Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
145525	18	10	8	1	3	3	3		Good job!	1			3		2	2	Animated light: 1, rendering only vertices: 1
210984	0	0	0														
218096	17.5	10	7.5	1	3	3	3			1	1		3	2		0.5	animation 0.5
292986	10	10	0	1	3	3	3		One commit doesn't imply the active use of VCS.	0							
351526	0	0	0														
353692	9.5	9.5	0	1	2.5	3	3		One commit is not enough for VCS point! R2 normal calculated wrong.	0							
362418	0	0	0														
424851	0	0	0														
425575	0	0	0														
426419	15	10	5	1	3	3	3		Good job on reqs! Viewport correction correction is a bit strange as it is glued to the left side of the window.		1		1	2	0.5	0.5	animation 0.5;
427230	0	0	0														
428925	13	10	3	1	3	3	3				1	1			0.5	0.5	animation 0.5
429461	14.5	9.5	5	1	2.5	3	3		R2: Normals calculated wrong.		1	1	2		1		
431006	0	0	0														
432788	22	9.5	12.5	0.5	3	3	3		R1: y- and z-axis translate missing.	1	1		3	2	2	3.5	animation 0.5; Support for PLY files 3
437631	0	0	0														
474898	10	10	0	1	3	3	3										
475389	0	0	0														
476883	13.5	9.5	4	1	2.5	3	3		R2 normal calculation wrong.	1	1			2			
478632	0	0	0														
506041	0	0	0														
506287	20.5	10	10.5	1	3	3	3			1	1	1	3			4.5	animation 0.5; Support for PLY files 4
520085	11	10	1	1	3	3	3			1							
525491	12	10	2	1	3	3	3		Good job!	1	1						
525750	0	0	0														
525941	9.5	9.5	0	1	2.5	3	3		R2 normal calculated wrong.								
527677	10	10	0	1	3	3	3										
530185	0	0	0														
530648	0	0	0														
552794	0	0	0														
552969	0	0	0														
565710	17.5	10	7.5	1	3	3	3			1	1		1	2	0.5	2	animation 0.5; Support for PLY files 1.5
570116	11.5	10	1.5	1	3	3	3				1					0.5	animation: 0.5

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
576149	9	9	0	0	3	3	3		R1 no axis control at all.								
577122	13.5	10	3.5	1	3	3	3			1						2.5	animation 0.5; Support for PLY files 2
585240	12	10	2	1	3	3	3			1	1						
586029	0	0	0														
586333	13	10	3	1	3	3	3			1	1	1					
587549	18	9.5	8.5	1	2.5	3	3		R2: Normals calculated wrong.	1	1		2	2	1.5	1	animation 0.5; A bit broken support for PLY format 0.5
589796	0	0	0														
589929	13	10	3	1	3	3	3				1			2			
590743	9.5	8.5	1	0	2.5	3	3		R1 no keys respond, no matrices build for operations. R2 normal inverted.	1							
592864	0	0	0														
592929	11	9.5	1.5	1	2.5	3	3		R2 inverted normals.							1.5	animation 0.5; rotate any axis 1;
595997	9.5	9.5	0	1	2.5	3	3		R2: Inverted normals.								
596174	0	0	0														
596747	11	10	1	1	3	3	3			1							
596815	0	0	0														
597322	11	10	1	1	3	3	3			1							
603025	0	0	0														
604244	0	0	0														
604312	16.5	10	6.5	1	3	3	3		Normals are unnormalized after transformation in the transform normals in the vertex shader extra Behaviour of trackball changes as window size is modified. You could could have used window size instead of hardcoded values. Also when calculating the rotation angle, the equation is missing dz as you call change. getXY() when comparing to the resource provided.	1	1		2.5	1.5		0.5	animation 0.5;
605560	11.5	9.5	2	1	2.5	3	3		R2 inverted normals.	1	1						
605722	10	10	0	1	3	3	3										
609265	13	10	3	1	3	3	3		Ply loader is kinda minimal.	1					0.5	1.5	animation 0.5; ply loader 1;
614580	10	10	0	1	3	3	3										

