



**Universidad Nacional Autónoma
de México**

Facultad de Ingeniería

Semestre 2025-1



Manual técnico de desarrollo

Asignatura:

Laboratorio de Computación Gráfica e Interacción
Humano Computadora

Profesor:

- Ing. José Roque Román Guadarrama

Integrantes:

- González Flores Francisco Eduardo 318074793
- López Betancourt Michelle 318309028
- Rivera Espinoza José Luis 31815252

Grupo laboratorio: 02

Grupo teoría: 06

Fecha de entrega: 2 de noviembre del 2024

→ Tablero

Para el caso de todos los elementos a incluir en el tablero lo primero que se hizo fue realizar las 40 texturas para las casillas, para este caso se crearon 120 para cuando es de día, de noche y cuando las casillas se iluminan cuando pasa el avatar, mediante código se cargaron estas texturas con if's se alterna entre texturas de día , de noche e iluminadas.

```
Proyecto
100 //texturas casillas
101 Texture mapaTexture;
102 Texture ATarbolesD;
103 Texture ATarbolesN;
104 Texture ATarbolesI;
105 Texture ATabejaD;
106 Texture ATabejaN;
107 Texture ATabejaI;
108 Texture ATbmoD;
109 Texture ATbmoN;
110 Texture ATbmoI;
111 Texture ATprismoD;
112 Texture ATprismoN;
113 Texture ATprismoI;
114 Texture ATbobertD;
115 Texture ATbobertN;
116 Texture ATbobertI;
117 Texture ATcaballoD;
118 Texture ATcaballoN;
119 Texture ATcaballoI;
120 Texture ATcasaD;
121 Texture ATcasaN;
122 Texture ATcasaI;
123 Texture ATflamaD;
124 Texture ATflamaN;
125 Texture ATflamaI;
126 Texture ATjakeD;
127 Texture ATjakeN;
128 Texture ATjakeI;
129 Texture ATmarceD;
130 Texture ATmarceN;
131 Texture ATmarceI;
132 Texture ATreinoD;
133 Texture ATreinoN;
```

Proyecto

```
133 Texture ATreinoN;  
134 Texture ATreinoI;  
135 Texture ATunicornD;  
136 Texture ATunicornN;  
137 Texture ATunicornI;  
138 Texture AThunterD;  
139 Texture AThunterN;  
140 Texture AThunterI;  
141 Texture FGbarD;  
142 Texture FGbarN;  
143 Texture FGbarI;  
144 Texture FGhorseD;  
145 Texture FGhorseN;  
146 Texture FGhorseI;  
147 Texture FGchangoD;  
148 Texture FGchangoN;  
149 Texture FGchangoI;  
150 Texture FGernieD;  
151 Texture FGernieN;  
152 Texture FGernieI;  
153 Texture FGperroD;  
154 Texture FGperroN;  
155 Texture FGperroI;  
156 Texture FGrupertD;  
157 Texture FGrupertN;  
158 Texture FGrupertI;  
159 Texture FGvinieD;  
160 Texture FGvinieN;  
161 Texture FGvinieI;  
162 Texture FGjasperD;  
163 Texture FGjasperN;  
164 Texture FGjasperI;  
165 Texture FGtoroD;  
166 Texture FGtoroN;
```

