



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° 02

NOMBRE COMPLETO: David Sánchez Gutiérrez

N° de Cuenta: 315596397

GRUPO DE LABORATORIO: 01

GRUPO DE TEORÍA: 04

SEMESTRE 2024-2

FECHA DE ENTREGA LÍMITE: 24 de febrero de 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.

- a. Dibujar las iniciales de sus nombres, cada letra de un color diferente.

- i. Código:

Se agregaron los mismos vértices de la práctica pasada, agregando el color para cada vértice según la letra.

```
96 void CrearLetrasyFiguras()
97 {
98     //[0]
99     GLfloat vertices_letras[] = {
100         //X      Y      Z      R      G      B
101         //0
102         -0.9f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
103         -0.9f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
104         -0.8f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
105
106         -0.9f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
107         -0.8f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
108         -0.8f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
109
110         -0.8f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
111         -0.8f,  0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
112         -0.6f,  0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
113
114         -0.8f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
115         -0.6f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
116         -0.6f,  0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
117
118         -0.8f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
119         -0.8f, -0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
120         -0.6f, -0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
121
122         -0.8f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
123         -0.6f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
124         -0.6f, -0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
125
126         -0.5f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
127         -0.4f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
128         -0.4f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
129
130         -0.5f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
131         -0.4f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
132         -0.5f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
133
134         -0.5f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
135         -0.6f,  0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
136         -0.6f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
137
138         -0.6f,  0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
139         -0.5f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
140         -0.4f,  0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
141
142         -0.5f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
143         -0.6f, -0.3f,  0.0f,  1.0f,  0.0f,  0.0f,
144         -0.6f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
145
146         -0.6f, -0.4f,  0.0f,  1.0f,  0.0f,  0.0f,
147         -0.5f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
148         -0.4f, -0.2f,  0.0f,  1.0f,  0.0f,  0.0f,
149
150         //S 36
151
152         -0.1f, -0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
153         -0.1f, -0.25f, 0.0f,  0.0f,  1.0f,  0.0f,
154         0.05f, -0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
155
156         0.05f, -0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
157         0.2f, -0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
158         0.2f, -0.25f, 0.0f,  0.0f,  1.0f,  0.0f,
159
160         -0.1f,  0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
161         -0.1f,  0.25f, 0.0f,  0.0f,  1.0f,  0.0f,
162         0.05f,  0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
163
164         0.05f,  0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
165         0.2f,  0.4f,  0.0f,  0.0f,  1.0f,  0.0f,
166         0.2f,  0.25f, 0.0f,  0.0f,  1.0f,  0.0f,
```

