CSC 204 - Lab 2: Writing Jeroo Methods

Now that we've worked with Jeroo for a few days, it's time to write some methods.

Goals:

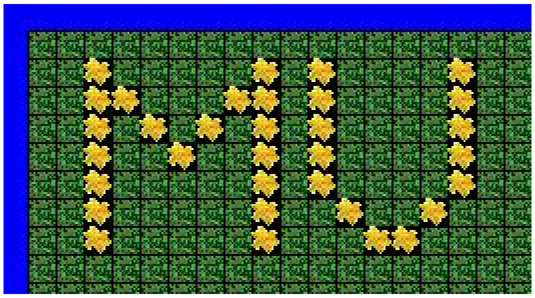
After this lab you should be able to create worlds in Jeroo and to create Jeroo programs which use action methods, Jeroo methods, and if and while statements. This is a lot! We'll spend most of the rest of this semester doing the same things in Java. If you get a good understanding of what's going on now, the pieces of the rest of the semester should fit together well.

Lab Setup:

Create a new folder named "Lab2" to work in today. It would be best if you create this Lab2 folder in your Orion folder. Copy the files from the folder Lab2 on Blackhawk into your Lab2 folder. You can also find a copy of these files on Google Drive. You should find:

- `NorthTest.jsc,`
- `Path1.jev,`
- `Path2.jev`, and
- `Path3.jev`

Sequences in Jeroo



As you have already seen with your Jeroo initials assignment, the simplest thing to do in Jeroo is to write a single main method with a long list of Jeroo instructions. But that can get very boring, very fast, so it should inspire us to want to write methods.

You should always plan your code so you don't waste a lot of time. (A favorite saying among programmers is "Weeks of programming can save you hours of planning." You might need to read that a few times to understand it.)

Jeroo Methods

faceNorth

There may be times when you want to have the Jeroo face in a particular direction, like North. Let's write a Jeroo method called `faceNorth`.

First, think about the problem. One way to solve this is below:

- If the Jeroo is already facing North, there's nothing to do.
- If the Jeroo is not facing North, it should turn left until it is facing North.

Be sure you understand this algorithm before going on.

This can be coded as the Jeroo code:

```
while (!isFacing (NORTH))
{
      turn (LEFT);
}
```

Open the file `NorthTest.jsc` in Jeroo. Before adding anything, try running the program and notice the error you get when you try to run a program that calls a method that is not defined.

Add your code for `faceNorth` and make sure it works. After running the program, all of your Jeroos should be facing north.

If you have questions about how this works, try running it step by step.

An aside: It would be nice if we could use arguments when writing Jeroo methods, so we could have `face(NORTH)` and `face(WEST)` but since Jeroo is a simplified system, we can't. Don't worry, we'll be allowed to do this in Java.

Exploring further

Once you get this program to run, try making intentional mistakes in it. Some things you might want to try (one at a time, please!) are:

- Change "`main`" to "`Main`".
- Change the order of the arguments to one of the constructors (try `Jeroo (EAST, 4, 4)` or `Jeroo (4, EAST)` for example).
- Change the spelling of `faceNorth` to something else.
- Remove a set of parentheses.
- Remove one of the curly braces.

All programmers will make errors like this, so the sooner you can recognize them and know how to fix them, the better. You don't need to turn anything in for this. Make sure your program runs when you're done.

followPath

A more difficult task to give your Jeroo is to follow a path of flowers. For this exercise, the path of flowers will have no branches and will only turn at 90 degree angles. There are three sample paths in the files `Path1.jev`, `Path2.jev`, and `Path3.jev.` Take a look at these.

The algorithm we're going to use, in English, is:

```
face an adjacent flower
while there is a flower in front of the jeroo
    hop
    pick the flower
    face an adjacent flower
```

It makes sense to write a method to face an adjacent flower. A first impulse would be to make it like the faceNorth method:

```
while (!isFlower (AHEAD))
{
      turn (LEFT);
}
```

This *will* work when there's a flower, but try it when there is not a flower next to the Jeroo. Be sure you can trace the code.

faceAFlower

Instead, make a method that will face an adjacent flower by just checking 3 times, using three if statements, since the Jeroo will not have to turn more than 3 times.

Once you can face a flower, create a method to follow a path that uses it and the algorithm above to follow a path. Finally, create a program with a Jeroo that will follow a path and try it on the paths given.

Save this work in the file `Path.jsc` .

Once you get your Jeroo to follow a path, try commenting out the line that picks the flower. How does the Jeroo's behavior change?

Deliverables:

Lab work is due by class time on Monday. When you are finished, copy your “Lab2” folder into your shared Google folder so your teacher can grade your work. Your entire Lab2 folder, and all of its contents, should be copied into your shared Google folder.