

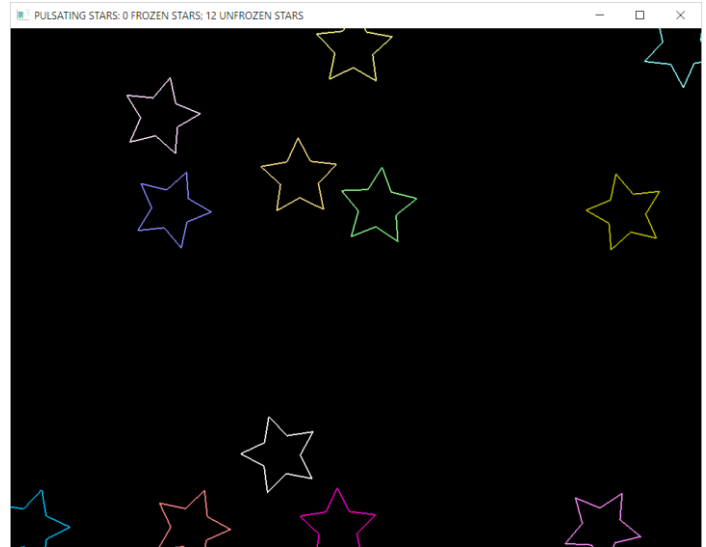
CS 382.001 - Game Design, Development, and Technology

Spring 2017

OpenGL Programming Assignment #1 (125 Points)

Due on your Moodle drop-box by 2:30 PM on Thursday, March 16, 2017

Your first programming assignment is designed to familiarize you with basic 2D animation in OpenGL, combining mouse-based graphical interaction, score display updates, and simple collision detection. An existing OpenGL application on the course Web site (<http://www.cs.siu.edu/wwhite/CS382/Syllabus.htm>) implements an interaction between the user and twelve spinning, pulsating star-shaped figures, as illustrated at right. The stars are assigned the same speed, base radius, and pulsation rate values, but varying colors and random initial positions and direction vectors. The user may use mouse clicks to select a particular star, producing a small beeping sound and causing the star to halt its forward progression, although it will continue to pulsate. The star will resume its journey (with an accompanying beep with a lower frequency) at the end of six seconds, or if the user mouse-clicks it a second time. A count of the number of “frozen” and “unfrozen” stars is kept updated in the window’s title bar, and the stars bounce off the window’s sides rather than experiencing wraparound.

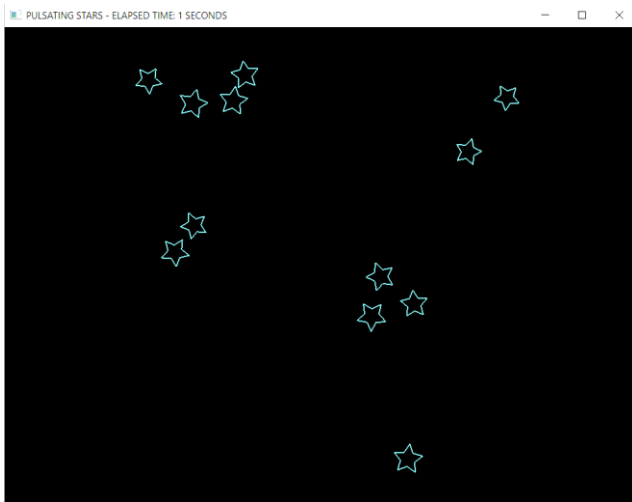


Your assignment is to alter this program in several ways. While the provided “game” amounts to little more than a dexterity test (i.e., can the user freeze all twelve stars at once?), the revised game will require the user to devise a survival strategy in an attempt to maximize the duration of the gameplay. Specifically, the following alterations will be required:

