|  |  |  |  |
| --- | --- | --- | --- |
| 1 | classdef PulseCombo < handle | 1 | classdef PulseCombo < handle |
| 2 |  | 2 |  |
| 3 | properties | 3 | properties |
| 4 |  | 4 |  |
| 5 | % Required props for all stim classes | 5 | % Required props for all stim classes |
| 6 | color | 6 | color |
|  |  | 7 | color\_white |
|  |  | 8 | color\_black |
| 7 | backgrndcolor | 9 | backgrndcolor |
| 8 |  | 10 |  |
| 9 | main\_trigger | 11 | main\_trigger |
| 10 | rep\_trigger | 12 | rep\_trigger |
| 11 |  | 13 |  |
| 12 | run\_date\_time | 14 | run\_date\_time |
| 13 | run\_time\_total | 15 | run\_time\_total |
| 14 |  | 16 |  |
| 15 | stim\_name | 17 | stim\_name |
| 16 |  | 18 |  |
| 17 | tmain0 | 19 | tmain0 |
| 18 | trep0 | 20 | trep0 |
| 19 |  | 21 |  |
| 20 | run\_duration | 22 | run\_duration |
| 21 | stim\_update\_freq | 23 | stim\_update\_freq |
| 22 | frames\_per\_halfcycle | 24 | frames\_per\_halfcycle |
| 23 |  | 25 |  |
| 24 | run\_script | 26 | run\_script |
| 25 |  | 27 |  |
| 26 | wait\_key | 28 | wait\_key |
| 27 | wait\_trigger | 29 | wait\_trigger |
| 28 |  | 30 |  |
| 29 | n\_repeats | 31 | n\_repeats |
| 30 | repeat\_num | 32 | repeat\_num |
| 31 |  | 33 |  |
| 32 |  | 34 |  |
| 33 | num\_reps % this controls how many reps we want to run | 35 | num\_reps % this controls how many reps we want to run |
| 34 |  | 36 |  |
| 35 | reps\_run % this records how many bar passes have already occured | 37 | reps\_run % this records how many bar passes have already occured |
| 36 |  | 38 |  |
| 37 |  | 39 |  |
| 38 | frametex | 40 | frametex |
| 39 |  | 41 |  |
| 40 | x\_cen\_offset | 42 | x\_cen\_offset |
| 41 | y\_cen\_offset | 43 | y\_cen\_offset |
| 42 |  | 44 |  |
| 43 | span\_width | 45 | span\_width |
| 44 | span\_height | 46 | span\_height |
| 45 |  | 47 |  |
| 46 | cen\_width | 48 | cen\_width |
| 47 | cen\_height | 49 | cen\_height |
| 48 |  | 50 |  |
| 49 | map\_filename | 51 | map\_filename |
| 50 | lut\_filename | 52 | lut\_filename |
| 51 |  | 53 |  |
| 52 | x\_start | 54 | x\_start |
| 53 | x\_end | 55 | x\_end |
| 54 |  | 56 |  |
| 55 | y\_start | 57 | y\_start |
| 56 | y\_end | 58 | y\_end |
| 57 |  | 59 |  |
| 58 | bar\_width | 60 | bar\_width |
| 59 | bar\_height | 61 | bar\_height |
| 60 |  | 62 |  |
| 61 | w | 63 | w |
| 62 | h | 64 | h |
| 63 |  | 65 |  |
| 64 |  | 66 |  |
| 65 | end % properties block | 67 | end % properties block |
| 66 |  | 68 |  |
| 67 |  | 69 |  |
| 68 | methods | 70 | methods |
| 69 |  | 71 |  |
| 70 | % Constructor method | 72 | % Constructor method |
| 71 | function[obj] = PulseCombo( stimuli, exp\_obj ) | 73 | function[obj] = PulseCombo( stimuli, exp\_obj ) |
| 72 |  | 74 |  |
| 73 | if ( stimuli.control\_flag == 1 ) | 75 | if ( stimuli.control\_flag == 1 ) |
| 74 | %--------------------------------------------------------------------------------------------------------------- ---- | 76 | %--------------------------------------------------------------------------------------------------------------- ---- |
| 75 | % Then we use the stimuli structure constructor mode for pulsing | 77 | % Then we use the stimuli structure constructor mode for pulsing |
| 76 | % cone isolating stim | 78 | % cone isolating stim |
| 77 |  | 79 |  |
| 78 | if (isfield(stimuli,'x\_cen\_offset')) | 80 | if (isfield(stimuli,'x\_cen\_offset')) |
| 79 | if (abs(stimuli.x\_cen\_offset) > (exp\_obj.monitor.width / 2)) | 81 | if (abs(stimuli.x\_cen\_offset) > (exp\_obj.monitor.width / 2)) |
| 80 | fprintf('\t RSM ERROR: X-position offset exceeds 1/2 display width. Please redfine and try again. \n'); | 82 | fprintf('\t RSM ERROR: X-position offset exceeds 1/2 display width. Please redfine and try again. \n'); |
| 81 | return | 83 | return |
| 82 | else | 84 | else |
| 83 | obj.x\_cen\_offset = stimuli.x\_cen\_offset; | 85 | obj.x\_cen\_offset = stimuli.x\_cen\_offset; |
| 84 | end | 86 | end |
| 85 | else | 87 | else |
| 86 | obj.x\_cen\_offset = 0; | 88 | obj.x\_cen\_offset = 0; |
| 87 | end | 89 | end |
| 88 |  | 90 |  |
| 89 |  | 91 |  |
| 90 | if (isfield(stimuli,'y\_cen\_offset')) | 92 | if (isfield(stimuli,'y\_cen\_offset')) |
| 91 | if (abs(stimuli.y\_cen\_offset) > (exp\_obj.monitor.height / 2)) | 93 | if (abs(stimuli.y\_cen\_offset) > (exp\_obj.monitor.height / 2)) |
| 92 | fprintf('\t RSM ERROR: Y-position offset exceeds 1/2 display height. Please redfine and try again. \n'); | 94 | fprintf('\t RSM ERROR: Y-position offset exceeds 1/2 display height. Please redfine and try again. \n'); |
| 93 | return | 95 | return |
| 94 | else | 96 | else |
| 95 | obj.y\_cen\_offset = stimuli.y\_cen\_offset; | 97 | obj.y\_cen\_offset = stimuli.y\_cen\_offset; |
| 96 | end | 98 | end |
| 97 | else | 99 | else |
| 98 | obj.y\_cen\_offset = 0; | 100 | obj.y\_cen\_offset = 0; |
| 99 | end | 101 | end |
| 100 |  | 102 |  |
| 101 |  | 103 |  |
| 102 | if (isfield(stimuli,'num\_reps')) | 104 | if (isfield(stimuli,'num\_reps')) |
| 103 | if (isfield(stimuli,'frames')) | 105 | if (isfield(stimuli,'frames')) |
| 104 |  | 106 |  |
| 105 | obj.num\_reps = stimuli.num\_reps; | 107 | obj.num\_reps = stimuli.num\_reps; |
| 106 | obj.frames\_per\_halfcycle = stimuli.frames; | 108 | obj.frames\_per\_halfcycle = stimuli.frames; |
| 107 |  | 109 |  |
| 108 | else | 110 | else |
| 109 | fprintf('\t RSM ERROR: stimulus update frequency ("interval") not recognized. Please define interval value and try again. \n'); | 111 | fprintf('\t RSM ERROR: stimulus update frequency ("interval") not recognized. Please define interval value and try again. \n'); |
