



Drexel-SDP GK-12 ACTIVITY

Cherry Picking Programming Activity

Scratch / Handiboard: Computer Programming and Mathematical Reasoning

Scratch Computer Programming

“To loop or not to Loop, that is multiplication!”



Grade Level _6_ (_5_-_7_)

Time Required: 2 Hours

Summary

In this activity, students will use the Scratch computer programming environment available for free download from MIT. The objective is for students understand the connection between “looping,” a common computer programming technique for encoding an action multiple times, and multiplication in context, an arithmetic technique as applied to a problem in the real world. The higher level objective is for students to strengthen their logical and mathematical reasoning skills while gaining some proficiency with computer programming and exposure to the career choice of computer science.

Students will receive a working “Cherry Picker” program, as depicted in the screen shot. The problem is as follows:

- the cherry picker can jump and pick 8 cherries at a time
- the cherry picker’s basket can only hold 100 cherries
- how many times can the cherry picker jump without overflowing his basket?

The point is for students to visualize multiplication/division and when to round up/down to the nearest integer. In other words, the cherry picker can only jump 9 times without overflowing his basket.

The Scratch programming environment allows children to build computer programs using colorful, easy-to-understand building blocks much like Legos that can be dragged and dropped onto the page.

Students will receive a working but “buggy” program in which the cherry picker is “looped” to jump to the tree too many times, thus overflowing his basket as shown in the picture above. The students will have the job that a real computer programmer often has, to “debug” the program by changing the number of times that the jumping activity is looped.

Evaluation and discussion is an important part of this activity. The activity will be followed by a discussion of how the students figured out how many times the cherry picker should have jumped (counting?? Subtracting?? Dividing??) and how they were able to fix the program. There will also be a discussion of how “looping,” whether in a computer program or in real life is just like multiplying and rounding in the appropriate direction. (e.g. suppose the students have a 4-1/2 day week at school. How many times do the students have to get up and get dressed for school? How many times do they pass through the door of the school?).

Engineering Connection

Students will be doing the job of a computer scientist or computer engineer – “debugging” a computer program.

Keywords

- Computer science
- Computer programming
- Technology
- Computer-aided instruction
- Scratch
- Computer
- Mathematical Reasoning

Educational Standards

- Science: Technological Devices 3.7 (A. Tools, C. Computer Software)
- Math: Mathematical Reasoning and Connections 2.4

Learning Objectives

After this lesson, students should be able to:

- Understand what a loop in computer programming does
- Understand how a loop in computer programming is like multiplication in math
- Be able to reason about when to round a number up or down, depending on the context of a problem.

Materials List

Each group needs:

- A computer with the Scratch programming environment installed

To share with the entire class:

- A projector for showing the image on a large screen

Introduction / Motivation

There is so much in children’s school, play, and home life that would not be possible without computers. Imagine not having any computer generated graphics in the movies. No *Batman*? No *Finding Nemo*? No *Transformers*? Unthinkable! Imagine life without calculators. Would we use an abacus? Our fingers? Paper and pencil?

Computer programs are made by people just like us who understand how to tell a computer what to do in language that the computer understands. The instructions have to be short and very, very simple. How would we tell a computer to find the lunchroom if it didn’t have “eyes?”

Computer programs use a lot of the logic and repeated instructions that we use every day and just don’t realize it. For example, how do you decide to take an umbrella? (“If it is raining, or if my Mom tells me that it might rain, then I take an umbrella. Else, I leave the umbrella at home.”) How do you know how many shoes to put on? (“For each foot, put on one shoe.”) The students already use the everyday logic that is necessary to tell a computer how to do it.

Vocabulary / Definitions

Word	Definition
Computer science	A field of study based on information storage and processing systems including hardware, software, basic design principles, and user requirements.
Computer programming	Creating a sequence of instructions that cause the computer to do something.
Looping	Executing the same set of instructions a fixed number of times OR until a desired outcome is achieved.
Debug	To find and remove errors in a computer program.
Rounding	A method of approximating a number.
Integer	A whole number.

Procedure

Background

Because all of the Drexel instructors for this activity are in some field of computing, talk briefly about our work, how we became interested in it, and why we enjoy it. Talk about all the great places (exotic conference locations, away to college) that our studies have taken us. Close with some statements about how the field of computing is changing rapidly as we see it, and all the opportunities for the students in their future (gaming, movie special effects, satellite programming, military operations, etc).

Before the Activity

- Define the vocabulary words and then discuss how the students will take on the job of a computer programmer for this activity.

With the Students

1. Explain that they will be given a working, but “buggy” program in which the cherry picker’s basket overflows. Point them in the direction to find the bug, to try to identify portions of the code that instruct the cherry picker to jump a repeated number of times.
2. Let them experiment with the code that is there, reminding them that there is always the original version to return to if they modify the code beyond their liking (e.g. REMEMBER TO HAVE THE STUDENTS SAVE THE PROGRAM UNDER THEIR OWN NAME SO THERE IS ALWAYS AN ORIGINAL VERSION TO RETURN TO).
3. After 30 minutes playing with the program, find out who was able to correctly modify the code and find out how they identified the correct portion of the code, and how they figured out the number of times to loop the cherry picking action. Also, ask the students who did NOT figure it what they had trouble with.
4. Discuss how looping in computer programs is like multiplication. Give the example cited in the summary above to start a discussion of how sometimes it is appropriate to round up to the nearest integer, and sometimes appropriate to round down.
5. Now, students can identify and derive the formula for each variable in the program. This is done by taking each number (the number of cherries picked so far, the number of cherries that can fit in the basket, the number of cherries picked on each jump, and the number of jumps needed to fill the basket as much as possible) and describing it in words. This translates into a Scratch variable for each item. Then, derive formulas for computing each variable using the others. This can be modeled in the Scratch program using the “Numbers” tab to create expressions that use the variables. To use a variable, simply drag it into the number oval of the expression. Students should modify the program so that the variables are used instead of hard coded values, and then test their program with various basket sizes and numbers of cherries picked on each jump. NOTE – if the number does not divide evenly, for example, 100 cherries in the basket with 8 cherries picked per jump, the program may compute the number of jumps needed as 12.5 instead of 12. It may be necessary to round the numbers down in Scratch.

Troubleshooting Tips

If the students get themselves into a code modification that will not run, if you have the knowledge to help them debug it, then do so, if not, tell the student to return to the original version of the code and start over.

Investigating Questions

What kinds of toys and entertainment do computer programmers have a role in? What would our life be like without computer programming and computer scientists? For how many years has the computer been available for everyday household use? What did the first computers look like?

Redirect URL

<http://scratch.mit.edu/download>

Owner

Drexel University GK-12 Program

Contributors

Bill Mongan, Dara Kusic, Quincy Brown, Elaine Gabarine

Copyright

Copyright 2007 Drexel University GK12 Program. Reproduction permission is granted for non-profit educational use

Version: Mar 2007