CS125 Section 3 **Maze algorithms and Finite State Machines**

Activity #1 **The tiler and the cheeser set out to solve a riddle… (about 20 minutes)**
Read the following carefully…
Check you have: a partner, some mazes, ONE cheese, FOUR action tiles: back, forward, left and right, TWO condition tiles ("facing-wall?" and "facing-space?"), ONE test sheet and ONE loop sheet.

You and your partner will write an algorithm using only the given tiles to help the cheese find its way out of the maze. You may not need all of the small tiles.

The loop sheet is a "While true" loop - it executes forever. All other tiles should be placed within the box on the loop sheet. You will also need to use the TEST (conditional) tile.

One orange condition may be placed over the "<condition>" on the TEST sheet.

The red tiles are actions (back, forward, left, right), which can be placed next to the YES /NO statements, or after the condition.

More than one tile can be placed next to the YES, NO positions.

The forward and back tiles instruct the cheese to move forward "one space", which is roughly one body-length, or one cell on the maze sheet. The right and left tiles instruct the cheese to turn right or left exactly 90 degrees.
Work together with your partner to construct, test, debug, and ultimately discover a correct algorithm that lets the cheese escape all of the mazes from either entry point. (You do not need to test on all four mazes, but it is a good idea to test your algorithm on a couple of mazes.)

Choose initial roles (either tiler or cheeser) but swap occasionally. The "tiler", constructs the algorithm with advice from the cheeser.
The cheeser carefully steps through the program, one statement at a time, and moves the cheese as dictated by your program. The tiler reads each statement and queries the cheese-master, who moves the cheese.
e.g. "Are you facing a wall?" "No." "Okay, move one step forward".

Testing: Can your cheese navigate from start to finish without getting stuck or moving onto a wall? An example, incomplete attempt might be:

facing-wall? YES: *back* 1 , NO: *forward* 1

CLEANUP Please make sure to return the six tiles back into the envelope, and to return the other sheets and the cheese token.

Activity #2 **Finite State Machines (about 20 minutes)**
The TA will give you instructions.

Attribution: The ideas and text are primarily from CS Unplugged and Prof. Pitt and Prof. Heeren at UIUC.

Today's Lab – Read the following carefully

Lab achievement points are based on the following -
"Impress us!" Present your HW0.
Look I can use the debugger! I can single-step and also use breakpoints to pause the execution.
Look I can switch perspectives (Java/SVN/Debugger) and know how to reset a perspective.

The first assignment "MP" Challenge1-DebugMe consists of three short programs - Factorial, Winner, QuizMaster.
You'll need to checkout the project from your subversion repository.
Then see the R-E-A-D-M-E--F-I-R-S-T that is included in the MP.
Make sure you COMMIT your changes. COMMIT means to commit your files back to subversion.
If you work on two computers you'll need to know about UPDATE too.

Lab Objectives
Announcements and demo (see below)
Ask all students to log in. During section ask each student to show you where they are with Turing's Craft.
Joined late? Sign up (pay $) for Turing's Craft if you haven't done so.
If necessary, demo complete TC by using the lecture page pre-reading sections.
Get students going with the first MP.
COMMIT code at end (always Checkout /Update ... Later Commit) - so that you can even work on two machines.

The lab assistants will demonstrate:
Checking out the project.
Running a unit tests. Running one test.
Starting the debugger.
Checking the autograder output and MP scores -

https://subversion.ews.illinois.edu/svn/sp16-cs125/YOUR-NETID/autograder/

You have a local autograder and units tests. Use them!
But MP Score is maximum of ALL grading runs that Lawrence runs.
i.e. Make sure you commit everything and there are no compile errors.
Each time the real autograder is run it creates a new subdirectory.
You can review the output of the tests there and the compile log.

Please log in to Turing's Craft. During the lab, the lab assistants will check your progress and help you if you're stuck.
Announcements
MP1 is out!
MP1 will be graded in one week's time - Monday night 8pm

I can click but can't seem to change Java files in Eclipse?

You're looking at the file on the subversion server. You need to check-out the project to the local computer first. Right click on the project and select Check-Out. Then use the Package view to select the file that you want to run or edit.

Other common problems and questions

I can edit the files but I can't run them! You need to check out "Challenge1-DebugMe" as a project, not your netid.

What do I need to do? Read the ReadMe file inside the project.

How do I finish? Make sure you commit your project back to the server. Remember there must be no (red) compile errors anywhere in your project.

I have a compile error in TextIO.java - change the compiler settings to 1.6

I have a compile error in a test class - these classes call your code. Modify your code to fix the test's expectations.

I'm lost dazed and confused - see help from the course assistants.

How do I work with two computers?

Before finishing today COMMIT your changes back to subversion.

When going back to a different computer use UPDATE to bring in the latest changes from subversion.