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
618117	12	10	2	1	3	3	3			1					0.5	0.5	animation 0.5
635187	0	0	0														
646655	10.5	9.5	1	1	2.5	3	3		R2: Normals calculated wrong.	1							
646927	11	10	1	1	3	3	3			1							
648530	16	10	6	1	3	3	3		Good job! Notice that you can get 3x3 matrix in FW by .getXYZ() call or likewise extract 3x3 matrix from 4x4 matrix in glsl by calling mat3 constructor on it. Normals are unnormalized after transformation in the transform normals in the vertex shader extra. Animation is a bit twitchy but works.	1	1		1	1.5	1	0.5	animation 0.5;
648653	24.5	10	14.5	1	3	3	3				1		3	2	2	6.5	animation: 0.5, position only PLY: 1, mesh simplification: 5
650227	10	10	0	1	3	3	3		Please fill in the README next time.								
650492	14	10	4	1	3	3	3				1		3				
652131	11	10	1	1	3	3	3				1						
652144	10	10	0	1	3	3	3										
652801	0	0	0														
653101	19.5	10	9.5	1	3	3	3		Good job on virtual trackball! Ply loader is kinda minimal and the animation extra is a bit twitchy.	1	1		3	2	1	1.5	animation 0.5; ply loader 1;
653460	0	0	0														
653509	12	10	2	1	3	3	3			1	1						
653758	12	10	2	1	3	3	3		One commit does not count as using VC. Return a log with more commits next round to get points. Scaling should be non-uniform along x-axis.	0	1	1					
653897	18	10	8	1	3	3	3			1	1	1	3		0.5	1.5	animation 0.5; Support for PLY files 1
654579	11	10	1	1	3	3	3			1							
655691	9.5	9.5	0	1	2.5	3	3		R2 inverted normals.								

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
656470	0	0	0	0	0	0	0		Empty readme. R1 you are setting the translation to last row instead of last column, which results in weird behaviour.								
656991	12	9.5	2.5	0.5	3	3	3		R1: z-axis translate missing. Scaling calculated wrong.	1	0.5		1				
665131	11	10	1	1	3	3	3			1							
666253	15.5	10	5.5	1	3	3	3		One commit does not count as using version control. Submit log with more commits next round to get points.	0	1		1.5	2	0.5	0.5	animation 0.5
667171	6	6	0	0.5	2.5	3	0		R1 no z-axis control. R2 inverted normals.								
672771	10.5	9.5	1	0.5	3	3	3		R1 no z-axis control.	1							
673987	29	10	19	1	3	3	3		Great work!	1	1	1	3		0.5	12.5	animation 0.5; Support for PLY files 4; Mesh simplification 8
675163	22.5	10	12.5	1	3	3	3			1	1		3	2	1	4.5	animation 0.5; support for PMX files 2
675765	0	0	0														
677721	0	0	0														
689395	16	9.5	6.5	0.5	3	3	3		R1: z-axis translate missing.	1	1		2	2		0.5	animation 0.5
700436	0	0	0														
705855	10	10	0	1	3	3	3		Git log missing. One commit does not count as using version control. Return one next round to get points.	0							
706003	10.5	9.5	1	0.5	3	3	3		R1: z-axis translate missing.	1							
706045	12	10	2	1	3	3	3			1	1						
706443	11.5	9.5	2	1	2.5	3	3		You seem to accidentally set translation when constructing modelToWorld matrix in the rotation part. R2 tip not at the origin as required.	1	1						
706595	10	10	0	1	3	3	3										

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
706919	9	9	0	1	2	3	3		R2 small mistake with indexing, should be (i+1) instead of i+1								
707138	12	10	2	1	3	3	3		As you accumulate modelToWorld in the eventloop. Random orderings of scale, rotation and transform can be performed. You'd want to scale in object space. -> Scale first and don't let it be dependant on prior transformations. Also pay attention the multiplication order of matrices!	1	0.5		0.5				
707277	0	0	0														
707316	17.5	10	7.5	1	3	3	3		Scaling was supposed to be non-uniform along x-axis.	1	1	1	3			1.5	animation 0.5; support for PLY files 1
707620	7	7	0	1	3	3	0		One commit does not count as using version control. Submit log with more commits next round to get points.	0							
708616	10	10	0	1	3	3	3										
708988	10.5	9.5	1	1	2.5	3	3		R2: Normals calculated wrong.	1							

