

```

1  global proc int innovScriptUNI_01(string $mainColour)
2  {
3
4  /* ***** */
5
6  /* SELECT EXISTING PARTICLE OBJECTS AND EMITTERS AND RENAME THEM */
7
8
9  string $oldEmitters[] = `ls -type "pointEmitter"`;
10
11  for($oldEmitterElement in $oldEmitters){
12
13      select -r $oldEmitterElement;
14      rename "oldEmitter";
15      select -cl;
16
17  }
18
19  string $oldParticles[] = `ls -type "particle"`;
20
21  for($oldParticleElement in $oldParticles){
22
23      select $oldParticleElement;
24      pickWalk -d up;
25      rename "oldParticle";
26
27  }
28
29  /* ***** */
30
31
32
33  /* ***** */
34
35
36
37  /* DELETE ANY CAMERAS THAT ARE ALREADY IN THE SCENE */
38
39  string $buffer[];
40  string $buffer2[];
41
42  string $camList[] = `ls -cameras`;
43
44  int $origNumCams = size($camList);
45
46  print $origNumCams;
47
48  for ($eachCam in $camList){
49
50      string $camList[] = `ls -cameras`;
51
52      int $numCams = size($camList);
53

```

```

54 print ("There are " + $numCams + " cameras\n");
55
56 print $camList;
57
58 if (($eachCam != "perspShape") && ($eachCam != "frontShape") && ($eachCam != "sideShape") && ($eachCam != "topShape")){
59
60
61     /* IF IT IS NOT ONE OF THE FOUR DEFAULT CAMERAS */
62
63     tokenize $eachCam "S" $buffer;
64
65     string $tempEnd = $buffer[1];
66
67     /* GET RID OF THE "Shape" PART OF THE NAME... */
68
69     tokenize $tempEnd "e" $buffer2;
70
71     /* ...AND GET THE NUMNER OF THE CAMERA */
72
73     string $start = $buffer[0];
74     string $end = $buffer2[1];
75
76     string $camToDelete = $start + $end;
77
78     /* ...AND JOIN TOGETHER TO GET THE TRANSFORM NODE OF THE CAMERA */
79
80     print ("***** CAMERA TO DELETE: = " + $camToDelete + "*****\n");
81
82     select -r $camToDelete;
83     doDelete;
84
85     /* AND DELETE IT */
86
87     select -cl;
88
89     string $camToDeleteGroup = ($camToDelete + "_group");
90
91     if (`objExists $camToDeleteGroup`)
92     {
93         select -r $camToDeleteGroup;
94         print ("***** CAMERA GROUP TO DELETE: = " + $camToDeleteGroup + "*****\n");
95         doDelete;
96         select -cl;
97     }
98
99
100 }
101
102 }
103
104
105
106

```

```

107  /* ***** */
108
109
110  string $selectionList[] = `ls -type "mesh" -type "nurbsSurface"`;
111
112      /* GET A LIST OF ALL THE POLYGONAL AND NURBS GEOMETRY IN THE SCENE */
113
114  print ("$selectionList is \n\n");
115  print $selectionList;
116  print ("\n\n\n\n");
117
118  int $i;
119  int $i_plusOne;
120
121  string $iString;
122  string $i_plusOneString;
123
124      /* USE THESE STRINGS LATER FOR CONCATENATING INTEGERS INTO STRINGS */
125
126  int $numberOfObjects = size($selectionList);
127
128      /* THE NUMBER OF OBJECTS TO WORK WITH */
129
130  print ("\n\n***** $numberOfObjects = " + $numberOfObjects + " *****\n\n");
131
132
133
134  /* ***** */
135
136  /* CREATE THE FIRST PARTICLE SYSTEM, FOR THE POSITIONS OF THE CAMERAS */
137
138  int $emitter1Check = `objExists "emitter1"`;
139
140  if ($emitter1Check)
141  {
142      select "emitter1";
143      rename "oldEmitter1";
144      select -cl;
145  }
146
147  select -cl;
148
149  emitter -n emitter1 -pos 0 0 0 -type volume -r 100 -sro 0 -nuv 0 -cye none -cyi 1 -spd 1 -srn 2 -nsp 1 -tsp 0.5 -mxd 0 -mnd 0 -dx 1 -dy 0 -dz 0 -sp
150  0 -vsh cube -vof 0 0 0 -vsw 360 -tsr 0.5 -afc 1 -afx 1 -arx 1 -alx 1 -rnd 1 -drs 0 -ssz 0 ;
151
152      /* CREATE THE FIRST PARTICLE EMITTER */
153
154  int $particle1Check = `particleExists "particle1"`;
155
156  if ($particle1Check)
157  {
158
159      select -r "particle1";

