Paper Robot [DRAFT]

Summary

Robots can be extremely useful and can do a lot of work for us. However, we need to be able to tell them what to do. We can do that using a coding language where the robot will do exactly what we say and exactly how we say it. In this activity, students will make their own paper robots that they code to draw pictures.

Grade Level: K-2nd

Subject: Computer Science

Length: 60+ min.

CSTA/Common Core Standards Alignment

CSTA - Algorithms and Programming - 1B-AP-08
Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
Algorithms to draw an image can be altered and refined to produce better results.

CSTA - Algorithms and Programming - 1A-AP-09
Create programs that use variables to store and modify data.
Use a symbolic language to store and manipulate data. The symbolic language can be used to create new instructions, which are combinations of others, by storing instruction data.

CSTA - Algorithms and Programming - 1A-AP-11
Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
The image can be separated into bits and then tackled one section at a time. Images allow the students to visualize decomposing a problem.

Computational Thinking Alignment

Problem Decomposition
The image can be separated into sections, or sub images. Each sub image can be approached individually and then combined to create the whole image.

CS@Mines
Contact us: cs@mines.edu

COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

Paper Robot [DRAFT]

Algorithmic Thinking
The process of developing an algorithm to draw or represent an image is the process of algorithmic thinking.

Data Representation
The image data is being represented as a series of cells on a paper.

Pattern Recognition
Recognizing patterns in the image can aid in solving the problem efficiently.

Objectives

In this class, students will learn about programming through a hands-on activity with paper robots. The students will first decorate their robots with various materials that they can later take home. They will then be given a programming language via a set of cards, where each card represents one programming instruction. The students will practice reading and writing code with this programming language by drawing and decoding pictures with their new robots. They will learn iterative programming through testing their code and refining it to make it better.

Timeline

Activity	**Time**	**Activity Location**
Introduction	10 min	Lesson Plan
Make Robots	15 min	Lesson Plan
Draw a Picture from Code	5 min	Example Execution Slides
Decode a Picture	5 min	Lesson Plan
Code Pictures	20 min	Lesson Plan
Discussion	5 min	Lesson Plan
Total	**60 min**	

Contact us: cs@mines.edu

Paper Robot [DRAFT]

Materials

- ❏ Dixie/Paper Cups — 2 per student
- ❏ Pipe Cleaners — one per student
- ❏ Hot glue gun — 1
- ❏ Hot glue sticks — 3-4
  - NOTE: You could replace the hot glue with masking tape.
- ❏ Other decoration materials (ideas below) — as needed
  - Tin foil, construction paper, glue, stickers, rhinestones, googly eyes, buttons for eyes
- ❏ Colored Pencils/Markers — one per student (more encouraged)
- ❏ ImageDrawingWorksheet.pdf — >1 per student
- ❏ RobotDrawingWorksheet.pdf — >1 per student
  - NOTE: The more images the students are able to draw, the more time this activity will take. Print more worksheets if you want the activity to last longer.
- ❏ CodeWritingWorksheet.pdf — 1 per student
- ❏ InstructionsPage.pdf — 1 set per student
- ❏ PracticeExampleSlides — 1
- ❏ ExampleExecutionSlides — 1

Attachments

- ImageDrawingWorksheet.pdf
- RobotDrawingWorksheet.pdf
- CodeWritingWorksheet.pdf
- InstructionsPage.pdf
- PracticeExample.pptx
- ExampleExecutionSlides.pptx

Preparation

Print and cut out the instruction set provided in the InstructionsPage.pdf. Each student should have a set for themselves. Consider using thicker paper for the instruction set so they can be re-used.

You may also consider preparing the paper robots in advance. Construction of each robot requires gluing two cups together with a hot glue gun. Or maybe prepare a glue station where students bring their robots to have you glue the two cups together. An alternative is to create the robot with masking tape instead of glue; glue is preferred for decoration reasons only!

CS@Mines
Contact us: cs@mines.edu

COLORADO SCHOOL OF MINES
EARTH • ENERGY • ENVIRONMENT

Procedure

Introduction

Consider showing this video to introduce what robots are and how they can be used https://youtu.be/UuPAImipntw.

Introduce what robots are at a general level. Ask your students what they think a robot is. Once they develop a general idea, discuss how robots are simply machines that can carry out physical work on their own. Robots can do complex work like assembling cars in a factory, or simple work like vacuuming your carpet.

Vocabulary

- Robot: Comes from the slavic word *rabota* which means *labor* or *work*
- Language: The set of instructions that the robot can follow
- Program: An ordered list of instructions that the robot will follow to draw an image

Robots do a task exactly as we tell them to. Ask your students to consider how we can talk to robots. Allow students to brainstorm a couple of ideas and stop if someone mentions coding. Since robots do exactly what we tell them to, we need a way to give them instructions.

