

## Appendix

### Appendix A: Experiment 1: Angle of Acceptance table results

Best Case Object		Worst Case Object	
Angle	Visibility Rating	Angle	Visibility Rating
0	5	0	5
10	5	10	4
20	5	20	3
30	5	30	2
40	5	40	2
50	5	50	1
55	5	60	1
60	3	70	1
65	2	80	1
70	1	90	1
75	1		
80	0		
90	0		

### Appendix B: Experiment 2: Dolly Capacity / Object Scale Robustness table results

Best Case Object	
Distance from picture origin	Visibility
0cm	5
5	5
10	5
15	5
20	5
25	5
30	5
35	5
40	4
45	4
50	4
55	4
60	4
65	2
70	1
75	1
80	0

Worst Case Object	
Distance from picture origin	Visibility
0cm	5
5	5
10	5
15	5
20	5

25	4
30	3
35	2
40	2
45	2
50	1
55	1
60	1
65	1
70	1
75	1
80	1
85	1
90	0

### **Appendix C: Experiment 3: Noise Limit table results**

Worst Case Object		
Objects added to scene	Camera FPS	Visibility
0	3	5
1	2.93	5
2	2.78	5
3	2.67	4
4	2.64	3
5	2.59	2
6	2.53	1
7	2.50	1
8	2.48	0

Best Case Object		
Objects added to scene	Camera FPS	Visibility
0	2.45	5
1	2.41	5
2	2.39	5
3	2.39	5
4	2.40	5
5	2.36	5
6	2.42	5
7	2.40	5
8	2.37	5
9	2.35	5
10	2.25	5
11	2.26	5
12	2.19	5
13	2.16	5
14	2.17	5
15	2.16	5
30	2.11	5

#### **Appendix D: Experiment 4: Library Limit table results**

Objects Loaded in Library	Camera FPS	Visibility Rating
1	3	5
2	1.12	5
3	0.65	3
4	0.57	2 (minima was 0)
5	0.32*	x
6	0.30	x
7	0.26	x
8	-	x

#### **Appendix E: Experiment 5: Object Distortion table results**

Object Rotation (°)	Visibility
0	5
45	5
90	5
135	4-5
180	5
225	5
270	4-5
315	5

#### **Appendix F: Neilson's Heuristics**

##### **Visibility of system status**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

##### **Match between system and the real world**

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

##### **User control and freedom**

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

##### **Consistency and standards**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

**Error prevention**

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

**Recognition rather than recall**

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

**Flexibility and efficiency of use**

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

**Aesthetic and minimalist design**

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

**Help users recognize, diagnose, and recover from errors**

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

**Help and documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.