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
709026	11.5	10	1.5	1	3	3	3		Rotate and scale extra: the idea of calculating the delta in eventloop is good, but you should keep scale, rotation and translation independant. Now for example I could first apply scale delta followed by rotation. ->This results in weird behaviour because of the matrix multiplication ordering you have set up, where the earlier scaling is dependant on the rotation applied afterwards. You'd want to scale in object space. Pay attention to matrix multiplication order.	1	0.5						
709178	10	8	2	1	2.5	1.5	3		R2 inverted normals. R3 unpacking normals incorrectly. You are now reading normals only by the index of the first vertex. R4, fixing R2 will fix the loading bug. Could not find VCS screenshot	0	1		1				
709903	12	10	2	1	3	3	3			1	1						
711111	17	10	7	1	3	3	3		Good job!	1	1		2.5	2		0.5	animation: 0.5,
711218	0	0	0						Late submission								
711263	9	9	0	0.5	2.5	3	3		R1 missing z-axis. R2 normals calculated wrong.								
711810	7	7	0	1	3	3	0										
712039	16	9.5	6.5	0.5	3	3	3		R1: x- and z-axis translations missing.	1	1		2	2		0.5	animation 0.5
713601	11	10	1	1	3	3	3				1						
713928	12	10	2	1	3	3	3			1	1						
714477	9.5	9.5	0	0.5	3	3	3		R1: z-axis translate missing. R4: Almost correct, you forgot to subtract one from the face indices.								

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
714574	18.5	10	8.5	1	3	3	3		Good job!	1	1		2	2	2	0.5	animation: 0.5,
716718	13	9.5	3.5	0.5	3	3	3		R1 missing z-axis. R2 calculating vertices multiple times? Rotate and scale extra has wrong multiplication order. Scaling is dependant of the rotation here. Scale in object space. Also there are no controls.	1	0		2			0.5	animation 0.5;
716792	10	10	0	1	3	3	3										
717377	14	10	4	1	3	3	3			1	1			2			
729637	16	10	6	1	3	3	3			1	1		3		0.5	0.5	animation 0.5
730105	10	10	0	1	3	3	3										
730448	21	10	11	1	3	3	3			1	1		3	2	2	2	Misc. improvements 1; Support for PLY files 1
730969	10.5	9.5	1	0.5	3	3	3		R1: z-axis translate missing.	1							
732381	11	10	1	1	3	3	3			1							
763282	12.5	10	2.5	1	3	3	3			1	1					0.5	animation 0.5
765662	11.5	9.5	2	1	2.5	3	3		Remember to mark completed parts with done! No deduction here. R2 normals are inverted. In the normal transform extra you are not supposed transform normals to clip space. This breaks the shading. Notice that light calculations are performed in world space here, so only the object space -> world space transform would have sufficed. If you want to transform normals in homogeneous coordinates, the w-component is 0	1	1	0					
766108	15	9.5	5.5	1	2.5	3	3		R2 inverted normals.	1	1			2	0.5	1	animation:0.5, WASD camera control: 0.5
768902	0	0	0														
770084	18	10	8	1	3	3	3			1	1		2	2	1	1	Support for STL 1