| 110 | return | 112 | return |
| 111 | end | 113 | end |
| 112 | else | 114 | else |
| 113 | fprintf('\t RSM ERROR: num\_rep not recognized. Please define num\_reps value and try again. \n'); | 115 | fprintf('\t RSM ERROR: num\_rep not recognized. Please define num\_reps value and try again. \n'); |
| 114 | return | 116 | return |
| 115 | end | 117 | end |
| 116 |  | 118 |  |
| 117 |  | 119 |  |
| 118 | if (isfield(stimuli,'map\_file\_name')) | 120 | if (isfield(stimuli,'map\_file\_name')) |
| 119 | if (isfield(stimuli,'lut\_file\_name')) | 121 | if (isfield(stimuli,'lut\_file\_name')) |
| 120 |  | 122 |  |
| 121 | map = load( fullfile(exp\_obj.map\_path, stimuli.map\_file\_name) ); | 123 | map = load( fullfile(exp\_obj.map\_path, stimuli.map\_file\_name) ); |
| 122 | load( stimuli.lut\_file\_name, 'lut' ); | 124 | load( stimuli.lut\_file\_name, 'lut' ); |
| 123 |  | 125 |  |
| 124 | else | 126 | else |
| 125 | fprintf('\t RSM ERROR: lut file name not recognized. Please define lut file name and try again. \n'); | 127 | fprintf('\t RSM ERROR: lut file name not recognized. Please define lut file name and try again. \n'); |
| 126 | return | 128 | return |
| 127 | end | 129 | end |
| 128 | else | 130 | else |
| 129 | fprintf('\t RSM ERROR: map file name not recognized. Please define map file name and try again. \n'); | 131 | fprintf('\t RSM ERROR: map file name not recognized. Please define map file name and try again. \n'); |
| 130 | return | 132 | return |
| 131 | end | 133 | end |
| 132 |  | 134 |  |
| 133 |  | 135 |  |
| 134 | %if (isfield(stimuli,'n\_repeats')) | 136 | %if (isfield(stimuli,'n\_repeats')) |
| 135 | % obj.n\_repeats = stimuli.n\_repeats; | 137 | % obj.n\_repeats = stimuli.n\_repeats; |
| 136 | %else | 138 | %else |
| 137 | % fprintf('\t RSM ERROR: n-repeats not recognized. Please define number of repeats value and try again. \n'); | 139 | % fprintf('\t RSM ERROR: n-repeats not recognized. Please define number of repeats value and try again. \n'); |
| 138 | % return | 140 | % return |
| 139 | %end | 141 | %end |
| 140 | obj.n\_repeats = 1; | 142 | obj.n\_repeats = 1; |
| 141 |  | 143 |  |
| 142 |  | 144 |  |
| 143 | obj.span\_width = exp\_obj.monitor.width; | 145 | obj.span\_width = exp\_obj.monitor.width; |
| 144 | obj.span\_height = exp\_obj.monitor.height; | 146 | obj.span\_height = exp\_obj.monitor.height; |
| 145 |  | 147 |  |
| 146 |  | 148 |  |
| 147 | obj.cen\_width = exp\_obj.monitor.cen\_width; | 149 | obj.cen\_width = exp\_obj.monitor.cen\_width; |
| 148 | obj.cen\_height = exp\_obj.monitor.cen\_height; | 150 | obj.cen\_height = exp\_obj.monitor.cen\_height; |
| 149 |  | 151 |  |
| 150 |  | 152 |  |
| 151 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; | 153 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; |
| 152 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); | 154 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
| 153 |  | 155 |  |
| 154 | obj.wait\_trigger = stimuli.wait\_trigger; | 156 | obj.wait\_trigger = stimuli.wait\_trigger; |
| 155 | obj.wait\_key = stimuli.wait\_key; | 157 | obj.wait\_key = stimuli.wait\_key; |
| 156 |  | 158 |  |
| 157 |  | 159 |  |
| 158 | obj.stim\_name = 'Pulse'; | 160 | obj.stim\_name = 'Pulse'; |
| 159 |  | 161 |  |
| 160 | obj.run\_date\_time = []; | 162 | obj.run\_date\_time = []; |
| 161 | obj.run\_time\_total = []; | 163 | obj.run\_time\_total = []; |
| 162 |  | 164 |  |
| 163 | obj.main\_trigger = 0; | 165 | obj.main\_trigger = 0; |
| 164 | obj.tmain0 = []; | 166 | obj.tmain0 = []; |
| 165 |  | 167 |  |
| 166 | obj.rep\_trigger = 0; | 168 | obj.rep\_trigger = 0; |
| 167 | obj.trep0 = []; | 169 | obj.trep0 = []; |
| 168 |  | 170 |  |
| 169 | obj.run\_script = 'Run\_Pulse\_Rep( exp\_obj.stimulus );'; | 171 | obj.run\_script = 'Run\_Pulse\_Rep( exp\_obj.stimulus );'; |
| 170 |  | 172 |  |
| 171 | obj.reps\_run = 0; | 173 | obj.reps\_run = 0; |
| 172 |  | 174 |  |
| 173 | obj.repeat\_num = 0; | 175 | obj.repeat\_num = 0; |
| 174 |  | 176 |  |
| 175 |  | 177 |  |
| 176 | % Convert LUT to linear vector form | 178 | % Convert LUT to linear vector form |
| 177 | num\_lut\_levels = size(lut, 1); | 179 | num\_lut\_levels = size(lut, 1); |
| 178 |  | 180 |  |
| 179 | lv\_i = 0; | 181 | lv\_i = 0; |
| 180 |  | 182 |  |
| 181 | for i = 1:num\_lut\_levels, | 183 | for i = 1:num\_lut\_levels, |
| 182 |  | 184 |  |
| 183 | for j = 1:3, | 185 | for j = 1:3, |
| 184 |  | 186 |  |
| 185 | lv\_i = lv\_i + 1; | 187 | lv\_i = lv\_i + 1; |
| 186 | lut\_vect(lv\_i) = lut(i,j); | 188 | lut\_vect(lv\_i) = lut(i,j); |
| 187 |  | 189 |  |
| 188 | end | 190 | end |
| 189 | end | 191 | end |
| 190 |  | 192 |  |
| 191 |  | 193 |  |
| 192 | % Condition inputs to Make\_Map | 194 | % Condition inputs to Make\_Map |
| 193 | map = uint8( map ); | 195 | map = uint8( map ); |
| 194 |  | 196 |  |
| 195 | lut\_vect = uint8( round( 255 \* lut\_vect )); | 197 | lut\_vect = uint8( round( 255 \* lut\_vect )); |
| 196 |  | 198 |  |
| 197 | backrgb = uint8( round( 255 \* obj.backgrndcolor)); | 199 | backrgb = uint8( round( 255 \* obj.backgrndcolor)); |
| 198 |  | 200 |  |
| 199 | % width, height, lut in mxarray, map\_in\_mxarray, backrgb vect (3 element) | 201 | % width, height, lut in mxarray, map\_in\_mxarray, backrgb vect (3 element) |
