CSCE XXX

CSCE XXX EXERCISE X: PATHFINDING I

Assigned: XXX
Due: XXX

OBJECTIVES

The objectives of this exercise:

• Computational:
  o Algorithmic thinking: (1) As a designer, developing a set of step by step instructions to generate a visual pattern on a base grid from straight line segments, and (2) As a responder to another group’s instructions, carry out their instructions methodically
  o Evaluation: Testing your instructions to make sure that another user can accurately recreate your pattern on a new base grid.
  o Pattern Recognition: As a designer, identifying similar (perhaps repeated) steps that create similar visual patterns on the grid and then using loops to simplify those steps, and (2) As a responder to another group’s instructions, identifying similar (perhaps repeated) steps that create similar visual patterns on the grid, and using short cuts to carry out those steps faster
  o Learning about boundary conditions by providing a set of instructions to draw a geometric pattern on a fixed grid
  o Learning how to use logical conditionals by providing different instructions depending on the (current) grid location
  o Developing a clear set of instructions that will allow others to create the geometric pattern

• Creative:
  o Surrounding: using your senses of touch and sight to follow a set of instructions and perceive that they produce the intended geometric pattern
  o Capturing: creating new outputs and using new ways to represent and save data by writing a description of a path which will generate a drawing of a geometric pattern
  o Challenging: looking at written descriptions in new ways as you both generate and follow them to recreate a drawing of a geometric design
  o Broadening: acquiring new information and skills by understanding how simple rules can generate complex patterns

• Collaborative:
  o Being open to all points of view and resolving conflicts in a constructive way.
Giving and receiving thoughtful and constructive feedback in order to develop your group project.

Meeting group deadlines, including completing your individual work in a timely manner.

Contributing substantively to the group process, using your skills, knowledge and experience.

Working together as a team to achieve a common goal; being able to both compete against and cooperate with other teams.

**Problem Description**

Using a grid, you will be designing a geometric visual pattern. You will then create a set of written instructions (an algorithm) for another group to follow which will accurately generate your geometric pattern on another grid.

Think of your pattern as a module that can be repeated, reflected and rotated to generate complex patterns. See Appendix A for visual examples of complex traditional quilt patterns using a simple module.

Each group will set up a wiki page on [agora.unl.edu](http://agora.unl.edu). The name of this page should be: “Path Finding I by <Course> Group <Name>” where <Course> is the course abbreviation and <Name> is your group name (e.g., Path Finding I by CSCE XXX Group Awesome).

*Any member may create the group page. Note that there should be only one page created per group. Before you create a new page, make sure that one doesn’t already exist.*

1. **Week One [20 points]**

   1.1. **Develop Base Pattern**

   Your group will generate a grid that is 3 x 3 with at least ¼ inch squares. You can generate this grid at [http://incompetech.com/graphpaper/lite/](http://incompetech.com/graphpaper/lite/) and download a PDF for printing if you do not have ¼ inch graph paper. Label your grid axes such that the origin (0,0) is in the upper left corner.

   Design your base pattern by drawing line segments on the grid. To simplify the set of instructions, please *only* use straight line segments (no curves) and restrict these line segments to horizontal, vertical or diagonal (45° angle) lines.
The proper use of **boundary conditions** is extremely important in computer science. When small errors in the boundary conditions are made, such as a mistake in array indexing or an improperly used pointer, the source code that you write may not compile or may generate an error message. Unfortunately, the severity of these boundary condition mistakes pales in comparison to the situation where your program still runs but returns the wrong output due to boundary condition mistakes. For small-scale programming projects, these mistakes are annoying and frequently require tedious debugging to locate and remove. On the other hand, in real-world applications written by hundreds of programmers, these mistakes may be extremely expensive taking hundreds of hours to track down and fix. Furthermore, these mistakes may not be fixable at all, for example, when they are found in an interplanetary probe.

**IMPORTANT:** The number of line segments in your base pattern must be at least 30% of all possible lines in a 3 x 3 grid. Note that a 3 x 3 grid can have at most 12 horizontal segments, 12 vertical segments, and 18 diagonal segments for a total of 42 line segments. After rounding up, this means that your base pattern must have at least 13 segments.

**NOTE:** Teams submitting patterns with fewer than 13 segments will lose points.

The base pattern should include some elements that repeat at least three times. These repeating elements can be rotated in the base pattern. Teams without a repeating element will lose points.

Here is a sample grid with a base pattern. In this example, the right triangle repeats six times with different rotations. It has 23 line segments, or \( \frac{23}{42} = 55\% \) of all possible line segments.

