



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e
INTERACCIÓN HUMANO COMPUTADORA



REPORTE DE PRÁCTICA N° 09

NOMBRE COMPLETO: Medrano Miranda Daniel Ulises

N° de Cuenta: 318045351

GRUPO DE LABORATORIO: 03

GRUPO DE TEORÍA: 04

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 19/Octubre/2024

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

1. Su dado de 10 caras cae sobre el piso, gira y cae en un número "random", se repite la tirada al presionar una tecla.
2. Por integrante del equipo elegirán un tipo de vehículo: terrestre o aéreo. Cada integrante del equipo creará un recorrido en el cual el vehículo se moverá alrededor de su tablero de Monopoly. Cada vehículo iniciará su recorrido a partir de una esquina diferente. (el vehículo terrestre no puede ser un carro o vehículo similar motorizado de 4 ruedas, se debe de tener movimiento de llantas o de hélices en sus vehículos.)

Código:

```
52 float movLetrero, giraLetrero, movCamion, rotllantaCamion, giraCamion;
53 float movCamion2, movCamion3, movCamion4;
54 float movLetreroOffset, giraLetreroOffset, movCamionOffset, rotllantaCamionOffset, giraCamionOffset;
55 bool baja, gira, avanzaCamion1, avanzaCamion2, avanzaCamion3, avanzaCamion4, vuelta;
56
57 float caeDado, giraDado;
58 float caeDadoOffset, giraDadoOffset;
59 bool caer, girar;
60
61
62
63
64
65
66
67 Texture brickTexture;
68 Texture dirtTexture;
69 Texture plainTexture;
70 Texture pisoTexture;
71 Texture AgaveTexture;
72
73 Texture Blanco;
74 Texture Casilla1;
75 Texture Casilla2;
76 Texture Casilla3;
77 Texture dadoDiezTexture;
78
79 Model Kitt_M;
80 Model Llanta_M;
81 Model Blackhawk_M;
82 Model Arco_M;
83 Model Letrero_M;
84 Model Camion_M;
85 Model LlantaCamion_M;
```

```

215 void Casillas() {
216     unsigned int Casillas1_indices[] = {
217         0,1,2, 0,2,3,
218         4,5,6, 4,6,7,
219
220         0,1,5, 0,4,5,
221         1,2,6, 1,5,6,
222         0,3,4, 3,4,7,
223         2,3,6, 3,6,7,
224     };
225     GLfloat Casillas1_vertices[] = {
226         1.0f, 0.2f, 1.0f, 1.0f,1.0f, 0.0f,-1.0f,0.0f,//A - 0
227         1.0f, 0.2f,-1.0f, 1.0f,0.0f, 0.0f,-1.0f,0.0f,//B - 1
228         -1.0f, 0.2f,-1.0f, 0.0f,0.0f, 0.0f,-1.0f,0.0f,//C - 2
229         -1.0f, 0.2f, 1.0f, 0.0f,1.0f, 0.0f,-1.0f,0.0f,//D - 3
230
231         1.0f, 0.0f, 1.0f, 1.0f,1.0f, 0.0f,-1.0f,0.0f,//E - 4
232         1.0f, 0.0f,-1.0f, 1.0f,0.0f, 0.0f,-1.0f,0.0f,//F - 5
233         -1.0f, 0.0f,-1.0f, 0.0f,0.0f, 0.0f,-1.0f,0.0f,//G - 6
234         -1.0f, 0.0f, 1.0f, 0.0f,1.0f, 0.0f,-1.0f,0.0f,//H - 7
235     };
236
237     Mesh* Casillas1 = new Mesh();
238     Casillas1->CreateMesh(Casillas1_vertices, Casillas1_indices, 64, 36);
239     meshList.push_back(Casillas1);
240
241     unsigned int Casillas2_indices[] = {
242         0,1,2, 0,2,3,
243         4,5,6, 4,6,7,
244
245         0,1,5, 0,4,5,
246         1,2,6, 1,5,6,
247         0,3,4, 3,4,7,
248         2,3,6, 3,6,7,
249     };
250     GLfloat Casillas2_vertices[] = {
251         1.0f, 0.2f, 0.5f, 1.0f,1.0f, 0.0f,-1.0f,0.0f,//A - 0
252         1.0f, 0.2f,-0.5f, 1.0f,0.0f, 0.0f,-1.0f,0.0f,//B - 1
253         -1.0f, 0.2f,-0.5f, 0.0f,0.0f, 0.0f,-1.0f,0.0f,//C - 2
254         -1.0f, 0.2f, 0.5f, 0.0f,1.0f, 0.0f,-1.0f,0.0f,//D - 3
255
256         1.0f, 0.0f, 0.5f, 1.0f,1.0f, 0.0f,-1.0f,0.0f,//E - 4
257         1.0f, 0.0f,-0.5f, 1.0f,0.0f, 0.0f,-1.0f,0.0f,//F - 5
258         -1.0f, 0.0f,-0.5f, 0.0f,0.0f, 0.0f,-1.0f,0.0f,//G - 6
259         -1.0f, 0.0f, 0.5f, 0.0f,1.0f, 0.0f,-1.0f,0.0f,//H - 7
260     };
261     Mesh* Casillas2 = new Mesh();
262     Casillas2->CreateMesh(Casillas2_vertices, Casillas2_indices, 144, 90);
263     meshList.push_back(Casillas2);
264 }
265
266 void CreateShaders()
267 {
268     Shader *shader1 = new Shader();
269     shader1->CreateFromFiles(vShader, fShader);
270     shaderList.push_back(*shader1);
271 }
272
273 void CrearDado10Lados()
274 {
275     unsigned int cubo10_indices[] = {
276         // Cara 1 - Numero 4
277         0, 1, 2,
278
279         // Cara 2 - Numero 6
280         3, 4, 5,

