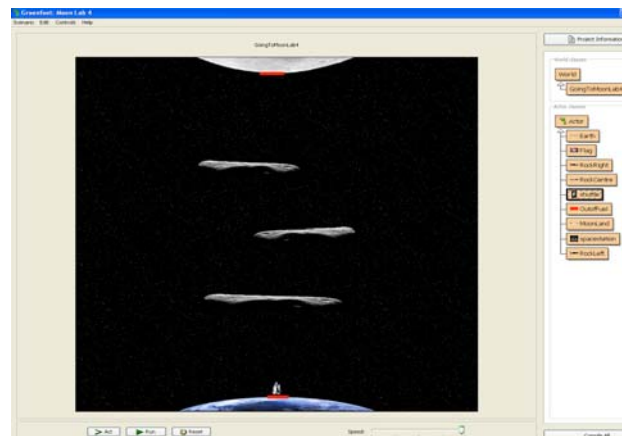


Figure 4: Starting scenario for Lab 4

### 5.1 Labs 1 & 2

In the first two labs the importance will be placed on the programming environment used for this course, Greenfoot. Students will be given the opportunity to download and install the Greenfoot environment from the Greenfoot website. The students will be introduced to some of the created scenarios predefined in Greenfoot. Students will be shown where and how changes are made to scenarios and how they are saved to disk. Students will be asked to run and play with some of the predefined Greenfoot scenarios to promote them to a comfort factor with the Greenfoot environment before they start their actual programming labs.

### 5.2 Labs 3 to 7

In these labs students will become familiar with basic Java programming concepts that will form the building blocks for the capstone project at the end of the term. Some of the basic concepts that will be discussed are: Simple data types, calling of functions and methods, loop structures and decision structures such as *if* and *switch*. In these labs counters as well as the prefix and postfix notations will be discussed and used along with other basic math functions. The use of the *new* operator and the constructors will be looked at and utilized as well as the concept of Objects and Inheritance. Reference materials for all Java labs

may come from the Java developer's website [7] or textbooks such as *Big Java* by Cay S. Horstmann [2] or other resources.

All of these labs will take place in the Greenfoot environment and will be on predefined small scenarios that are made up of individual parts of the trip to the moon. Such things as keeping track of your fuel, maneuvering around objects and keeping track of your position in space will be a few of the lessons learned in Labs 3 to 7.

### 5.3 Labs 8 to 10

In these labs more advanced topics will be used. The students will start to look at the graphics involved with the Greenfoot environment and manipulate the structure of existing graphics and even create their own. Sounds and other advanced Greenfoot operations such as keyboard control will also be experimented with in these later labs.

## 6. Current Evaluation

To validate the project idea and to see what computing students would have to say about the Going to the Moon approach, we turned to 3<sup>rd</sup> and 4<sup>th</sup> year computing students at the University of Guelph-Humber, because they have gone through our CS1 course without using Greenfoot or the Going to the Moon project. 42 students were given a survey that had 10 questions regarding their personal experience with the current technologies, environments and methods used in the labs. Questions covered topics such as entertainment value, assignment preparation, interest in programming and their final grade.

In examining the results from this short survey we made the following seven observations about the Going to the Moon concept and the currently used command line IDE.

1. Students like the lab work.
2. Students don't dislike the idea of command line but they would prefer a more entertaining and interactive programming environment if one is available.
3. Students use the lab sessions as preparation for their given assignments.
4. Students want and like to be entertained.
5. If they are entertained they become more interested in what they are doing.
6. Overwhelming majority said that the "Going to the Moon" labs and project excite them and make them look forward to what they can learn in future courses.
7. Based on the students marks from question 7, and the fact that they have made it this far, the command line programming environment does work but leaves something to be desired.

## 7. Future Evaluation

In the next few months the Going to the Moon project will be put to the ultimate test. A group of graduating high school students will be the first to put the project through its paces in a special credit project. Following the high school students, the Going to the Moon project will be used as the main curriculum in an introductory Java programming course at Humber College in Toronto in the fall of 2008. These two tests should give us some

good feedback and will hopefully prove that this idea and concept will work in a post-secondary environment.

## 8. Preliminary Conclusions and Future Work

Learning to program is a challenging task and requires a lot of hard work and dedication. Unfortunately for today's student this is considered quite boring. The students of today prefer video games to board games, so with that in mind their educational institutions should change accordingly. To get the best out of the students, they have to be challenged, tested and entertained. Using the Java development environment, Greenfoot, our project, Going to the Moon, should provide the entertainment that the current post-secondary students need to keep them interested and involved in their own learning process. In the end the hope is to graduate more students and ones that have a real interest in the skill of programming and not just students that are going through the motions to get their diploma or degree.

Once proven as an effective technique to teach Java, the future development of this project will revolve around the creation of new Greenfoot scenarios that follow the same formula as the Going to the Moon concept and therefore adding a variety of choices to the educators that choose to use our method in their classrooms. We are currently in negotiations with various Professors and Program Coordinators who have expressed interest in the project and wish to utilize the Going to the Moon package in their classrooms and labs. We hope to receive feedback that will help us in improving this scenario and in making the labs more easily understood for the beginner programmer. We are also investigating the possibility of having the Going to the Moon scenarios used as a series of labs for students in distance education. Being self-contained in a simple environment it lends itself to the possibility of making a very good distance education tool.

## 9. Acknowledgments

Many thanks to the students who participated in the survey that provided us with valuable feedback for our "Going to the Moon" project. We also like to thank fellow professors who have shown interest and support in this project.

## 10. References

- [1] Java Platform, Standard Edition: <http://java.sun.com/javase>.
- [2] Horstmann Cay S., *Big Java 3<sup>rd</sup> edition*, Wiley Publishing, 2008.
- [3] Kouznetsova, S., Using BlueJ and Blackjack to Teach Object Oriented Design Concepts in CS1. CCSC Central Plains Conference April 2007, <http://www.bluej.org/papers/2007-04-kouznetsova.pdf>.
- [4] Greenfoot: <http://www.greenfoot.org>.
- [5] University of Kent Computer Science Department: <http://www.cs.kent.ac.uk>.
- [6] BlueJ Home: <http://www.bluej.org>.
- [7] Java Developer Resources: <http://java.sun.com>.
- [8] Greenfoot Programming Competition at JavaOne2007: <http://www.greenfoot.org/javaone/competition-flyer.html>.
- [9] Greenfoot Competition Winners at JavaOne2007: <http://www.greenfoot.org/events/javaone.html>.