- The dozen stars will all begin with the same small radius, the same large pulsation factor, and the same color (cyan). Each star will have its own randomly generated pulsation rate, speed, initial position, and direction vectors, all within reasonable bounds.
- Every time a star collides with another star, the stars will exchange direction vectors. In addition, both stars will advance through a series of radius values (small, medium-small, medium, medium-large, large, very-large), pulsation factors (very-fast, fast, medium-fast, medium, medium-low, and low), and colors (cyan, blue, violet, red, orange, and yellow). The freeze duration (i.e., the amount of time a star remains frozen when mouse-clicked by the user) will also be reduced with each collision with another star (5, 4, 3, 2, 1, and 0 seconds). A beep will also sound whenever a collision occurs, with the beep’s frequency based upon the maximum number of collisions experienced by the two colliding stars.
- The title bar display will keep a running count of how many seconds have elapsed since the beginning of gameplay, with the player’s goal being to maximize that number. This is made difficult because of the increased likelihood of collisions as the game progresses. Once a star experiences a collision, it gets larger and its freeze duration is reduced. In addition, it obtains a more “dangerous” color and emits a higher-pitched beep with each successive collision, potentially adding to the pressure experienced by the player. Upon its fifth collision, a star becomes yellow and cannot be frozen at all, making it very difficult for the user to prevent collisions with it.

- The game is over when all twelve stars are yellow, at which time the title bar displays the player's final score (the number of seconds the game lasted) and the clearing of the display before each refresh is halted, resulting in a "smearing" of the animation of the twelve yellow stars.

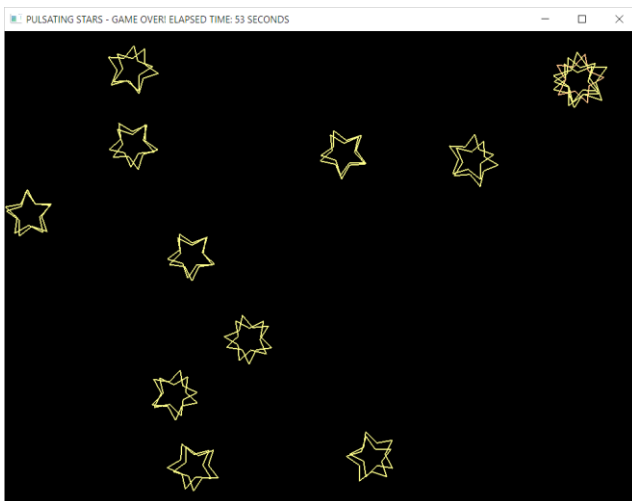
Screenshots of the provided executable are displayed below:



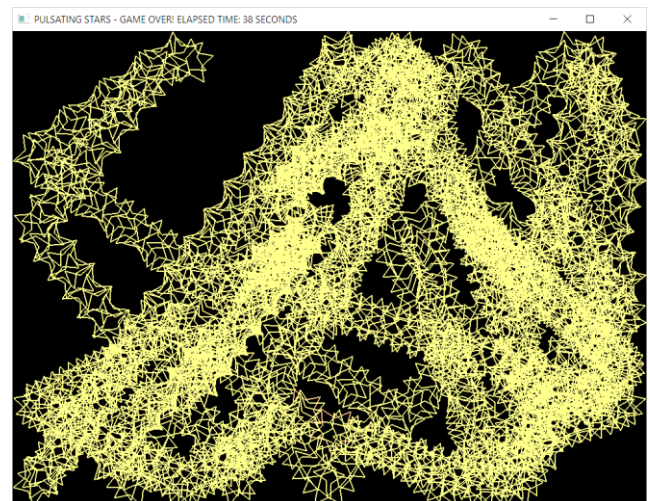
Initial Configuration At Start Of Gameplay



Configuration After Several Seconds Of Gameplay



Configuration As Game Ends



Smearing Effect After End Of Game

Your program is expected to behave exactly like the provided executable. Please consult the instructor for assistance with your OpenGL coding. Any code obtained from classmates, tutors, other individuals, or on-line sources is considered academic misconduct. You must write your own code, and no one but the instructor may see your code.

Zip-compress a folder containing all of your .cpp and .h files (not the entire project folder) and copy it to your dropbox by Thursday, March 16, 2017, at 2:30 PM. Late assignments are not accepted without verifiable medical documentation.