|  |  |  |  |
| --- | --- | --- | --- |
| 1 | classdef Moving\_Bar < handle | 1 | classdef Moving\_Bar < handle |
| 2 | % Moving\_Bar: Presents simple hard-edged rectangle as a moving bar. | 2 | % Moving\_Bar: Presents simple hard-edged rectangle as a moving bar. |
| 3 | % | 3 | % |
| 4 | % $Id: Moving\_Bar VER\_ID DATA-TIME vinje $ | 4 | % $Id: Moving\_Bar VER\_ID DATA-TIME vinje $ |
| 5 | % usage: NAME(Args) | 5 | % usage: NAME(Args) |
| 6 | % by: william vinje | 6 | % by: william vinje |
| 7 | % date: Date | 7 | % date: Date |
| 8 | % copyright: (c) Date William Vinje, Eduardo Jose Chichilnisky (GPL see RSM/COPYING) | 8 | % copyright: (c) Date William Vinje, Eduardo Jose Chichilnisky (GPL see RSM/COPYING) |
| 9 | % NB: when using mglQuad my convention is to start in upper left as 0, 0 | 9 | % NB: when using mglQuad my convention is to start in upper left as 0, 0 |
| 10 | % then always work in clockwise manner for sub-quads | 10 | % then always work in clockwise manner for sub-quads |
| 11 | % within each quad or sub-quad vertices are also described in a clockwise | 11 | % within each quad or sub-quad vertices are also described in a clockwise |
| 12 | % manner. | 12 | % manner. |
| 13 |  | 13 |  |
| 14 | properties | 14 | properties |
| 15 |  | 15 |  |
| 16 | stim\_name | 16 | stim\_name |
| 17 |  | 17 |  |
| 18 | run\_date\_time | 18 | run\_date\_time |
| 19 | run\_time\_total | 19 | run\_time\_total |
| 20 |  | 20 |  |
| 21 | main\_trigger | 21 | main\_trigger |
| 22 | tmain0 | 22 | tmain0 |
| 23 |  | 23 |  |
| 24 | rep\_trigger | 24 | rep\_trigger |
| 25 | trep0 | 25 | trep0 |
| 26 |  | 26 |  |
| 27 | run\_duration | 27 | run\_duration |
| 28 | stim\_update\_freq | 28 | stim\_update\_freq |
| 29 |  | 29 |  |
| 30 | run\_script | 30 | run\_script |
| 31 |  | 31 |  |
| 32 | color | 32 | color |
| 33 | backgrndcolor | 33 | backgrndcolor |
| 34 |  | 34 |  |
| 35 | bar\_width | 35 | bar\_width |
| 36 | bar\_height | 36 | bar\_height |
| 37 |  | 37 |  |
| 38 | x\_start | 38 | x\_start |
| 39 | x\_end | 39 | x\_end |
| 40 | y\_start | 40 | y\_start |
| 41 | y\_end | 41 | y\_end |
| 42 |  | 42 |  |
| 43 | num\_reps % this controls how many reps we want to run | 43 | num\_reps % this controls how many reps we want to run |
| 44 |  | 44 |  |
| 45 | reps\_run % this records how many bar passes have already occured | 45 | reps\_run % this records how many bar passes have already occured |
| 46 |  | 46 |  |
| 47 | x\_delta | 47 | x\_delta |
| 48 | y\_delta | 48 | y\_delta |
| 49 | frames | 49 | frames |
| 50 |  | 50 |  |
| 51 | wait\_trigger | 51 | wait\_trigger |
| 52 | wait\_key | 52 | wait\_key |
| 53 | repeat\_num | 53 | repeat\_num |
|  |  | 54 |  |
|  |  | 55 | direction |
|  |  | 56 | interval |
| 54 |  | 57 |  |
| 55 | end % properties block | 58 | end % properties block |
| 56 |  | 59 |  |
| 57 |  | 60 |  |
| 58 |  | 61 |  |
| 59 | methods | 62 | methods |
| 60 |  | 63 |  |
| 61 | % Constructor method | 64 | % Constructor method |
| 62 | function[obj] = Moving\_Bar( stimuli ) | 65 | function[obj] = Moving\_Bar( stimuli ) |
|  |  | 66 | global display |
| 63 |  | 67 |  |
| 64 |  | 68 | if (isfield(stimuli,'bar\_width')) |
| 65 | if (isfield(stimuli,'x\_start')) | 69 | obj.bar\_width = stimuli.bar\_width; |
| 66 | if (isfield(stimuli,'x\_end')) |  |  |