```

```

281
282 // Cara 3 - Numero 2
283 6, 7, 8,
284
285 // Cara 4 - Numero 8
286 9, 10, 11,
287
288 // Cara 5 - Numero 10
289 12, 13, 14,
290
291 // Cara 6 - Numero 5
292 15, 16, 17,
293
294 // Cara 7 - Numero 7
295 18, 19, 20,
296
297 // Cara 8 - Numero 1
298 21, 22, 23,
299
300 // Cara 9 - Numero 3
301 24, 25, 26,
302
303 // Cara 10 - Numero 9
304 27, 28, 29
305 };
306
307 GLfloat cubo10_vertices[] = {
308 // Cara 1 - Numero 4
309 // X      Y      Z      S      T      NX      NY      NZ
310 -4.55f,   0.0f,   4.81f,  0.01f,  0.51f,  1.0f,  -1.0f,  -1.0f,  //0
311  1.63f,   0.0f,   4.66f,  0.19f,  0.51f,  1.0f,  -1.0f,  -1.0f,  //1
312 -1.56f,   5.0f,   0.48f,  0.10f,  0.74f,  1.0f,  -1.0f,  -1.0f,  //2
313
314 // Cara 2 - Numero 6
315 // X      Y      Z      S      T      NX      NY      NZ
316  1.63f,   0.0f,   4.66f,  0.21f,  0.51f,  -1.0f,  -1.0f,  -1.0f,  //3
317  3.41f,   0.0f,  -1.26f,  0.39f,  0.51f,  -1.0f,  -1.0f,  -1.0f,  //4
318 -1.56f,   5.0f,   0.48f,  0.30f,  0.74f,  -1.0f,  -1.0f,  -1.0f,  //5
319
320 // Cara 3 - Numero 2
321 // X      Y      Z      S      T      NX      NY      NZ
322  3.41f,   0.0f,  -1.26f,  0.41f,  0.51f,  -1.0f,  -1.0f,   1.0f,  //6
323 -1.68f,   0.0f,  -4.78f,  0.59f,  0.51f,  -1.0f,  -1.0f,   1.0f,  //7
324 -1.56f,   5.0f,   0.48f,  0.50f,  0.74f,  -1.0f,  -1.0f,   1.0f,  //8
325
326 // Cara 4 - Numero 8
327 // X      Y      Z      S      T      NX      NY      NZ
328 -1.68f,   0.0f,  -4.78f,  0.61f,  0.51f,  1.0f,  -1.0f,   1.0f,  //9
329 -6.60f,   0.0f,  -1.03f,  0.79f,  0.51f,  1.0f,  -1.0f,   1.0f,  //10
330 -1.56f,   5.0f,   0.48f,  0.70f,  0.74f,  1.0f,  -1.0f,   1.0f,  //11
331
332 // Cara 5 - Numero 10
333 // X      Y      Z      S      T      NX      NY      NZ
334 -6.60f,   0.0f,  -1.03f,  0.81f,  0.51f,  1.0f,  -1.0f,  -1.0f,  //12
335 -4.55f,   0.0f,   4.81f,  0.99f,  0.51f,  1.0f,  -1.0f,  -1.0f,  //13
336 -1.56f,   5.0f,   0.48f,  0.90f,  0.74f,  1.0f,  -1.0f,  -1.0f,  //14
337
338 // Cara 6 - Numero 5
339 // X      Y      Z      S      T      NX      NY      NZ
340 -4.55f,   0.0f,   4.81f,  0.01f,  0.49f,  1.0f,   1.0f,  -1.0f,  //15
341  1.63f,   0.0f,   4.66f,  0.19f,  0.49f,  1.0f,   1.0f,  -1.0f,  //16
342 -1.56f,  -5.0f,   0.48f,  0.10f,  0.26f,  1.0f,   1.0f,  -1.0f,  //17
343
344 // Cara 7 - Numero 7
345 // X      Y      Z      S      T      NX      NY      NZ
346  1.63f,   0.0f,   4.66f,  0.21f,  0.49f,  -1.0f,   1.0f,  -1.0f,  //18