Proyecto

```
166 Texture FGtoroN;  
167 Texture FGtoroI;  
168 Texture FGbrayanD;  
169 Texture FGbrayanN;  
170 Texture FGbrayanI;  
171 Texture FGcasaD;  
172 Texture FGcasaN;  
173 Texture FGcasaI;  
174 Texture FGarbolD;  
175 Texture FGarbolN;  
176 Texture FGarbolI;  
177 Texture FGranaD;  
178 Texture FGranaN;  
179 Texture FGranaI;  
180 Texture FGnavidadD;  
181 Texture FGnavidadN;  
182 Texture FGnavidadI;  
183 Texture RMescuelaD;  
184 Texture RMescuelaN;  
185 Texture RMescuelaI;  
186 Texture RMcasaD;  
187 Texture RMcasaN;  
188 Texture RMcasaI;  
189 Texture RMgarageD;  
190 Texture RMgarageN;  
191 Texture RMgarageI;  
192 Texture RMgatoD;  
193 Texture RMgatoN;  
194 Texture RMgatoI;  
195 Texture RMinsectoD;  
196 Texture RMinsectoN;  
197 Texture RMinsectoI;  
198 Texture RMmeeseekD;  
199 Texture RMmeeseekN;  
200 Texture RMmeeseekI;  
201 Texture RMpaisaje1D;  
202 Texture RMpaisaje1N;
```

```
oyecto
202 Texture RMpaisaje1N;
203 Texture RMpaisaje1I;
204 Texture RMpaisaje2D;
205 Texture RMpaisaje2N;
206 Texture RMpaisaje2I;
207 Texture RMpepinilloD;
208 Texture RMpepinilloN;
209 Texture RMpepinilloI;
210 Texture RMportalD;
211 Texture RMportalN;
212 Texture RMportalI;
213 Texture RMpulpoD;
214 Texture RMpulpoN;
215 Texture RMpulpoI;
216 Texture MRickD;
217 Texture MRickN;
218 Texture MRickI;
219 Texture RMjessicaD;
220 Texture RMjessicaN;
221 Texture RMjessicaI;
222 Texture RMfleebD;
223 Texture RMfleebN;
224 Texture RMfleebI;
225 Texture RMsnowballD;
226 Texture RMsnowballN;
227 Texture RMsnowballI;
228 Texture start;
229
```

```
538 //Texturas casillas
539 ATarbolesD = Texture("Textures/ATarbolesD.png");
540 ATarbolesD.LoadTextureA();
541 ATarbolesN = Texture("Textures/ATarbolesN.png");
542 ATarbolesN.LoadTextureA();
543 ATarbolesI = Texture("Textures/ATarbolesI.png");
544 ATarbolesI.LoadTextureA();
545
546 ATabejaD = Texture("Textures/ATabejaD.png");
547 ATabejaD.LoadTextureA();
548 ATabejaN = Texture("Textures/ATabejaN.png");
549 ATabejaN.LoadTextureA();
550 ATabejaI = Texture("Textures/ATabejaI.png");
551 ATabejaI.LoadTextureA();
552
553 ATbmoD = Texture("Textures/ATbmoD.png");
554 ATbmoD.LoadTextureA();
555 ATbmoN = Texture("Textures/ATbmoN.png");
556 ATbmoN.LoadTextureA();
557 ATbmoI = Texture("Textures/ATbmoI.png");
558 ATbmoI.LoadTextureA();
559
560 ATbobotD = Texture("Textures/ATbobotD.png");
561 ATbobotD.LoadTextureA();
562 ATbobotN = Texture("Textures/ATbobotN.png");
563 ATbobotN.LoadTextureA();
564 ATbobotI = Texture("Textures/ATbobotI.png");
565 ATbobotI.LoadTextureA();
566
567 ATprismoD = Texture("Textures/ATprismoD.png");
568 ATprismoD.LoadTextureA();
569 ATprismoN = Texture("Textures/ATprismoN.png");
570 ATprismoN.LoadTextureA();
571 ATprismoI = Texture("Textures/ATprismoI.png");
572 ATprismoI.LoadTextureA();
573
574 AThunterD = Texture("Textures/AThunterD.png");
```

Proyecto (Ámbito global)

```
574     ATHunterD = Texture("Textures/ATHunterD.png");
575     ATHunterD.LoadTextureA();
576     ATHunterN = Texture("Textures/ATHunterN.png");
577     ATHunterN.LoadTextureA();
578     ATHunterI = Texture("Textures/ATHunterI.png");
579     ATHunterI.LoadTextureA();
580
581     ATcaballoD = Texture("Textures/ATcabaD.png");
582     ATcaballoD.LoadTextureA();
583     ATcaballoN = Texture("Textures/ATcabaN.png");
584     ATcaballoN.LoadTextureA();
585     ATcaballoI = Texture("Textures/ATcabaI.png");
586     ATcaballoI.LoadTextureA();
587
588     ATcasaD = Texture("Textures/ATcasarbolD.png");
589     ATcasaD.LoadTextureA();
590     ATcasaN = Texture("Textures/ATcasarbolN.png");
591     ATcasaN.LoadTextureA();
592     ATcasaI = Texture("Textures/ATcasarbolI.png");
593     ATcasaI.LoadTextureA();
594
595     ATflamaD = Texture("Textures/ATflamaD.png");
596     ATflamaD.LoadTextureA();
597     ATflamaN = Texture("Textures/ATflamaN.png");
598     ATflamaN.LoadTextureA();
599     ATflamaI = Texture("Textures/ATflamaI.png");
600     ATflamaI.LoadTextureA();
601
602     ATjakeD = Texture("Textures/ATjakeD.png");
603     ATjakeD.LoadTextureA();
604     ATjakeN = Texture("Textures/ATjakeN.png");
605     ATjakeN.LoadTextureA();
606     ATjakeI = Texture("Textures/ATjakeI.png");
607     ATjakeI.LoadTextureA();
608
609     ATmarceD = Texture("Textures/ATmarceD.png");
610     ATmarceD.LoadTextureA();
```

