Late Day Clarifications

Pseudocode

The syllabus says that you can use Late Days for late projects - however, the syllabus does not explicitly say how this works with pseudocode. The SLs and I talked it over, and we decided on the following policy:

- Late days cannot be spent on pseudocode. Turn it in on time, or you’ll just miss that part of the project grade.
- You must turn in your pseudocode using turnin; email will no longer be accepted.

(We were flexible about both of these points in the first two projects - but starting with Project 3, we’re going to enforce this.)

Java code

Starting with Project 3, if you email us your code (instead of using turnin), it will count as a Late Day, even if you do it before the deadline. Please use turnin. And remember: if you are having trouble connecting from home, it’s always possible to come to the 228 lab, and upload your file using a thumbdrive.

1 Introduction

In this assignment, you will implement a simple game. Using Scanner to read from the keyboard and some simple StdDraw commands to draw a picture, your game will have a player which walks toward a target. When the player reaches the target, the game ends.

I will be posting a video to Panopto, which shows my solution to the program. I would encourage you to go watch that video first, and then come back and read this spec.

2 Program Overview

In this game, the player (represented by a blue square) will walk toward a target (represented by a red square). The player will type commands on the keyboard
(n,s,e,w) to move the player in one of the four directions. After each move, your program will redraw the screen, so that the player can see their own position, and that of the target. When the player reaches the target, you will end the loop, print a congratulatory message, and then end the program.

2.1 Pseudocode

As is normal, you are required to submit pseudocode two days before the project’s due date. Your pseudocode doesn’t need to deal with the details of exactly how Scanner or StdDraw work, but it must show where you will do certain things - including any relevant variables.

For instance, when you want to draw the player on the screen, the following line of pseudocode would be appropriate:

\[
\text{draw player at (x,y)}
\]

(assuming that x and y are the variables that you are using to store the player’s current position).

Another place to watch out for in your pseudocode is in the loop condition - how will you create a loop which will end if either of two different things happen? (Your program must end if the user closes the keyboard. But it must also end if the player reaches the target.)

2.2 Requirements

To receive most (but not all) of the points for this assignment, your program must:

- Use Scanner to read from the keyboard
- Call StdDraw.setScale(int,int) to set the scale to (-10,10) (see the section on StdDraw below).
- Start the player at the location (0,0) (middle of the screen).
- Choose a target location for the player; you may hard-code it for partial credit.
- Recognize the commands “n”, “s”, “e”, “w” - and move the player in the appropriate direction.
  (For partial credit, it is OK to ignore any invalid commands.)
- After each command, redraw the screen (see the section on StdDraw below).
- End the loop as soon as the user reaches the target. Also end the loop if the user hits Ctrl-D (that is, if hasNext() returns false).
- After the loop ends, print the message "Game Over" to the user.
To receive full credit, your program must also do the following:

- Randomly generate the target location. (It should be on the visible board, and not equal to (0,0), since the player starts there.)
- Support the commands “north”, “south”, “east”, “west” (in addition to the four shorter versions).
- If the player types an invalid command, print an error message, but do not terminate the program. Instead, just move on to read the next command.
- Replace the red and blue squares with more interesting shapes (you get to choose what to draw). The player must be represented by at least three different StdDraw objects - such as three lines, three circles, or a mix - and likewise for the target.
- Draw the screen, with the player and target in the proper position, before the user types the first command.

3  StdDraw

In this program, you’ll be writing your own StdDraw code for the first time. This section tells you what you need to know about StdDraw for this project - but there’s lots more at the StdDraw documentation.

3.1  StdDraw.setScale(int,int)

Call StdDraw.setScale(int,int) to set the scale of your picture - that is, to set the minimum and maximum values for the coordinates. For instance, if you call

    StdDraw.setScale(-10,10);

then the screen will go from -10 to 10 in both dimensions. That is, the bottom-left corner of the screen will be (-10,-10), and the upper-right corner will be (10,10).

    You only need to call this once per program. Call it at the beginning of your program, before any loops or drawing operations.

3.2  StdDraw.clear()

Call StdDraw.clear() to clear the screen back to plain white. Call this each time that you plan to redraw the screen from scratch.
3.3 StdDraw.setPenColor(Color)

Call StdDraw.setPenColor(Color) to set the color that StdDraw will draw with; this call affects all of the drawing operations which follow it, until you call StdDraw.setPenColor(Color) to change it to something else.

For the argument, pass one of the colors provided by StdDraw. For instance, to set the pen color to red, do this:

```
StdDraw.setPenColor(StdDraw.RED);
```

or to set it to blue, do this:

```
StdDraw.setPenColor(StdDraw.BLUE);
```

StdDraw provides many colors. This is a partial list:

- BLACK
- WHITE
- RED
- BLUE
- GREEN
- YELLOW
- GRAY
- LIGHT_GRAY
- DARK_GRAY
- ORANGE
- PINK

3.4 StdDraw.filledSquare(double,double, double)

Call StdDraw.filledSquare(double,double, double) to draw a filled square on the screen. The first two parameters are the x and y coordinates of the center of the square; the third parameter is the half-width. (The half-width is half of the width of the square.)

So, the following call will draw a 1x1 square with its lower-left corner at (0,0):

```
StdDraw.filledSquare(.5,.5, .5);
```

whereas this call will draw a 2x2 square with its lower-left corner at (-1,-1):

```
StdDraw.filledSquare(0,0, 1);
```

3.5 square, filledCircle, circle, filledRectangle, rectangle

In addition to filledSquare(double,double, double), StdDraw also provides calls to draw filled circles and rectangles. (For a rectangle, the size is specified by the half-width and half-height of the shape.)

In addition, StdDraw has matching functions, which do not fill in the figure:
• circle(double, double, double)
• square(double, double, double)
• rectangle(double, double, double, double)

3.6 StdDraw.line(double, double, double, double)
Call StdDraw.line(double, double, double, double) to draw a line. The first two parameters are the (x, y) coordinates of one end of the line; the next two are the (x, y) coordinates of the other end.

4 Pseudocode

Download the latest pseudocode requirements from:
http://www.cs.arizona.edu/classes/cs127a/spring16/projects/
Project 3 will be graded based on v1.1 of the requirements.
For hints about writing pseudocode, check the Project 1 spec.

5 Java code

After you complete the pseudocode (and turn it in!), start working on translating it to Java. (For a list of steps for converting the pseudocode to Java, see the Project 1 spec.)
Your file must be named Wanderer.java, and the class inside it must be named Wanderer.

5.1 Header Comment

Every Java file that you submit must have a Header Comment - which is a simple comment, at the top of your file, which gives basic information about it. A header comment must include:
• The name of the Java class
• The name of this class (“CSc 127A Spring 16”)
• The assignment name
• Your name
• Your Section Leader’s name (or letter)
• A quick description of the Java class.
Here’s my suggested header comment for Program 3:
6 Turning in Your Programs

See the class website for information about how to upload your files to lectura, and how to use the ‘turnin’ utility.

6.1 Pseudocode

Your pseudocode file must be either TXT or PDF files (one per program), but you may name them whatever you want. You must turn them in using the assignment name

\texttt{cs127a_s16_sX_proj03_pseudocode}

(replace the \texttt{X} with your section letter). \textbf{REMEMBER:} The pseudocode is due two days before the Java code!

6.2 Java Code

You must turn in file(s) named exactly this:

\texttt{Wanderer.java}

You must turn in the Java file(s) using the assignment name

\texttt{cs127a_s16_sX_proj03}

(replace the \texttt{X} with your section letter).