```

```

347         3.41f,    0.0f,   -1.26f,    0.39f,  0.49f,   -1.0f,  1.0f,   -1.0f, //19
348         -1.56f,   -5.0f,    0.48f,    0.30f,  0.26f,   -1.0f,  1.0f,   -1.0f, //20
349
350         // Cara 8 - Numero 1
351         // X      Y      Z      S      T      NX      NY      NZ
352         3.41f,    0.0f,   -1.26f,    0.41f,  0.49f,   -1.0f,  1.0f,   1.0f, //21
353         -1.68f,    0.0f,   -4.78f,    0.59f,  0.49f,   -1.0f,  1.0f,   1.0f, //22
354         -1.56f,   -5.0f,    0.48f,    0.50f,  0.26f,   -1.0f,  1.0f,   1.0f, //23
355
356         // Cara 9 - Numero 3
357         // X      Y      Z      S      T      NX      NY      NZ
358         -1.68f,    0.0f,   -4.78f,    0.61f,  0.49f,    1.0f,  1.0f,   1.0f, //24
359         -6.60f,    0.0f,   -1.03f,    0.79f,  0.49f,    1.0f,  1.0f,   1.0f, //25
360         -1.56f,   -5.0f,    0.48f,    0.70f,  0.26f,    1.0f,  1.0f,   1.0f, //26
361
362         // Cara 10 - Numero 9
363         // X      Y      Z      S      T      NX      NY      NZ
364         -6.60f,    0.0f,   -1.03f,    0.81f,  0.49f,    1.0f,  1.0f,   -1.0f, //27
365         -4.55f,    0.0f,    4.81f,    0.99f,  0.49f,    1.0f,  1.0f,   -1.0f, //28
366         -1.56f,   -5.0f,    0.48f,    0.90f,  0.26f,    1.0f,  1.0f,   -1.0f, //29
367
368     };
369
370     Mesh* dado10lados = new Mesh();
371     dado10lados->CreateMesh(cubo10_vertices, cubo10_indices, 300, 30);
372     meshList.push_back(dado10lados);
373 }

