Zane Jacobs

1607 Cherokee Trail Plano, TX 75023 | (385) 290-8867 | zane.jacobs.0@gmail.com | zanejacobs.me

Technical Skills

Languages	Frameworks & Libraries	IDEs	Exposure to	Theory
Java	ASP.NET MVC	Visual Studio	Android	Object-Oriented Design
C#	Windows GDI	NetBeans	Python	Mathematics
C++	Swing	Eclipse	SQL	Software Design Principles
JavaScript	WPF		NoSQL	Software Development Life Cycle
HTML		**Source Control**	Scrum	Design of Data Structures
CSS		Git	Agile	Design of Algorithms

Project Experience

Complex Function Animator - A set of complex numbers, the input set, are sent through a hard-coded complex-valued function to define an output set. The input set is plotted on screen as points. These points move to their respective output positions along a path defined either by polar or linear interpolation. The input set is defined in-code, as well as the domain of the view. This was built in C++ and uses Windows GDI for the graphics. A custom complex number class was created with many functions such as sine, cosine, hyperbolic sine, hyperbolic cosine, exp (Euler's number to the power of a complex number), pow (raise a complex number to the power of another complex number), functions which calculate a complex number's argument and absolute value, as well as basic arithmetic (add, subtract, multiply, divide, negate).

Fractal Generator - The Mandelbrot set is rendered using the Escape-Time algorithm. For each pixel, a complex number is derived based on the domain of the view. This number is plugged into a function which is iterated. A color is derived from the number of iterations it takes to determine whether or not the value of the iterated function is bounded. The set of colors used in the rendering is defined in-code. The location and zoom of the view can be manipulated at runtime through a rudimentary GUI. This was built in C++ and uses Windows GDI for the graphics and user input.

3D Chess - Two-player chess with a 3D view. The board and pieces are rendered using perspective projection. The location and angle of the view can be manipulated at runtime. The faces of the board and pieces are shaded using diffuse-lighting. Pieces are moved by clicking on a piece's square and then clicking on a square which is valid for that piece to move to. The models used for the pieces are Wavefront OBJ files loaded when the program starts. This was done in Java and uses Swing for the graphics. The perspective projection is done "by hand", Swing is only used for drawing the 2D data resulting from the projection.

Relevant College-Level Courses Completed

Algorithms & Data Structures I & II
Principles of Software Engineering
Application Development
Linear Algebra
Open Source Platforms Development
Databases I & II
Persistence Applications
C++ Programming I & II
Logic
Calculus
Object Oriented Programming and Design
Leadership & Problem Solving
Introduction to Information Modeling
Business & Information Systems Practices
Introduction to Information Technology
Introduction to Web Presentation and Development
Networking I
College Algebra
Sets, Probability, and Number Systems
Survey of Robotics

References

Steve Halladay
Computer Science Program Chair, Neumont University
shallada@gmail.com
(303) 641-8915

Mark Herrera
Instructor, Fortis College
markaherrera@gmail.com
(801) 573-6600