| 200 | map = map'; % NB: The transposing of the matrix was estabilished by comparison to the older style code that read in the | 202 | map = map'; % NB: The transposing of the matrix was estabilished by comparison to the older style code that read in the |
| 201 | % map to build up the image mat within matlab. | 203 | % map to build up the image mat within matlab. |
| 202 |  | 204 |  |
| 203 | image\_mat = Make\_Map(size(map,1), size(map,2), lut\_vect, map, backrgb); | 205 | image\_mat = Make\_Map(size(map,1), size(map,2), lut\_vect, map, backrgb); |
| 204 |  | 206 |  |
| 205 | %obj.frametex = mglCreateTexture( image\_mat, [], 0, {'GL\_TEXTURE\_MAG\_FILTER','GL\_NEAREST'} ); | 207 | %obj.frametex = mglCreateTexture( image\_mat, [], 0, {'GL\_TEXTURE\_MAG\_FILTER','GL\_NEAREST'} ); |
| 206 |  | 208 |  |
| 207 | obj.frametex = mglCreateTexture( image\_mat ); | 209 | obj.frametex = mglCreateTexture( image\_mat ); |
| 208 |  | 210 |  |
| 209 | obj.x\_start = []; | 211 | obj.x\_start = []; |
| 210 | obj.x\_end = []; | 212 | obj.x\_end = []; |
| 211 |  | 213 |  |
| 212 | obj.y\_start = []; | 214 | obj.y\_start = []; |
| 213 | obj.y\_end = []; | 215 | obj.y\_end = []; |
| 214 |  | 216 |  |
| 215 | obj.bar\_width = []; | 217 | obj.bar\_width = []; |
| 216 | obj.bar\_height = []; | 218 | obj.bar\_height = []; |
| 217 |  | 219 |  |
| 218 | obj.run\_duration = []; | 220 | obj.run\_duration = []; |
| 219 |  | 221 |  |
| 220 |  | 222 |  |
| 221 |  | 223 |  |
| 222 | elseif ( stimuli.control\_flag == 2 ) | 224 | elseif ( stimuli.control\_flag == 2 ) |
| 223 | %--------------------------------------------------------------------------------------------------------------- ---- | 225 | %--------------------------------------------------------------------------------------------------------------- ---- |
| 224 | % Then we use the S\_file constructor mode | 226 | % Then we use the S\_file constructor mode |
| 225 | % Contents of parsed\_S for Pulse stim | 227 | % Contents of parsed\_S for Pulse stim |
| 226 | % parsed = | 228 | % parsed = |
| 227 | % spec: [1x1 struct] | 229 | % spec: [1x1 struct] |
| 228 | % pulses: [1x3650 struct] | 230 | % pulses: [1x3650 struct] |
| 229 | % numcones: 4 | 231 | % numcones: 4 |
| 230 | % rgbs: {1x3650 cell} | 232 | % rgbs: {1x3650 cell} |
| 231 | % | 233 | % |
| 232 | % parsed.spec | 234 | % parsed.spec |
| 233 | % type: 'PULSE' | 235 | % type: 'PULSE' |
| 234 | % frames: 15 | 236 | % frames: 15 |
| 235 | % delay\_frames: 0 | 237 | % delay\_frames: 0 |
| 236 | % x\_start: 100 | 238 | % x\_start: 100 |
| 237 | % x\_end: 700 | 239 | % x\_end: 700 |
| 238 | % y\_start: 0 | 240 | % y\_start: 0 |
| 239 | % y\_end: 600 | 241 | % y\_end: 600 |
| 240 | % index\_map: ':2012-09-13-2:1234d01:map-0000.txt' | 242 | % index\_map: ':2012-09-13-2:1234d01:map-0000.txt' |
| 241 | % | 243 | % |
| 242 | % parsed.pulses | 244 | % parsed.pulses |
| 243 | % 1x3650 struct array with fields: | 245 | % 1x3650 struct array with fields: |
| 244 | % rgbs | 246 | % rgbs |
| 245 | % parsed.pulses(1) | 247 | % parsed.pulses(1) |
| 246 | % | 248 | % |
| 247 | % rgbs: {{1x3 cell} {1x3 cell} {1x3 cell} {1x3 cell}} | 249 | % rgbs: {{1x3 cell} {1x3 cell} {1x3 cell} {1x3 cell}} |
| 248 | % | 250 | % |
| 249 | % parsed.pulses(1).rgbs{1} | 251 | % parsed.pulses(1).rgbs{1} |
| 250 | % | 252 | % |
| 251 | %                [0.2400] [0.2400] [0.2400] | 253 | %                [0.2400] [0.2400] [0.2400] |
| 252 | % | 254 | % |
| 253 | % parsed.rgbs{1} | 255 | % parsed.rgbs{1} |
| 254 | % | 256 | % |
| 255 | % 0.2400 0.2400 0.2400 | 257 | % 0.2400 0.2400 0.2400 |
| 256 | % 0 0 0 | 258 | % 0 0 0 |
| 257 | % 0 0 0 | 259 | % 0 0 0 |
| 258 | % -0.4800 -0.4800 -0.4800 | 260 | % -0.4800 -0.4800 -0.4800 |
| 259 | % | 261 | % |
| 260 | % each rgbs entry defines a lut appropriate for a single pulse | 262 | % each rgbs entry defines a lut appropriate for a single pulse |
| 261 | % num cones matches the number of rows in the lut matrix for a pulse | 263 | % num cones matches the number of rows in the lut matrix for a pulse |
| 262 |  | 264 |  |
| 263 | obj.x\_cen\_offset = 0; | 265 | obj.x\_cen\_offset = 0; |
| 264 |  | 266 |  |
| 265 | obj.y\_cen\_offset = 0; | 267 | obj.y\_cen\_offset = 0; |
| 266 |  | 268 |  |
| 267 | obj.num\_reps = 1; | 269 | obj.num\_reps = 1; |
| 268 |  | 270 |  |
| 269 | %period = stimuli.parsed\_S.spec.frames / exp\_obj.monitor.screen\_refresh\_freq; | 271 | %period = stimuli.parsed\_S.spec.frames / exp\_obj.monitor.screen\_refresh\_freq; |
| 270 | %obj.stim\_update\_freq = 1/period; | 272 | %obj.stim\_update\_freq = 1/period; |
| 271 | obj.frames\_per\_halfcycle = stimuli.parsed\_S.spec.frames; | 273 | obj.frames\_per\_halfcycle = stimuli.parsed\_S.spec.frames; |
| 272 |  | 274 |  |
| 273 | obj.map\_filename = cat(2, stimuli.map\_file\_path, stimuli.map\_file\_name); | 275 | obj.map\_filename = cat(2, stimuli.map\_file\_path, stimuli.map\_file\_name); |
| 274 |  | 276 |  |
| 275 | obj.n\_repeats = 1; | 277 | obj.n\_repeats = 1; |
| 276 |  | 278 |  |
| 277 | obj.span\_width = exp\_obj.monitor.width; | 279 | obj.span\_width = exp\_obj.monitor.width; |
| 278 | obj.span\_height = exp\_obj.monitor.height; | 280 | obj.span\_height = exp\_obj.monitor.height; |
| 279 |  | 281 |  |
| 280 |  | 282 |  |
| 281 | obj.cen\_width = exp\_obj.monitor.cen\_width; | 283 | obj.cen\_width = exp\_obj.monitor.cen\_width; |
| 282 | obj.cen\_height = exp\_obj.monitor.cen\_height; | 284 | obj.cen\_height = exp\_obj.monitor.cen\_height; |
| 283 |  | 285 |  |