```

```

380     CreateObjects();
381     Casillas();
382     CrearDado10Lados();
383     CreateShaders();

```

```

398     dadoDiezTexture = Texture("Textures/dado10lados.tga");
399     dadoDiezTexture.LoadTextureA();
400
401     Blanco = Texture("Textures/Blanco.tga");
402     Blanco.LoadTextureA();
403     Casillal = Texture("Textures/Casillal.tga");
404     Casillal.LoadTextureA();
405     Casilla2 = Texture("Textures/Casilla2.tga");
406     Casilla2.LoadTextureA();
407     Casilla3 = Texture("Textures/Casilla3.tga");
408     Casilla3.LoadTextureA();
409
410
411
412     Kitt_M = Model();
413     Kitt_M.LoadModel("Models/kitt_optimizado.obj");
414     Llanta_M = Model();
415     Llanta_M.LoadModel("Models/llanta_optimizada.obj");
416     Blackhawk_M = Model();
417     Blackhawk_M.LoadModel("Models/uh60.obj");
418     Arco_M = Model();
419     Arco_M.LoadModel("Models/Arco_obj.obj");
420     Letrero_M = Model();
421     Letrero_M.LoadModel("Models/Letrero_obj.obj");
422     Camion_M = Model();
423     Camion_M.LoadModel("Models/Camion_obj.obj");
424     LlantaCamion_M = Model();
425     LlantaCamion_M.LoadModel("Models/LlantaCamion_obj.obj");

```

```

492     movCamion = 0.0f;
493     movCamionOffset = 0.05f;
494     rotllantaCamion = 0.0f;
495     rotllantaCamionOffset = 1.5f;
496     giraCamion = 0.0;
497     giraCamionOffset = 0.37f;
498     avanzaCamion1 = true;
499     vuelta = false;
500
501     caeDado = 0.0f;
502     caeDadoOffset = 0.05f;
503     giraDado = 0.0f;
504     giraDadoOffset = 0.5f;
505     caer = true;
506     girar = true;
507
508     glfwSetTime(0);

```

```

586     glm::mat4 model(1.0);
587     glm::mat4 modelaux(1.0);
588     glm::mat4 modelauxLetrero(1.0);
589     glm::mat4 modelauxCamion(1.0);
590     glm::vec3 color = glm::vec3(1.0f, 1.0f, 1.0f);

```

```

1066 // ----- CAMIÓN 6 RUEDAS ----- //
1067
1068 float contador = glfwGetTime(), lugarCamion = 0.0f;
1069
1070 printf("%f\n", glfwGetTime());
1071 rotllantaCamion += rotllantaCamionOffset * deltaTime;
1072 if (avanzaCamion1)
1073 {
1074     if (glfwGetTime() < 15.0) {
1075         movCamion += movCamionOffset * deltaTime;
1076     }
1077     else {
1078         avanzaCamion1 = !avanzaCamion1;
1079         glfwSetTime(0);
1080     }
1081 }
1082 else {
1083
1084     if (glfwGetTime() < 2) {
1085         giraCamion += giraCamionOffset * deltaTime ;
1086     }
1087     else {
1088         avanzaCamion1 = !avanzaCamion1;
1089         glfwSetTime(0);
1090     }
1091 }
1092
1093 //Carroceria
1094 model = glm::mat4(1.0);
1095 model = glm::rotate(model, 90 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
1096
1097 model = glm::rotate(model, giraCamion * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
1098
1099 model = glm::translate(model, glm::vec3(15.0f, -1.0f, 5.0 - movCamion));
1100
1101 model = glm::scale(model, glm::vec3(0.15f, 0.15f, 0.15f));
1102
1103 modelauxCamion = model;
1104 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1105 Camion_M.RenderModel();
1106
1107 // Llantas //
1108 //Izquierdas
1109 model = modelauxCamion;
1110 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, -23.0));
1111 model = glm::rotate(model, rotllantaCamion * toRadians, glm::vec3(-1.0f, 0.0f, 0.0f));
1112 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1113 LlantaCamion_M.RenderModel();
1114
1115 model = modelauxCamion;
1116 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, 8.45));
1117 model = glm::rotate(model, rotllantaCamion * toRadians, glm::vec3(-1.0f, 0.0f, 0.0f));
1118 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1119 LlantaCamion_M.RenderModel();
1120
1121 model = modelauxCamion;
1122 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, 19.6));
1123 model = glm::rotate(model, rotllantaCamion * toRadians, glm::vec3(-1.0f, 0.0f, 0.0f));
1124 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1125 LlantaCamion_M.RenderModel();
1126
1127 //Derechas
1128 model = modelauxCamion;
1129 model = glm::rotate(model, 180 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
1130 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, 23.0));
1131 model = glm::rotate(model, rotllantaCamion * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));

