

Amazing Mazes

Materials and Teaching Checklist – lesson 9

Lesson Name: Advanced Walker Programming 1

Date to be taught: 5/7/2013

“I Can” Skills:

Last Time	This Time	Next time
Create and test maze walker programs with loops and conditions	Create and test more advanced maze walker programs (breadcrumb algorithm) Start creating display and presentations for the WOW!	Finish creating and testing more advanced maze walker programs (breadcrumb algorithm) Finish creating display and presentations for the WOW!

Before the Lesson:

Copies to Make	Materials to Bring	Visuals to Make
	<ul style="list-style-type: none"> - Projector - Computers with Internet connectivity - 	

During the Lesson:

Section	Time (min)	I Say / Do	They Say / Do
Hook	5	Review and Teach-back <ul style="list-style-type: none"> - Quick review and teach-back of last lesson: - Student volunteering to role-play maze walker. Class instruct student in executing left-hand walk, using class desks as “maze walls” 	<ul style="list-style-type: none"> - Students guide/instruct volunteer student to execute a left-hand walk in class
Activity 1	10	Summary of left/right-hand algorithms <ul style="list-style-type: none"> * The teacher copies and pastes the maze below the program at http://employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html (this is a “replicated H shape” from previous lessons). * The teacher creates a maze walker at (1,5), and asks the students to call out the location of the target which would result in the longest search (path, number of steps) by the walker. ○ This is a trick question, since the target location that's “hidden the best” depends on the walker algorithm. ○ For a left-hand walk the best 	<ul style="list-style-type: none"> - Students call out the “best location” for the target. But this actually depends on the walker algorithm/walk that we select to execute: - for left-hand walk (3,4) - for right-hand walk (3,6)

		<p>target location is at (3,4)</p> <ul style="list-style-type: none"> ○ For a right-hand walk the best target location is at (3,6) <p>* The teacher places the target in both places and shows how both algorithms/walks find the target</p>	
Mini Lesson	10	<p>The motivation/questions for this mini lesson:</p> <p>When is left/right-hand walk NOT successful?</p> <ul style="list-style-type: none"> - Using the programming algorithms applet http://employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html the teacher draws a simple square maze (e.g. one corner of the square at (1,1), and the length of each side 5), and then adds one intersecting line at (3,3)(3,8) <p>draw-path (1,1)(1,6)</p> <p>draw-path (1,6)(6,6)</p> <p>draw-path (6,6)(6,1)</p> <p>draw-path (6,1)(1,1)</p> <p>draw-path (3,3)(3,8)</p> <ul style="list-style-type: none"> - Then teacher asks students to volunteer and place the walker and target in such a way where the walker will not find the target using a left-hand walk - In many cases, if a left-hand walk doesn't work, a right-hand walk will work. <ul style="list-style-type: none"> ○ Show a couple of cases that work ○ Ask a volunteer student to place the walker and target in a way that BOTH walks don't succeed 	<ul style="list-style-type: none"> - Students should try different combinations of walker and target. For example: <ul style="list-style-type: none"> ○ walker at (1,1) and target at (3,3) will not work for a left-hand walk - For a walker to fail to find the target with both a left and right walk, place the walker at (3,3) and the target at (3,8)
Activity 2	15	<p>Bread-crumbs algorithm</p> <ul style="list-style-type: none"> - From the last activity, it is clear that there is a need for a better walking algorithm. Ask the students for ideas. - They will probably end up with the idea for the bread-crumbs algorithm: <ul style="list-style-type: none"> ○ As the walker goes through the maze, it will leave a trail of "bread-crumbs" wherever it goes. ○ As it gets to any intersection the walker will check to see which direction has not bread-crumbs (i.e., it hasn't been there yet). 	<ul style="list-style-type: none"> - Ask students to come up with ideas for a better maze walking algorithm (compared to left/right-hand walk) - Assuming they will propose something close to a "bread-crumbs" algorithm, ask students to explain the details and reason for it to always work.

		<ul style="list-style-type: none"> ○ If all paths at an intersection have been traversed before (i.e. all have bread-crumbs), the walker will chose to go in the direction with the “least crumbs” (meaning the least times traversed) 	
Activity 3	15	<p>Testing out the bread-crumbs maze walking algorithm</p> <ul style="list-style-type: none"> • Ask the students - Using the programming algorithms applet http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-ultimate-algorithm-1.html copy and paste the “grid maze” below the applet (it is the one with most “maze loops” and therefore difficult for left/right-hand walk algorithms, but not for the bread-crumbs algorithm) • Place the walker anywhere and the target someplace else • Run the bread-crumbs algorithm, and show that it will fail to reach the target 	<ul style="list-style-type: none"> - Students will copy and paste the grid algorithm, and place walker and target in multiple places, then run the bread-crumbs algorithm and convince themselves that it finds the target every time!
Activity 4	15	<p>Prepare for the WOW!</p> <ul style="list-style-type: none"> - Using http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-drawing-saving.html students will create the “fanciest” maze for including in the slide-show to be displayed at the WOW! - They will save the maze by uploading it to the server, giving it a unique name (e.g. the concatenation of the first names of the students in each pair) 	<ul style="list-style-type: none"> - Students create a “fancy” maze and upload/save it on the server with a unique name.
Exit Tix	10	<p>Teacher to make sure that each pair has uploaded at least one maze that they have created for inclusion in the WOW!</p> <p>-</p>	
Dismiss		<p>Remind students of our goal regarding the WOW!:</p> <p>We will show how we learned to build mazes and how to program “fool proof” programs (programs that don't fail or break)..</p>	

Thumbnails lesson outline:

- **Review and teach-back** of lesson – ask students to model with classroom desks a left-hand-walk maze walking program.
- **Teacher - NetLogo -Programming Algorithms – show longest path for walker to find the target in the “replicated H” maze**
 - **Trick question:** left-hand or right-hand algorithms require different positioning of the target, in order to find the target in the largest number of steps
- **teacher – show how left/right-hand algorithms fail in a simple square+line maze - NetLogo - Programming Algorithms**
 - motivate students to find/define the bread-crumb algorithm idea
- **Students - NetLogo – Ultimate Algorithm – test the bread-crumbs algorithm**
 - Use the “grid maze” which is difficult (or impossible) for left/right-hand algorithms
- **Students prepare for WOW! - NetLogo – Drawing-Saving**
 - Draw a fancy maze and save it on the server – to be included in the WOW! Slide show