

Homework 4

Assigned in Laboratory 4

Due Start of Laboratory 6

Please perform the following activities in groups of up to three people. Although you are allowed to talk with other people outside your group regarding assignment requirements and debugging, the work of each group must be its own—code must not be shared from one group to another.

Assignment files

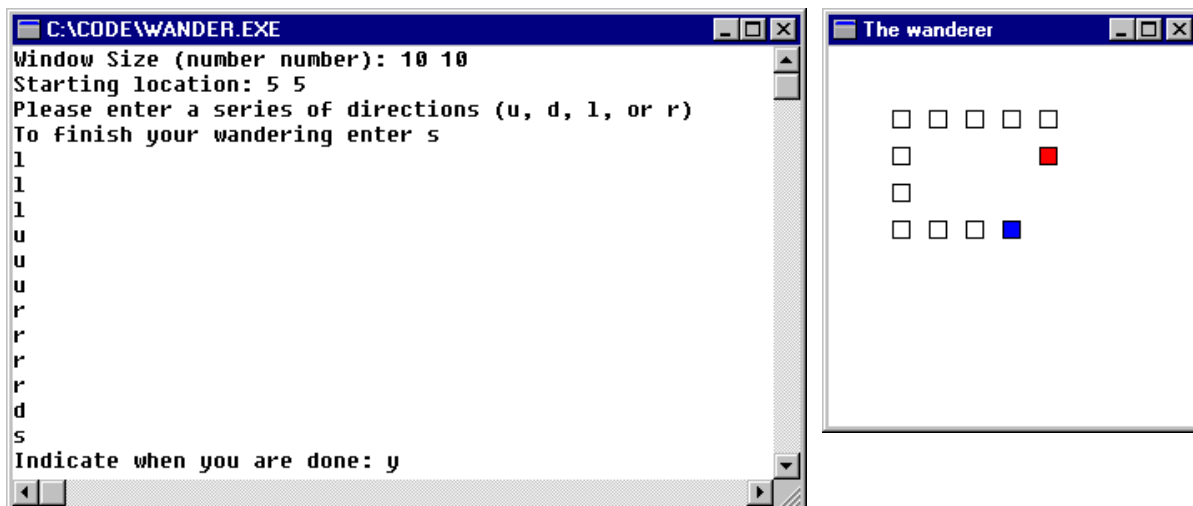
A self-extracting archive `hw04.exe` has been created for this assignment. The archive will produce two files `wander.ide` and `wander.cpp`. Archive `hw04.exe` can be acquired by accessing the course home page `www.cs.virginia.edu/cs101` or on many of our university servers in the directory `F:\public\csfiles\cs101\hw`. If you are copying this file using a web browser, first copy the file to a hard drive and then use the file manager to copy the file to a floppy if need be.

Objective

Gain experience with graphical programs, conditional and looping constructs, data validation, and control programs. In a control program, an external device is manipulated. In particular, you will develop a program that controls the movement of a simulated robot.

Problem

To extract a series of requests that determine the path of our “robot”. The robot will be represented using a square that is 0.25 centimeters on a side. The initial inputs are the size of the rectangular terrain our robot is allowed to wander. The terrain width and length are to be integer positive numbers less than or equal to 15 centimeters. The next inputs are the initial location of the robot. These values are also integers. The starting location must lie on our terrain. The initial position of the robot is given by a blue rectangle. The next inputs are a series of requests indicating in which the direction the robot is to take a step. These requests are given as a characters: *l* for left, *r* for right, *d* for down, and *u* for up. Each step brings the robot 0.5 centimeters in the indicated direction. The current position of the robot after each step is given using a white rectangle. When you are done controlling the robot, issue the command *s* for stop. The last position of the robot is to be redrawn over using a red rectangle. To terminate the program type an additional character. A sample input/output behavior along with the graphical output associated with this input/output behavior is given below. Observe that rectangles are drawn using a black perimeter (therefore we can see white rectangles).



If the terrain size or initial robot position is illegal, the program should display an appropriate error message and terminate. You can cause your program to terminate while within `ApiMain()` by using a return statement—return statements are not limited to being the last statement in a function body.

If a robot step would take it out of the terrain, an appropriate message should be displayed and the program should terminate.

By defining a `RectangleShape` object inside the body of the loop and then drawing that object, a robot path can be displayed.

Other notes

- Source file `wander.cpp` is to be completed for your assignment. A hard copy of `wander.cpp` is to be handed in at the start of the scheduled laboratory and an electronic version is to be submitted either prior to or at the start of that same lab. No other files should be submitted. If you have inadvertently submitted other files, please delete them. Do not delete the source files that you submitted for previous assignments.
- Your program file should contain the standard header. It should properly identify all members of your group. It must contain the pledge. All group partners must be from the same lab section. There should be only one submission per group.
- When you run the program, it is suggested before typing inputs in reaction to the prompt, that you resize the console window, so that it and the graphical window are both visible and do not overlap.
- If you decide to develop your program on a non-ITC machine, you must make modifications to the project file. Instructions on how to modify a project file were given in the handout for the second programming assignment.
- The program should follow course programming style guidelines. The guidelines are contained in the course workbook and on the course web page.
- The grading criteria for this assignment can be found on our course web page.