Proyecto (Ámbito global)

```
610     ATmarceD.LoadTextureA();
611     ATmarceN = Texture("Textures/ATmarceN.png");
612     ATmarceN.LoadTextureA();
613     ATmarceI = Texture("Textures/ATmarceI.png");
614     ATmarceI.LoadTextureA();
615
616     ATreinoD = Texture("Textures/ATreiniD.png");
617     ATreinoD.LoadTextureA();
618     ATreinoN = Texture("Textures/ATreiniN.png");
619     ATreinoN.LoadTextureA();
620     ATreinoI = Texture("Textures/ATreiniI.png");
621     ATreinoI.LoadTextureA();
622
623     ATunicornD = Texture("Textures/ATunicornD.png");
624     ATunicornD.LoadTextureA();
625     ATunicornN = Texture("Textures/ATunicornN.png");
626     ATunicornN.LoadTextureA();
627     ATunicornI = Texture("Textures/ATunicornI.png");
628     ATunicornI.LoadTextureA();
629
630     FGbarD = Texture("Textures/FGbarD.png");
631     FGbarD.LoadTextureA();
632     FGbarN = Texture("Textures/FGbarN.png");
633     FGbarN.LoadTextureA();
634     FGbarI = Texture("Textures/FGbarI.png");
635     FGbarI.LoadTextureA();
636
637     FGhorseD = Texture("Textures/FGcaballoD.png");
638     FGhorseD.LoadTextureA();
639     FGhorseN = Texture("Textures/FGcaballoN.png");
640     FGhorseN.LoadTextureA();
641     FGhorseI = Texture("Textures/FGcaballoI.png");
642     FGhorseI.LoadTextureA();
643
644     FGchangoD = Texture("Textures/FGchangoD.png");
645     FGchangoD.LoadTextureA();
646     FGchangoN = Texture("Textures/FGchangoN.png");
```



```
Proyecto (Ámbito global)
646 FGchangoN = Texture("Textures/FGchangoN.png");
647 FGchangoN.LoadTextureA();
648 FGchangoI = Texture("Textures/FGchangoI.png");
649 FGchangoI.LoadTextureA();
650
651 FGernieD = Texture("Textures/FGernieD.png");
652 FGernieD.LoadTextureA();
653 FGernieN = Texture("Textures/FGernieN.png");
654 FGernieN.LoadTextureA();
655 FGernieI = Texture("Textures/FGernieI.png");
656 FGernieI.LoadTextureA();
657
658 FGperroD = Texture("Textures/FGperroD.png");
659 FGperroD.LoadTextureA();
660 FGperroN = Texture("Textures/FGperroN.png");
661 FGperroN.LoadTextureA();
662 FGperroI = Texture("Textures/FGperroI.png");
663 FGperroI.LoadTextureA();
664
665 FGrupertD = Texture("Textures/FGrupertD.png");
666 FGrupertD.LoadTextureA();
667 FGrupertN = Texture("Textures/FGrupertN.png");
668 FGrupertN.LoadTextureA();
669 FGrupertI = Texture("Textures/FGrupertI.png");
670 FGrupertI.LoadTextureA();
671
672 FGvinieD = Texture("Textures/FGvinieD.png");
673 FGvinieD.LoadTextureA();
674 FGvinieN = Texture("Textures/FGvinieN.png");
675 FGvinieN.LoadTextureA();
676 FGvinieI = Texture("Textures/FGvinieI.png");
677 FGvinieI.LoadTextureA();
678
679 FGjasperD = Texture("Textures/FGjasperD.png");
680 FGjasperD.LoadTextureA();
681 FGjasperN = Texture("Textures/FGjasperN.png");
682 FGjasperN.LoadTextureA();
```

```
Proyecto (Ámbito global)
691 FGtoroI.LoadTextureA();
692
693 FGbrayanD = Texture("Textures/FGbrayanD.png");
694 FGbrayanD.LoadTextureA();
695 FGbrayanN = Texture("Textures/FGbrayanN.png");
696 FGbrayanN.LoadTextureA();
697 FGbrayanI = Texture("Textures/FGbrayanI.png");
698 FGbrayanI.LoadTextureA();
699
700 FGcasaD = Texture("Textures/FGcasaD.png");
701 FGcasaD.LoadTextureA();
702 FGcasaN = Texture("Textures/FGcasaN.png");
703 FGcasaN.LoadTextureA();
704 FGcasaI = Texture("Textures/FGcasaI.png");
705 FGcasaI.LoadTextureA();
706
707 FGarbolD = Texture("Textures/FGarbolD.png");
708 FGarbolD.LoadTextureA();
709 FGarbolN = Texture("Textures/FGarbolN.png");
710 FGarbolN.LoadTextureA();
711 FGarbolI = Texture("Textures/FGarbolI.png");
712 FGarbolI.LoadTextureA();
713
714 FGranaD = Texture("Textures/FGranaD.png");
715 FGranaD.LoadTextureA();
716 FGranaN = Texture("Textures/FGranaN.png");
717 FGranaN.LoadTextureA();
718 FGranaI = Texture("Textures/FGranaI.png");
719 FGranaI.LoadTextureA();
720
721 FGnavidadD = Texture("Textures/FGnavidadD.png");
722 FGnavidadD.LoadTextureA();
723 FGnavidadN = Texture("Textures/FGnavidadN.png");
724 FGnavidadN.LoadTextureA();
725 FGnavidadI = Texture("Textures/FGnavidadI.png");
726 FGnavidadI.LoadTextureA();
727
```

