

CMRoboBits (15-491) Fall 2007

Homework 2

Due Wednesday, September 19th 2007 at the beginning of lab (12:30pm)

<http://www.andrew.cmu.edu/course/15-491/>

Submission

Your entire homework should be submitted by copying it into the “**dropbox/lab02**” folder on your personal AFS space before the due time. Please read the website for more information about your AFS dropbox. If there is anything in particular that you would like us to know about your submission, please place a README.txt in the same directory.

Part 1. Custom Activities in VPL

a) Create a new VPL activity

In part 2 of your first homework you created a VPL diagram to implement a custom turning behavior. To make this code more re-usable, you will now encapsulate it in a VPL “Activity”.

Your VPL activity should contain a single action named “Turn”. This action should have the same three Input values as your program in part 2 of the last homework:

DriveSpeed (type double, range 0.0-1.0): the speed of the robot.

TurnRadius (type double, in centimeters): the turning radius of the robot.

TurnRange (type double, in degrees): how far to turn.

Your action should have a **single output value named “Success” of type bool**.

When calling the “Turn” action, your activity should perform the turn as implemented in your last homework. It will then wait until the turn is completed (until the drive is stopped) and finally return the boolean “Success” output. Generally this output should be true, unless e.g. the DiffDrive returned false for some reason.

This part should be fairly straightforward and you should be able to simply cut and paste most parts of your last homework into the VPL activity.

b) Draw the Fibonacci Spiral

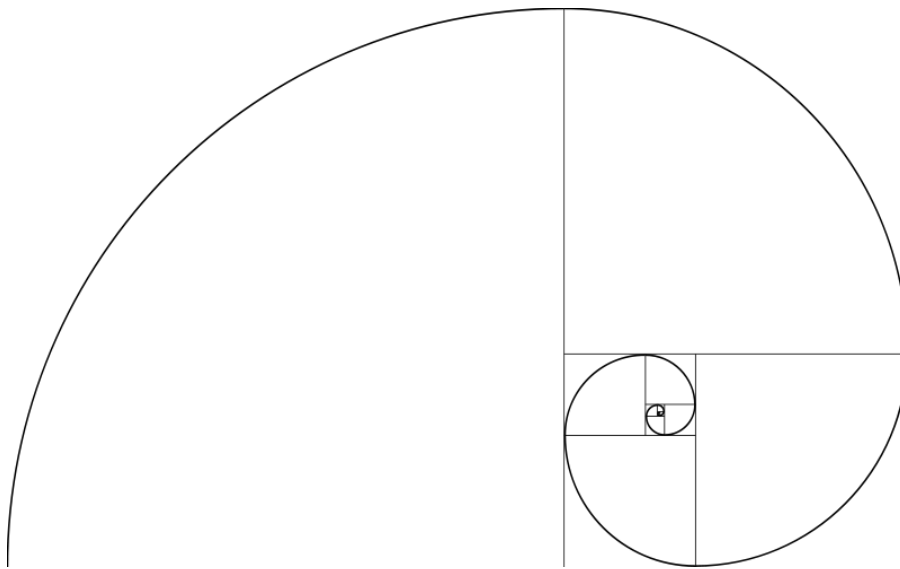
Now that you have encapsulated your turn-behavior into a re-usable VPL activity, it is time to actually make use of it. In this part of the assignment you should make your robot draw a “Fibonacci Spiral”. A Fibonacci spiral is drawn by consecutive 90 degree arcs (quarter-circles) with radii corresponding to the numbers in the Fibonacci sequence.

Recall the definition of the Fibonacci sequence:

$$F(n) := \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

Your program should ask for a single input of type int (by using a simple dialog) representing the number of iterations after which the robot should stop. On each iteration your program should compute the current Fibonacci number and use the turn activity from part a) to drive the corresponding 90-degree turn with a radius of the particular Fibonacci number. **You should use the Scribbler robot for this assignment.**

The resulting path of your robot should look like this:



Your program should compute the current Fibonacci number on the fly for each iteration. Please do so efficiently by always storing the previous two results in variables, thus only requiring to add and update them on each iteration. You will need to use VPL variables for this. You will also need to use a VPL variable to be used as the loop counter. Your robot may draw the spiral at whatever drive-speed works best for you. Also, if your spirals grow too large too quickly, it might be a good idea to simply scale down the result of the Fibonacci sequence. For example use $F(n)/10$ instead of $F(n)$ as the radius...this would mean $F(n)$ would correspond to millimeters rather than centimeters, which might make for a more dense, and thus more interesting looking spiral. Feel free to experiment with this.

IMPORTANT NOTE: Dry-Erase Pens, a large roll of paper, and tape are available at the lab. Feel free to tape sheets of the paper on top of the carpet so the scribbler can draw on it with the dry-erase markers. **Please use the supplies sparingly!** Use the paper multiple times (use different colored markers). Flip the paper around and use the other side when one side is full.

ART-SHOW AT LAB-TIME: Once you know that your robot does the correct thing, take a blank piece of paper of the size you need, and let the robot perform its spiral on it. Please bring your sheet with the finished Fibonacci Spiral and your name on it, to lab on Wednesday, September 19th. If you forget to do so, you will need to demo it during lab-time.

Please save your project (including the activity) in the dropbox/lab02/part1 folder on AFS.

Part 2. Sensors + Closed Loop Control: Building a line-follower

To be demonstrated at lab-time!

This will be your first implementation of a closed-loop controlled robot. You will use a scribbler to create a simple line-follower. In particular, your robot should do two things:

1. It should follow a line on the floor using its bottom Line-Sensors (use the “Scribbler Line Sensor” service).
2. It should check its front-sensors (“Scribbler Bumper” service) to scan for obstacles (e.g. walls). If it detects an obstacle, it should stop, perform a 180 degree turn in place and resume following the line (now going the other way on it). Thus, if one were to create a line with walls at each end, the robot should infinitely go back and forth on it.

It will be your job to figure out how to make the line-following the most reliable. Depending on what data the sensor returns (which is also for you to figure out), you might want to correct the robot’s path depending on how far you are away from the line. We would like to see reliability over speed, so if you need to reduce the DriveSpeed of the robot to deal with latency issues, that’s ok.

Please save your project in the dropbox/lab02/part2 folder on AFS.

Please create a write-up describing the approach you took, and place it in a README.txt file to be placed in the “dropbox/lab02/” folder.

IMPORTANT: Be ready to demo your solution at lab-time on Wednesday, September 19th!

Good Luck!