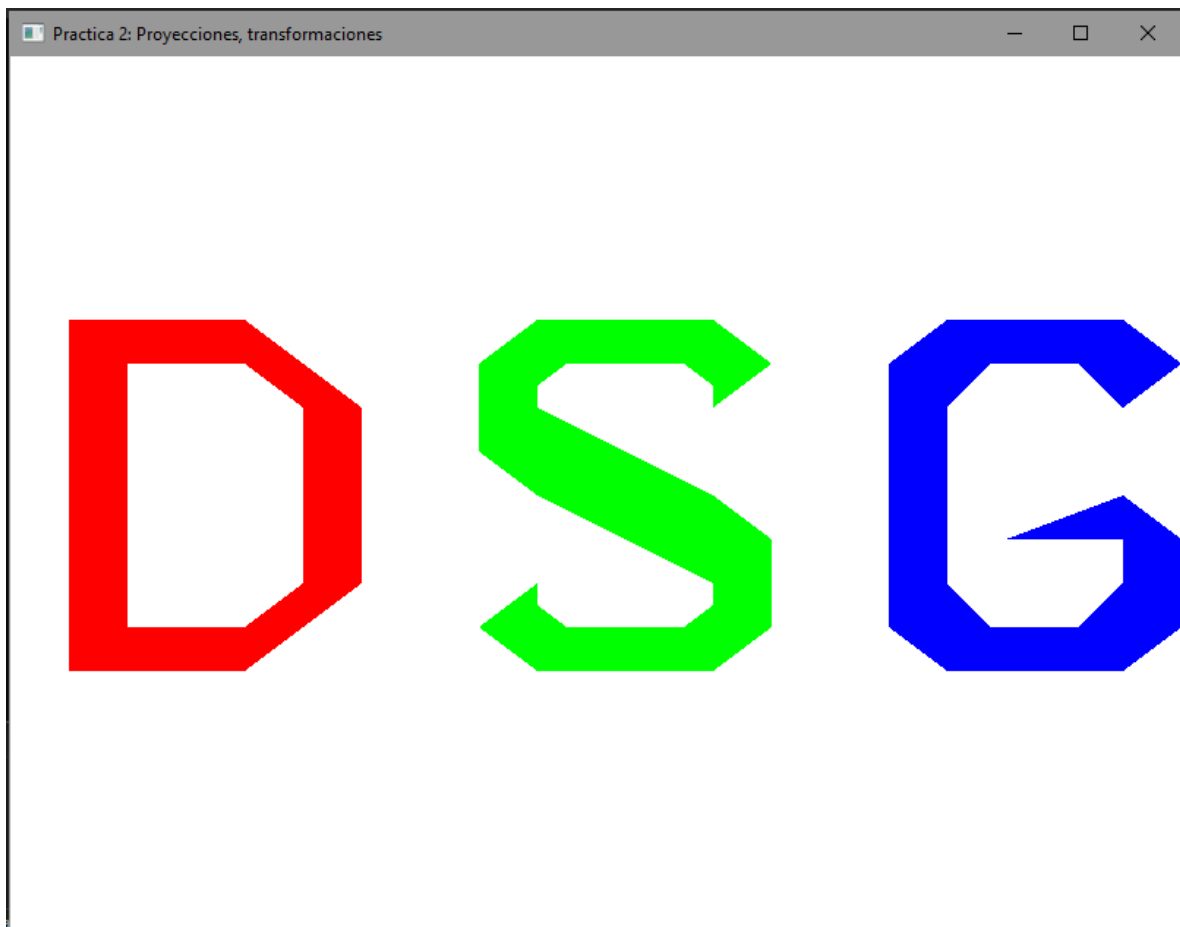
168	0.05f,	-0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
169	0.15f,	-0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
170	-0.05f,	-0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
171						
172	0.05f,	0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
173	0.15f,	0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
174	-0.05f,	0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
175						
176	-0.1f,	0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
177	-0.2f,	0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
178	-0.2f,	0.1f,	0.0f,	0.0f,	1.0f,	0.0f,
179						
180	-0.1f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
181	-0.2f,	0.1f,	0.0f,	0.0f,	1.0f,	0.0f,
182	-0.1f,	0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
183						
184	0.2f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
185	0.3f,	-0.1f,	0.0f,	0.0f,	1.0f,	0.0f,
186	0.3f,	-0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
187						
188	0.2f,	-0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
189	0.2f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
190	0.3f,	-0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
191						
192	0.2f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
193	0.2f,	-0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
194	-0.1f,	0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
195						
196	-0.1f,	0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
197	-0.1f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
198	0.2f,	-0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
199						
200	0.2f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
201	0.2f,	-0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
202	-0.1f,	0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
203						
204	-0.1f,	0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
205	-0.1f,	0.0f,	0.0f,	0.0f,	1.0f,	0.0f,
206	0.2f,	-0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
207						
208	-0.1f,	-0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
209	-0.1f,	-0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
210	-0.2f,	-0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
211						
212	0.2f,	0.2f,	0.0f,	0.0f,	1.0f,	0.0f,
213	0.2f,	0.4f,	0.0f,	0.0f,	1.0f,	0.0f,
214	0.3f,	0.3f,	0.0f,	0.0f,	1.0f,	0.0f,