proyecto (Ámbito global)

```
727
728     RMescuelaD = Texture("Textures/RMescuelaD.png");
729     RMescuelaD.LoadTextureA();
730     RMescuelaN = Texture("Textures/RMescuelaN.png");
731     RMescuelaN.LoadTextureA();
732     RMescuelaI = Texture("Textures/RMescuelaI.png");
733     RMescuelaI.LoadTextureA();
734
735
736     RMcasaD = Texture("Textures/RMcasaD.png");
737     RMcasaD.LoadTextureA();
738     RMcasaN = Texture("Textures/RMcasaN.png");
739     RMcasaN.LoadTextureA();
740     RMcasaI = Texture("Textures/RMcasaI.png");
741     RMcasaI.LoadTextureA();
742
743     RMgarageD = Texture("Textures/RMgarajeD.png");
744     RMgarageD.LoadTextureA();
745     RMgarageN = Texture("Textures/RMgarajeN.png");
746     RMgarageN.LoadTextureA();
747     RMgarageI = Texture("Textures/RMgarajeI.png");
748     RMgarageI.LoadTextureA();
749
750     RMgatoD = Texture("Textures/RMgatoD.png");
751     RMgatoD.LoadTextureA();
752     RMgatoN = Texture("Textures/RMgatoN.png");
753     RMgatoN.LoadTextureA();
754     RMgatoI = Texture("Textures/RMgatoI.png");
755     RMgatoI.LoadTextureA();
756
757     RMinsectoD = Texture("Textures/RMinsectoD.png");
758     RMinsectoD.LoadTextureA();
759     RMinsectoN = Texture("Textures/RMinsectoN.png");
760     RMinsectoN.LoadTextureA();
761     RMinsectoI = Texture("Textures/RMinsectoI.png");
762     RMinsectoI.LoadTextureA();
763
```

```
Proyecto (Ámbito global)
796 RMpulpoI = Texture("Textures/RMpulpoI.png");
797 RMpulpoI.LoadTextureA();
798
799 RMrickD = Texture("Textures/RMrickD.png");
800 RMrickD.LoadTextureA();
801 RMrickN = Texture("Textures/RMrickN.png");
802 RMrickN.LoadTextureA();
803 RMrickI = Texture("Textures/RMrickI.png");
804 RMrickI.LoadTextureA();
805
806 RMsnowballD = Texture("Textures/RMsnowballD.png");
807 RMsnowballD.LoadTextureA();
808 RMsnowballN = Texture("Textures/RMsnowballN.png");
809 RMsnowballN.LoadTextureA();
810 RMsnowballI = Texture("Textures/RMsnowballI.png");
811 RMsnowballI.LoadTextureA();
812
813 RMjessicaD = Texture("Textures/RMjessicaD.png");
814 RMjessicaD.LoadTextureA();
815 RMjessicaN = Texture("Textures/RMjessicaN.png");
816 RMjessicaN.LoadTextureA();
817 RMjessicaI = Texture("Textures/RMjessicaI.png");
818 RMjessicaI.LoadTextureA();
819
820 RMfleebeD = Texture("Textures/RMfleebeD.png");
821 RMfleebeD.LoadTextureA();
822 RMfleebeN = Texture("Textures/RMfleebeN.png");
823 RMfleebeN.LoadTextureA();
824 RMfleebeI = Texture("Textures/RMfleebeI.png");
825 RMfleebeI.LoadTextureA();
826
827 start = Texture("Textures/start.png");
828 start.LoadTextureA();
829 // modelos
830
```

```
1384 //c1
1385 model = glm::mat4(1.0);
1386 model = glm::translate(model, glm::vec3(94.5f, 1.0f, -129.2f));
1387 model = glm::scale(model, glm::vec3(22.5f, 1.0f, 41.5f));
1388 modelaux = model;
1389 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1390 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1391 if (dia) {
1392     ATarbolesD.UseTexture();
1393 }
1394 if (noche) {
1395     ATarbolesN.UseTexture();
1396 }
1397 if (iluminada) {
1398     ATarbolesI.UseTexture();
1399 }
1400 //ATarbolesD.UseTexture();
1401 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1402 meshList[4]->RenderMesh();
1403
1404 //c2
1405 model = modelaux;
1406 model = glm::translate(model, glm::vec3(-1.05f, 0.0f, 0.0f));
1407 if (dia) {
1408     RMgarageD.UseTexture();
1409 }
1410 if (noche) {
1411     RMgarageN.UseTexture();
1412 }
1413 if (iluminada) {
1414     RMgarageI.UseTexture();
1415 }
1416 //RMgarageD.UseTexture();
1417 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1418 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1419 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1420 meshList[4]->RenderMesh();
```

```
1420 meshList[4]->RenderMesh();
1421
1422 //c3
1423 model = modelaux;
1424 model = glm::translate(model, glm::vec3(-2.1f, 0.0f, 0.0f));
1425 if (dia) {
1426     RMfleebeD.UseTexture();
1427 }
1428 if (noche) {
1429     RMfleebeN.UseTexture();
1430 }
1431 if (iluminada) {
1432     RMfleebeI.UseTexture();
1433 }
1434 //RMfleebeD.UseTexture();
1435 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1436 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1437 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1438 meshList[4]->RenderMesh();
1439
1440 //c4
1441 model = modelaux;
1442 model = glm::translate(model, glm::vec3(-3.15f, 0.0f, 0.0f));
1443 if (dia) {
1444     RMjessicaD.UseTexture();
1445 }
1446 if (noche) {
1447     RMjessicaN.UseTexture();
1448 }
1449 if (iluminada) {
1450     RMjessicaI.UseTexture();
1451 }
1452 //RMjessicaD.UseTexture();
1453 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1454 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1455 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1456 meshList[4]->RenderMesh();
```

```
1456     meshList[4]->RenderMesh();
1457
1458     //c5
1459     model = modelaux;
1460     model = glm::translate(model, glm::vec3(-4.20f, 0.0f, 0.0f));
1461     if (dia) {
1462         RMmeeseekD.UseTexture();
1463     }
1464     if (noche) {
1465         RMmeeseekN.UseTexture();
1466     }
1467     if (iluminada) {
1468         RMmeeseekI.UseTexture();
1469     }
1470     //RMmeeseekD.UseTexture();
1471     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1472     glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1473     Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1474     meshList[4]->RenderMesh();
1475
1476     //c6
1477     model = modelaux;
1478     model = glm::translate(model, glm::vec3(-5.25f, 0.0f, 0.0f));
1479     if (dia) {
1480         RMrickD.UseTexture();
1481     }
1482     if (noche) {
1483         RMrickN.UseTexture();
1484     }
1485     if (iluminada) {
1486         RMrickI.UseTexture();
1487     }
1488     //RMrickD.UseTexture();
1489     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1490     glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1491     Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1492     meshList[4]->RenderMesh();
```

```
Proyecto (Ámbito global)
1492 meshList[4]->RenderMesh();
1493
1494 //c7
1495 model = modelaux;
1496 model = glm::translate(model, glm::vec3(-6.3f, 0.0f, 0.0f));
1497 if (dia) {
1498     RMpepinilloD.UseTexture();
1499 }
1500 if (noche) {
1501     RMpepinilloN.UseTexture();
1502 }
1503 if (iluminada) {
1504     RMpepinilloI.UseTexture();
1505 }
1506 //RMpepinilloD.UseTexture();
1507 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1508 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1509 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1510 meshList[4]->RenderMesh();
1511
1512 //c8
1513 model = modelaux;
1514 model = glm::translate(model, glm::vec3(-7.35f, 0.0f, 0.0f));
1515 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1516 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
1517 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1518 if (dia) {
1519     ATprismoD.UseTexture();
1520 }
1521 if (noche) {
1522     ATprismoN.UseTexture();
1523 }
1524 if (iluminada) {
1525     ATprismoI.UseTexture();
1526 }
1527 //ATprismoD.UseTexture();
1528 meshList[4]->RenderMesh();
```