| 67 |  |  |  |
| 68 | % flip around if needed for proper ordering |  |  |
| 69 | if (stimuli.x\_start > stimuli.x\_end) |  |  |
| 70 | temp = stimuli.x\_start; |  |  |
| 71 | stimuli.x\_end = temp; |  |  |
| 72 | stimuli.x\_start = stimuli.x\_end; |  |  |
| 73 | clear temp |  |  |
| 74 | end |  |  |
| 75 |  |  |  |
| 76 | obj.x\_start = stimuli.x\_start; |  |  |
| 77 | obj.x\_end = stimuli.x\_end; |  |  |
| 78 |  |  |  |
| 79 |  |  |  |
| 80 | obj.bar\_width = stimuli.x\_end - stimuli.x\_start; |  |  |
| 81 |  |  |  |
| 82 | else |  |  |
| 83 | fprintf('\t RSM ERROR: x-end not recognized. Please define x\_end value and try again. \n'); |  |  |
| 84 | return |  |  |
| 85 | end |  |  |
| 86 | else | 70 | else |
| 87 | fprintf('\t RSM ERROR: x-start not recognized. Please define x\_start value and try again. \n'); | 71 | fprintf('\t RSM ERROR: bar\_width not recognized. Please define bar\_width value and try again. \n'); |
| 88 | return | 72 | return |
| 89 | end | 73 | end |
| 90 |  | 74 |  |
| 91 |  |  |  |
| 92 |  |  |  |
| 93 | if (isfield(stimuli,'y\_start')) |  |  |
| 94 | if (isfield(stimuli,'y\_end')) |  |  |
| 95 |  |  |  |
| 96 | % flip around if needed for proper ordering |  |  |
| 97 | if (stimuli.y\_start > stimuli.y\_end) |  |  |
| 98 | temp = stimuli.y\_start; |  |  |
| 99 | stimuli.y\_end = temp; |  |  |
| 100 | stimuli.y\_start = stimuli.y\_end; |  |  |
| 101 | clear temp |  |  |
| 102 | end |  |  |
| 103 |  |  |  |
| 104 | obj.y\_start = stimuli.y\_start; |  |  |
| 105 | obj.y\_end = stimuli.y\_end; |  |  |
| 106 |  |  |  |
| 107 | obj.bar\_height = stimuli.y\_end - stimuli.y\_start; |  |  |
| 108 |  |  |  |
| 109 | else |  |  |
| 110 | fprintf('\t RSM ERROR: y-end not recognized. Please define y\_end value and try again. \n'); |  |  |
| 111 | return |  |  |
| 112 | end |  |  |
| 113 | else |  |  |
| 114 | fprintf('\t RSM ERROR: y-start recognized. Please define y\_start value and try again. \n'); |  |  |
| 115 | return |  |  |
| 116 | end |  |  |
| 117 |  |  |  |
| 118 |  |  |  |
| 119 | if (isfield(stimuli,'x\_delta')) |  |  |
| 120 | if (isfield(stimuli,'y\_delta')) |  |  |
| 121 |  |  |  |
| 122 | obj.x\_delta = stimuli.x\_delta; |  |  |
| 123 | obj.y\_delta = stimuli.y\_delta; |  |  |
| 124 |  |  |  |
| 125 | else |  |  |
| 126 | fprintf('\t RSM ERROR: y-delta not recognized. Please define y\_delta value and try again. \n'); |  |  |
| 127 | return |  |  |
| 128 | end |  |  |
| 129 | else |  |  |
| 130 | fprintf('\t RSM ERROR: x-delta recognized. Please define x\_delta value and try again. \n'); |  |  |
| 131 | return |  |  |
| 132 | end |  |  |
| 133 |  |  |  |
| 134 |  |  |  |
| 135 | if (isfield(stimuli,'frames')) |  |  |
| 136 | obj.frames = stimuli.frames; |  |  |
| 137 | else |  |  |
| 138 | fprintf('\t RSM ERROR: frames not recognized. Please define frames value and try again. \n'); |  |  |
| 139 | return |  |  |
| 140 | end |  |  |
| 141 |  |  |  |
| 142 |  |  |  |
| 143 | if (isfield(stimuli,'rgb')) | 75 | if (isfield(stimuli,'rgb')) |
| 144 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; | 76 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; |
| 145 | obj.color = Color\_Test( obj.color ); | 77 | obj.color = Color\_Test( obj.color ); |
| 146 | else | 78 | else |