| 284 | % Forced to use default background since Sfile doesn't carry | 286 | % Forced to use default background since Sfile doesn't carry |
| 285 | % necessary info. | 287 | % necessary info. |
| 286 | obj.backgrndcolor = [exp\_obj.monitor.backgrndcolor(1); exp\_obj.monitor.backgrndcolor(2); exp\_obj.monitor.backgrndcolor(3)]; | 288 | obj.backgrndcolor = [exp\_obj.monitor.backgrndcolor(1); exp\_obj.monitor.backgrndcolor(2); exp\_obj.monitor.backgrndcolor(3)]; |
| 287 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); | 289 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
| 288 |  | 290 |  |
| 289 | obj.wait\_trigger = 0; | 291 | obj.wait\_trigger = 0; |
| 290 | obj.wait\_key = 0; | 292 | obj.wait\_key = 0; |
| 291 |  | 293 |  |
| 292 |  | 294 |  |
| 293 | obj.stim\_name = 'Pulse'; | 295 | obj.stim\_name = 'Pulse'; |
| 294 |  | 296 |  |
| 295 | obj.run\_date\_time = []; | 297 | obj.run\_date\_time = []; |
| 296 | obj.run\_time\_total = []; | 298 | obj.run\_time\_total = []; |
| 297 |  | 299 |  |
| 298 | obj.main\_trigger = 0; | 300 | obj.main\_trigger = 0; |
| 299 | obj.tmain0 = []; | 301 | obj.tmain0 = []; |
| 300 |  | 302 |  |
| 301 | obj.rep\_trigger = 0; | 303 | obj.rep\_trigger = 0; |
| 302 | obj.trep0 = []; | 304 | obj.trep0 = []; |
| 303 |  | 305 |  |
| 304 | obj.run\_script = 'Run\_Pulse\_Rep( exp\_obj.stimulus );'; | 306 | obj.run\_script = 'Run\_Pulse\_Rep( exp\_obj.stimulus );'; |
| 305 |  | 307 |  |
| 306 | obj.reps\_run = 0; | 308 | obj.reps\_run = 0; |
| 307 |  | 309 |  |
| 308 | obj.repeat\_num = 0; | 310 | obj.repeat\_num = 0; |
| 309 |  | 311 |  |
| 310 | lut = stimuli.parsed\_S.rgbs{stimuli.index}; | 312 | lut = stimuli.parsed\_S.rgbs{stimuli.index}; |
| 311 |  | 313 |  |
| 312 | % Convert LUT to linear vector form | 314 | % Convert LUT to linear vector form |
| 313 | num\_lut\_levels = size(lut, 1); | 315 | num\_lut\_levels = size(lut, 1); |
| 314 |  | 316 |  |
| 315 | lv\_i = 0; | 317 | lv\_i = 0; |
| 316 |  | 318 |  |
| 317 | for i = 1:num\_lut\_levels, | 319 | for i = 1:num\_lut\_levels, |
| 318 |  | 320 |  |
| 319 | for j = 1:3, | 321 | for j = 1:3, |
| 320 |  | 322 |  |
| 321 | lv\_i = lv\_i + 1; | 323 | lv\_i = lv\_i + 1; |
| 322 | lut\_vect(lv\_i) = (lut(i,j) + obj.backgrndcolor(j)); | 324 | lut\_vect(lv\_i) = (lut(i,j) + obj.backgrndcolor(j)); |
| 323 | lut\_vect(lv\_i) = Color\_Test( lut\_vect(lv\_i) ); | 325 | lut\_vect(lv\_i) = Color\_Test( lut\_vect(lv\_i) ); |
| 324 |  | 326 |  |
| 325 | end | 327 | end |
| 326 | end | 328 | end |
| 327 |  | 329 |  |
| 328 | lut\_vect = uint8( round( 255 \* lut\_vect )); | 330 | lut\_vect = uint8( round( 255 \* lut\_vect )); |
| 329 |  | 331 |  |
| 330 | backrgb = uint8( round( 255 \* obj.backgrndcolor)); | 332 | backrgb = uint8( round( 255 \* obj.backgrndcolor)); |
| 331 |  | 333 |  |
| 332 | % width, height, lut in mxarray, map\_in\_mxarray, backrgb vect (3 element) | 334 | % width, height, lut in mxarray, map\_in\_mxarray, backrgb vect (3 element) |
| 333 |  | 335 |  |
| 334 | image\_mat = Make\_Map(size(stimuli.map,1), size(stimuli.map,2), lut\_vect, stimuli.map, backrgb); | 336 | image\_mat = Make\_Map(size(stimuli.map,1), size(stimuli.map,2), lut\_vect, stimuli.map, backrgb); |
| 335 |  | 337 |  |
| 336 | obj.frametex = mglCreateTexture( image\_mat ); | 338 | obj.frametex = mglCreateTexture( image\_mat ); |
| 337 |  | 339 |  |
| 338 | obj.x\_start = []; | 340 | obj.x\_start = []; |
| 339 | obj.x\_end = []; | 341 | obj.x\_end = []; |
| 340 |  | 342 |  |
| 341 | obj.y\_start = []; | 343 | obj.y\_start = []; |
| 342 | obj.y\_end = []; | 344 | obj.y\_end = []; |
| 343 |  | 345 |  |
| 344 | obj.run\_duration = []; | 346 | obj.run\_duration = []; |
| 345 |  | 347 |  |
| 346 |  | 348 |  |
| 347 | elseif ( stimuli.control\_flag == 3 ) | 349 | elseif ( stimuli.control\_flag == 3 ) |
| 348 | %--------------------------------------------------------------------------------------------------------------- ---- | 350 | %--------------------------------------------------------------------------------------------------------------- ---- |
| 349 |  | 351 |  |
| 350 | if (isfield(stimuli,'rgb')) | 352 | if (isfield(stimuli,'rgb')) |
| 351 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; | 353 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; |
| 352 | obj.color = Color\_Test( obj.color ); | 354 | obj.color = Color\_Test( obj.color ); |
| 353 | else | 355 | else |
| 354 | fprintf('\t RSM ERROR: rgb not recognized. Please define rgb value and try again. \n'); | 356 | fprintf('\t RSM ERROR: rgb not recognized. Please define rgb value and try again. \n'); |
| 355 | return | 357 | return |
| 356 | end | 358 | end |
| 357 |  | 359 |  |
| 358 | if (isfield(stimuli,'back\_rgb')) | 360 | if (isfield(stimuli,'back\_rgb')) |
| 359 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; | 361 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; |
| 360 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); | 362 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
| 361 | else | 363 | else |
| 362 | fprintf('\t RSM ERROR: background rgb not recognized. Please define backgrndcolor value and try again. \n'); | 364 | fprintf('\t RSM ERROR: background rgb not recognized. Please define backgrndcolor value and try again. \n'); |
| 363 | return | 365 | return |
| 364 | end | 366 | end |
| 365 |  | 367 |  |
| 366 |  | 368 |  |
| 367 | if (isfield(stimuli,'num\_reps')) | 369 | if (isfield(stimuli,'num\_reps')) |