215						
216	//G 48					
217						
218	0.5f,	0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
219	0.5f,	-0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
220	0.6f,	-0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
221						
222	0.6f,	-0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
223	0.5f,	0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
224	0.6f,	0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
225						
226	0.6f,	0.2f,	0.0f,	0.0f,	0.0f,	1.0f,
227	0.6f,	0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
228	0.75f,	0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
229						
230	0.6f,	-0.2f,	0.0f,	0.0f,	0.0f,	1.0f,
231	0.6f,	-0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
232	0.75f,	-0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
233						
234	0.75f,	0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
235	0.6f,	0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
236	0.9f,	0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
237						
238	0.75f,	-0.4f,	0.0f,	0.0f,	0.0f,	1.0f,
239	0.6f,	-0.3f,	0.0f,	0.0f,	0.0f,	1.0f,
240	0.9f,	-0.3f,	0.0f,	0.0f,	0.0f,	1.0f,

```

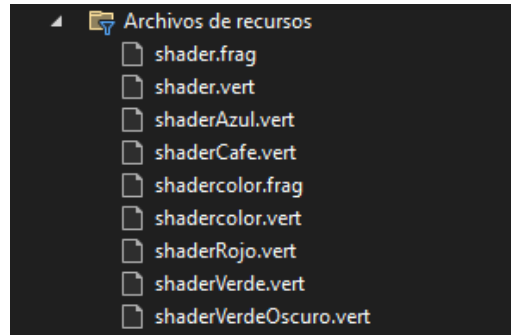
241
242      0.9f, 0.2f, 0.0f, 0.0f, 0.0f, 1.0f,
243      0.9f, 0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
244      0.75f, 0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
245
246      0.9f, -0.2f, 0.0f, 0.0f, 0.0f, 1.0f,
247      0.9f, -0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
248      0.75f, -0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
249
250      0.9f, 0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
251      0.9f, 0.2f, 0.0f, 0.0f, 0.0f, 1.0f,
252      1.0f, 0.3f, 0.0f, 0.0f, 0.0f, 1.0f,
253
254      0.9f, -0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
255      0.9f, 0.0f, 0.0f, 0.0f, 0.0f, 1.0f,
256      1.0f, -0.1f, 0.0f, 0.0f, 0.0f, 1.0f,
257
258      0.9f, -0.4f, 0.0f, 0.0f, 0.0f, 1.0f,
259      1.0f, -0.1f, 0.0f, 0.0f, 0.0f, 1.0f,
260      1.0f, -0.3f, 0.0f, 0.0f, 0.0f, 1.0f,
261
262      0.9f, 0.0f, 0.0f, 0.0f, 0.0f, 1.0f,
263      0.7f, -0.1f, 0.0f, 0.0f, 0.0f, 1.0f,
264      0.9f, -0.1f, 0.0f, 0.0f, 0.0f, 1.0f
265    };
266    MeshColor* letras = new MeshColor();
267    letras->CreateMeshColor(vertices_letras, 720); //120
268    meshColorList.push_back(letras);
269