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
772040	15	10	5	1	3	3	3		You should transform normals with inverse transpose of the modelToWorld in the transform normals in shader extra. Not taking the inverse transpose results in strange normals/shading with non uniform scaling. Also remember to normalize normals after transformation. Great job on the trackball!	1	1	0	3				
778109	18.5	10	8.5	1	3	3	3		One commit does not count as using version control. Submit log with more commits next round to get points.	0	1		3	2	2	0.5	animation 0.5
779661	13	10	3	1	3	3	3			1	1	1					
779959	11.5	9.5	2	1	2.5	3	3		R2: Normals calculated wrong.	1			1				
780210	10	10	0	1	3	3	3										
780223	19	10	9	1	3	3	3		Animation indeed mesmerizing. One commit does not count as using version control. Submit log with more commits next round to get points.	0	1		3	2	2	1	animation+effects 1
781468	0	0	0														
781866	19.5	10	9.5	1	3	3	3			1	1		3	2	2	0.5	animation 0.5
782124	12	10	2	1	3	3	3			1			1				
782182	15	10	5	1	3	3	3		Normal transformation in shader extra incorrect. When transforming normals use inverse transpose of the modelToWorld matrix. Also if you want to use homogeneous coordinates with transformations, normal's w component is 0. Also remember to renormalize normals! Nice camera additions overall.	1	1	0	3				

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
782357	9.5	9.5	0	0.5	3	3	3		R1: z-axis translate missing. If you get version control working, submit a log with multiple comments next round to get points.								
782700	10	10	0	1	3	3	3										
783288	10	10	0	1	3	3	3										
783301	0	0	0														
783356	14.5	10	4.5	1	3	3	3			1	1	1	1			0.5	animation 0.5
785707	12.5	10	2.5	1	3	3	3		Animation is pretty fast. You could have made rotate and scale more interactive like, e.g., the translation in R1.		1		1			0.5	animation 0.5
786146	10.5	9.5	1	0.5	3	3	3		R1: z-axis translate missing.	1							
786874	0	0	0														
787226	0	0	0														
788254	34.5	10	24.5	1	3	3	3		Great work!	1	1		3	2	0.5	17	Better reference grid 1; Phong shading + shader refactorization 4; Support for STL files 4; Mesh simplification 8
788539	16	10	6	1	3	3	3			1	1	1	2			1	vertex-only ply loading: 1
788788	0	0	0														
789017	0	0	0														
791610	9	9	0	0.5	2.5	3	3		R1 only y-axis implemented. R2 inverted normals. Positions work as face edges in this case but not generally.								
791678	0	0	0														
793621	11.5	10	1.5	1	3	3	3		Scaling and rotation in wrong order, causes a bit of distortion.	1	0.5						
795658	13.5	9.5	4	1	2.5	3	3		R2: Normals calculated wrong.	1	1	1	1				
795755	15	10	5	1	3	3	3		Animation extra did not work. PLY loader doesn't display anything, so no points :(.	1	1			2	1		
818409	16	10	6	1	3	3	3		Nice virtual trackball!	1	1	1	3				
819754	0	0	0														

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
829948	10	10	0	1	3	3	3	0									
831907	12	10	2	1	3	3	3			1					0.5	0.5	animation: 0.5
838191	9.5	9.5	0	1	2.5	3	3		R2 normal calculation wrong.								
853723	0	0	0														
871831	8	7	1	1	3	3	0			1							
882972	0	0	0														
883353	10	10	0	1	3	3	3										
885665	16.5	10	6.5	1	3	3	3		In the normal transform in the shader with uniform extra the normals are unnormalized.	1	1		3	1.5			
887799	9.5	8.5	1	1	1.5	3	3		R2 tip is not at the origin and face normals are calculated incorrectly - the actual cross product approach is correct, but you recalculate the normal for each vertex with some of them inverted. Pay attention to the direction of the normals!	1							
892179	16	10	6	1	3	3	3		Good job on the trackball camera! Animation could have used a toggle button.	1	1		3		0.5	0.5	animation 0.5;
892292	0	0	0														
892412	13.5	10	3.5	1	3	3	3		Scaling was supposed to be non-uniform along x-axis.	1	1	1				0.5	0.5 mouse scroll to scale model
892690	12	10	2	1	3	3	3		One commit does not count as using version control. Submit log with more commits next round to get points.	0	1	1					
897572	10	10	0	1	3	3	3		One commit does not count as using version control. Submit log with more commits next round to get points.	0							
897925	16	10	6	1	3	3	3		Attempted virtual trackball.	1	1	1	2		0.5	0.5	animation 0.5
903929	0	0	0														
905833	15.5	10	5.5	1	3	3	3		VCS point requires proof of multiple commits.	0	1	1	2			1.5	animation: 0.5, PLY loader with only positions: 1
906971	12	10	2	1	3	3	3			1	1						

