

COURSE CODE: CGT 215

COURSE TITLE: Graphics Programming I

INSTRUCTOR: David Whittinghill

COURSE LEARNING OBJECTIVES:

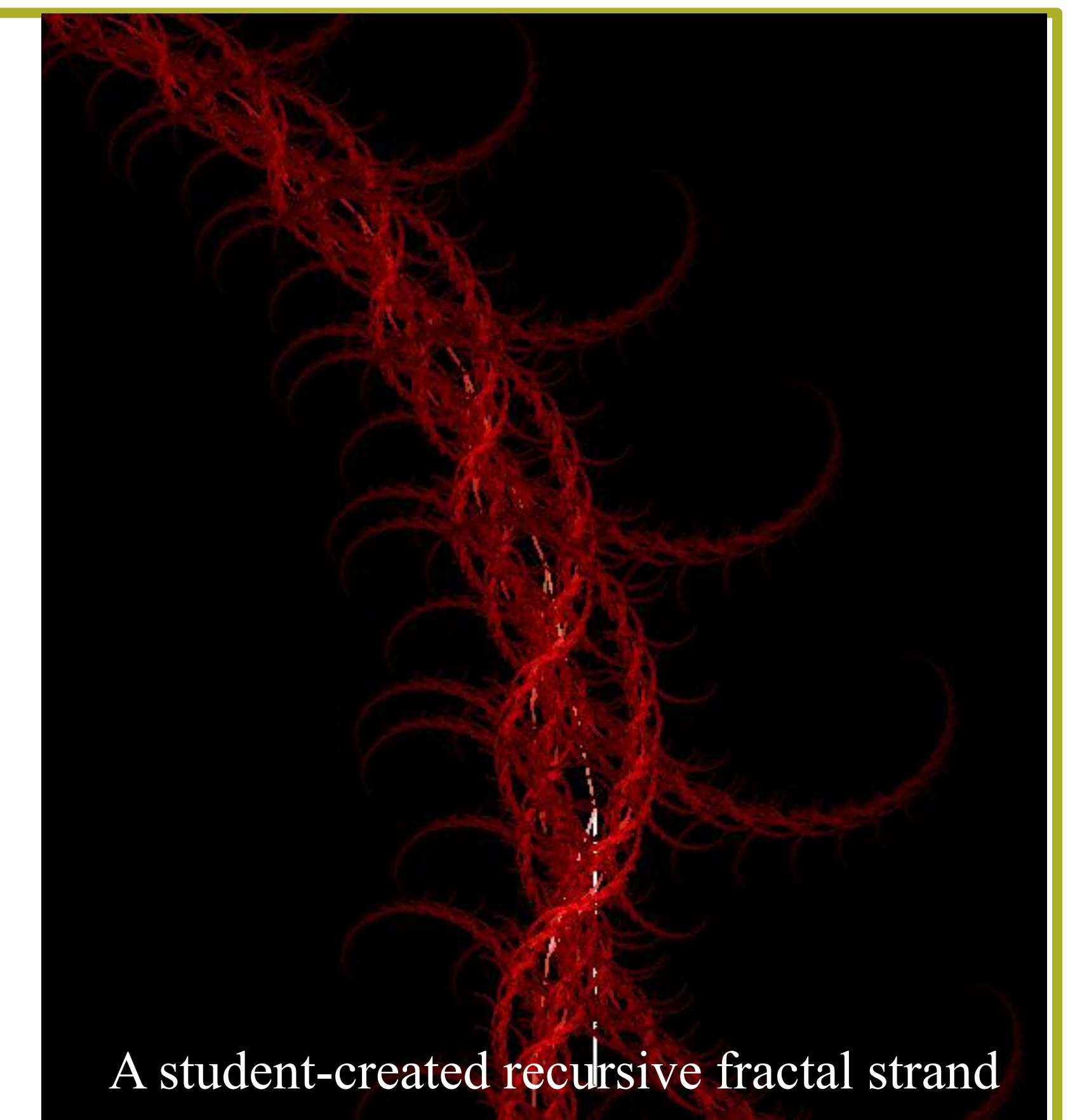
1. Decompose any process and transform it into a pseudocode representation
2. Compose a C++ program
3. Demonstrate a working knowledge of computer graphics drawing, animation, and interactive techniques
4. Demonstrate a working knowledge of how matrices, vectors, and the Matrix Stack are applied in Computer Graphics
5. Analyze a C++ program and determine how it behaves
6. Recognize programming fundamentals in a programming language other than C++

WHY IS THE COURSE BEING TRANSFORMED?

Research and past experience teaching the course corroborate the notion that computer programming is typically best taught when students can get hands-on experience as early and often as possible. Students are aware of this need as well, as a consistent plea from outgoing students in previous semesters was for more homework, or more practice time than provided in a once-weekly lab session. Further, lecturing, though not without value, places students in a passive mental mode which is not as helpful when learning active, operational, hands-on concepts. I sought a new teaching model that minimized passive listening in favor of more opportunities for students to test the knowledge shared immediately upon delivery, and to do so at a much greater frequency than the traditional lecture/lab model previously used.

RESEARCH QUESTION ADDRESSED WITH COURSE REDESIGN

Does the IMPACT model result in differences in students' assessment scores, attitudes toward the discipline of programming, and self-efficacy beliefs regard their ability to program, once the class has ended?



A student-created recursive fractal strand

IMPROVEMENTS TO STUDENT LEARNING I HOPE TO FOSTER

```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
            Console.WriteLine("Unable to connect to server.");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while(true){
            input = Console.ReadLine();
            if (input == "exit") break;
            newchild.Properties["ID"].Add("Auditing Department");
            if (input == "commit") newchild.CommitChanges();
            newchild.Close();
        }
    }
}
```

- Goal 1: Affective improvement. Computer programming can be difficult to learn and students often come to the class with self-efficacy beliefs that are almost at the level of phobia. I hope that students will develop neutral-to-positive feelings about programming.
- Goal 2: Knowledge retention. By providing increased opportunities to practice, I hope that students will gain a more resilient understanding of the practices of programming.
- Goal 3: Higher assessment scores. I hope that students have improved scores on all programming assessments.

GREATEST BENEFITS FROM PARTICIPATING IN THE IMPACT PROGRAM

- Benefit 1: Increased rigor of my pedagogy. I feel that my pedagogy is more orderly, better validated, better quantified, and more amenable to continuous research and improvement.
- Benefit 2: Improved affective and learning outcomes for my students. I believe that my students will be more capable programmers as a result of the IMPACT changes.
- Benefit 3: Improved consistency of content delivery. By making the content more organized and rigorous, it has also become more maintainable and modular. Using this framework, it is much easier for other content experts to contribute while maintaining the fundamental pedagogical framework of the course.