| 368 | if (isfield(stimuli,'frames')) | 370 | if (isfield(stimuli,'frames')) |
| 369 |  | 371 |  |
| 370 | obj.num\_reps = stimuli.num\_reps; | 372 | obj.num\_reps = stimuli.num\_reps; |
| 371 | obj.frames\_per\_halfcycle = stimuli.frames; | 373 | obj.frames\_per\_halfcycle = stimuli.frames; |
| 372 |  | 374 |  |
| 373 | else | 375 | else |
| 374 | fprintf('\t RSM ERROR: Frames per half-cycle not recognized. Please define value and try again. \n'); | 376 | fprintf('\t RSM ERROR: Frames per half-cycle not recognized. Please define value and try again. \n'); |
| 375 | return | 377 | return |
| 376 | end | 378 | end |
| 377 | else | 379 | else |
| 378 | fprintf('\t RSM ERROR: num\_rep not recognized. Please define num\_reps value and try again. \n'); | 380 | fprintf('\t RSM ERROR: num\_rep not recognized. Please define num\_reps value and try again. \n'); |
| 379 | return | 381 | return |
| 380 | end | 382 | end |
| 381 |  | 383 |  |
| 382 |  | 384 |  |
| 383 | obj.wait\_trigger = stimuli.wait\_trigger; | 385 | obj.wait\_trigger = stimuli.wait\_trigger; |
| 384 | obj.wait\_key = stimuli.wait\_key; | 386 | obj.wait\_key = stimuli.wait\_key; |
| 385 |  | 387 |  |
| 386 | % The following setup values are not under direct user control | 388 | % The following setup values are not under direct user control |
| 387 | % via the setup script | 389 | % via the setup script |
| 388 | obj.stim\_name = 'Pulse'; % formerly 'Flashing\_Color'; | 390 | obj.stim\_name = 'Pulse'; % formerly 'Flashing\_Color'; |
| 389 |  | 391 |  |
| 390 | obj.run\_date\_time = []; | 392 | obj.run\_date\_time = []; |
| 391 | obj.run\_time\_total = []; | 393 | obj.run\_time\_total = []; |
| 392 |  | 394 |  |
| 393 | obj.main\_trigger = 0; | 395 | obj.main\_trigger = 0; |
| 394 | obj.tmain0 = []; | 396 | obj.tmain0 = []; |
| 395 |  | 397 |  |
| 396 | obj.rep\_trigger = 0; | 398 | obj.rep\_trigger = 0; |
| 397 | obj.trep0 = []; | 399 | obj.trep0 = []; |
| 398 |  | 400 |  |
| 399 | obj.reps\_run = 0; | 401 | obj.reps\_run = 0; |
| 400 |  | 402 |  |
| 401 | obj.run\_script = 'Run\_Flashing\_Color( exp\_obj.stimulus );'; | 403 | obj.run\_script = 'Run\_Flashing\_Color( exp\_obj.stimulus );'; |
| 402 |  | 404 |  |
| 403 | obj.w = exp\_obj.monitor.width; | 405 | obj.w = exp\_obj.monitor.width; |
| 404 | obj.h = exp\_obj.monitor.height; | 406 | obj.h = exp\_obj.monitor.height; |
| 405 | obj.repeat\_num = 0; | 407 | obj.repeat\_num = 0; |
| 406 |  | 408 |  |
| 407 | obj.run\_duration = []; | 409 | obj.run\_duration = []; |
|  |  | 410 |  |
|  |  | 411 |  |
|  |  | 412 | elseif ( stimuli.control\_flag == 5 ) |
|  |  | 413 | %--------------------------------------------------------------------------------------------------------------- ---- |
|  |  | 414 |  |
|  |  | 415 | if (isfield(stimuli,'rgb\_black')) |
|  |  | 416 | obj.color\_black = [stimuli.rgb\_black(1); stimuli.rgb\_black(2); stimuli.rgb\_black(3)]; |
|  |  | 417 | obj.color\_black = Color\_Test( obj.color\_black ); |
|  |  | 418 | else |
|  |  | 419 | fprintf('\t RSM ERROR: rgb black not recognized. Please define rgb black value and try again. \n'); |
|  |  | 420 | return |
|  |  | 421 | end |
|  |  | 422 |  |
|  |  | 423 | if (isfield(stimuli,'rgb\_white')) |
|  |  | 424 | obj.color\_white = [stimuli.rgb\_white(1); stimuli.rgb\_white(2); stimuli.rgb\_white(3)]; |
|  |  | 425 | obj.color\_white = Color\_Test( obj.color\_white ); |
|  |  | 426 | else |
|  |  | 427 | fprintf('\t RSM ERROR: rgb white not recognized. Please define rgb white value and try again. \n'); |
|  |  | 428 | return |
|  |  | 429 | end |
|  |  | 430 |  |
|  |  | 431 | if (isfield(stimuli,'back\_rgb')) |
|  |  | 432 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; |
|  |  | 433 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
|  |  | 434 | else |
|  |  | 435 | fprintf('\t RSM ERROR: background rgb not recognized. Please define backgrndcolor value and try again. \n'); |
|  |  | 436 | return |
|  |  | 437 | end |
|  |  | 438 |  |
|  |  | 439 |  |
|  |  | 440 | if (isfield(stimuli,'num\_reps')) |
|  |  | 441 | if (isfield(stimuli,'frames')) |
|  |  | 442 |  |
|  |  | 443 | obj.num\_reps = stimuli.num\_reps; |
|  |  | 444 | obj.frames\_per\_halfcycle = stimuli.frames; |
|  |  | 445 |  |
|  |  | 446 | else |
|  |  | 447 | fprintf('\t RSM ERROR: Frames per half-cycle not recognized. Please define value and try again. \n'); |
|  |  | 448 | return |
|  |  | 449 | end |
|  |  | 450 | else |
|  |  | 451 | fprintf('\t RSM ERROR: num\_rep not recognized. Please define num\_reps value and try again. \n'); |
|  |  | 452 | return |
|  |  | 453 | end |
|  |  | 454 |  |
|  |  | 455 |  |
|  |  | 456 | obj.wait\_trigger = stimuli.wait\_trigger; |
|  |  | 457 | obj.wait\_key = stimuli.wait\_key; |
|  |  | 458 |  |
|  |  | 459 | % The following setup values are not under direct user control |
|  |  | 460 | % via the setup script |
|  |  | 461 | obj.stim\_name = 'Pulse'; % formerly 'Flashing\_Color'; |
|  |  | 462 |  |
|  |  | 463 | obj.run\_date\_time = []; |
|  |  | 464 | obj.run\_time\_total = []; |
|  |  | 465 |  |
|  |  | 466 | obj.main\_trigger = 0; |
|  |  | 467 | obj.tmain0 = []; |
|  |  | 468 |  |
|  |  | 469 | obj.rep\_trigger = 0; |
|  |  | 470 | obj.trep0 = []; |
|  |  | 471 |  |
|  |  | 472 | obj.reps\_run = 0; |
|  |  | 473 |  |
|  |  | 474 | obj.run\_script = 'Run\_Pulses( exp\_obj.stimulus );'; |
|  |  | 475 |  |
|  |  | 476 | obj.w = exp\_obj.monitor.width; |