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
913498	16	9.5	6.5	1	2.5	3	3		R2: Normals calculated wrong. Pay attention in which order you apply scaling, rotation and translation. As matrix multiplication does not commute.	1	1	1	3			0.5	animation 0.5
913511	0	0	0														
913540	0	0	0														
943413	0	0	0														
952352	0	0	0														
963354	11.5	9.5	2	0.5	3	3	3		R1 no z-axis control.	1	1						
976260	9	9	1	0.5	2.5	3	3	-1	I could not build the assignment. Your submission is missing all framework related source files and utility.hpp. R1 missing z-axis. R2 inverted normals.	1							
976503	17	10	7	1	3	3	3		Please detail your changed keybindings in the readme. Rotate and scale extra has rotation and scaling done in wrong order resulting in distortion.	1	0.5		2	2	1	0.5	animation: 0.5
995212	17	10	7	1	3	3	3			1	1		1	2	1.5	0.5	animation 0.5
995270	18	9	9	0.5	2.5	3	3		R1 only y-axis implemented. See the example! R2 inverted normals. In normal transform extra with uniform normals are unnormalized. Viewport - Nice to freely choose fov! I hope i didn't miss any extra.	1	1		1	1.5	2	2.5	ply loader 2.0; animation 0.5;
995319	16.5	10	6.5	1	3	3	3			1	1		1	2	1	0.5	animation 0.5
995762	15	10	5	1	3	3	3			1	1	1	2				
998743	10	10	0	1	3	3	3		One commit is not enough for VCS point.	0							

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
999263	10.5	9	1.5	0.5	2.5	3	3		R1 missing z-axis. R2 tip not at the origin. Rotate and scale extra wrong ordering of scale, rotation and translation -> applying scale is dependant on translation. Scale in object space.	1	0.5						
1000203	9.5	9.5	0	1	2.5	3	3		R2 inverted normals.								
1001163	9	9	0	1	3	3	2		R4: Almost correct, for faces you need to use the sink like "iss >> f[0] >> sink >> f[1] >> f[2] >> sink >> ...", also the >> syntax simplifies the code significantly.								
1002450	10	9	1	0.5	2.5	3	3		R1 no control for x-axis. R2 cone is missing the last piece. No proof found for usage of version control, please include a couple commit hashes.	0			1				
1002696	1	1	0	1	0	0	0		Git log missing. Add one next round to get points.	0							
1010138	25	9.5	15.5	0.5	3	3	3		R1: z-axis translate missing.	1	1		3	2	2	6.5	animation 0.5; Support for PLY files 2; Mesh simplification 4
1010921	15.5	10	5.5	1	3	3	3		In the shader extra you can call mat3 contructor on modelToWorld to get 3x3 upper matrix. You should also renormalize! Trackball - Approximation is good but does not provide full points! Works well though. Perspective 0.5 points from the easy extra.	1	1	0.5	2		0.5	0.5	animation 0.5;
1011166	10.5	9.5	1	1	2.5	3	3		R2 normal should be the same for all.	1							

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
1015515	9.5	9.5	0	0.5	3	3	3		R1 missing z-axis. 1 commit does not imply the active use of VCS!	0							
1031418	11	9.5	1.5	1	2.5	3	3		R2 inverted normals. For rotate and scale extra, the ordering is wrong. You'd want to do the scaling in object space making the scale result not be dependant on rotation and the new basis -> scale first.	1	0.5						
1034897	11.5	9	2.5	0.5	2.5	3	3		R1: z-axis translate missing. R2: normals calculated wrong	1	1					0.5	animation 0.5
1034907	13	10	3	1	3	3	3			1	1		1				
100063675	10.5	9.5	1	0.5	3	3	3		R1 only one axis control.	1							
100065699	11.5	10	1.5	1	3	3	3			1	0.5						
100077632	10	10	0	1	3	3	3		One commit not enough for VCS point.	0							
100080195	14.5	10	4.5	1	3	3	3			1	1			2		0.5	animation: 0.5
100082119	10	10	0	1	3	3	3		Loaded models looked strange, but the loading code was solid.								
100083817	11	10	1	1	3	3	3			1							
100084638	9.5	8.5	1	0.5	3	3	2		One commit is not enough for VCS point. R1 no z-axis control. R4 indices read in wrong order.	0	1						
100085828	12	10	2	1	3	3	3		Nice FOV animation, good job!	1						1	FOV animation: 1
100087363	9.5	9	0.5	1	2	3	3		R2 normal calculated wrong.							0.5	animation: 0.5,