```

```

160         rename "oldParticle1";
161         select -cl;
162
163     }
164
165     particle -n "particle1";
166
167     select -r "particle1";
168     pickWalk -d down;
169     rename "particleShape1";
170     select -cl;
171
172     /* CREATE THE FIRST PARTICLE OBJECT */
173
174     connectDynamic -em emitter1 particle1;
175
176     /* AND CONNECT THE TWO TOGETHER */
177
178     setAttr "emitter1.scaleX" 10;
179     setAttr "emitter1.scaleY" 10;
180     setAttr "emitter1.scaleZ" 10;
181
182     setAttr "particleShape1.maxCount" $numberOfObjects;
183
184
185
186     setAttr "particleShape1.conserve" 0.7438;
187
188
189
190     /* ***** */
191
192     /* CREATE A SECOND PARTICLE SYSTEM, FOR THE CAMERA SECOND GOAL OBJECT*/
193
194     int $emitter2Check = `objExists "emitter2"`;
195
196     if ($emitter2Check)
197     {
198         select "emitter2";
199         rename "oldEmitter2";
200         select -cl;
201     }
202
203     select -cl;
204
205
206     emitter -pos 0 0 0 -type volume -name "aimSecondGoalEmitter#" -r 20 -sro 0 -nuv 0 -cye none -cyi 1 -spd 1 -srn 3 -nsp 1 -tsp 0.5 -mxd 0 -mnd
207     0 -dx 1 -dy 0 -dz 0 -sp 0 -vsh cube -vof 0 0 0 -vsw 360 -tsr 0.5 -afc 2 -afx 1 -arx 2 -alx 2 -rnd 2 -drs 0 -ssz 0 ;
208
209     int $particle2Check = `particleExists "particle2"`;
210
211     if ($particle2Check)
212     {

```

```

213
214         select -r "particle2";
215         rename "oldParticle2";
216         select -cl;
217
218     }
219
220 particle -n "particle2";
221
222     select -r "particle2";
223     pickWalk -d down;
224     rename "particleShape2";
225     select -cl;
226
227     connectDynamic -em aimSecondGoalEmitter1 particle2;
228
229     setattr "aimSecondGoalEmitter1.scaleX" 10;
230     setattr "aimSecondGoalEmitter1.scaleY" 10;
231     setattr "aimSecondGoalEmitter1.scaleZ" 10;
232
233     setattr "particleShape2.maxCount" $numberOfObjects;
234
235     setattr "particleShape2.conserve" 0.7438;
236
237
238
239 /* ***** */
240
241 /* CREATE A THIRD PARTICLE SYSTEM, TO CREATE THE CAMERA AIM CONSTRAINTS */
242
243 int $emitter3Check = `objExists "emitter3"`;
244
245 if ($emitter3Check)
246 {
247     select "emitter3";
248     rename "oldEmitter3";
249     select -cl;
250 }
251
252 select -cl;
253
254 emitter -pos 0 0 0 -type volume -name "aimConstraintEmitter#" -r 70 -sro 0 -nuv 0 -cye none -cyi 1 -spd 1 -srn 3 -nsp 1 -tsp 0.5 -mxd 0 -mnd 0
255 -dx 1 -dy 0 -dz 0 -sp 0 -vsh cube -vof 0 0 0 -vsw 360 -tsr 0.5 -afc 2 -afx 1 -arx 2 -alx 2 -rnd 2 -drs 0 -ssz 0 ;
256
257 int $particle3Check = `particleExists "particle3"`;
258
259 if ($particle3Check)
260 {
261
262     select -r "particle3";
263     rename "oldParticle3";
264     select -cl;
265

```

```

266     }
267
268
269 particle -n "particle3";
270
271 select -r "particle3";
272 pickWalk -d down;
273 rename "particleShape3";
274 select -cl;
275
276 connectDynamic -em aimConstraintEmitter1 particle3;
277
278 setAttr "aimConstraintEmitter1.scaleX" 10;
279 setAttr "aimConstraintEmitter1.scaleY" 10;
280 setAttr "aimConstraintEmitter1.scaleZ" 10;
281
282 setAttr "particleShape3.maxCount" $numberOfObjects;
283
284 setAttr "particleShape3.conserve" 0.7438;
285
286 /* ***** */
287 /* ***** */
288
289 /* CREATE INDIVIDUAL PARTICLE SYSTEM FOR EACH AIM CONSTRAINT */
290
291 for($i=1; $i<=$numberOfObjects; $i++)
292 {
293     select -cl;
294     string $i_toString = $i;
295
296     emitter -pos 0 0 0 -type volume -name ("particleAIM_systemEmitter" + $i_toString) -r 70 -sro 0 -nuv 0 -cye none -cyi 1 -spd 1 -srn 3
297 -nsp 1 -tsp 0.5 -mxd 0 -mnd 0 -dx 1 -dy 0 -dz 0 -sp 0 -vsh cube -vof 0 0 0 -vsw 360 -tsr 0.5 -afc 2 -afx 1 -arx 2 -alx 2 -rnd 2 -drs 0 -ssz 0;
298
299     select -cl;
300
301     particle -n ("particleAIM_systemParticle" + $i_toString);
302
303     select -cl;
304
305     connectDynamic -em ("particleAIM_systemEmitter" + $i_toString) ("particleAIM_systemParticle" + $i_toString);
306
307     setAttr ("particleAIM_systemEmitter" + $i_toString + ".scaleX") 10;
308     setAttr ("particleAIM_systemEmitter" + $i_toString + ".scaleY") 10;
309     setAttr ("particleAIM_systemEmitter" + $i_toString + ".scaleZ") 10;
310
311     setAttr ("particleAIM_systemParticle" + $i_toString + ".maxCount") 1;
312
313     setAttr ("particleAIM_systemParticle" + $i_toString + ".conserve") 0.7438;
314
315     select -cl;
316
317 }
318