→ Recorrido en el tablero

Para los elementos a incluir en el recorrido se descargaron modelos de los dados y se cargaron a OpenGL, mediante código se realizaron las respectivas rotaciones para que al dado girara y con casos para saber dónde se ubica cada número, mediante números aleatorios es como se genera la cara dónde cae el dado.

```
61
62 //dados
63 float movdado;
64 float rotardadoX;
65 float rotardadoY;
66 float rotardadoZ;
67
68 //dado 8 caras
69 float movdado8;
70 float rotardado8X;
71 float rotardado8Y;
72 float rotardado8Z;
73
```



```

74 Model Paha_M;
75 */
76 Model dado4_M;
77 Model dado4ap_M;
78 Model Stewie_M;
79 Model dado8_M;
80 Model lampRM_M;

```

```

970
971     dado4_M = Model();
972     dado4_M.LoadModel("Models/dado4.obj");
973
974     dado8_M = Model();
975     dado8_M.LoadModel("Models/octa.obj");
976
977     dado4ap_M = Model();
978     dado4ap_M.LoadModel("Models/dado4ap.obj");
979

```

```

1163     }
1164     if (mainWindow.getKeys()[GLFW_KEY_H]) {
1165         dadoResultado = rand() % 3; // Genera un número entre 1 y 4
1166         dadoResultado2 = rand() % 7; // Genera un número entre 1 y 8
1167
1168         rotardadoX = 0.0f; // Reiniciar la rotación
1169         rotardadoY = 0.0f; // Reiniciar la rotación
1170         rotardadoZ = 0.0f; // Reiniciar la rotación
1171         movdado = 100.0f;
1172
1173         rotardado8X = 0.0f; // Reiniciar la rotación
1174         rotardado8Y = 0.0f; // Reiniciar la rotación
1175         rotardado8Z = 0.0f; // Reiniciar la rotación
1176         movdado8 = 100.0f;
1177
1178     }
1179
1180     }
1181     if (movdado > 6.3f) {
1182         movdado -= 0.2f * deltaTime;
1183
1184         // Rotación según el resultado del dado
1185         switch (dadoResultado) {
1186             case 0:
1187                 rotardadoX += 6.15f * deltaTime;
1188                 break;
1189             case 1:
1190                 rotardadoZ += 6.38f * deltaTime;
1191                 break;
1192             case 2:

```

```

1192     case 2:
1193         rotardadoX += 1.0f * deltaTime;
1194         rotardadoY += 2.0f * deltaTime;
1195         break;
1196     case 3:
1197         rotardadoX += 6.14f * deltaTime;
1198         break;
1199     }
1200 }
1201
1202 if (movdado8 > 6.3f) {
1203     movdado8 -= 0.2f * deltaTime;
1204
1205     // Rotación según el resultado del dado
1206     switch (dadoResultado2) {
1207         //cuando sale 1
1208     case 0:
1209         rotardado8X += 4.75f * deltaTime;
1210         rotardado8Y += 8.38f * deltaTime;
1211
1212         break;
1213         //cuando sale 2
1214     case 1:
1215         rotardado8X -= 9.80f * deltaTime;
1216         rotardado8Z -= 8.52f * deltaTime;
1217         rotardado8Y -= 4.13f * deltaTime;
1218
1219         break;
1220
1221         //cuando sale 3
1222     case 2:
1223         rotardado8Y += 10.3f * deltaTime;
1224         rotardado8X += 7.0f * deltaTime;
1225
1226
1227