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
100088595	16.5	9	7.5	0	3	3	3		R1 only x-axis implemented. Scale and rotate extra, you were supposed to scale along x and rotate around y. Read instructions carefully. Transform normals in vertex shader via uniform. When transforming normals in homogeneous coordinates w-component is 0. Also resulting normal is unnormalized. Viewport - I cannot freely choose the fov I want with each screensize. Trackball - math seems correct but once I start spinning there's no going back. It's not too user friendly.	1	0.5		2.5	1	1	1.5	animation 0.5; other formats ply 1;
100088812	12	10	2	1	3	3	3			1	1						
100090114	13.5	10	3.5	1	3	3	3		Please provide a better listing of extra credit done with the corresponding keys. No scaling in rotate and scale extra?		0.5		2.5			0.5	animation: 0.5,
100097625	0	0	0														
100098349	10.5	9.5	1	0.5	3	3	3		R1 z-axis control missing.	1							
100119587	0	0	0														
100126376	13	9.5	3.5	1	2.5	3	3		R2 tip not at the origin. Rotate and scale has wrong ordering of matrix multiplication as you want to scale in object space -> scale first. Normals trans. in uniform. Unnormalized normals after transform. What if scaling is applied? Ply loader is kinda minimal.	1				1.5		1	ply loader 1;

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
100135985	15.5	9.5	6	1	2.5	3	3		Computation of face normals was asked in R2. Smoothing only the lower edge seems strange. In normal transform extra with uniform the normals are unnormalized.	1	1		2	1.5		0.5	smooth translation R1 0.5;
100153873	0	0	0														
27028M	2.5	2.5	0	0	2.5	0	0		In R1 you are using the translation vector in wrong place when creating modelToWorld matrix The translation vector should the on the last column: (tx,ty, tz,1). R2 inverted normals. In the rotate and scale extra scaling was asked to be performed non uniformally only on x-axis.You are scaling uniformally all axes. Scaling is also broken with the incorrect construction of modelToWorld.		0						
35564T	13	10	3	1	3	3	3			1	1	1					

Student number	point total	req total	extra total	R1 moving (1p)	R2 cone (3p)	R3 unpack (3p)	R4 loader (3p)	mod	notes / wtf / ...	VCS (1p)	Rotate and scale (1p)	Normal trans. in shader (1p)	Camera (max 3p)	Normal trans. in uniform (2p)	Viewport & perspective (0.5-2p)	other (points)	other (what)
46477D	10.5	8.5	2	1	1.5	3	3		R2 incorrect normal calculation, tip is not at the origin. To calculate face normals, use edges of the face, i.e get perpendicular direction to the basis of the face. Think of what kind of a triangle the position vectors in their current setup create. Rotate and scale extra, wrong ordering. You'd want to scale in object space so that it is not dependant on world space rotation and translation. -> Scale first. Notice the matrix multiplication order. Ply loader doesn't load the provided cube mesh correctly.	1	0.5					0.5	other fileformats ply 0.5;
68933B	12.5	10	2.5	1	3	3	3		Unnormalized normals in transform normals in vertex shader extra. What if scale was applied?	1	1	0.5					
81616N	0	0	0														
82085F	10	10	0	1	3	3	3										
83818L	18.5	10	8.5	1	3	3	3			1	1		2	2	2	0.5	animation: 0.5
k28342	1	0	1							1							
k90993	11	10	1	1	3	3	3			1							