```

```
319
320
321
322
323
324
325
326 /* **** */
327 /* **** */
328
329
330
331
332
333 /* ADD A VORTEX FIELD TO THE SECOND PARTICLE SYSTEM */
334
335 select -r particle2 ;
336 vortex -pos 0 0 0 -m 5 -att 1 -ax 0 -ay 1 -az 0 -mxd -1 -vsh none -vex 0 -vof 0 0 0 -vsw 360 -tsr 0.5 ;
337
338 connectDynamic -f vortexField1 particle2;
339
340 setattr "vortexField1.magnitude" 100;
341
342 /* ADD A TURBULENCE FIELD TO THE SECOND PARTICLE SYSTEM */
343
344 select -r particle2;
345
346 turbulence -pos 0 0 0 -m 500 -att 1 -f 32 -phaseX 0 -phaseY 0 -phaseZ 0 -noiseLevel 6 -noiseRatio 0.707 -mxd -1 -vsh none -vex 0 -vof 0 0 0
347 -vsw 360 -tsr 0.5 ;
348
349 connectDynamic -f turbulenceField1 particle2;
350
351 setattr "turbulenceField1.magnitude" 300;
352
353
354 /* ADD A RADIAL FIELD TO THE SECOND PARTICLE SYSTEM */
355
356 select -r particle2;
357
358 radial -pos 0 0 0 -m 100 -att 0 -typ 0 -mxd -1 -vsh none -vex 0 -vof 0 0 0 -vsw 360 -tsr 0.5 ;
359
360 connectDynamic -f radialField1 particle2;
361
362 setattr "radialField1.magnitude" 0;
363
364
365
366
367
368
369
370 /* ADD PER OBJECT ATTRIBUTES TO THE PARTICLE OBJECT TO HOLD THE NAMES OF THE OBJECTS IN THE SCENE */
371
```

[illegible]


```

425
426 /* ***** */
427
428
429
430
431
432
433
434 /* ***** */
435
436
437
438 /* PARTICLE RUNTIME AFTER DYNAMICS EXPRESSION TO SET POSITION OF CAMERA TO PARTICLE */
439
440 /* and to move camera aim constraint to point at particlar object */
441
442 dynExpression -s "int $pId = particleShape1.particleId;\r\n$pId = $pId + 1;\r\n\r\nint $maxCount = `getAttr particleShape1.maxCount`; \r\n\r\nif
443 ($pId > $maxCount){\r\n\t$pId = $maxCount;\r\n}\r\n\r\nvector $position = particleShape1.position;\r\nfloat $pX = $position.x;\r\nfloat $pY =
444 $position.y;\r\nfloat $pZ = $position.z;\r\n\r\nsetAttr ("camera" + $pId + ".tx") $pX;\r\nsetAttr ("camera" + $pId + ".ty") $pY;\r\nsetAttr
445 ("camera" + $pId + ".tz") $pZ;\r\n\r\n\r\n-rad particleShape1;
446
447
448
449
450
451 /* PARTICLE CREATRION EXPRESSION FOR SECOND PARTICLE OBJECT */
452
453 dynExpression -s "int $pId = particleShape2.particleId;\r\n$pId = $pId + 1;\r\n\r\nint $maxCount = particleShape2.maxCount;\r\n\r\nstring
454 $maxCount_toString = $maxCount;\r\n\r\nstring $lastAimSphere = ("aimSphere" + $maxCount_toString);\r\n\r\nprint
455 (" \r\n\r\n*****$lastAimSphere = " + $lastAimSphere + "*****\r\n\r\n");\r\n\r\nvector $position = particleShape2.position;\r\nfloat $pX
456 = $position.x;\r\nfloat $pY = $position.y;\r\nfloat $pZ = $position.z;\r\n\r\nif ( `objExists $lastAimSphere` ) == 0 )\r\n{\r\n\tisphere -n
457 ("aimSphere" + $pId);\r\n\tsetAttr ("aimSphere" + $pId + ".tx") $pX;\r\n\tsetAttr ("aimSphere" + $pId + ".ty") $pY;\r\n\tsetAttr
458 ("aimSphere" + $pId + ".tz") $pZ;\r\n\tsetAttr ("aimSphere" + $pId + ".visibility") 0;\r\n}\r\n-c particleShape2;
459
460
461
462
463 /* PARTICLE RUNTIME EXPRESSION FOR SECOND PARTICLE OBJECT */
464
465 dynExpression -s "int $maxCount = particleShape2.maxCount;\r\n\r\nstring $maxCount_toString = $maxCount;\r\n\r\nstring $lastAimSphere =
466 ("aimSphere" + $maxCount_toString);\r\n\r\n\r\nint $pId = particleShape2.particleId;\r\n\r\n$pId = $pId + 1;\r\n\r\nvector $position =
467 particleShape2.position;\r\nfloat $pX = $position.x;\r\nfloat $pY = $position.y;\r\nfloat $pZ = $position.z;\r\n\r\n\r\n\r\nsetAttr ("aimSphere" +
468 $pId + ".tx") $pX;\r\nsetAttr ("aimSphere" + $pId + ".ty") $pY;\r\nsetAttr ("aimSphere" + $pId + ".tz") $pZ;" -rad particleShape2;
469
470
471
472
473
474
475 /* CREATE CREATION EXPRESSION FOR THIRD PARTICLE OBJECT TO CREATE polyCube cameraAimObjects */
476
477 dynExpression -s "int $pId = particleShape3.particleId;\r\n\r\n$pId = $pId + 1;\r\n\r\n\r\nvector $position = particleShape3.position;\r\n\r\nfloat $posX =

