

(<http://csmatters.org>) 0 - 3

0b0 - 0b11

Intelligent Paper

Unit 0. Introduction

Revision Date: Jul 22, 2019

Duration: 1 50-minute session



Lesson Summary

Summary: This lesson is a basic introduction to algorithms and the nature of intelligence. Students will play tic-tac-toe (noughts and crosses is the British version) between a “highly intelligent piece of paper” and a human. Students will explore how to create an algorithm and the concept of computer intelligence.

Outcomes

- Students will play tic-tac-toe using a specific algorithm.
- Students will create and test a new algorithm for playing tic-tac-toe.

Overview

1. Getting Started (5 min) - Journal
2. Introduction (5 min) - PowerPoint
3. Guided Activities (35 min) - Students play tic-tac-toe with a given algorithm, then practice designing and implementing their own.
4. Wrap-up (5 min) - Students create definitions and assign homework for Lesson 1-1
5. Optional Activity (5 min) - Students examine an actual "AI" (simple game playing program) created for tic-tac-toe in Python

Source: This lesson is adapted from a lesson created by Paul Curzon, Queen Mary, University of London.

Learning Objectives

CSP Objectives

- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*

- LO CRD-2.C - Identify input(s) to a program.
- LO CRD-2.D - Identify output(s) produced by a program.
- *EU AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*
 - LO AAP-2.A - Express an algorithm that uses sequencing without using a programming language.
 - LO AAP-2.G - Express an algorithm that uses selection without using a programming language.
- *EU IOC-1 - While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.*
 - LO IOC-1.A - Explain how an effect of a computing innovation can be both beneficial and harmful.

Common Core ELA:

- WHST 12.2 - Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes

NGSS Practices:

- 8. Obtaining, evaluation, and communicating information

Essential Questions

- How do computer programs implement algorithms?
- How does computing enhance human communication, interaction, and cognition?

Teacher Resources

Student computer usage for this lesson is: **none**

A PowerPoint for this lesson is included in the Lesson Resources folder - IntelligentPaper.pptx and IntelligentPaper.pdf

Copies for student pairs of "intelligent paper directions" with tic-tac-toe directions on one side, and blank on the other - in the Lesson Resources folder - IntelligentPaperDirections.pdf

The wrap-up questions are available in the Lesson Resources folder as Questions To Consider.docx

Optional: a musical greeting card, a paper folded into a fortune teller (http://en.wikipedia.org/wiki/Paper_fortune_teller (http://en.wikipedia.org/wiki/Paper_fortune_teller)), a page of equations

The Python program for the optional activity is located in the Lesson Resources Folder - TicTacToeAI.py

Lesson Plan

Getting Started (5 min) - Journal

What could make a piece of paper intelligent? (Think-Pair-Share)

Introduction of Content (5 min)

(Use IntelligentPaper.pptx in the Lesson Resources folder to help deliver this lesson.)

Challenge the students by saying that you have a piece of paper that is at least as smart as any human. (Show the blank side of the paper, don't tell the students yet, but it has directions on how to play tic-tac-toe on the back.) Ask if anybody believes that this is possible.

Show students examples of "smart papers," such as:

- A page full of equations
- A paper folded into a fortune teller http://en.wikipedia.org/wiki/Paper_fortune_teller (http://en.wikipedia.org/wiki/Paper_fortune_teller)
- A greeting card that plays music when you open it. It's made of paper, is it intelligent? (it's the chip inside that's smart, and somebody had to program it to make it do something... same with any paper, a person needs to put the information on there to "give" it intelligence)

Encourage discussion and debate, prod students to argue their point for or against intelligence, and get them to develop their own criteria and definition for intelligence. Write the class definition and criteria on the board.

Guided Activities (35 min)

Part 1 - Game Activity (5 min)

Tell the class that the paper has never lost a game: it has perfect intelligence.

Challenge students to play a game against the paper. The paper is peripherally challenged (it has no arms, and thus needs somebody to do its work for it). One person represents humankind, while the other person represents the paper. Play tic-tac-toe with a partner. The paper must begin the game.

Possible outcomes:

- Paper wins or draws.
- Humans give up. (They often do.)
- Try again and see if it's just luck?
- Humans cheat. (They sometimes do.)
- Humans cause an error. (It happens.)

But, the paper WILL NOT LOSE.

