

CMSC131

Introduction to Computational Thinking
(with links to some external Scratch exercises)

How do you accomplish a task?

When faced with a task, we often need to undertake several steps to accomplish it.

We can think of this as a set of instructions, a recipe, a strategy, an algorithm, etc. but regardless of the term we use, it is a way of expressing a series of steps to follow.

What language to use?

Just as there are many natural languages in which we can present a series of instructions to humans (English, French, Russian, etc.) there are many computer languages available. The language is a tool that you use to convey your instructions.

In CMSC131, you will be learning about how to construct instructions for a computer to follow as well as the **Java** language which you will use to express these instructions. In this first exercise, we will use a visual language called **Scratch**.

The Walking Fed

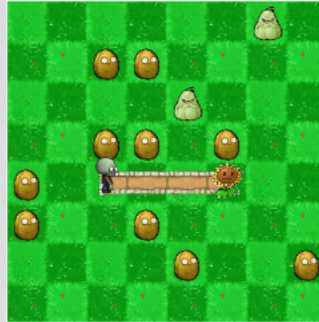
Imagine you needed to give directions to a generally mindless, yet somehow trained, zombie who wanted to feed on a sunflower but avoid things that would kill it. Your directions would need to be very precise since this zombie would do what you said, and only what you said.

Imagine for now that you only had 3 atomic instructions available in the programming language; (1) **move forward one space**, (2) **turn right 90°**, and (3) **turn left 90°**.

Stay on the path!

While you might want to tell the zombie in this puzzle to “walk on the path until you get to the sunflower,” that is not an atomic instruction.

Your goal is to keep the zombie on the path?
What would you give as the instructions?



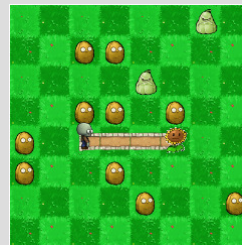
Stay on the path! Solution.




In this case, keeping the zombie on the path and having it reach the sunflower would require the following steps:

Move Forward

Move Forward

Move Forward



Visit ter.ps/Zombie01 and drag and attach three of the  bricks into the workspace area, the first connected to the  and then the next two under the previous one, and then click  to see it work.

Dealing with turns!

In the next challenge, your zombie needs to follow a path with turns along the way.

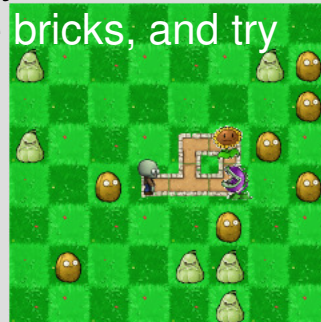
Not only do we want to get the zombie to the flower, we also want to get it there with the fewest instructions! What would they be?



Dealing with turns! Give it a try.

Visit the following page and drag and click the appropriate bricks for your solution onto the workspace and then click “Run” to see it work. <http://ter.ps/Zombie02>

If it doesn't work the first time, you can click on “Reset” and then change the bricks, and try again.



Dealing with turns! Solution.

After the first step forward, your zombie needs to turn. You have to imagine you are the zombie when giving directions, so the zombie must turn to its left.

The zombie is then ready to take another step forward. At that point it will need to turn again. This time, to its right.

It is then ready for one more step forward and a yummy sunflower.

Must walk further!

In the next challenge, your zombie needs to follow a path with turns along the way, and the path is longer.

Visit <http://ter.ps/Zombie03> and give it a try.



That was exhausting...

You may have noticed that, similar to in the first puzzle, you had to tell your zombie to move forward several times in a row.

One of the topics that we will cover is called “iteration” where you can have a program do the same line(s) of code several times, which can make a program more compact and more flexible.

Bigger Challenges

Thinking through how to instruct a computer to solve a challenge is the first stage. Then, we use a programming language to express our instructions to a computer, and test that we are correct.

There are a variety of computational thinking exercises online in Scratch, often with coding blocks made to fit the problem. You might even find the challenges at ter.ps/wonderw interesting later in the semester.

Programming Tools

For this semester, we will explore computational thinking, problem solving, coding, and software testing using a general purpose programming language called **Java** to both implement and test your solutions to various challenges.

As we introduce each new programming design concept, via Java, there will be new challenges assigned whose solutions can be expressed utilizing these tools. For each, your task will be to think through how you will instruct a computer to solve it, using Java.

Copyright © 2015-2017 : Evan Golub