```

```

1228
1229         break;
1230         //cuando sale 4
1231
1232     case 3:
1233         rotardado8X += 2.15f * deltaTime; //esta mal
1234
1235
1236         break;
1237         //cuando sale 5
1238
1239     case 4:
1240         //// queda pendiente
1241
1242
1243         break;
1244         //cuando sale 6
1245
1246     case 5:
1247         rotardado8Y += 2.2f * deltaTime;
1248         rotardado8X += 2.8f * deltaTime;
1249
1250
1251         break;
1252         //cuando sale 7
1253
1254     case 6:
1255         //// queda pendiente
1256
1257         break;
1258         //cuando sale 8
1259
1260     case 7:
1261         rotardado8X += 2.0f * deltaTime;
1262         rotardado8Y += 7.3f * deltaTime;
1263         rotardado8Z += 7.0f * deltaTime;
1264         printf("entro a 7 \n");

```

```

1259
1260         case 7:
1261             rotardado8X += 2.0f * deltaTime;
1262             rotardado8Y += 7.3f * deltaTime;
1263             rotardado8Z += 7.0f * deltaTime;
1264             printf("entro a 7 \n");
1265
1266             break;
1267
1268         }
1269
1270
1271     }

```

→ Ciudad

Para el caso de la ciudad se modificó el skybox y se dejó un piso en dónde se mantiene el tablero de mesa.

Además de poner algunas texturas en el fondo para ver un poco mas que estamos sobre un plano y no volando.

```
2274 //ediificos alrededor
2275 model = glm::mat4(1.0);
2276 model = glm::translate(model, glm::vec3(225.0f, 45.0f, 140.0f));
2277 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
2278 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2279 model = glm::scale(model, glm::vec3(180.0f, 1.0f, 90.0f));
2280 modelaux = model;
2281 casag.UseTexture();
2282 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2283 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
2284 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
2285 meshList[4]->RenderMesh();
2286
2287 //paisaje
2288 model = modelaux;
2289 model = glm::translate(model, glm::vec3(1.27f, 0.0f, 0.0f));
2290 model = glm::scale(model, glm::vec3(1.55f, 1.0f, 1.0f));
2291 modelaux = model;
2292 pai.UseTexture();
2293 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2294 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
2295 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
2296 meshList[4]->RenderMesh();
2297
2298 //edifi
2299 model = glm::mat4(1.0);
2300 model = glm::translate(model, glm::vec3(0.0f, 45.0f, 225.0f));
2301 model = glm::rotate(model, 90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
2302 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, -1.0f, 0.0f));
2303 model = glm::scale(model, glm::vec3(90.0f, 1.0f, 450.0f));
2304 edif.UseTexture();
2305 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2306 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
2307 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
2308 meshList[4]->RenderMesh();
2309
2310 //lim
2311 model = glm::mat4(1.0);
2312 model = glm::translate(model, glm::vec3(-225.0f, 45.0f, 0.0f));
2313 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 0.0f, -1.0f));
2314 model = glm::rotate(model, 180 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
2315 model = glm::scale(model, glm::vec3(90.0f, 1.0f, 450.0f));
2316 lim.UseTexture();
2317 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2318 glUniform3fv(uniformColor, 1, glm::value_ptr(color));
2319 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
2320 meshList[4]->RenderMesh();
2321
2322 //paisajeRM
2323 model = glm::mat4(1.0);
2324 model = glm::translate(model, glm::vec3(0.0f, 45.0f, -225.0f));
2325 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 0.0f, -1.0f));
2326 model = glm::rotate(model, -90 * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
2327 model = glm::scale(model, glm::vec3(90.0f, 1.0f, 450.0f));
2328 pai2.UseTexture();
```

1 % No se encontraron problemas.

→ Iluminación

Para el caso de la iluminación lo que se hizo fue descargar 3 modelos de lámparas del universo de cada integrante y mediante líneas de código se implementaron luces puntuales en las 4 esquinas del tablero , en cada una de estas esquinas se agregaron los 3 modelos de lámparas.

```
278     Model dado8_M;  
279     Model dado8_M;  
280     Model lampRM_M;  
281     Model lampHA_M;  
282     Model lampFG_M;  
283
```

```
0     Pana_M.LoadModel("Models/Pana.obj");  
1  
2     lampRM_M = Model();  
3     lampRM_M.LoadModel("Models/lampRM.obj");  
4  
5     lampHA_M = Model();  
6     lampHA_M.LoadModel("Models/lampHA.obj");  
7  
8     lampFG_M = Model();  
9     lampFG_M.LoadModel("Models/lampFG.obj");  
0  
1     */  
2  
3     //personajes en movimiento  
4     entity_M = Model();
```

```

93     material_opaco = Material(0.5f, 4);
94
95
96     //luz direccional, sólo 1 y siempre debe de existir
97     mainLight = DirectionalLight(1.0f, 1.0f, 1.0f,
98         0.3f, 0.3f,
99         0.0f, 0.0f, -1.0f);
100    //contador de luces puntuales
101    unsigned int pointLightCount = 0;
102
103    //Declaración de primer luz puntual
104    pointLights[0] = PointLight(0.5f, 0.5f, 0.5f,
105        0.0f, 1.0f,
106        145.0f, 8.5f, 145.0f,
107        0.05f, 0.005f, 0.001f);
108    pointLightCount++;
109
110    pointLights[1] = PointLight(0.5f, 0.5f, 0.5f,
111        0.0f, 1.0f,
112        -145.0f, 8.5f, -145.0f,
113        0.05f, 0.005f, 0.001f);
114    pointLightCount++;
115
116    pointLights[2] = PointLight(0.5f, 0.5f, 0.5f,
117        0.0f, 1.0f,
118        145.0f, 8.5f, -145.0f,
119        0.05f, 0.005f, 0.001f);
120    pointLightCount++;
121
122    unsigned int spotLightCount = 2;
123
124    //linterna
125    spotLights[0] = SpotLight(1.0f, 1.0f, 1.0f,
126        1.0f, 2.0f,
127        0.0f, 0.0f, 0.0f,

