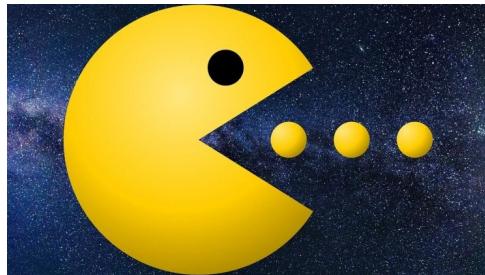


CSCE 4813 – Programming Project 2
Due Date – Friday 02/20/2026 at 11:59pm

1. Problem Statement:

The goal of this programming assignment is to give students experience with user interaction and OpenGL callback functions. Your task is to create an interactive game of your own design where Pac Man flies around in space eating food while being chased by space monsters. Your detailed requirements are listed below.



- A) Create simple geometric models for Pac Man, food, and one or more monsters out of 2D polygons. You are welcome to use any colors and shapes you like for the food and monsters, but Pac Man should be the traditional yellow disk with an open mouth. Display your geometric models in random locations on the screen.
- B) Use the OpenGL idle or timer callbacks to animate the food and monsters so they move slowly across the screen with random starting directions. When the food and monster objects reach the edges of the display window you can either have them bounce off in the opposite direction or teleport to the opposite side and reappear. It might be fun to have the monsters change directions at random intervals so they are less predictable.
- C) Use the OpenGL keyboard and mouse callbacks to control the motion of Pac Man. Your goal is to be able to move the Pac Man in any direction (not just up, down, left, right) so they can chase and eat the food while avoiding the monsters. Try not to make the Pac Man too fast and powerful, so it is a challenge to eat the food and avoid the monsters.
- D) Your final task will be to implement the game logic that detects when Pac Man collides with food or one of the monsters. Once you have done this, you can add some sort of scoring scheme to the game. For example, Pac Man could gain points by eating food, and lose one of their lives if they collide with by one of the monsters. You can decide when the game ends based on your scoring scheme.

For this project you ARE allowed to use generative AI to learn about OpenGL and to help you complete two of four tasks above. You are required to keep track of all of your AI prompts in a separate document to submit with your project report.

2. Design:

The design of this interactive game is up to you. Your goal is to invent game characters that look interesting but are easy to display. You are welcome to use the idle, timer, keyboard and mouse callbacks any way you want to create your game. My only advice is to keep it simple until you have something working, and then you can add features as time permits. Keep track of any significant decisions you make along the way, and describe them in your project report.

3. Implementation:

This semester we will be using C++ and OpenGL to implement all of our programming projects. The instructions for installing OpenGL are posted in README file the “Source Code” page of the class website. Once you have OpenGL installed, you can compile your graphics program using “g++ -Wall game.cpp -o game -lGL -lGLU -lglut”.

You are encouraged to look at sample OpenGL programs to see how the “main” function and the “display” function are normally implemented. As always, you should break the code into appropriate functions, and then add code incrementally writing comments, adding code, compiling, debugging, a little bit at a time.

Remember to use good programming style when creating your program. Choose good names for variables and constants, use proper indenting for loops and conditionals, and include clear comments in your code. Also, be sure to save backup copies of your program somewhere safe. Otherwise, you may end up retying your whole program if something goes wrong.

4. Testing:

Test your program with different random number generator seeds until you get some images that look fun/interesting. Take a screen shot of these images to include in your project report. You may also want to show some bad/ugly images that illustrate what happens if there is a problem somewhere. You can discuss how you corrected these problems in your project report.

5. Documentation:

When you have completed your C++ program, write a short report using the project report template describing what the objectives were, what you did, and the status of the program. Be sure to include several output images. Finally, describe any known problems and/or your ideas on how to improve your program. Save this report to be submitted electronically via Blackboard.

6. Project Submission:

In this class, we will be using electronic project submission to make sure that all students hand their programming projects and labs on time, and to perform automatic plagiarism analysis of all programs that are submitted. When you have completed the tasks above go to Blackboard to upload your documentation (a single docx or pdf file), and all of your C++ program files. Do NOT upload an executable version of your program.

The dates on your electronic submission will be used to verify that you met the due date above. All late projects will receive reduced credit:

- 25% off if less than 1 day late,
- 50% off if less than 2 days late,
- 75% off if less than 3 days late,
- no credit if more than 3 days late.

You will receive partial credit for all programs that compile even if they do not meet all program requirements, so handing projects in on time is highly recommended.

7. Academic Honesty Statement:

Students are expected to submit their own work on all programming projects, unless group projects have been explicitly assigned. Students are NOT allowed to distribute code to each other, or copy code from another individual or website. Students ARE allowed to use any materials on the class website, or in the textbook, or ask the instructor and/or GTAs for assistance.

For this project you ARE allowed to use generative AI to learn about OpenGL and to help you write the part of the code above. You are required to keep track of all of your AI prompts in a separate document to submit with your project report. Note: You can ask the AI to keep track of all of your prompts for you and then ask it to produce this document.

This course will be using highly effective program comparison software to calculate the similarity of all programs to each other, and to homework assignments from previous semesters. Please do not be tempted to plagiarize from another student.

Violations of the policies above will be reported to the Provost's office and may result in a ZERO on the programming project, an F in the class, or suspension from the university, depending on the severity of the violation and any history of prior violations.