| 147 | fprintf('\t RSM ERROR: rgb not recognized. Please define rg value and try again. \n'); | 79 | fprintf('\t RSM ERROR: rgb not recognized. Please define rg value and try again. \n'); |
| 148 | return | 80 | return |
| 149 | end | 81 | end |
| 150 |  | 82 |  |
| 151 |  | 83 |  |
| 152 | if (isfield(stimuli,'num\_reps')) | 84 | if (isfield(stimuli,'num\_reps')) |
| 153 | obj.num\_reps = stimuli.num\_reps; | 85 | obj.num\_reps = stimuli.num\_reps; |
| 154 | else | 86 | else |
| 155 | fprintf('\t RSM ERROR: number of repetitions not recognized. Please define num\_reps value and try again. \n'); | 87 | fprintf('\t RSM ERROR: number of repetitions not recognized. Please define num\_reps value and try again. \n'); |
| 156 | return | 88 | return |
| 157 | end | 89 | end |
|  |  | 90 |  |
|  |  | 91 | if (isfield(stimuli,'interval')) |
|  |  | 92 | obj.interval = stimuli.interval; |
|  |  | 93 | else |
|  |  | 94 | fprintf('\t RSM ERROR: interval not recognized. Please define interval value and try again. \n'); |
|  |  | 95 | return |
|  |  | 96 | end |
|  |  | 97 |  |
|  |  | 98 | if (isfield(stimuli,'direction')) |
|  |  | 99 | obj.direction = stimuli.direction; |
|  |  | 100 | L = 3000; % Length of the bar, the idea of making this number large is to ensure that it covers the whole display. |
|  |  | 101 | if stimuli.direction >= 0 && stimuli.direction < 90 |
|  |  | 102 | r0 = [0, display.height]; % coordinates of one display corner. |
|  |  | 103 | obj.x\_start = [0; 0; stimuli.bar\_width; stimuli.bar\_width]; % bar was first set to be either vertical or horizontal (base on moving direction) |
|  |  | 104 | obj.y\_start = [L; -L; -L; L]; |
|  |  | 105 | [obj.x\_start, obj.y\_start] = rotateData(obj.x\_start, obj.y\_start, r0(1), r0(2), stimuli.direction\*pi/180); % then the bar will be rotated around a corner of display by a certain degree (base on moving direction) |
|  |  | 106 | elseif stimuli.direction >= 90 && stimuli.direction < 180 |
|  |  | 107 | r0 = [display.width, display.height]; |
|  |  | 108 | obj.x\_start = [-L; -L; L; L]; |
|  |  | 109 | obj.y\_start = [display.height-stimuli.bar\_width; display.height; display.height; display.height-stimuli.bar\_width]; |
|  |  | 110 | [obj.x\_start, obj.y\_start] = rotateData(obj.x\_start, obj.y\_start, r0(1), r0(2), (stimuli.direction-90)\*pi/180); |
|  |  | 111 | elseif stimuli.direction >= 180 && stimuli.direction < 270 |
|  |  | 112 | r0 = [display.width, 0]; |
|  |  | 113 | obj.x\_start = [display.width-stimuli.bar\_width; display.width-stimuli.bar\_width; display.width; display.width]; |
|  |  | 114 | obj.y\_start = [L; -L; -L; L]; |
|  |  | 115 | [obj.x\_start, obj.y\_start] = rotateData(obj.x\_start, obj.y\_start, r0(1), r0(2), (stimuli.direction-180)\*pi/180); |
|  |  | 116 | elseif stimuli.direction >= 270 && stimuli.direction < 360 |
|  |  | 117 | r0 = [0, 0]; |
|  |  | 118 | obj.x\_start = [-L; -L; L; L]; |
|  |  | 119 | obj.y\_start = [0; stimuli.bar\_width; stimuli.bar\_width; 0]; |
|  |  | 120 | [obj.x\_start, obj.y\_start] = rotateData(obj.x\_start, obj.y\_start, r0(1), r0(2), (stimuli.direction-270)\*pi/180); |
|  |  | 121 | else |
|  |  | 122 | fprintf('\t RSM ERROR: invalid direction. Please define valid direction value and try again. \n'); |
|  |  | 123 | return |
|  |  | 124 | end |
|  |  | 125 | obj.x\_start = obj.x\_start'; |