```

```

478 $position.x;\r\nfloat $posY = $position.y;\r\nfloat $posZ = $position.z;\r\n\nint $maxCount = particleShape3.maxCount;\n\nstring
479 $maxCount_toString = $maxCount;\n\nstring $lastAimObject = ("cameraAimObject\" + $maxCount_toString);\n\nprint
480 ("\\n\\n*****$lastAimObject = \" + $lastAimObject + \"*****\\n\\n\");\n\nif ( `objExists $lastAimObject` ) == 0 )\n{\n\n\tpolyCube
481 -n ("cameraAimObject\" + $pId);\r\n\n\tsetAttr ("cameraAimObject\" + $pId + \".tx\") $posX;\r\n\tsetAttr ("cameraAimObject\" + $pId +
482 \".ty\") $posY;\r\n\tsetAttr ("cameraAimObject\" + $pId + \".tz\") $posZ;\n\n\tsetAttr ("cameraAimObject\" + $pId + \".visibility\") 0;\n}" -c
483 particleShape3;
484
485
486
487
488 /* STEP FORWARD, SO AIM CONSTRAINT OBJECTS ARE CREATED */
489
490 playButtonStepForward;
491 playButtonStepForward;
492 playButtonStepForward;
493 playButtonStepForward;
494 playButtonStepForward;
495 playButtonStepForward;
496 playButtonStepForward;
497 playButtonStepForward;
498 playButtonStepForward;
499 playButtonStepForward;
500 playButtonStepForward;
501 playButtonStepForward;
502 playButtonStepForward;
503 playButtonStepForward;
504 playButtonStepForward;
505 playButtonStepForward;
506 playButtonStepForward;
507 playButtonStepForward;
508 playButtonStepForward;
509 playButtonStepForward;
510 playButtonStepForward;
511 playButtonStepForward;
512 playButtonStepForward;
513 playButtonStepForward;
514 playButtonStepForward;
515 playButtonStepForward;
516 playButtonStepForward;
517 playButtonStepForward;
518 playButtonStepForward;
519 playButtonStepForward;
520 playButtonStepForward;
521 playButtonStepForward;
522 playButtonStepForward;
523 playButtonStepForward;
524 playButtonStepForward;
525 playButtonStepForward;
526 playButtonStepForward;
527 playButtonStepForward;
528 playButtonStepForward;
529 playButtonStepForward;
530 playButtonStepForward;

```

```

531 playButtonStepForward;
532 playButtonStepForward;
533 playButtonStepForward;
534 playButtonStepForward;
535 playButtonStepForward;
536
537 /* THEN CREATE GOALS WITH A DEFAULT WEIGHT OF 0.3, FOR NOW */
538
539 for($i=1; $i <= $numberOfObjects; $i++)
540 {
541
542     string $i_toString = $i;
543
544     goal -w 0.3 -utr 1 -g $selectionList[$i-1] ("particleAIM_systemParticle" + $i_toString);          /* HAS GOAL INDEX 0
545 */
546     goal -w 0.3 -utr 1 -g ("aimSphere" + $i_toString) ("particleAIM_systemParticle" + $i_toString);          /* HAS GOAL
547 INDEX 1 */
548
549 }
550
551
552
553 /* ***** */
554
555 /* GET LIST OF SHADERS */
556
557 string $objectShaderArray[];
558
559 for($i=0; $i < $numberOfObjects; $i++)
560 {
561
562     string $tempShaderList[] = `listConnections -source true -type shadingEngine $selectionList[$i]`;
563
564                                     /* eg, "blinn2SG */
565
566     print ("***** ***** ** \n\n$tempShaderList: ");
567     print $tempShaderList;
568
569     string $tempShaderList2[] = `listConnections -source true $tempShaderList[0]`;
570
571     print ("***** ***** ** \n\n$tempShaderList2: ");
572     print $tempShaderList2;
573
574
575                                     /* HERE, YOU NEED TO CHECK THAT WHAT YOU'RE GETTING FROM $tempShaderList2[] IS A
576 MATERIAL NODE */
577
578     for ($currentElement in $tempShaderList2)
579     {
580
581
582         select -r $currentElement;
583         string $checkType[] = `ls -sl -showType`;

```

```

584         print ("\n\n&&&&&&&&&&&&&&&&& $checkType[1] for " + $currentElement + " is:\n");
585         print $checkType[1];
586         print ("\n&&&&&&&&&&&&&&&&&\n\n");
587
588         if(($checkType[1] == "lambert")||( $checkType[1] == "blinn")||( $checkType[1] == "phong"))
589             {
590
591                 $objectShaderArray[$i] = $currentElement;
592                 /* eg, "blinn4" or "lambert2" or "boxShader" */
593             }
594
595     }
596
597 }
598
599 print ("\n\n\n*****\n\n\n");
600 print ("**^^^ **^^^ ** \n\n$objectShaderArray: ");
601 print $objectShaderArray;
602 print ("\n\n");
603
604          /* GET COLOUR COMPONENTS OF SHADERS, TO BASE GOAL WEIGHTS ON LATER */
605
606 vector $colourComponents[];
607
608          /* SET THE DOMINANT COLOR TO BE WHAT THE USER WANTS IT TO BE */
609
610 int $dominantColour;
611
612 if ($mainColour == "red")
613     $dominantColour = 0;
614
615     else if ($mainColour == "green")
616         $dominantColour = 1;
617
618     else $dominantColour = 2;
619
620 print ("\nDominant Colour is " + $dominantColour + "\n");
621
622          /* ARRAY TO BE USED TO SET GOAL WEIGHTS LATER */
623
624 float $dominantColourGoalWeights[];
625 float $setColourGoalWeight;
626 vector $tempColourComponents;
627
628 for($i=0; $i < $numberOfObjects; $i++)
629     {
630
631         /* GET THE COLOUR COMPONENTS OF THE MATERIAL INTO A VECTOR ARRAY */
632
633         string $tempColourAttr = ($objectShaderArray[$i] + ".color");
634         float $tempColourVals[] = `getAttr $tempColourAttr`;
635         $colourComponents[$i] = <<$tempColourVals[0], $tempColourVals[1], $tempColourVals[2]>>;
636         print $colourComponents[$i];