```

ii. Captura de pantalla



- b. Generar el dibujo de la casa de la clase, pero en lugar de instanciar triangulos y cuadrados será instanciando piramides y cubos, para esto se requiere crear shaders diferentes de los colores: rojo, verde, azul, café y verde oscuro en lugar de usar el shader con el color clamp.
 - i. Código:



Se generan shaders de color para las pirámides y cubos.

```
1  #version 330
2
3  layout (location =0) in vec3 pos;
4  out vec4 vColor;
5  uniform mat4 model;
6  uniform mat4 projection;
7
8  void main()
9  {
10     gl_Position = projection * model * vec4(pos,1.0f);
11     vColor=vec4(0.0f,0.0f,1.0f,1.0f);
12 }
```

Se usa el código de shader y dentro del main se coloca el color en formato RGB.

```
//Cuadrado rojo
shaderList[3].useShader();
uniformModel = shaderList[3].getModelLocation();
uniformProjection = shaderList[3].getProjectLocation();

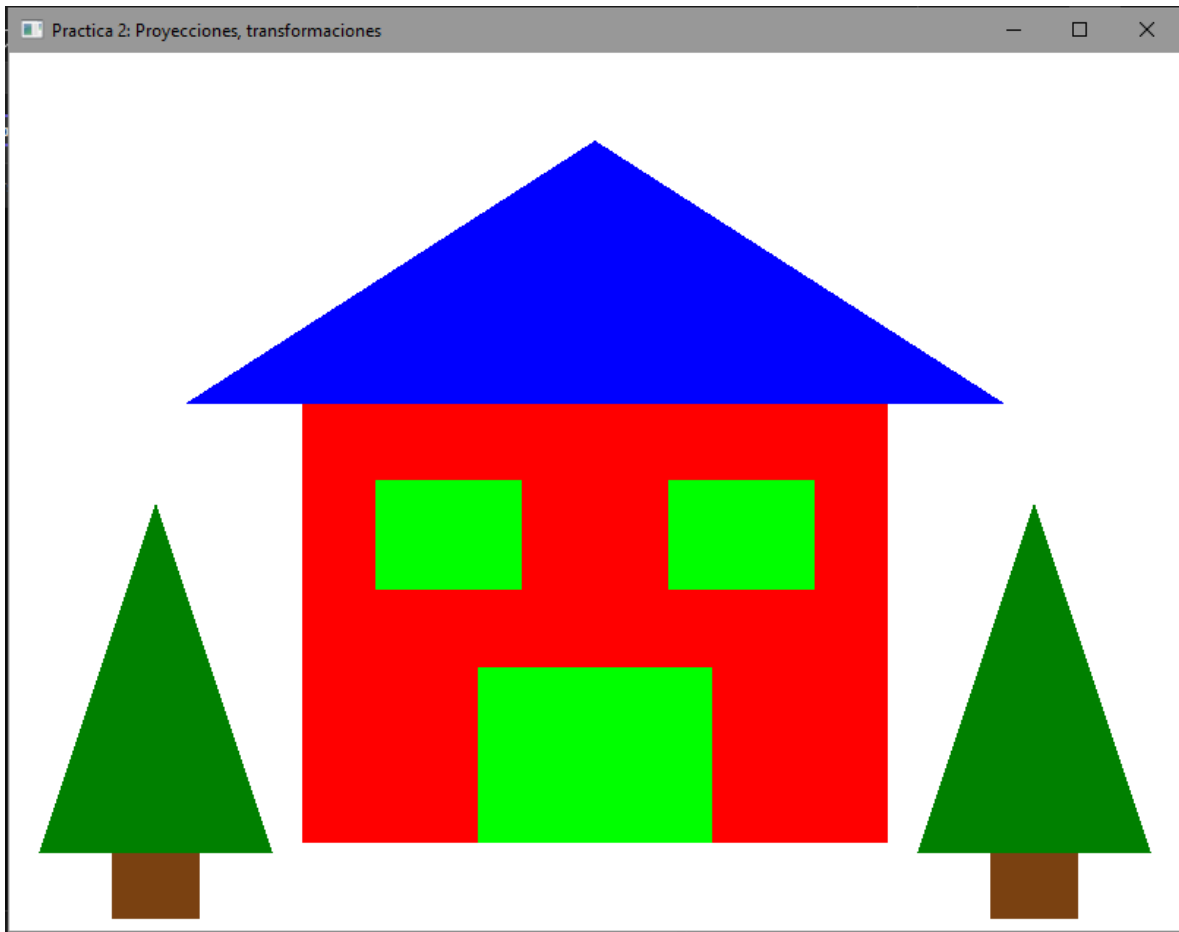
model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, -0.3f, -4.0f));
model = glm::scale(model, glm::vec3((1.0f, 1.0f, 1.0f)));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
meshList[1]->RenderMesh();

//Triangulo azul
shaderList[5].useShader();
uniformModel = shaderList[5].getModelLocation();
uniformProjection = shaderList[5].getProjectLocation();
angulo += 0.3;

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f, 0.5f, -4.0f));
model = glm::scale(model, glm::vec3(1.4f, 0.6f, 2.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model)); //FALSE ES PARA QUE NO SEA TRANSPUESTA
glUniformMatrix4fv(uniformProjection, 1, GL_FALSE, glm::value_ptr(projection));
meshList[0]->RenderMesh();
```

Y se genera el código como en el ejercicio de clase.

ii. Captura de pantalla:



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 no hubo ningún problema para ejecutar los ejercicios de práctica propuestos.

3. Conclusión:

- a. Los ejercicios del reporte: Complejidad, Explicación.

No tuve ningún problema con los ejercicios del reporte. Se explico de forma correcta y considero que logré realizarlos.

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

El desarrollo de la práctica fue bueno ya que pude completar los ejercicios propuestos.

c. Conclusión.

El uso de shaders para el color y uso de clases para generar figuras hace más fácil el trabajo de dibujo en el API de OpenGL.

4. Bibliografía en formato APA

- a. De Programación, T. (2016a, abril 24). Tutorial OpenGL - Cámara. *Tutorial OpenGL*. Recuperado 17 de febrero de 2024, de <https://acodigo.blogspot.com/2016/04/tutorial-opengl-camara.html>
- b. De Programación, T. (2016b, septiembre 25). GLSL Variables Uniform. *Tutor de programación*. Recuperado 17 de febrero de 2024, de <https://acodigo.blogspot.com/2016/09/glsl-variables-uniform.html>
- c. *Tutorial 3 : Matrices*. (s. f.-a). <https://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-3-matrices/#la-matriz-vista>
- d. *Tutorial 3 : Matrices*. (s. f.-b). <https://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-3-matrices/#la-matriz-proyecci%C3%B3n>
- e. *Tutorial 3 : Matrices*. (s. f.-c). Opnengl-tutorial. Recuperado 17 de febrero de 2024, de <https://www.opengl-tutorial.org/es/beginners-tutorials/tutorial-3-matrices/#rotaci%C3%B3n-de-matrices>
- f. *Why do I need extern?* (s. f.). Stack Overflow. <https://stackoverflow.com/questions/12433534/why-do-i-need-extern>