|  |  | 126 | obj.y\_start = obj.y\_start'; |
|  |  | 127 |  |
|  |  | 128 | % make the bar moving either vertically or horizontally |
|  |  | 129 | % based on which distance is shorter |
|  |  | 130 | x\_dis = (abs(tan(stimuli.direction\*pi/180)\*display.height)+display.width); |
|  |  | 131 | y\_dis = (abs(display.width/tan(stimuli.direction\*pi/180))+display.height); |
|  |  | 132 | [dis, I] = min([x\_dis, y\_dis]); |
|  |  | 133 | if (isfield(stimuli,'delta')) |
|  |  | 134 | switch I |
|  |  | 135 | case 1 |
|  |  | 136 | obj.x\_delta = stimuli.delta/cos(stimuli.direction\*pi/180); |
|  |  | 137 | obj.y\_delta = 0; |
|  |  | 138 | obj.frames = abs(dis/obj.x\_delta) + stimuli.interval; |
|  |  | 139 | case 2 |
|  |  | 140 | obj.y\_delta = stimuli.delta/sin(stimuli.direction\*pi/180); |
|  |  | 141 | obj.x\_delta = 0; |
|  |  | 142 | obj.frames = abs(dis/obj.y\_delta) + stimuli.interval; |
|  |  | 143 | end |
|  |  | 144 |  |
|  |  | 145 | else |
|  |  | 146 | fprintf('\t RSM ERROR: delta recognized. Please define delta value and try again. \n'); |
|  |  | 147 | return |
|  |  | 148 | end |
|  |  | 149 |  |
|  |  | 150 |  |
|  |  | 151 | else |
|  |  | 152 | fprintf('\t RSM ERROR: direction not recognized. Please define direction value and try again. \n'); |
|  |  | 153 | return |
|  |  | 154 | end |
| 158 |  | 155 |  |
| 159 | % Note: color is rgb vector in [0-1] format, vertical vect for mglQuad | 156 | % Note: color is rgb vector in [0-1] format, vertical vect for mglQuad |
| 160 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; | 157 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; |
| 161 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); | 158 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
| 162 |  | 159 |  |
| 163 | obj.wait\_trigger = stimuli.wait\_trigger; | 160 | obj.wait\_trigger = stimuli.wait\_trigger; |
| 164 | obj.wait\_key = stimuli.wait\_key; | 161 | obj.wait\_key = stimuli.wait\_key; |
| 165 |  | 162 |  |
| 166 |  | 163 |  |
| 167 | % The following setup values are not under direct user control | 164 | % The following setup values are not under direct user control |
| 168 | % via the setup script | 165 | % via the setup script |
| 169 |  | 166 |  |
| 170 | obj.stim\_name = 'Moving\_Bar'; | 167 | obj.stim\_name = 'Moving\_Bar'; |
| 171 |  | 168 |  |
| 172 | obj.run\_date\_time = []; | 169 | obj.run\_date\_time = []; |
| 173 | obj.run\_time\_total = []; | 170 | obj.run\_time\_total = []; |
| 174 |  | 171 |  |
| 175 | obj.run\_duration = []; % By setting this to empty we indicate we switch to end condition being a fixed number of reps | 172 | obj.run\_duration = []; % By setting this to empty we indicate we switch to end condition being a fixed number of reps |
| 176 | obj.stim\_update\_freq = []; % By setting this to empty we remove artificial delay in main execution while loop | 173 | obj.stim\_update\_freq = []; % By setting this to empty we remove artificial delay in main execution while loop |
| 177 |  | 174 |  |
| 178 | obj.main\_trigger = 0; | 175 | obj.main\_trigger = 0; |
| 179 | obj.tmain0 = []; | 176 | obj.tmain0 = []; |
| 180 |  | 177 |  |
| 181 | obj.rep\_trigger = 0; | 178 | obj.rep\_trigger = 0; |
| 182 | obj.trep0 = []; | 179 | obj.trep0 = []; |
| 183 |  | 180 |  |