Tell students to think about how you might tell your friend to tie their shoes. You would start with “grab both ends of the laces” then “cross them over each other” then “pass one under the other” then “make bunny ears” and so on. If you miss a step, your friend isn’t going to be able to tie their shoes correctly! This is the same for robots. We will use our robot language in this activity to instruct our robot to draw a picture. Each image represents an action. Go over the actions with the students; explain that the arrows represent an absolute direction of travel on the board and not a relative direction.

Then discuss how robots will follow the instructions given to them exactly. Even if the instructions are wrong, the robot will still follow that instruction. This can cause problems for robots and computers; we call these errors “bugs” in our code.

Contact us: cs@mines.edu

Paper Robot [DRAFT]

Body of Lesson and Activities

Make the Robots

Starting with two paper dixie cups, put a hole through the top and the bottom of the cup for our drawing utensil to go through, in order to allow our robot to draw pictures. Depending on the age group of the students, you may want to complete this step for them. Next, glue the two cups together such that the top of each cup is touching. Now we have the body of our robot! Allow the students to decorate their robot as they would like. Some ideas are putting on googly eyes, using pipe cleaners for arms, “color” it with tin foil or construction paper, and so on.

Introduce the Language

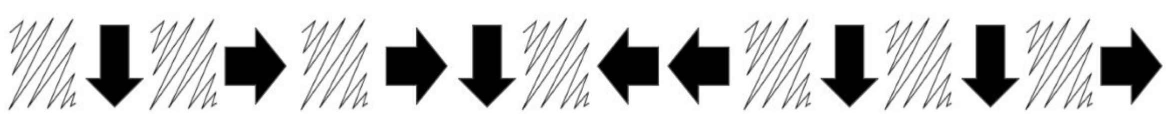
The language used to write programs for the robots consists of five instructions:

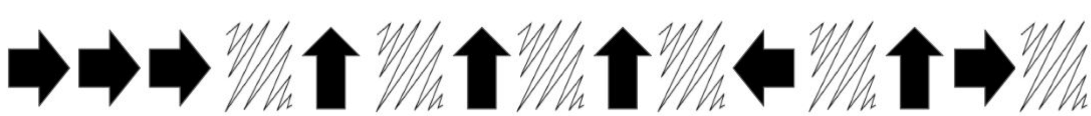
- Move Up
- Move Down
- Move Left
- Move Right
- Fill Cell

The **movement instructions** will move the robot one cell in the specified direction. Movement instructions that cause the robot to move off the grid are not valid in the language.
Fill instructions will fill a cell. If the cell is already filled, the robot will do nothing.

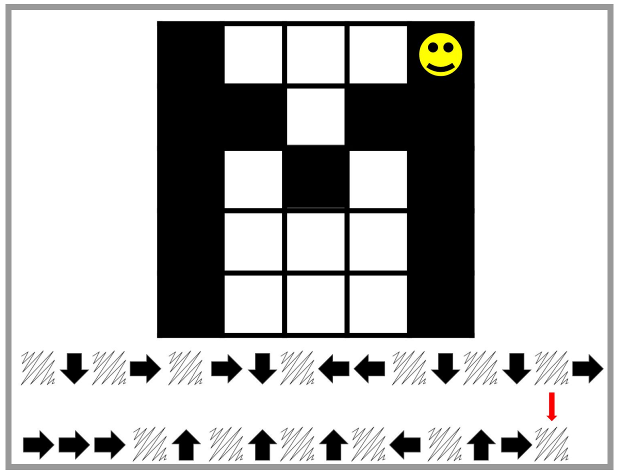
Activity 1 - Draw a Picture from Code

In order to learn the robot language being used, ask students to draw a picture from code that is given to them. They will draw their picture on the RobotDrawingWorksheet. The code given to them can either be printed on worksheets and handed out, or shown on the board using a projector (see PracticeExample; slide 2). Example slides are also available to demonstrate the execution of our instructions (see ExampleExecutionSlides).



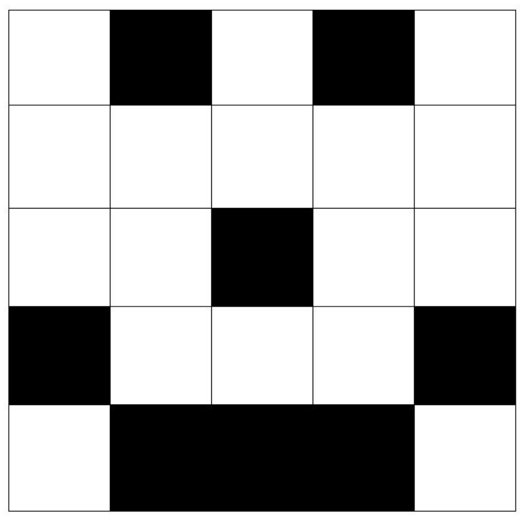


The robots will “draw” the pictures by moving across the grids and filling in specific squares (see RobotDrawingWorksheet). The students will move the robot according to the instructions given to them. Remind the students that robots follow exactly the instructions listed. Read one instruction, then execute the instruction exactly how it is written, then move on to the next instruction. Demonstrate this with the first 2 or 3 instructions in the example to show the students how it will work. Then let the students try it on their own. When the robot is moving, suggest the students draw squiggly lines on the grid paper for “fill cell” instructions; later, students can fully color in the squares where squiggly lines are.