```

```

637         print("\n\n");
638     }
639 }
640
641 for($i=0; $i < $numberOfObjects; $i++)
642 {
643
644     /* ASSIGN EACH COLOUR VECTOR TO TEMPORARY VARIABLE TO ACCESS IT */
645
646     $tempColourComponents = $colourComponents[$i];
647
648     /* GET THE APPROPRIATE COLOUR COMPONENT, BASED ON THE DOMINANT COLOUR CHOSEN */
649
650     if($dominantColour == 0)
651         $setColourGoalWeight = $tempColourComponents.x;
652
653     else if ($dominantColour == 1)
654         $setColourGoalWeight = $tempColourComponents.y;
655
656     else if ($dominantColour == 2)
657         $setColourGoalWeight = $tempColourComponents.z;
658
659     /* PUT THE REQUIRED COLOUR COMPONENTS INTO AN ARRAY FOR LATER */
660
661     $dominantColourGoalWeights[$i] = $setColourGoalWeight;
662
663 }
664
665 print("*****\n\n");
666 print("Dominant colour goal weights are ");
667 print $dominantColourGoalWeights;
668 print("\n\n");
669
670
671 /* ***** */
672
673     /* ADD ATTRIBUTES TO PARTICLE SHAPE NODE TO HOLD GOAL WEIGHT VALUES FOR
674     NECESSARY OBJECTS */
675
676     for($i=0; $i<$numberOfObjects; $i++)
677     {
678
679         string $iplusOne_to_string = ($i + 1);
680
681         addAttr -ln ("goalValueObject" + $iplusOne_to_string) -at double -min 0.0 -max 1.0 -dv ($dominantColourGoalWeights[$i]) |
682         particle1;
683         setAttr -e -keyable true ("|particle1.goalValueObject" + $iplusOne_to_string);
684
685     }
686
687
688
689

```

```

690  /* ***** */
691
692
693  /* ADD ATTRIBUTE TO particle1 TO HOLD A VALUE FOR THE OSCILLATION OF THE GOAL WEIGHTS */
694
695  addAttr -ln oscValue -at double -min 0 -max 30 -dv 1 |particle1;
696  setAttr -e -keyable true |particle1.oscValue;
697
698
699  /* ***** */
700
701
702  /* AND GOAL SMOOTHNESS FOR PARTICLE SYSTEM */
703
704  addAttr -ln goalPower -at double -min 0 -max 10 -dv 3 |particle1;
705  setAttr -e -keyable true |particle1.goalPower;
706
707
708  /* CREATE PARTCILE RUNTIME AFTER DYNAMICS EXPRESSIONS FOR AIM SYSTEM PARTICLES */
709
710  /* AND SET THE PARTICLE SYSTEM GOAL SMOOTHNESS TO BE THE USER DEFINED particle1.goalPower */
711
712  /* AND MOVE polyCubes AND camera_aims */
713
714  for ($i=1; $i<=$numberOfObjects; $i++)
715  {
716
717      string $i_to_string = $i;
718
719      dynExpression -s ("int $pId = id;\nstring $pId_to_string = $pId;\nint $objNumber = " +
720
721          $i_to_string +
722
723          "\n\nfloat $oscValue = `getAttr particle1.oscValue`; \n\nstring $objNumber_to_string = $objNumber; \n\nstring
724  $tempParticleAttrName = (`particle1.goalValueObject` + $objNumber_to_string); \n\nfloat $colourGoalVal = `getAttr
725  $tempParticleAttrName`; \n\nfloat $timeGoalVal = abs(sin((time+($objNumber*10))*$oscValue)); \n\n\nfloat $tempGoalSmoothness =
726  particle1.goalPower; \n\nsetAttr `particleAIM_systemParticle` +
727
728      $i_to_string +
729
730      "Shape.goalSmoothness\" $tempGoalSmoothness; \n\n\nsetAttr `particleAIM_systemParticle` +
731
732      $i_to_string +
733
734      "Shape.goalWeight[0]\" $colourGoalVal; \n\n\nsetAttr `particleAIM_systemParticle` +
735
736      $i_to_string +
737
738      "Shape.goalWeight[1]\" $timeGoalVal; \n\n\n\n\n\n\nselect -cl; \n\n/* MOVE polyCubes */ \n\n\n$pId = $pId + 1; \n\nvector $position =
739  position; \n\nfloat $posX = $position.x; \n\nfloat $posY = $position.y; \n\nfloat $posZ = $position.z; \n\n\nsetAttr (`cameraAimObject` + $i_to_string +
740  ".tx\" ) $posX; \n\nsetAttr (`cameraAimObject` + $i_to_string + ".ty\" ) $posY; \n\nsetAttr (`cameraAimObject` + $i_to_string + ".tz\" ) $posZ; \n\n\n/*
741  MOVE CAMERA AIM */ \n\n\nsetAttr (`camera` + $i_to_string + "_aim.tx\" ) $posX; \n\nsetAttr (`camera` + $i_to_string + "_aim.ty\" )
742

