Discrete Mathematics
November 10, 2011

Calculator Programming Project

Today we will begin a group project creating calculator programs for "client" calculus students to use. This will count as a major assessment. Your finished products will be given to other calculus students at Deep Run (and potentially beyond). You will also get a chance to see others' projects. These projects will be due on Thursday, November 17. Your goal today is to choose a project, choose team roles, and begin your investigation of the problem and your methods. The first step is to **read this document**.

Your groups

You will be placed in three (or four) person groups. All of you have some programming experience and all of you are in BC Calculus now or were in center Precalculus last year, so all of you are qualified. The groups have been chosen so that each team has at least one member with AP Comp Sci experience and at least one member with Calculus BC experience.

Team roles

The major team role will likely be Team leader, Lead programmer, Mathematics leader, Researcher, and Product manager. You may have some or all of these roles on your team.
Team leader – Make sure that everyone is contributing and that the project is completed well and on time.
Lead programmer – Many of you can program, but it may be easiest if one person sees to it that all of the code is put together.
Mathematics leader – No matter how slick the code, your program needs to be accurate. The lead programmer may need calculations to be made or formulae found. If an advanced topic is chosen, then one of the first jobs will be for a group member to teach the topic to others in the group. Do not proceed without everyone knowing the topic you are presenting. Researcher – Several or all of you on a team may need to be researchers. This involves investigating the mathematical problem and researching how to program different things on a calculator. I expect that even the most skillful programmers will need research on implementing their ideas in a calculator. Product manager – This is a crucial role. A major portion of the success of this project will be a successful user interface. Each team should have a person responsible for seeing to it that your program is as easy to use and clear as possible. This person would then give the lead programmer suggestions on how to improve the interface. Remember, your clients are assumed to have no prior programming knowledge. This person should also ask how else or better the information could be presented. Your clients will judge the success of your project by how easily they can use it, and by an impressive display.

The problem

Your purpose is to demonstrate a major concept of calculus/precalculus via a calculator program. The intended audience is a calculus (or precalculus) student without any programming skills. Here are some possible topics.

- Secant lines (whose slope represents an average rate of change) approaching a tangent line
  This can present an especially impressive visual display of the "meaning" of the derivative at a point.
- Calculation of difference quotients
  Another means for presenting the fundamental idea behind calculus.
- Limits
  The TI-89 can evaluate limits, but it would be useful to "see" how output values changed as "input" values approached a given value (or grew larger).
- Series
  The TI-89 can evaluate some infinite series, but it would be nice to see how these sums are found. Partials sums could be shown numerically or graphically.
- The binomial theorem
  Always an important topic. A number of different properties could be shown.
- Newton's method
  Approximating roots can be presented graphically, numerically, or in a table.
- Euler's method
  The approximate solution at a point to a differential equation can be presented in a number of formats. Note that the TI-89 has a built in graphical solution, but a tabular solution would be very useful to BC students.
- Taylor Polynomials
  A numerical or graphical demonstration of "best fit polynomials."
- Riemann sums
  The classic introduction to area under a curve.
- Your own idea
  What would you like to show other students?

Where it runs

All of you have some experience programming in Java. However, these programs are being written for an audience they may not be able to run Java. Also, the calculators afford an opportunity for impressive visual displays. As programmers, being able to implement your ideas in different environments and in different languages is invaluable.

The expectation is that your programs will be written for the TI-89. However, you may opt to write for a TI-84 or a Cassio.

Input

The user of your program can input whatever functions or values you need in a variety of ways. Values may be inputed to a TI-89 program as arguments of the program when it is run, information may be taken from a prompt, functions may be entered into "Y=," or information can be read in from a list. Values can also be read in from graphs.

Make sure that it is clear to a user how to run your program. Directions can be made part of your program, a handout can be made for users, or you can create a webpage to supplement your program.

Output

The output from your program – what it shows to the user – can be a function, can be output on the program output screen, can be a graph, can be a list or can be a table. Your output can also have a time element, such as a series of outputs that (with appropriate pauses) the user can see change. You might want to use several outputs or to prompt the user for a chosen display method.

Please note that more than just the graph of a function can be displayed in the graphing screen. You can also show a variety of Plots and there are numerous commands to Draw objects within the window. (You can also always take a pixel by pixel approach to draw anything, limited only by resolution.) A mixture of these methods can all be displayed at once.

Workspace

You can transfer programs from one calculator to another using cables. This is how the programs will likely be disseminated to the calculus students. When working together; however, you may find it much easier to use the emulators that our part of the hcps image. There are emulators for a TI-89, a TI-84, and a Cassio. You can save the states in these emulators and transfer these saved states to other group members. Programs can be downloaded from a computer to a calculator or vice versa. I also have a keyboard that can be plugged into a calculator, if that will help.

Resources

There are many TI-89 resources online. One reference is at `http://www.prenhall.com/divisions/esm/app/calculator/medialib/ReferenceCenter/framesets/TechFeat89.html`. A searchable TI-89 guidebook can be found at `http://education.ti.com/guidebooks/graphing/89ti/TI89TitaniumGuidebook_Part2_EN.pdf`, but it can be slow to download. A copy is in virtual share. The index begins on page 914. There are also two small hardcopy guidebooks in the room.

What you will turn in

A completed project will consist of three items; a description of your project and directions for the user, a separate description of what role each team member played in the finished product, and a written copy of your code. Of course, you will also transfer your program to me.

Grading Rubric

Below is a breakdown for how your project will be graded. The breakdown below is for 90 points with numerous opportunities for more. Please note that an A project should be in some way exceptional, instead of meeting minimum standards. Project completed late, without prior approval, will be marked down.

Mathematics
Is the mathematics accurate? (18 pts.)
Is the mathematics appropriate for the audience? (5 pts.)
Have you presented something new? (+)
Is the mathematics aesthetically thrilling? (+)

Programming
Does it run and do what it should without glitches? (20 pts.)
Does the program run efficiently? (4 pts.)
Is your code beautiful? (2 pts.)

Usefulness
Can your program be used in a calculus or precalculus class? (10 pts)
How easy is it for students to use the program? (10 pts.)
Does the program have a wow factor for those using it? (6 pts.)
Does the program do multiple things/have multiple outputs? (+)
Does the user have meaningful options? (+)
Can the user "see" something happen? (+)

Teamwork
Did all team members contribute in a substantial way? (10 pts.)
Did your team work harmoniously? (5 pts.)