Try letting humankind go first. (Wait and try it: The paper will lose. Why?)

Part 2 - Independent Activity (15 min)

Challenge students to write out detailed directions (an algorithm) that will never lose the game whether it goes first or second. Point out that algorithms are made by putting detailed steps in order (sequencing), making choices depending on the situation (selection) and repeating certain sections (iteration).

Part 3 - Algorithms, Programs and Computers (15 min)

Students should use their new algorithm to play against each other. Follow the same model for the paper versus the human game.

Discuss:

Computers programs implement algorithms.

Computer programs carry out algorithms by controlling the computer memory, central processing unit, input and output. When we carried out our tic tac toe algorithm what did we do that corresponds to each computer component?

Memory:

Central Processing Unit:

Input:

Output:

Additional Possible Activities and Discussions Time Permitting:

- Students investigate how many different, unique games can possibly be played in tic-tac-toe.
- Try and represent all the possibilities.
- Point out, after they give it a good try, that using symmetry cuts out a lot of redundant possible moves.
- Have students share how they kept track of the many different possible games there are.
- Discuss machine learning and artificial intelligence.
- You can show them some ideas if you just do an image search for tic-tac-toe:
 - For example, <http://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Tic-tac-toe-game-tree.svg/300px-Tic-tac-toe-game-tree.svg.png>
(<http://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Tic-tac-toe-game-tree.svg/300px-Tic-tac-toe-game-tree.svg.png>)
- What about 3D tic-tac-toe? http://en.wikipedia.org/wiki/3-D_Tic-Tac-Toe
(http://en.wikipedia.org/wiki/3-D_Tic-Tac-Toe)

Wrap-up (5 min)

See if students know what a CPU is. If not, define. [

CPU is the abbreviation for **central processing unit** or **processor**, the brains of the computer where most calculations take place.

Reflection: (written in journal or discussed as a class)

How is a computer CPU (processor) similar to/different from a human brain? (both process information, can use logic and math, can follow instructions; computers only do what they are told, can work without need of food or sleep, etc.)

A human only has 1 brain but a computer can have multiple CPUs. How could a computer with multiple processors (CPUs) divide up a task like playing tic-tac-toe? (This is a good optional exercise if there is extra time, to divide up what the steps are and see if any can be done in parallel to save time. example: one processor checks rows, one checks columns to see if there is a winner)

Have students write their own definitions for the four words at the end of the presentation:

- Computer program
- Artificial intelligence
- Peripheral
- Algorithm

Optional Activity (5 min)

(Use PyCharm or some other Python environment to show the TicTacToeAI.py program from the Lesson Resources folder.)

- Follow slides 14-16 of the PowerPoint to review the Python "AI" written for TicTacToe. Encourage discussion on how a computer can "think" in order to win the game.
- Play a few rounds against the program and see whether a student can figure out how to beat it.
- Discuss how the program could have been better designed.

Homework

Assign homework for Lesson 1-1: Provide students a copy of the "Questions to Consider" in the resources folder and assign the reading:

Blown to Bits – Chapter 1, can be found here <http://www.bitsbook.com/wp-content/uploads/2008/12/chapter1.pdf> (<http://www.bitsbook.com/wp-content/uploads/2008/12/chapter1.pdf>) and is available in the lesson resources folder for Unit 0 Lesson 3.

Options for Differentiated Instruction

Extension: If you have extra time, have a championship contest between one set of student-generated instructions and another, alternating who goes 1st and 2nd. You can work in groups of three, with one person acting as the judge if desired.

Evidence of Learning

Formative Assessment

Vocabulary entries in journals from the end of the PowerPoint presentation

Group participation in interactive activity

Writeup about a more general solution



(<http://www.umbc.edu/>)



(<http://www.umd.edu/>)



(<http://www.nsf.gov/>)

Authored by: CS Matters in Maryland

Website: csmatters.org (<http://csmatters.org>)

Email: csmattersinmaryland@gmail.com (<mailto:csmattersinmaryland@gmail.com>)

This work is licensed under a
Creative Commons Attribution-ShareAlike 3.0 United States License
(<http://creativecommons.org/licenses/by-sa/3.0/us/>)

by University of Maryland, Baltimore County (<http://umbc.edu>) and University of Maryland, College Park
(<http://umd.edu>).