|  |  | 477 | obj.h = exp\_obj.monitor.height; |
|  |  | 478 | obj.repeat\_num = 0; |
|  |  | 479 |  |
|  |  | 480 | obj.run\_duration = []; |
| 408 |  | 481 |  |
| 409 |  | 482 |  |
| 410 |  | 483 |  |
| 411 | else | 484 | else |
| 412 | %--------------------------------------------------------------------------------------------------------------- ---- | 485 | %--------------------------------------------------------------------------------------------------------------- ---- |
| 413 | % This is the case to handle simple solid color | 486 | % This is the case to handle simple solid color |
| 414 | if (isfield(stimuli,'x\_start')) | 487 | if (isfield(stimuli,'x\_start')) |
| 415 | if (isfield(stimuli,'x\_end')) | 488 | if (isfield(stimuli,'x\_end')) |
| 416 |  | 489 |  |
| 417 | % flip around if needed for proper ordering | 490 | % flip around if needed for proper ordering |
| 418 | if (stimuli.x\_start > stimuli.x\_end) | 491 | if (stimuli.x\_start > stimuli.x\_end) |
| 419 | temp = stimuli.x\_start; | 492 | temp = stimuli.x\_start; |
| 420 | stimuli.x\_end = temp; | 493 | stimuli.x\_end = temp; |
| 421 | stimuli.x\_start = stimuli.x\_end; | 494 | stimuli.x\_start = stimuli.x\_end; |
| 422 | clear temp | 495 | clear temp |
| 423 | end | 496 | end |
| 424 |  | 497 |  |
| 425 | obj.x\_start = stimuli.x\_start; | 498 | obj.x\_start = stimuli.x\_start; |
| 426 | obj.x\_end = stimuli.x\_end; | 499 | obj.x\_end = stimuli.x\_end; |
| 427 |  | 500 |  |
| 428 |  | 501 |  |
| 429 | obj.bar\_width = stimuli.x\_end - stimuli.x\_start; | 502 | obj.bar\_width = stimuli.x\_end - stimuli.x\_start; |
| 430 |  | 503 |  |
| 431 | else | 504 | else |
| 432 | fprintf('\t RSM ERROR: x-end not recognized. Please define x\_end value and try again. \n'); | 505 | fprintf('\t RSM ERROR: x-end not recognized. Please define x\_end value and try again. \n'); |
| 433 | return | 506 | return |
| 434 | end | 507 | end |
| 435 | else | 508 | else |
| 436 | fprintf('\t RSM ERROR: x-start not recognized. Please define x\_start value and try again. \n'); | 509 | fprintf('\t RSM ERROR: x-start not recognized. Please define x\_start value and try again. \n'); |
| 437 | return | 510 | return |
| 438 | end | 511 | end |
| 439 |  | 512 |  |
| 440 | if (isfield(stimuli,'y\_start')) | 513 | if (isfield(stimuli,'y\_start')) |
| 441 | if (isfield(stimuli,'y\_end')) | 514 | if (isfield(stimuli,'y\_end')) |
| 442 |  | 515 |  |
| 443 | % flip around if needed for proper ordering | 516 | % flip around if needed for proper ordering |
| 444 | if (stimuli.y\_start > stimuli.y\_end) | 517 | if (stimuli.y\_start > stimuli.y\_end) |
| 445 | temp = stimuli.y\_start; | 518 | temp = stimuli.y\_start; |
| 446 | stimuli.y\_end = temp; | 519 | stimuli.y\_end = temp; |
| 447 | stimuli.y\_start = stimuli.y\_end; | 520 | stimuli.y\_start = stimuli.y\_end; |
| 448 | clear temp | 521 | clear temp |
| 449 | end | 522 | end |
| 450 |  | 523 |  |
| 451 | obj.y\_start = stimuli.y\_start; | 524 | obj.y\_start = stimuli.y\_start; |
| 452 | obj.y\_end = stimuli.y\_end; | 525 | obj.y\_end = stimuli.y\_end; |
| 453 |  | 526 |  |
| 454 | obj.bar\_height = stimuli.y\_end - stimuli.y\_start; | 527 | obj.bar\_height = stimuli.y\_end - stimuli.y\_start; |
| 455 |  | 528 |  |
| 456 | else | 529 | else |
| 457 | fprintf('\t RSM ERROR: y-end not recognized. Please define y\_end value and try again. \n'); | 530 | fprintf('\t RSM ERROR: y-end not recognized. Please define y\_end value and try again. \n'); |
| 458 | return | 531 | return |
| 459 | end | 532 | end |
| 460 | else | 533 | else |
| 461 | fprintf('\t RSM ERROR: y-start recognized. Please define y\_start value and try again. \n'); | 534 | fprintf('\t RSM ERROR: y-start recognized. Please define y\_start value and try again. \n'); |
| 462 | return | 535 | return |
| 463 | end | 536 | end |
| 464 |  | 537 |  |
| 465 |  | 538 |  |
| 466 | if (isfield(stimuli,'rgb')) | 539 | if (isfield(stimuli,'rgb')) |
| 467 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; | 540 | obj.color = [stimuli.rgb(1); stimuli.rgb(2); stimuli.rgb(3)]; |
| 468 | obj.color = Color\_Test( obj.color ); | 541 | obj.color = Color\_Test( obj.color ); |
| 469 | else | 542 | else |
| 470 | fprintf('\t RSM ERROR: rgb not recognized. Please define rgb value and try again. \n'); | 543 | fprintf('\t RSM ERROR: rgb not recognized. Please define rgb value and try again. \n'); |
| 471 | return | 544 | return |
| 472 | end | 545 | end |
| 473 |  | 546 |  |
| 474 | if (isfield(stimuli,'back\_rgb')) | 547 | if (isfield(stimuli,'back\_rgb')) |
| 475 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; | 548 | obj.backgrndcolor = [stimuli.back\_rgb(1); stimuli.back\_rgb(2); stimuli.back\_rgb(3)]; |
| 476 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); | 549 | obj.backgrndcolor = Color\_Test( obj.backgrndcolor ); |
| 477 | else | 550 | else |
| 478 | fprintf('\t RSM ERROR: background rgb not recognized. Please define backgrndcolor value and try again. \n'); | 551 | fprintf('\t RSM ERROR: background rgb not recognized. Please define backgrndcolor value and try again. \n'); |
| 479 | return | 552 | return |
| 480 | end | 553 | end |
| 481 |  | 554 |  |
| 482 | obj.stim\_update\_freq = []; | 555 | obj.stim\_update\_freq = []; |
| 483 |  | 556 |  |
| 484 | obj.num\_reps = 1; | 557 | obj.num\_reps = 1; |
| 485 | obj.repeat\_num = 0; | 558 | obj.repeat\_num = 0; |
| 486 |  | 559 |  |