```

```

Proyecto (Ambito global)
1027        0.0f, 0.0f, 0.0f,
1028        0.0f, -1.0f, 0.0f,
1029        0.8f, 0.0f, 0.0f,
1030        30.0f);
1031    spotLightCount++;
1032
1033    //luz fija
1034    spotLights[1] = SpotLight(0.0f, 1.0f, 0.0f,
1035        1.0f, 2.0f,
1036        5.0f, 10.0f, 0.0f,
1037        0.0f, -5.0f, 0.0f,
1038        1.0f, 0.0f, 0.0f,
1039        15.0f);
1040    spotLightCount++;
1041

```

```

2539
2540 //lampara RM
2541 model = glm::mat4(1.0);
2542 model = glm::translate(model, glm::vec3(-145.0f, 2.3f, -145.0f));
2543 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2544 model = glm::scale(model, glm::vec3(50.0f, 50.0f, 50.0f));
2545 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2546 lampRM_M.RenderModel();
2547
2548 //lampara HA
2549 model = glm::mat4(1.0);
2550 model = glm::translate(model, glm::vec3(145.0f, 2.3f, 145.0f));
2551 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2552 model = glm::scale(model, glm::vec3(17.0f, 17.0f, 17.0f));
2553 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2554 lampHA_M.RenderModel();
2555
2556 //lampara FG
2557 model = glm::mat4(1.0);
2558 model = glm::translate(model, glm::vec3(145.0f, 2.3f, -145.0f));
2559 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2560 model = glm::scale(model, glm::vec3(30.0f, 30.0f, 30.0f));
2561 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2562 lampFG_M.RenderModel();
2563

```

→ Animación

Para el caso de la animación mediante código logramos que el los avatares realizarán el recorrido alrededor del tablero , así como también el movimiento de los dados.

Mediante if's se creó el movimiento del personaje en el tablero dándole los valores máximos de nuestro tablero, así como para el caso de los dados de 4 y 8 caras se crearon casos para delimitar dónde se encontraban cada una de las caras de ambos dados y mediante números aleatorios es como se obtiene el valor de los dados.

```

1111     }
1112     //actualizarSkybox(día, noche);
1113
1114     //movimiento de personajes
1115     if (stewie) {
1116         if (caminaZ) {
1117             if (movPerZ > -250) movPerZ -= movOffset * deltaTime;
1118             else if (rotPer < 90 && rotPer >= 0) rotPer += rotOffset * deltaTime;
1119             else caminaX = true;
1120         }
1121
1122         else {
1123             if (movPerZ < 0) movPerZ += movOffset * deltaTime;
1124             else
1125                 if (rotPer < 270 && rotPer >= 180) rotPer += rotOffset * deltaTime;
1126                 else caminaX = false;
1127         }
1128
1129         if (caminaX) {
1130             if (movPerX < 250) movPerX += movOffset * deltaTime;
1131             else if (rotPer < 180 && rotPer >= 90) rotPer += rotOffset * deltaTime;
1132             else caminaZ = false;
1133         }
1134         else {
1135             if (movPerX > 0) movPerX -= movOffset * deltaTime;
1136             else if (rotPer < 360 && rotPer >= 270) rotPer += rotOffset * deltaTime;
1137             else caminaZ = true;
1138         }
1139         if (rotPer > 360) rotPer = 0.0f;
1140         //aquí se debe agregar que cada que el personaje este en una casilla a esta se le asigna :
1141         // luminada=true; o cuando se detenga (para cada personaje), aunque esto no es correcto de
1142         /*Cada que termine este personaje de caminar se asigna :
1143         stewie=false;

```

```

        */
    }
    if (mainWindow.getKeys()[GLFW_KEY_H]) {
        dadoResultado = rand() % 3; // Genera un número entre 1 y 4
        dadoResultado2 = rand() % 7; // Genera un número entre 1 y 8

        rotardadoX = 0.0f; // Reiniciar la rotación
        rotardadoY = 0.0f; // Reiniciar la rotación
        rotardadoZ = 0.0f; // Reiniciar la rotación
        movdado = 100.0f;

        rotardado8X = 0.0f; // Reiniciar la rotación
        rotardado8Y = 0.0f; // Reiniciar la rotación
        rotardado8Z = 0.0f; // Reiniciar la rotación
        movdado8 = 100.0f;

    }
    if (movdado > 6.3f) {
        movdado -= 0.2f * deltaTime;

        // Rotación según el resultado del dado
        switch(dadoResultado) {
        case 0:
            rotardadoX += 6.15f * deltaTime;
            break;
        case 1:
            rotardadoZ += 6.38f * deltaTime;
            break;

