

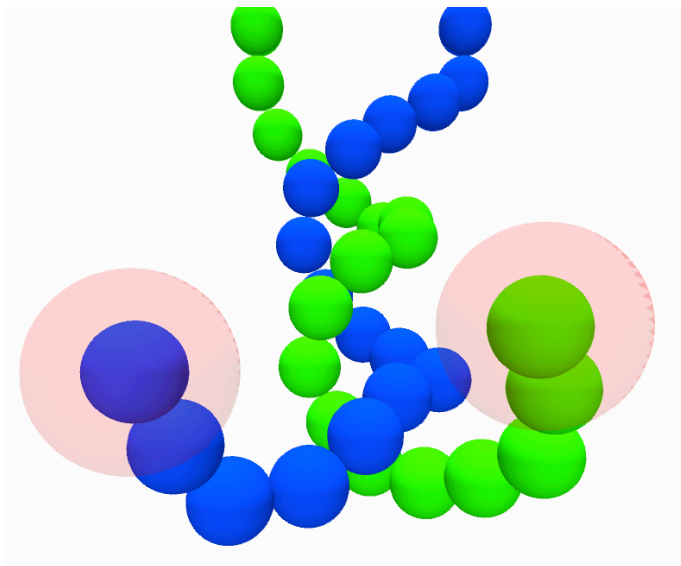
## CS3451 - Fall 2009 - Project 3: Knots

Overall objective: Produce a tool for designing and teaching the art of tying knots. The user of your posted applet should be able to press a key to circulate through 3 different animations, each one teaching how to tie a different knot. The user should be able to pause/play your animation and rotate the view. The user should also be able to design a new animation and save it to file. The process of designing the animation for a simple knot, such as the square knot, should take only a few minutes.



The project is to be designed and implemented individually. It is divided in 4 phases. You should email the TA the URL for this project page, which should contain the title of the project ("CS3451 Fall 2009 - Project 3 - Knots"), your name and picture, a link to a new applet for each phase (each phase adding functionality to the previous one), and a brief explanation of your approach. The applet page for each phase for each phase should contain a clear and concise explanation of what the user should do. If you decide to structure your project differently, you must still split it into 4 phases, explain what the phases are and follow the deadlines below.

Oct 27 - Phase 1 (DESIGN): Provide an interactive tool where the user may create and edit two curves (the spines of the ropes) in three dimensions. Display the two curves in a way that clearly conveys depth and occlusion as a series of spheres, each tangent to the previous one along the rope. You may start with a default (straight lines) for the curves or let the user grow them as desired. But you should provide reasonable tools for editing the grown curves with a mouse. To help you get started displaying balls in 3D and manipulating the view, you may start with the bouncing ball applet <http://www.gvu.gatech.edu/~jarek/demos/bounce/> or, for more advanced programmers, with the isosurface applet <http://www.gvu.gatech.edu/~jarek/demos/isosurface/> that has tools for picking balls graphically. Implement functions ('W' and 'G') for writing the spines to file and for getting them back.



Nov 3 - Phase 2 (PHYSICS): Implement three actions: Smoothing, Repulsion, and Length and key actions ('S', 'R', 'L') that invoke them. Smoothing moves the vertices of the spine by using bi-Laplace smoothing (tuck&untuck) in 3D. Repulsion detects pairs of spheres that intersect and displace them away from each other. The Length action moves each sphere towards a position where it would be tangent to the previous and next sphere (hence it tries to preserve the length of each rope). During these motions, the first and last spheres of each string may remain fixed.

The amounts of displacement for each action should be a parameter that you should fine-tune. Finally, provide a mode (toggled by pressing "P") where all three actions are performed at each frame.

Nov 10 – Phase 3 (RENDERING): Implement a four-point subdivision of the two spines (and its activation by pressing 'F'). The first and last edges may be subdivided by inserting a mid-edge point. Then, compute a quad-mesh that bounds each rope subdivided twice. When the user presses 'Q', toggle between rendering the balls of the original un-subdivided ropes or rendering the quad-mesh of the subdivided rope. (When switching to the quad-mesh mode, your program should re-compute the quad-mesh.) Render the quad-meshes each as a checkerboard using two complementary colors. Use a different color pair for each rope. For extra credit, you may provide the option 'T' to render each rope using a different texture. Make sure that each ring of quads is aligned with the previous one, so as to avoid unnecessary twists.



Nov 17 – Phase 4 (ANIMATION): Implement a tool for producing an animation that starts with the two ropes as straight lines and ends up with the ropes in the designed position for the given knot. The animation should be physically valid, in that the two ropes should not interfere with each other and remain free from self-intersection at all times. You are free to implement the animation as an automatic process or as a user-assisted process (for example using key-framing). If the user is involved or if the automatic process is slow, make sure that you save on file and reload the data needed to replay the animation.

Nov 24 – Final deadline for late submission of the complete project, including write up.

Dec 3 – Project presentations.

GRADING:

A good (easy to use, effective, and clearly visualized) implementation of Phase 1, 2, and 3 and clear usage instructions and explanations of the solutions will bring you up to 10 points per phase. These 30 points will have as much weight on your final grade as the total of the first two projects.

Phase 4 will bring you extra credit towards your final. A complete and effective implementation of phase 4 (either fully automatic or one that lets the user design correct animations for interesting knots in a few minutes) and a clear and detailed write up may bring you up to 30 extra credit points.