| 184 | obj.run\_script = 'Run\_Bar\_Rep( exp\_obj.stimulus );'; | 181 | obj.run\_script = 'Run\_Bar\_Rep( exp\_obj.stimulus );'; |
| 185 |  | 182 |  |
| 186 | obj.reps\_run = 0; | 183 | obj.reps\_run = 0; |
| 187 | obj.repeat\_num = 0; | 184 | obj.repeat\_num = 0; |
| 188 |  | 185 |  |
| 189 |  | 186 |  |
| 190 | end % constructor | 187 | end % constructor |
| 191 |  | 188 |  |
| 192 |  | 189 |  |
| 193 |  | 190 |  |
| 194 | function Run\_Bar\_Rep( obj ) | 191 | function Run\_Bar\_Rep( obj ) |
| 195 |  | 192 |  |
| 196 | x\_lead\_new = obj.x\_end; | 193 | x\_new = obj.x\_start; |
| 197 | y\_lead\_new = obj.y\_end; | 194 | y\_new = obj.y\_start; |
| 198 |  | 195 |  |
| 199 | % OK: Time to tell DAQ we are starting | 196 | % OK: Time to tell DAQ we are starting |
| 200 | Pulse\_DigOut\_Channel; | 197 | Pulse\_DigOut\_Channel; |
| 201 |  | 198 |  |
| 202 | for frame\_num = 1:obj.frames, | 199 | for frame\_num = 1:obj.frames, |
| 203 |  | 200 |  |
| 204 | % Set up x | 201 | % Set up x |
| 205 | if (obj.x\_delta > 0) | 202 | % Then case is Left 2 right motion |
| 206 | % Then case is Left 2 right motion | 203 |  |
| 207 |  | 204 | % update x\_vertices |
| 208 | % update x\_vertices | 205 | x\_new = x\_new + obj.x\_delta; |
| 209 | x\_lead\_new = x\_lead\_new + obj.x\_delta; | 206 | y\_new = y\_new - obj.y\_delta; |
| 210 | x\_lag\_new = x\_lead\_new - obj.bar\_width; |  |  |
| 211 |  |  |  |
| 212 | else |  |  |
| 213 | x\_lead\_new = x\_lead\_new + obj.x\_delta; |  |  |
| 214 | x\_lag\_new = x\_lead\_new - obj.bar\_width; |  |  |
| 215 |  |  |  |
| 216 | end % x step |  |  |
| 217 |  |  |  |
| 218 |  |  |  |
| 219 | if (obj.y\_delta > 0) |  |  |
| 220 | % Then case is up to down |  |  |
| 221 |  |  |  |
| 222 | % update x\_vertices |  |  |
| 223 | y\_lead\_new = y\_lead\_new + obj.y\_delta; |  |  |
| 224 | y\_lag\_new = y\_lead\_new - obj.bar\_height; |  |  |
| 225 |  |  |  |
| 226 | else |  |  |
| 227 | y\_lead\_new = y\_lead\_new + obj.y\_delta; |  |  |
| 228 | y\_lag\_new = y\_lead\_new - obj.bar\_height; |  |  |
| 229 |  |  |  |
| 230 | end % y step |  |  |
| 231 |  |  |  |
| 232 |  |  |  |
| 233 |  | 207 |  |
| 234 | x\_vertices = [x\_lag\_new; x\_lead\_new; x\_lead\_new; x\_lag\_new]; | 208 | x\_vertices = x\_new; |
| 235 |  | 209 |  |
| 236 | y\_vertices = [y\_lag\_new; y\_lag\_new; y\_lead\_new; y\_lead\_new]; | 210 | y\_vertices = y\_new; |
| 237 |  | 211 |  |
| 238 |  | 212 |  |
| 239 | % Draw the quad | 213 | % Draw the quad |
| 240 | mglClearScreen( obj.backgrndcolor ); | 214 | mglClearScreen( obj.backgrndcolor ); |
|  |  | 215 |  |
|  |  | 216 | new\_color = obj.color + obj.backgrndcolor; |
|  |  | 217 | [ new\_color ] = Color\_Test( new\_color ); |
| 241 |  | 218 |  |
| 242 | mglQuad(x\_vertices, y\_vertices, (obj.color + obj.backgrndcolor), 0); | 219 | mglQuad(x\_vertices, y\_vertices, new\_color, 0); |
| 243 |  | 220 |  |
| 244 | mglFlush(); | 221 | mglFlush(); |
| 245 |  | 222 |  |
| 246 |  | 223 |  |
| 247 | end % loop through frames | 224 | end % loop through frames |
| 248 |  | 225 |  |
| 249 | end % Run bar rep | 226 | end % Run bar rep |
| 250 |  | 227 |  |
| 251 |  | 228 |  |
| 252 | end % methods block | 229 | end % methods block |
| 253 |  | 230 |  |
| 254 |  | 231 |  |
| 255 | end % Moving Bar class | 232 | end % Moving Bar class |