```



```

188         break;
189     case 1:
190         rotardadoZ += 6.38f * deltaTime;
191         break;
192     case 2:
193         rotardadoX += 1.0f * deltaTime;
194         rotardadoY += 2.0f * deltaTime;
195         break;
196     case 3:
197         rotardadoX += 6.14f * deltaTime;
198         break;
199     }
200 }
201
202 if (movdado8 > 6.3f) {
203     movdado8 -= 0.2f * deltaTime;
204
205     // Rotación según el resultado del dado
206     switch (dadoResultado2) {
207         //cuando sale 1
208     case 0:
209         rotardado8X += 4.75f * deltaTime;
210         rotardado8Y += 8.38f * deltaTime;
211
212         break;
213         //cuando sale 2
214     case 1:
215         rotardado8X -= 9.80f * deltaTime;
216         rotardado8Z -= 8.52f * deltaTime;
217         rotardado8Y -= 4.13f * deltaTime;
218
219         break;
220
221         //cuando sale 3
222     case 2:

```

```
oyecto (Ambito global)
224 case 2:
225     rotardado8Y += 10.3f * deltaTime;
226     rotardado8X += 7.0f * deltaTime;
227     |
228
229     break;
230     //cuando sale 4
231
232 case 3:
233     rotardado8X += 2.15f * deltaTime; //esta mal
234
235
236     break;
237     //cuando sale 5
238
239 case 4:
240     //// queda pendiente
241
242
243     break;
244     //cuando sale 6
245
246 case 5:
247     rotardado8Y += 2.2f * deltaTime;
248     rotardado8X += 2.8f * deltaTime;
249
250
251     break;
252     //cuando sale 7
253
254 case 6:
255     //// queda pendiente
256
257     break;
258     //cuando sale 8
259
260 case 7:
```

```
258     //cuando sale 8
259
260 case 7:
261     rotardado8X += 2.0f * deltaTime;
262     rotardado8Y += 7.3f * deltaTime;
263     rotardado8Z += 7.0f * deltaTime;
264     printf("entro a 7 \n");
265
266     break;
267
268 }
269
270
271 }
272
```

```

2497 //jupiter_RenderModel();
2498
2499 //dado4
2500 if (dadoResultado != 3){
2501     model = glm::mat4(1.0f);
2502     model = glm::translate(model, glm::vec3(0.0f, movdado, 0.0f));
2503     model = glm::rotate(model, rotardadoX * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
2504     model = glm::rotate(model, rotardadoY * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2505     model = glm::rotate(model, rotardadoZ * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
2506
2507     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2508     dado4_M.RenderModel();
2509 }
2510 else {
2511     model = glm::mat4(1.0f);
2512     model = glm::translate(model, glm::vec3(0.0f, movdado, 0.0f));
2513     model = glm::rotate(model, rotardadoX * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
2514     model = glm::rotate(model, rotardadoY * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
2515     model = glm::rotate(model, rotardadoZ * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
2516
2517     glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
2518     dado4ap_M.RenderModel();
2519 }
2520

```

- **Conclusiones**

- **González Flores Francisco Eduardo:**

El desarrollo del proyecto nos permitió familiarizarnos con distintas técnicas de animación y renderizado en tiempo real, así como con el uso de texturas y efectos de iluminación. A pesar de no completar todos los objetivos, el aprendizaje obtenido nos equipará mejor para futuros proyectos similares, en los que podremos implementar de manera más eficiente estas herramientas.

- **López Betancourt Michelle:**

Después de haber realizado el proyecto puedo concluir que aprendimos a usar cada una de los aspectos solicitados, aunque en nuestro caso no logramos terminar todo lo que se pedía, logramos comprender e implementar el recorrido del tablero, el uso de texturas para día, noche y casillas iluminadas, el movimiento de los modelos.

- **Rivera Espinoza José Luis:**

A lo largo del proyecto, enfrentamos diversos obstáculos técnicos que requirieron analizar, investigar y aplicar soluciones creativas. Aunque no completamos todos los objetivos, el proceso nos ayudó a desarrollar habilidades para identificar y abordar problemas de manera autónoma, mejorando nuestra capacidad para adaptarnos a los desafíos y aprender de ellos. Esta experiencia fortalecerá nuestra habilidad para resolver problemas complejos en futuros proyectos.

- **Conclusión grupal**

A lo largo del proyecto, logramos integrar diversos elementos, fortaleciendo nuestras habilidades en texturizado, animación, iluminación y manipulación de modelos en OpenGL. Cada integrante contribuyó con ideas y conocimientos específicos, lo cual permitió desarrollar un tablero interactivo con alternancia de texturas para día, noche e iluminación, recorrido de avatares y dados, y su respectiva iluminación basado en modelos de lámparas personalizados. Aunque enfrentamos desafíos en la implementación, la experiencia de trabajo en equipo y la resolución de problemas nos permitió comprender y aplicar conceptos avanzados de programación gráfica.