| 487 |  | 560 |  |
| 488 | obj.wait\_trigger = stimuli.wait\_trigger; | 561 | obj.wait\_trigger = stimuli.wait\_trigger; |
| 489 | obj.wait\_key = stimuli.wait\_key; | 562 | obj.wait\_key = stimuli.wait\_key; |
| 490 |  | 563 |  |
| 491 | % The following setup values are not under direct user control | 564 | % The following setup values are not under direct user control |
| 492 | % via the setup script | 565 | % via the setup script |
| 493 | obj.stim\_name = 'Pulse'; | 566 | obj.stim\_name = 'Pulse'; |
| 494 |  | 567 |  |
| 495 | obj.run\_date\_time = []; | 568 | obj.run\_date\_time = []; |
| 496 | obj.run\_time\_total = []; | 569 | obj.run\_time\_total = []; |
| 497 |  | 570 |  |
| 498 | obj.main\_trigger = 0; | 571 | obj.main\_trigger = 0; |
| 499 | obj.tmain0 = []; | 572 | obj.tmain0 = []; |
| 500 |  | 573 |  |
| 501 | obj.rep\_trigger = 0; | 574 | obj.rep\_trigger = 0; |
| 502 | obj.trep0 = []; | 575 | obj.trep0 = []; |
| 503 |  | 576 |  |
| 504 | obj.reps\_run = 0; | 577 | obj.reps\_run = 0; |
| 505 | obj.n\_repeats = 1; | 578 | obj.n\_repeats = 1; |
| 506 |  | 579 |  |
| 507 | obj.run\_script = 'Run\_SimplePulse\_Rep( exp\_obj.stimulus );'; | 580 | obj.run\_script = 'Run\_SimplePulse\_Rep( exp\_obj.stimulus );'; |
| 508 |  | 581 |  |
| 509 | obj.run\_duration = []; | 582 | obj.run\_duration = []; |
| 510 |  | 583 |  |
| 511 | end % % stimuli vs Sfile if-then | 584 | end % % stimuli vs Sfile if-then |
| 512 |  | 585 |  |
| 513 | end % constructor methods | 586 | end % constructor methods |
| 514 |  | 587 |  |
| 515 |  | 588 |  |
| 516 | function Run\_Pulse\_Rep( obj ) | 589 | function Run\_Pulse\_Rep( obj ) |
| 517 |  | 590 |  |
| 518 |  | 591 |  |
| 519 | % blit the texture | 592 | % blit the texture |
| 520 | mglBltTexture( obj.frametex, [(obj.cen\_width + obj.x\_cen\_offset), (obj.cen\_height + obj.y\_cen\_offset), obj.span\_width, obj.span\_height] ); % should be centered | 593 | mglBltTexture( obj.frametex, [(obj.cen\_width + obj.x\_cen\_offset), (obj.cen\_height + obj.y\_cen\_offset), obj.span\_width, obj.span\_height] ); % should be centered |
| 521 |  | 594 |  |
| 522 | mglFlush(); | 595 | mglFlush(); |
| 523 | Pulse\_DigOut\_Channel; | 596 | Pulse\_DigOut\_Channel; |
| 524 |  | 597 |  |
| 525 | RSM\_pulsepause\_nframes(obj, 0); | 598 | RSM\_pulsepause\_nframes(obj, 0); |
| 526 |  | 599 |  |
| 527 | % clear the screen | 600 | % clear the screen |
| 528 | mglClearScreen( obj.backgrndcolor ); | 601 | mglClearScreen( obj.backgrndcolor ); |
| 529 |  | 602 |  |
| 530 | mglFlush(); | 603 | mglFlush(); |
| 531 | Pulse\_DigOut\_Channel; | 604 | Pulse\_DigOut\_Channel; |
| 532 |  | 605 |  |
| 533 | RSM\_pulsepause\_nframes(obj, 1); | 606 | RSM\_pulsepause\_nframes(obj, 1); |
| 534 |  | 607 |  |
| 535 | end % run single flash on or off | 608 | end % run single flash on or off |
| 536 |  | 609 |  |
| 537 |  | 610 |  |
| 538 |  | 611 |  |
| 539 | function Run\_SimplePulse\_Rep( obj ) | 612 | function Run\_SimplePulse\_Rep( obj ) |
| 540 |  | 613 |  |
| 541 | x\_vertices = [obj.x\_start; obj.x\_end; obj.x\_end; obj.x\_start]; | 614 | x\_vertices = [obj.x\_start; obj.x\_end; obj.x\_end; obj.x\_start]; |
| 542 |  | 615 |  |
| 543 | y\_vertices = [obj.y\_end; obj.y\_end; obj.y\_start; obj.y\_start]; | 616 | y\_vertices = [obj.y\_end; obj.y\_end; obj.y\_start; obj.y\_start]; |
| 544 |  | 617 |  |
| 545 | % OK: Time to tell DAQ we are starting | 618 | % OK: Time to tell DAQ we are starting |
| 546 | Pulse\_DigOut\_Channel; | 619 | Pulse\_DigOut\_Channel; |
| 547 |  | 620 |  |
| 548 | % Draw the quad | 621 | % Draw the quad |
| 549 | mglClearScreen( obj.backgrndcolor ); | 622 | mglClearScreen( obj.backgrndcolor ); |
| 550 |  | 623 |  |
| 551 | mglQuad(x\_vertices, y\_vertices, (obj.color + obj.backgrndcolor), 0); | 624 | mglQuad(x\_vertices, y\_vertices, (obj.color + obj.backgrndcolor), 0); |
| 552 |  | 625 |  |
| 553 | mglFlush(); | 626 | mglFlush(); |
| 554 |  | 627 |  |
| 555 | fprintf('\n'); | 628 | fprintf('\n'); |
| 556 | disp('Hit any key to clear screen and return to session stim selection menu: '); | 629 | disp('Hit any key to clear screen and return to session stim selection menu: '); |
| 557 | pause | 630 | pause |
| 558 | fprintf('\n'); | 631 | fprintf('\n'); |
| 559 |  | 632 |  |
| 560 |  | 633 |  |
| 561 | end % simple pulse rep | 634 | end % simple pulse rep |
| 562 |  | 635 |  |
| 563 |  | 636 |  |
| 564 | function[ ] = RSM\_pulsepause\_nframes(obj , backgrnd\_flag) | 637 | function[ ] = RSM\_pulsepause\_nframes(obj , backgrnd\_flag) |
| 565 |  | 638 |  |
| 566 | if (~backgrnd\_flag) | 639 | if (~backgrnd\_flag) |
| 567 |  | 640 |  |
| 568 | for i = 1:obj.frames\_per\_halfcycle, | 641 | for i = 1:obj.frames\_per\_halfcycle, |
| 569 |  | 642 |  |
| 570 | % blit the texture | 643 | % blit the texture |
| 571 | mglBltTexture( obj.frametex, [(obj.cen\_width + obj.x\_cen\_offset), (obj.cen\_height + obj.y\_cen\_offset), obj.span\_width, obj.span\_height] ); % should be centered | 644 | mglBltTexture( obj.frametex, [(obj.cen\_width + obj.x\_cen\_offset), (obj.cen\_height + obj.y\_cen\_offset), obj.span\_width, obj.span\_height] ); % should be centered |
| 572 |  | 645 |  |
| 573 | mglFlush(); | 646 | mglFlush(); |
| 574 | end % loop through number of frames | 647 | end % loop through number of frames |
| 575 |  | 648 |  |
| 576 | else | 649 | else |
| 577 |  | 650 |  |
| 578 | for i = 1:obj.frames\_per\_halfcycle, | 651 | for i = 1:obj.frames\_per\_halfcycle, |