Show the students the picture their robots should have drawn. Did their robots draw it right? What went wrong, and how can they fix it for their next picture?

Activity 2 - Write Code from a Picture



Show the students the example picture (right OR in PracticeExample; slide 3) and have them create the code that the robot would use to draw this picture. Students will create their code by placing the instructions, given in InstructionsPage, inside the lines of the CodeWritingWorksheet.

Considering the entire picture may be overwhelming, so show the students where the starting point is (upper left hand corner), then break down the picture into smaller parts. These smaller parts could be straight lines, curving sections, or individual squares at a time. It might also be helpful for the students to write a few instructions at a time, then have their robot execute them. It is easier to test a couple instructions at a time than test the whole picture; testing a few instructions at a time allows students to catch errors in their code, called bugs, early in the coding process. When students are done, show them the cirect answer. What could they do to improve their code next time?

Contact us: cs@mines.edu

Activity 3 - Write Your Own Code!

Now we get to the fun and creative part! Allow the students to create their own pictures and then write code for those pictures. Hand out the ImageDrawingWorksheet. These worksheets have a grid, similar to the one the robot travels on, which the students can use to draw their own picture. The goal will be to write code to draw their picture, then have another student/student's robot draw the picture, then refine their code based on if their partner drew the correct picture.

Step 1: Draw the Picture
The students should first draw their picture on the given worksheet. Encourage students to start with a simple picture that they can add to later (if desired). A simpler picture will assure they have time for another student to test the code and then refine it.

Step 2: Decode the Picture
Similar to Activity 2, the students should write the code that draws their picture. The students should place their code on the CodeWritingWorksheet. Just like before, students should plan to have the robot begin executing the code in the upper left hand corner.

Step 3: Test the Code with a Partner
When a student has their code done, have the student find another student in the class who is also done with their code. Each student will then have their robot execute their partner's instructions to draw the picture. Make sure students don't leave their initial picture (step 1) near their code! (We don't want the student's partner to know what the picture should look like!) Once each of the partners are done, compare the picture drawn by "the robot" to what the picture should have looked like. Students should ask:

- Did their robot draw the code correctly?
- Where did the picture go wrong?
- Is there an error in the code or in the drawing of the code?

Step 4: Refine Code
If there were "bugs" or errors in the student's code, then the student should fix their code and test again. Have the students think about strategies to narrow down bugs in their code such as (1) stepping through instruction by instruction or (2) looking at where the image started to divert from the original image.

If a student didn't have errors in their code, the student can work on shortening their code with new instructions. These new instructions can represent combinations of other instructions. They should not include robot movements that did not exist before. An example of an instruction that the student could make might be a "Draw line up" command, which is the combination of multiple “move up” and “fill cell” commands. This new instruction allows the student to easily draw a line up, which is useful if the student’s picture has many multiple vertical lines.

Step 5: Keep Coding

As time permits, students can create code for another picture! They should follow the same process as they did for their first picture. Their images can get more complex and challenging. Students can also create more combination commands in order to shorten their code.

Discussion

1. What are some examples of robots you have seen in the real world?
2. Do robots always follow instructions perfectly?
    a. Yes they do! But that doesn’t mean the instructions are what was intended.
3. Did anyone make a new instruction that was a combination of others?
    a. If they did, great! Discuss why they did, and what the outcome was of using the new instruction. Ideally, they were able to shorten their code by using their new instructions.
    b. If no one had time to develop a new combined instruction, talk about how/why we would create one, hinting towards the concept of a variable. Creating a new instruction allows us to repeat actions much easier than writing out each instruction separately. Imagine we wanted to draw a box; we could condense the drawing of each side into single instructions that are combinations of multiple move right/down/left/right instructions. What would have been a long line of code has now been reduced down to two instructions. In this way, we can use our new instructions as variables to store combinations of other existing operations.

Modification

The instruction set can easily be modified in the activity. As an example, the “fill cell” instruction could be changed to a pen-up/pen-down instruction, which would require more knowledge of the state of the robot to operate. Additionally, an “erase cell” instruction could add extra challenge to the activity if the students were looking for something more challenging.