```

```

743 $posY;\nsetAttr (\\"camera" + $i_to_string + \"_aim.tz\\") $posZ;")
744
745     -rad ("particleAIM_systemParticle" + $i_to_string + "Shape");
746
747
748 }
749
750
751         /* CREATE A WINDOW WITH A SLIDER TO CHANGE THE VALUE OF particle1.goalPower */
752
753
754
755 window -width 1600 -height 300 -te 60 -le 40 -title "Set Render Settings then batch render" mattWindow2;
756
757 columnLayout -width 1600;
758
759
760
761         /* GOAL POWER SLIDER */
762
763 attrFieldSliderGrp -l "Camera speed" -min 0 -max 10 -at "particle1.goalPower";
764
765         /* PARTICLE CONSERVE MOTION SLIDER */
766
767 attrFieldSliderGrp -l "Camera Aim Speed" -min 0 -max 1 -at "particleShape2.conserve";
768
769         /* particle1.ocsValue SLIDER */
770
771 attrFieldSliderGrp -l "Oscillation Speed" -min 0 -max 30 -at "particle1.oscValue";
772
773         /* VORTEX MAGNITUDE SLIDER */
774
775 attrFieldSliderGrp -l "Vortex strength" -min 0 -max 2000 -at "vortexField1.magnitude";
776
777         /* TURBULENCE MAGNITUDE SLIDER */
778
779 attrFieldSliderGrp -l "Turbulence strength" -min 0 -max 6000 -at "turbulenceField1.magnitude";
780
781         /* RADIAL MAGNITUDE SLIDER */
782
783 attrFieldSliderGrp -l "Radial strength" -min -10 -max 10 -at "radialField1.magnitude";
784
785
786
787         /* CREATE A TEXT BOX WITH THE SHAKE SCRIPT TO LAYER EVERYTHING TOGETHER */
788
789 string $filePrefixBoxName = `textFieldGrp -label "File prefix" -width 800 -text "Enter image file prefix here"`;
790
791 sphere -name "nameSphere";
792
793         /* CREATE A SPHERE TO ADD ATTRIBUTES TO */
794
795 addAttr -ln filePrefix -dt "string" lnameSphere;

```

```

796
797             /* ADD AN ATTRIBUTE TO HOLD THE FILE PREFIX */
798
799 string $newFilePrefix = `textFieldGrp -q -text $filePrefixBoxName`;
800
801             /* GET THE FILE PREFIX FROM THE TEXT BOX... */
802
803 setAttr -type "string" nameSphere.filePrefix $newFilePrefix;
804
805             /* ...AND ASSIGN IT TO THE SPHERE ATTRIBUTE FOR LATER */
806
807 setAttr nameSphere.visibility 0;
808
809             /* MAKE THE SPHERE INVISIBLE */
810
811
812 /* ***** */
813
814
815
816 string $enterFilePrefixButtonName = `button -width 400 -label "Enter file prefix";
817
818             /* BUTTON TO ENTER NEW FILE PREFIX */
819
820 string $enterFilePrefixButtonCmd = ("string $newFilePrefix = `textFieldGrp -q -text " + $filePrefixBoxName + "`;\n" +
821
822             /* QUERIES THE FILE PREFIX THAT THE USER ENTERED... */
823
824 "setAttr -type \"string\" nameSphere.filePrefix $newFilePrefix;\n");
825
826             /* AND ASSIGNS IT TO THE NAME SPHERE */
827
828 button -edit -command $enterFilePrefixButtonCmd $enterFilePrefixButtonName;
829
830             /* ASSIGNS THIS TO THE BUTTON TO ENTER THE FILE PREFIX */
831
832 text -align "center" -label "This is the Shake command that will be executed" -width 500;
833
834 int $startFrame = `getAttr defaultRenderGlobals.startFrame`;
835 int $endFrame = `getAttr defaultRenderGlobals.endFrame`;
836
837             /* GETS THE START AND END FRAMES FROM THE RENDER GLOBALS */
838
839 /* ***** */
840
841             /* START CREATING STRINGS TO BE CONCATENATED INTO A SHAKE COMMAND LATER ON */
842
843 string $newFilePrefix = `getAttr nameSphere.filePrefix`;
844
845             /* CREATES A STRING WITH THE FILE PREFIX IN IT */
846
847 string $shakeTimeRange = ("-t " + $startFrame + "-" + $endFrame);
848

```



```

849      /* CREATES A STRING WITH THE START AND END FRAMES IN IT */
850
851      string $shakeCommand = ("goShake FILE_PREFIX.@@@.tif -label BG1 -fi FILE_PREFIX.@@@.tif -over BG1 " + $shakeTimeRange);
852
853      /* CREATES A DEFAULT START SHAKE COMMAND TO PUT INTO THE TEXT BOX */
854
855      /* ***** */
856
857      string $shakeCommandButtonName = `textFieldGrp -text $shakeCommand -width 600`;
858
859      /* CREATES A TEXT BOX WITH THE DEFAULT SHAKE COMMAND IN IT, FOR NOW */
860
861      string $updateShakeCommandButtonName = `button -width 400 -label "Update Shake command"`;
862
863      /* BUTTON TO UPDATE SHAKE COMMAND */
864
865      string $runShakeButtonName = `button -width 400 -label "Run Shake"`;
866
867      /* BUTTON TO RUN THE SHAKE 'system' COMMAND */
868
869      /* ***** */
870
871      /* CONCATENATE NEW SHAKE COMMAND STRING */
872
873      string $updateShakeCommandButtonCmd_part[100];
874
875      /* STRING ARRAY TO HOLD EACH PART OF THE SHAKE COMMAND, ONE FOR EACH RENDERED
876      SEQUENCE */
877
878      $updateShakeCommandButtonCmd_part[1] =
879
880
881      /* FIRST PART OF THE STRING */
882
883      (
884
885          "int $startFrame = `getattr defaultRenderGlobals.startFrame`;\\n" +
886          "int $endFrame = `getattr defaultRenderGlobals.endFrame`;\\n" +
887
888          "string $newTimeRange = (\\n-t \\n" + $startFrame + "\\n" + $endFrame);\\n" +
889
890          "string $mattFilePrefix = `getattr nameSphere.filePrefix`;\\n" +
891
892          /* WHICH GETS TIME RANGE AND FILE PREFIX */
893
894          "print $mattFilePrefix;\\n" +
895
896          "string $newMattString[];\\n" +
897
898          /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
899
900          /* AND CREATES ANOTHER STRING, WITHIN THIS ONE... */
901