| 579 |  | 652 |  |
| 580 | % clear the screen | 653 | % clear the screen |
| 581 | mglClearScreen( obj.backgrndcolor ); | 654 | mglClearScreen( obj.backgrndcolor ); |
| 582 |  | 655 |  |
| 583 | mglFlush(); | 656 | mglFlush(); |
| 584 |  | 657 |  |
| 585 | end % loop through number of frames | 658 | end % loop through number of frames |
| 586 |  | 659 |  |
| 587 | end % background if-then | 660 | end % background if-then |
| 588 |  | 661 |  |
| 589 | end % end pulsepause | 662 | end % end pulsepause |
| 590 |  | 663 |  |
| 591 |  | 664 |  |
| 592 |  | 665 |  |
| 593 | function Run\_Flashing\_Color( obj ) | 666 | function Run\_Flashing\_Color( obj ) |
| 594 |  | 667 |  |
| 595 |  | 668 |  |
| 596 | x\_vertices = [0; obj.w; obj.w; 0]; | 669 | x\_vertices = [0; obj.w; obj.w; 0]; |
| 597 | y\_vertices = [0; 0; obj.h; obj.h]; | 670 | y\_vertices = [0; 0; obj.h; obj.h]; |
| 598 |  | 671 |  |
| 599 | % First phase: turn on colored flash. | 672 | % First phase: turn on colored flash. |
| 600 | mglQuad(x\_vertices, y\_vertices, obj.color, 0); | 673 | mglQuad(x\_vertices, y\_vertices, obj.color, 0); |
| 601 |  | 674 |  |
| 602 | mglFlush(); | 675 | mglFlush(); |
| 603 | Pulse\_DigOut\_Channel; | 676 | Pulse\_DigOut\_Channel; |
| 604 |  | 677 |  |
| 605 | % Now make sure the second buffer is loaded with the | 678 | % Now make sure the second buffer is loaded with the |
| 606 | % fore-ground | 679 | % fore-ground |
| 607 | mglQuad(x\_vertices, y\_vertices, obj.color, 0); | 680 | mglQuad(x\_vertices, y\_vertices, obj.color, 0); |
| 608 | mglFlush(); | 681 | mglFlush(); |
| 609 |  | 682 |  |
| 610 | RSM\_Pause(obj.frames\_per\_halfcycle-1); | 683 | RSM\_Pause(obj.frames\_per\_halfcycle-1); |
| 611 |  | 684 |  |
| 612 |  | 685 |  |
| 613 | % Now the second phase of the cycle, return to background | 686 | % Now the second phase of the cycle, return to background |
| 614 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); | 687 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
| 615 |  | 688 |  |
| 616 | mglFlush(); | 689 | mglFlush(); |
| 617 | Pulse\_DigOut\_Channel; | 690 | Pulse\_DigOut\_Channel; |
| 618 |  | 691 |  |
| 619 | % Now make sure the second buffer is loaded with the | 692 | % Now make sure the second buffer is loaded with the |
| 620 | % background | 693 | % background |
| 621 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); | 694 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
| 622 | mglFlush(); | 695 | mglFlush(); |
| 623 |  | 696 |  |
| 624 | RSM\_Pause(obj.frames\_per\_halfcycle-1); | 697 | RSM\_Pause(obj.frames\_per\_halfcycle-1); |
| 625 |  | 698 |  |
|  |  | 699 | end % run single flash on or off |
|  |  | 700 |  |
|  |  | 701 | function Run\_Pulses( obj ) |
|  |  | 702 |  |
|  |  | 703 |  |
|  |  | 704 | x\_vertices = [0; obj.w; obj.w; 0]; |
|  |  | 705 | y\_vertices = [0; 0; obj.h; obj.h]; |
|  |  | 706 | % First phase: turn on background. |
|  |  | 707 | Pulse\_DigOut\_Channel; |
|  |  | 708 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
|  |  | 709 |  |
|  |  | 710 | mglFlush(); |
|  |  | 711 |  |
|  |  | 712 |  |
|  |  | 713 | % Now make sure the second buffer is loaded with the |
|  |  | 714 | % background |
|  |  | 715 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
|  |  | 716 | mglFlush(); |
|  |  | 717 |  |
|  |  | 718 | RSM\_Pause(obj.frames\_per\_halfcycle-2); |
|  |  | 719 |  |
|  |  | 720 |  |
|  |  | 721 |  |
|  |  | 722 |  |
|  |  | 723 | % Now the second phase of the cycle, turn on white flash |
|  |  | 724 | Pulse\_DigOut\_Channel; |
|  |  | 725 | mglQuad(x\_vertices, y\_vertices, obj.color\_white, 0); |
|  |  | 726 |  |
|  |  | 727 | mglFlush(); |
|  |  | 728 |  |
|  |  | 729 |  |
|  |  | 730 | % Now make sure the second buffer is loaded with the |
|  |  | 731 | % fore-ground |
|  |  | 732 | mglQuad(x\_vertices, y\_vertices, obj.color\_white, 0); |
|  |  | 733 | mglFlush(); |
|  |  | 734 |  |
|  |  | 735 | RSM\_Pause(obj.frames\_per\_halfcycle-2); |
|  |  | 736 |  |
|  |  | 737 | % third phase: return to background. |
|  |  | 738 | Pulse\_DigOut\_Channel; |
|  |  | 739 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
|  |  | 740 |  |
|  |  | 741 | mglFlush(); |
|  |  | 742 |  |
|  |  | 743 |  |
|  |  | 744 | % Now make sure the second buffer is loaded with the |
|  |  | 745 | % background |
|  |  | 746 | mglQuad(x\_vertices, y\_vertices, obj.backgrndcolor, 0); |
|  |  | 747 | mglFlush(); |
|  |  | 748 |  |
|  |  | 749 | RSM\_Pause(obj.frames\_per\_halfcycle-2); |
|  |  | 750 |  |
|  |  | 751 |  |
|  |  | 752 |  |
|  |  | 753 | % Now the fourth phase of the cycle, go to black |
|  |  | 754 | Pulse\_DigOut\_Channel; |
|  |  | 755 | mglQuad(x\_vertices, y\_vertices, obj.color\_black, 0); |
|  |  | 756 |  |
|  |  | 757 | mglFlush(); |
|  |  | 758 |  |
|  |  | 759 |  |
|  |  | 760 | % Now make sure the second buffer is loaded with the |
|  |  | 761 | % fore-ground |
|  |  | 762 | mglQuad(x\_vertices, y\_vertices, obj.color\_black, 0); |
|  |  | 763 | mglFlush(); |
|  |  | 764 |  |
|  |  | 765 | RSM\_Pause(obj.frames\_per\_halfcycle-2); |
|  |  | 766 |  |
| 626 | end % run single flash on or off | 767 | end % run single flash on or off |
| 627 |  | 768 |  |
| 628 |  | 769 |  |
| 629 |  | 770 |  |
| 630 | end % methods block | 771 | end % methods block |
| 631 |  | 772 |  |
| 632 |  | 773 |  |
| 633 | end % PaintByNumbers class def. | 774 | end % PaintByNumbers class def. |