**IMPORTANT:** Feel free to use the forum on the wiki page to discuss your pattern, but do **NOT** upload the base pattern to your wiki page. Instead, email a copy of the base pattern you decide
Developing a clear set of instructions is extremely important in computer science. For small-scale programming projects, programmers may be able to look at in-line comments or source code and understand immediately the purpose for the module in question. However, real-world applications often contain thousands or even millions of lines of code making this understanding impossible to achieve. In these real-world applications, modules lacking a clear set of instructions are extremely difficult to use properly, and may result in missed deadlines or additional expense for a company. Furthermore, one must always keep in mind that the end users of the vast majority of real-world applications are not software developers. These end users need a clear set of instructions on how to use each module in the application or they will become upset very quickly and stop using it. Nevertheless, both developers and end users are often impatient and unwilling to spend hours reading hundreds of pages of instructions. As a result, programmers need to think creatively on how to provide a set of instructions that provides all the details while being as concise as possible.

Please post your initial set of instructions on the main wiki page. Please provide a header, as in the example, to specify that these are the initial set of instructions.

2. **Week Two [20 points]**

2.1. **Generate Base Patterns from Instructions by Other Groups**

As discussed in Week 1, start by generating a 3x3 grid.
Then, each group should navigate to a different group’s wiki page and “claim” their pattern instructions by posting on their wiki page. (Each group’s pattern instructions may have only ONE group attempting to follow it.) “Claim” a group’s pattern instructions by posting on that group’s page that “<Course> Group <Name of group> claims these pattern instructions to follow.” Each group will then try to follow the written instructions of another group by drawing on a blank grid labeled following the same (0,0) numbering. When you think you have followed the other group’s instructions correctly, post an image of the other group’s base pattern (a jpg of your grid drawing) on that group’s page. If you think that the other group’s instructions are invalid and you cannot complete the pattern, post your reasoning on that group’s page along with an image of your incomplete pattern.

NOTE: Groups will lose points in Week Two for not generating another group’s pattern from that group’s instructions or for not correctly following the base pattern instructions of another group.

2.2. ANALYSIS AND REFLECTION [20 POINTS]

Post your Analysis and Reflection responses in the Discussion area of your page, NOT in the body of the page.

You are expected to discuss these analysis and reflection questions among your group. One member must start a new topic for EACH Analysis or Reflection by selecting “New Comment.” In the Topic area, type “Analysis” or “Reflection” and in the Comment area, paste in the Analysis or Reflection questions. Using the Analysis or Reflection questions as prompts, each member will post his or her responses as a reply to the original comment. This process will keep the group’s Analysis and Reflection in separate threads and make it easier to follow the development of your answers.

You will be graded individually based upon your contributions to the group Analysis or Reflection. In order to receive individual credit for Week 2, each group member must contribute to the answers to these questions. Group members who do not contribute to the Analysis or Reflection Discussion will not receive points.

Analysis [10 points]. Respond to these questions: (1) Analyze your group’s instructions. Are they ambiguous? Can they be simplified? How could you simplify your instructions using loops? (2) If you combine two sets of instructions, your group’s and another group’s, to create a composite pattern, how would you make the combination a more efficient set by removing steps that draw the same line segments? Would you be able to retain certain subsets of steps without changing them at all? Why or why not?

Reflection [10 points]. Respond to these questions: (1) These instructions for drawing patterns are essentially steps of an algorithm. Do you see parts of all these algorithms
(including those of the other groups) that are common (i.e., creating the same shape or same combination of line segments)? If yes, please highlight them here. If no, please explain. (2) Again, these instructions for drawing patterns are essentially steps of an algorithm. Do you see parts of your pattern repeated in other groups’ patterns? You should. Please identify at least one part that consists of more than four segments. Is this part achieved using the same “sequence of steps” in other groups’ instructions? Likely it is not. Please explain why different groups designed their instructions differently for the same common sub-pattern.

**DEADLINES AND HAND-IN**

**Week 1 Deadline – [XXX, 11:59 p.m.]:** You should have completed creating your base pattern and have posted the written instructions for generating it to your wiki page. You should have attempted to follow the instructions of other groups and posted your drawn solutions to their wiki pages.

**Week 2 Deadline – [XXX, 11:59 p.m.]:** Your analysis and reflection responses are due by the Week 2 deadline above. Your individual Analysis and Reflection comments must be posted in the Discussion area of your group’s page.

**GRADING**

Week 1. Group credit for generating the base pattern instructions. Instructions must meet the stated requirements and points are deducted for if the patterns do not start at (0,0), do not contain the required number of segments or repeating elements or if the instructions are unclear.

Week 2. Pattern Generation from another group’s instructions: Group credit for generating and posting another group’s pattern from their base instructions OR for identifying the flaws in the other group’s instructions and posting the resulting incomplete pattern. Analysis and Reflection: graded individually. Each member must post in the Discussion with a minimum of 3-5 coherent, relevant sentences for full credit.

Late work will not be graded.

**APPENDIX A. EXAMPLES OF TRADITIONAL QUILTS**
END OF EXERCISE