```

```

902 "$newMattString[1] = (\`goShake \` + $mattFilePrefix + \`.@@@.tif" + \` -label BG1 \` + $newTimeRange + \` \`);\n"
903
904 /* WHICH IS THE FIRST PART OF THE ACTUAL SHAKE COMMAND THAT WILL BE EXECUTED */
905
906 /* WHICH GETS THE FIRST IMAGE SEQUENCE, LABELS IT 'BG1', AND SETS THE TIME RANGE */
907
908 /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
909
910 );
911
912 /* ***** */
913
914 /* FOR EACH OBJECT, AND THEREFORE EACH RENDERED SEQUENCE OF IMAGES, CREATE A NEW PART TO */
915 /* THE STRING THAT WILL BE USED TO UPDATE THE SHAKE COMMAND LATER ON */
916
917 for ($i=2; $i <= $numberOfObjects; $i++){
918
919     string $i_toString = $i;
920
921     /* A STRING VERSION OF THE CURRENT ITERATION NUMBER */
922
923     int $i_minusOne = $i-1;
924     string $i_minusOne_to_string = $i_minusOne;
925
926     /* A STRING VERSION OF THE PREVIOUS ITERATION NUMBER */
927
928     $updateShakeCommandButtonCmd_part[$i] =
929
930     (
931
932         "string $mattFilePrefix = `getAttr nameSphere.filePrefix`; \n" +
933
934         /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
935
936         /* AND WITHIN THIS STRING, WE CREATE ANOTHER STRING */
937
938         "$newMattString[" + $i + "] = (\` -fi \` + $mattFilePrefix + \`_cameraShape" + $i_toString + \`.@@@.tif" +
939
940         " -over BG" + $i_minusOne_to_string + " -label BG" + $i_toString + "\`);"
941
942         /* WHICH IS THE NEXT PART OF THE ACTUAL SHAKE COMMAND THAT WILL BE EXECUTED */
943
944         /* WHICH CREATES A 'fileIn' NODE TO GET THE NEXT IMAGE SEQUENCE THAT WAS RENDERED, */
945         /* AND CREATES AN 'over' NODE TO LAYER IT OVER THE PREVIOUS 'BG' LAYER ( eg. 'BG4') */
946         /* AND THEN LABELS THE NEW LAYER (eg. 'BG5') */
947
948         /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
949
950     );
951
952 }
953
954 /* ***** */

```

```

955
956 /* CONCATENATE TOGETHER ALL THE STRINGS THAT HAVE JUST BEEN CREATED */
957 /* TO CREATE THE FINAL SHAKE COMMAND */
958
959 string $updateShakeCommandButtonCmd_ALL = "";
960
961             /* START WITH AN EMPTY STRING */
962
963 for ($i=1; $i <= $numberOfObjects; $i++){
964
965     $updateShakeCommandButtonCmd_ALL += $updateShakeCommandButtonCmd_part[$i];
966
967             /* AND FOR EACH OF THE OBJECTS, AND THEREFORE EACH OF THE RENDERED SEQUENCES */
968             /* AND EACH PART OF THE SHAKE COMMAND, ADD EACH PART TO THE PREVIOUS */
969
970     }
971
972
973
974 /* ***** */
975
976 /* CONCATENATE TOGETHER A STRING THAT WILL ITSELF JOIN TOGETHER AS MANY */
977 /* PARTS OF THE ABOVE SHAKE COMMAND STRINGS AS THERE ARE */
978
979 string $concatenate = "";
980
981             /* START WITH AN EMPTY STRING */
982
983 for($i=1; $i<=$numberOfObjects; $i++){
984
985     string $i_toString = $i;
986
987     if ($i == 1){
988         $concatenate += ("string $newStringFinal = $newMattString[1]");
989     }
990
991             /* FIRST PART GETS THE START OF THE SHAKE COMMAND */
992             /* AND ASSIGNS IT TO THE STRING '$newStringFinal' */
993
994     else{
995         $concatenate += (" + $newMattString[" + $i + "]");
996     }
997
998             /* NEXT PARTS GET THE NEXT PIECES OF THE SHAKE COMMAND */
999
1000     if ($i == $numberOfObjects){
1001
1002         $concatenate += (";\n");
1003
1004             /* END OF THE CONCATENATION STRING */
1005
1006     }
1007 }

