EXPANDED LEARNING BY CITIZEN SCHOOLS

Amazing Mazes

Materials and Teaching Checklist – lesson 6

Lesson Name: Maze Walking Algorithms
Date to be taught: 4/16/2013

"I Can" Skills:

Last Time	This Time	Next time
Create a basic maze walker program to control the walkers	**Create and test maze walker programs with left-hand and right-hand algorithms**	**Create and test more advanced maze walker programs (breadcrumb algorithm)**

Before the Lesson:

Copies to Make	Materials to Bring	Visuals to Make
	- Projector - Computers with Internet connectivity -	

During the Lesson:

Section	Time (min)	I Say / Do	They Say / Do
Hook	10	**Review and Teach-back** - Quick review and **teach-back** of inside-maze view commands and programs: Ask for a student volunteer. With the help of the class, write on the board all the inside-maze view commands we covered: - forward, turn-left, turn-right, turn-back Using the maze walker programming algorithms program http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html, ask the volunteer to copy and paste from the webpage one of the mazes (either replicated H from previous lessons, or the simple grid maze) Ask the class to call out a position for the walker, and the volunteer will create it. Same for a position for the target. Ask the volunteer to open the command-line, and ask the class to call out the inside-maze view commands necessary to have the walker go from the initial position to the target. The volunteer will slow down the execution (using the slider at the top), and run the program to see if the walker gets to the target.	- **One student volunteers to "drive the program at http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html** - **Students participate in creating the walker, target, and inside-maze view program for the walker.**

Activity 1	10	**Hands-on: creating a maze and a matching inside-maze view maze walking program:** **- Ask the students, using** maze walker programming algorithms program http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html, **to create a maze, then create 1 maze walker, and 1 target anywhere in their maze, and write an inside view program to walk their maze to the target.** **Note: in order for the walker to be able to walk the maze, all the maze paths/lines need to be either horizontal or vertical, but cannot be diagonal!** - Teacher needs to make sure that the program works, by asking the students to delete the maze walker, recreate and reposition it in the same initial position and rerun their walker program, to show that it repeatedly/reliably works.	- **Students use the program at** http://www.employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html,**to write a maze walker program.** - **Students rerun the walker program a couple of times to make sure it works.**
Activity 2	15	**Maze Walker algorithms** **Motivation: Now that we know how to write an inside view walker program, will it always work (find the target)?** **Can a maze-walking program really find the target in any maze?** * Left-hand walk inside a maze – student participation: – example 1: * arrange 3 classroom desks in a U shape, and declare a maze target at the end of one arm of the U shaped "maze" (it can be on the outside of the arm) * ask for a student volunteer and place them at the end of the other arm of the U shaped maze. * ask the volunteer to put their left hand on the edge of the table, close their eyes and just follow the edge, always staying in touch with the tables ("the maze wall") until they reach the target – example 2: * draw a simple H shaped maze on the board, where each part of the shape is wide like a path or "street" * put a target at the end of one arm of the H, and with a marker placed as the walker at the end of a different arm, trace a left-hand walk all the way to the target	- **Students volunteer and participate in the activities**

		– Example 3: * ask a student to draw another, more complex maze on the board, with wide paths (like "streets"), and have another volunteer trace a left-hand walk from a starting point to a target. – Teacher explains that an algorithm is a certain way and a sequence of steps to solve a problem in a way that the computer can execute. An algorithm is like a recipe for a dish that can be executed by a cook/chef.	
Activity 3	15	**Inside maze view program effectiveness – playing a game: "hide that cheese" (target)** * Teacher shows the new Java applet at http://employees.org/~hmark/courses/amazingmazes/amazing-mazes-12-programming-algorithms-1.html and explains the new elements on the User Interface: ○ "Draw with mouse", "Left-hand walk", "Right-hand walk" * Then the teacher will explain the game "Hide that cheese" (target): – Students will build a maze of their own design, either using the "draw-path" button of the User Interface, or the "Draw with mouse" button. – One student in the pair will then place 1 walker anywhere in the maze, and the other student will place the target in the maze, in a way that they think will take the walker the most time/steps to reach. – Then they will start the left-hand walk algorithm and count the steps the walker takes to get to the target. – The students will switch roles, and see if they can find a longer path (basically trying to "hide" the target from the walker). – Teacher will tabulate the results (longest path) for each pair. The longest (i.e. the cheese is hidden the best) is the winner.	- **Students will play the game "Hide that cheese" (target)** - **Students will try to hide the cheese so that it takes the walker the longest to find.** - **The teacher will tabulate the longest paths, and declare a winning pair.**
Activity 4	15	**Summary of left/right-hand algorithms** * 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	- 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)

		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 target location is at (3,4) ○ For a right-hand walk the best target location is at (3,6) * The teacher places the target in both places and shows how both algorithms/walks find the target	
Exit Tix	10	If we have time: - Hands-on activity: ○ the students can try to create a maze and hide the target in a way that **neither a left-hand walker nor a right-hand walker will find the target** (it is possible to create such a maze!). -	**Students in pairs, build a maze and try to hide the target so that neither walking algorithm hits/finds it.**
Dismiss		Remind students of our goal regarding programming: ○ We will learn in the next lessons how to program "fool proof" programs (programs that don't fail or break), but we'll do it in stages, refining and strengthening our programs every time.	

Thumbnails lesson outline:

- **Review and teach-back** of lesson – **Volunteer** – **NetLogo Algorithms program** - copy & past maze, create walker and target. **Class** - call out inside-maze view program to successfully walk the walker to the target

- **Students - NetLogo - Algorithms program** – draw their own maze, walker, target, and program an inside-maze view program to walk the walker

- **Students** - NetLogo – **Algorithms program** – create **their own maze**, a walker, a target
  - Program the walker using inside maze view commands to run through their maze

- **Walking algorithms** – demo with Students
  - rearrange tables, Teacher draws "street maze" on board, Students draw "street maze" on board
  - Definition of algorithm

- **Student Game "Hide the cheese" - NetLogo -Programming Algorithms**
  - **Students create a complex maze** – 1 student places walker, 1 places target
  - see if left-hand or right-hand algorithms can find the target in the largest number of steps

- **Teacher demos/summarizes "Hide the cheese" - NetLogo -Programming Algorithms**
  - using the "replicated H shaped maze", place walker at (1,5), ask students where the best location for hiding the target is? (depends on which type of walk/algorithm)

- **If we have time:**
  - **students program Inside Maze view program for a maze, trying to create a maze and target location which both left and right hand walks will not find**