```

```

1132 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1133 LlantaCamion_M.RenderModel();
1134
1135 model = modelauxCamion;
1136 model = glm::rotate(model, 180 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
1137 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, -8.45));
1138 model = glm::rotate(model, rotLlantaCamion * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
1139 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1140 LlantaCamion_M.RenderModel();
1141
1142 model = modelauxCamion;
1143 model = glm::rotate(model, 180 * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
1144 model = glm::translate(model, glm::vec3(-9.0f, 5.0f, -19.6));
1145 model = glm::rotate(model, rotLlantaCamion * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
1146 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1147 LlantaCamion_M.RenderModel();
1148
1149 // ----- DADO DE 10 LADOS ----- //
1150 if (glfwGetTime() > 4 and glfwGetTime() < 5.5) {
1151     if (caer) {
1152         caeDado += caeDadoOffset * deltaTime;
1153     }
1154     else {
1155         caer = !caer;
1156     }
1157 }
1158 else if (glfwGetTime() > 5.5 and glfwGetTime() < 10) {
1159     if (girar) {
1160         giraDado += giraDadoOffset * deltaTime;
1161     }
1162     else {
1163         girar = !girar;
1164     }
1165 }
1166 model = glm::mat4(1.0);
1167 model = glm::translate(model, glm::vec3(-15.0f, 5.0f - caeDado, 15.0));
1168 model = glm::scale(model, glm::vec3(0.3f, 0.3f, 0.3f));
1169 model = glm::rotate(model, giraDado * toRadians, glm::vec3(1.0f, 1.0f, 1.0f));
1170 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1171 dadoDizTexture.UseTexture();
1172 meshList[6]->RenderMesh();
1173
1174 glUseProgram(0);
1175 mainWindow.swapBuffers();
1176 }
1177
1178 return 0;
1179 }
1180

```

Link al video demostrativo:

https://mega.nz/file/VtU1IAQD#mtU7P8ios9fnQtq6vKfWeYcjGBeP5fyCR1FNHx_9F88

2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla

En esta ocasión los problemas presentados fueron en el giro del camión, ya que si posicionaba la rotación antes de la traslación al llegar a la esquina giraba, pero comenzaba a moverse de lado”, por otro lado, si colocaba la rotación después de la traslación avanza siempre hacia enfrente, pero gira de manera “extraña”. Además de que siempre se regresa a la posición inicial.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.

Puedo decir que esta práctica fue un poco más complicada porque requiere usar el tiempo de ejecución como una entrada a las acciones de los modelos que estamos utilizando, por lo que hay que estar reiniciando el tiempo de ejecución, pero si hay algo que también lo está usando se arruina.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica

Quizá explicar un poco más que hacer si 2 objetos van a usar un tiempo de ejecución diferente y requieren reiniciarse cada uno por separado.

c. Conclusión

En esta ocasión no se cumplieron al 100% los objetivos de la práctica ya que mi código funciona pero no realiza del todo las acciones necesarias.