```

```

1008 print ("*****\n");
1009 print ("$concatenate = " + $concatenate + "\n\n");
1010 print ("*****\n");
1011
1012
1013
1014
1015 /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
1016
1017 /* NEED THESE STRINGS LATER BECAUSE YOU ARE CONCATENATING STRINGS WITHIN CONCATENATING STRINGS */
1018 /* SO YOU NEED TO ESCAPE ESCAPED CHARACTERS */
1019
1020 string $systemStart = "system(\\\"";
1021
1022 string $systemEnd = "\\\")";
1023
1024 /* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! */
1025
1026 /* ***** */
1027
1028
1029
1030 /* ADD, TO THE ABOVE STRING, A BIT THAT WILL ASSIGN THE STRING '$newStringFinal' */
1031 /* THAT WAS CREATED WITHIN THE ABOVE STRING, TO THE TEXT FIELD THAT DISPLAYS */
1032 /* THE SHAKE COMMAND TO BE EXECUTED */
1033
1034 string $updateShakeCommandButtonCmd_partEND =
1035 (
1036 $concatenate +
1037
1038 "textFieldGrp -e -text $newStringFinal " + $shakeCommandButtonName + ";\n"
1039 );
1040
1041
1042
1043
1044
1045
1046
1047 /* ***** */
1048
1049 /* JOIN ALL BITS TOGETHER */
1050
1051 string $updateShakeCommandButtonCmd = $updateShakeCommandButtonCmd_ALL + $updateShakeCommandButtonCmd_partEND;
1052
1053 /* print ("*****\n");
1054 print ("$updateShakeCommandButtonCmd_ALL = " + $updateShakeCommandButtonCmd_ALL + "\n\n");
1055 print ("*****\n"); */
1056
1057
1058 /* ***** */
1059
1060 /* AND ASSIGN THE FINAL CONCATENATED COMMAND TO THE BUTTON THAT UPDATES THE SHAKE COMMAND */

```

```

1061
1062 button -edit -command $updateShakeCommandButtonCmd $updateShakeCommandButtonName;

1063
1064 /* SO THAT WHEN THIS BUTTON IS PRESSED, IT WILL UPDATE THE SHAKE COMMAND, TAKING INTO ACCOUNT */
1065 /* THE FILE PREFIX, TIME RANGE, AND HOW MANY SEQUENCES OF RENDERED IMAGES THERE ARE */

1066
1067
1068
1069 /* ***** */

1070
1071 /* STRING FOR BUTTON THAT GETS TEXT FROM TEXT FIELD AND EXECUTES IT ( USING THE 'system' COMMAND TO RUN
1072 SHAKE ) */

1073
1074 string $runShakeButtonCmd =

1075
1076 ("string $actualTextToExecute = `textFieldGrp -q -text " + $shakeCommandButtonName + "`;" +

1077
1078 /* HERE, ADD THE NECESSARY BITS TO MAKE IT A SYSTEM COMMAND, BY ADDING STRINGS TO THE START AND END */

1079
1080 "$actualTextToExecute = \"\" + $systemStart + "\"" + $actualTextToExecute + "\"" + $systemEnd + "\"";" +

1081
1082
1083
1084 /* ***** */

1085
1086 /* PRINT SYSTEM COMMAND... */

1087
1088 "print $actualTextToExecute;" +

1089
1090 /* ...BEFORE EXECUTING IT */

1091
1092 "eval $actualTextToExecute;"

1093
1094
1095
1096
1097 );

1098
1099
1100
1101
1102 /* ASSIGN THE FINISHED COMMAND TO THE BUTTON THAT RUNS SHAKE */

1103
1104 button -edit -command $runShakeButtonCmd $runShakeButtonName;

1105
1106
1107
1108
1109
1110 /* DISPLAY THE WINDOW */

1111
1112 showWindow mattWindow2;

1113

```

```

1114
1115
1116
1117 /* **** */
1118
1119 /* **** */
1120
1121 /*          R E N D E R I N G          */
1122
1123 /* **** */
1124
1125 /* GO THROUGH ALL THE CAMERAS AND MAKE THEM ALL UNRENDERABLE, INCLUDING PERSPECTIVE... */
1126 /* ...TOP, SIDE AND FRONT */
1127
1128 string $allCameras[] = `ls -type "camera"`;
1129 string $cam;
1130
1131 for($cam in $allCameras){
1132
1133     setattr ($cam + ".renderable") 0;
1134 }
1135
1136
1137
1138 /* **** */
1139
1140 /* NOW CREATE A RENDER LAYER WITH THE ORIGINAL GEOMETRY AND CREATED CAMERAS IN IT */
1141
1142 select -r $selectionList;
1143
1144 for($i=1; $i<=$numberOfObjects; $i++){
1145
1146     string $iString = $i;
1147     string $renderCamera = ("camera" + $iString);
1148
1149     /* MAKE THE CAMERAS YOU ARE GOING TO USE, RENDERABLE */
1150
1151     setattr ($renderCamera + ".renderable") 1;
1152
1153     select -add $renderCamera;
1154
1155 }
1156
1157
1158 /* CREATE THE NEW RENDER LAYER WITH THE SELECTION ( ALL THE OBJECTS AND CREATED CAMERAS ) */
1159
1160 createRenderLayer -name "layer1" -number 1 -noRecurse `ls -selection`;
1161
1162
1163
1164
1165 /* MAKE THE masterLayer UNRENDERABLE */
1166

```

```
1167 layerEditorLayerButtonRenderabilityChange defaultRenderLayer;
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177 return 1;
1178 }
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
```