Graphics Programming II – Assignment #1 Due on 1/25/2012

In this assignment you will generate camera control in a simple scene. Initially, a set of positions that the camera must "hit" are supplied. In the first iteration, linearly interpolate among these position.

In the second iteration create a set of Bézier curves that also pass through these end points. Evaluate the curves through time to generate camera positions. To do this, you will have to implement the bezier_curve::position method.

In the third iteration generate a new set of Bézier curves from the original set. The new set of curves must be G^1 continuous. Evaluate the G^1 curves through time to generate camera positions.

In the fourth iteration generate a new set of Bézier curves from the original set. The new set of curves must be C^1 continuous. Evaluate the C^1 curves through time to generate camera positions.

Note: Depending on the algorithm you select, you make skip generation of the G^1 curves and just generate C^1 curves.

In addition, modify the supplied shaders, phong.vert and phong.frag, to perform per-fragment lighting. Extra credit will be given if lighting is performed in surface-space. This will also give you a start on the next assignment.

There are three main programming elements to this exercise:

- Evaluation of positions on a Bézier curve.
- Algorithmic modification of piecewise Bézier curves to attain higher orders of continuity.
- Per-fragment lighting.

Criteria	Excellent	Good	Satisfactory	Unacceptable
Completion	Program correctly im-	Program implements	Program implements	Many required
	plements all required	all required elements,	most required ele-	elements are
	elements in a manner	but some elements	ments. Some of the	missing. User
	that is readily appar-	may not function	implemented elements	interface is in-
	ent when the program	correctly. User inter-	may not function	complete or is
	is executed. User	face is complete and	correctly. User inter-	not responsive
	interface is complete	responsive to input.	face is complete and	to input.
	and responsive to in-		responsive to input.	
	put. Program doc-			
	uments user interface			
	functionality.			
Correctness	Program executes	Program executes	Program executes	Program does
	without errors. Pro-	without errors. Pro-	without errors. Pro-	not execute due
	gram handles all	gram handles most	gram handles some	to errors. Lit-
	special cases. Pro-	special cases.	special cases.	tle or no error
	gram contains error			checking code
P.00 +	checking code.		D 1	included.
Efficiency	Program uses solution	Program uses an ef-	Program uses a log-	Program uses
	that is easy to under-	ficient and easy to	ical solution that is easy to follow, but it is	a difficult and inefficient
	stand and maintain.	follow solution (i.e.,	not the most efficient.	and inefficient solution. Pro-
	Programmer has analysed many alternate	no confusing tricks). Programmer has con-	Programmer has con-	grammer has
	solutions and has cho-	sidered alternate solu-	sidered alternate solu-	not consid-
	sen the most efficient.	tion and has chosen	tions.	ered alternate
	Programmer has in-	the most efficient.	tions.	solutions.
	cluded the reasons for	the most emercia.		solutions.
	the solution chosen.			
Presentation &	Program code is for-	Program code is	Program code is for-	Program code
Organization	matted in a consistent	formatted in mostly	matted with multi-	is formatted
	manner. Variables,	consistent with occa-	ple styles. Variables,	in an inconsis-
	functions, and data	sional inconsistencies.	functions, and data	tent manner.
	structures are named	Variables, functions,	structures are named	Variables, func-
	in a logical, consistent	and data structures	in a logical but incon-	tions, and data
	manner. Use of white	are named in a logi-	sistent manner. Use	structures are
	space improves code	cal, mostly consistent	of white space neither	poorly named.
	readability.	manner. Use of white	helps or hurts code re-	Use of white
		space neither helps or	ability.	space hurts code
D / /:	C 1 1 1 1 C	hurts code reability.	C 1 1 4 1	reability.
Documentation	Code clearly and ef-	Code documented	Code documented	No useful doc-
	fectively documented	including descrip-	including descriptions	umentation ex-
	including descriptions	tions of most global	of the most important	ists.
	of all global variables and all non-obvious lo-	variables and most non-obvious local	global variables and	
	cal variables. The spe-	variables. The spe-	the most important local variables. The	
	cific purpose of each	cific purpose of each	specific purpose of	
	data type is noted.	data type is noted.	each data type is	
	The specific purpose	The specific purpose	noted. The spe-	
	of each function is	of each function is	cific purpose of each	
	noted, as are the input	noted, as are the	function is noted.	
	requirements and out-	input requirements	Tailouou io iiouou.	
	put results.	and output results.		
his rubric is based loosely on the "Rubric for the Assessment of Computer Programming" used by Queens				

This rubric is based loosely on the "Rubric for the Assessment of Computer Programming" used by Queens University (http://educ.queensu.ca/compsci/assessment/Bauman.html).