

Proiect 1 - Grafică pe Calculator  
- Depășire între 2 dreptunghiuri -

Linte Robert Ovidiu  
Popescu Paullo Robertto Karloss  
Grupa 331

Conf. Dr. Stupariu Mihai-Sorin

- 10 Noiembrie 2022 -

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
1.1	Modul de organizare al echipei . . . . .	2
1.2	Obiectivele proiectului . . . . .	2
1.3	Vizionarea proiectului . . . . .	2
<b>2</b>	<b>Desenarea obiectelor</b>	<b>3</b>
2.1	Prezentarea tablei de joc . . . . .	3
2.2	Cum a fost construită tabla de joc . . . . .	3
2.3	Cod sursă . . . . .	4
<b>3</b>	<b>Adăugarea translațiilor</b>	<b>18</b>
3.1	Prezentarea Translației 1 . . . . .	18
3.1.1	Cod sursă . . . . .	18
3.2	Prezentarea Translației 2 . . . . .	20
3.2.1	Cod sursă . . . . .	21
3.3	Prezentarea Translației 3 . . . . .	22
3.3.1	Cod sursă . . . . .	22
<b>4</b>	<b>Afișarea câștigătorului</b>	<b>23</b>
4.1	Prezentarea înainte de translația finală . . . . .	23
4.1.1	Cod sursă . . . . .	23
4.2	Prezentare după translație . . . . .	26
4.2.1	Cod sursă . . . . .	26
<b>5</b>	<b>Codul Sursă Complet</b>	<b>27</b>
<b>6</b>	<b>Codul pentru Shader</b>	<b>50</b>
<b>7</b>	<b>Referințe</b>	<b>51</b>

# Introducere

## 1.1 Modul de organizare al echipei

Echipa noastră:

- Linte Robert Ovidiu
- Popescu Paullo Robertto Karloss

Se va specifica **numele** membrului care a **contribuit** la realizarea fiecărei *etape*.

## 1.2 Obiectivele proiectului

Simularea unei "depășiri":

- O mașină (un dreptunghi) se deplasează pe o șosea uniform (print translație)
- O altă mașină (alt dreptunghi) vine din spate (tot prin translații)
- La un moment dat a doua mașină intră în depășire
- A doua mașină trece în fața primei mașini
- Se afișează la final câștigătorul "cursei"

Aprofundarea cunoștințelor în OpenGL prin:

- Utilizarea translațiilor
- Desenarea obiectelor
- Utilizarea culorilor

## 1.3 Vizionarea proiectului

Puteți viziona *demo-ul* proiectului **aici**, iar *repository-ul* de pe Github **aici**.

# Desenarea obiectelor

Această etapă a fost realizată de *Popescu Paullo Robertto Karloss*.

## 2.1 Prezentarea tablei de joc

Tabla de joc conține:

- Șoseua propiu-zisă
- O linie punctată pe post de marcaj rutier
- Două mașini de culori diferite (prima roșie, a doua albastră) care au forma unor dreptunghiuri
- Iarbă pe marginea șoselei
- Un text la finalul șoselei cu mesajul "FINISH", pentru scoate în evidență câștigătorul "cursei"

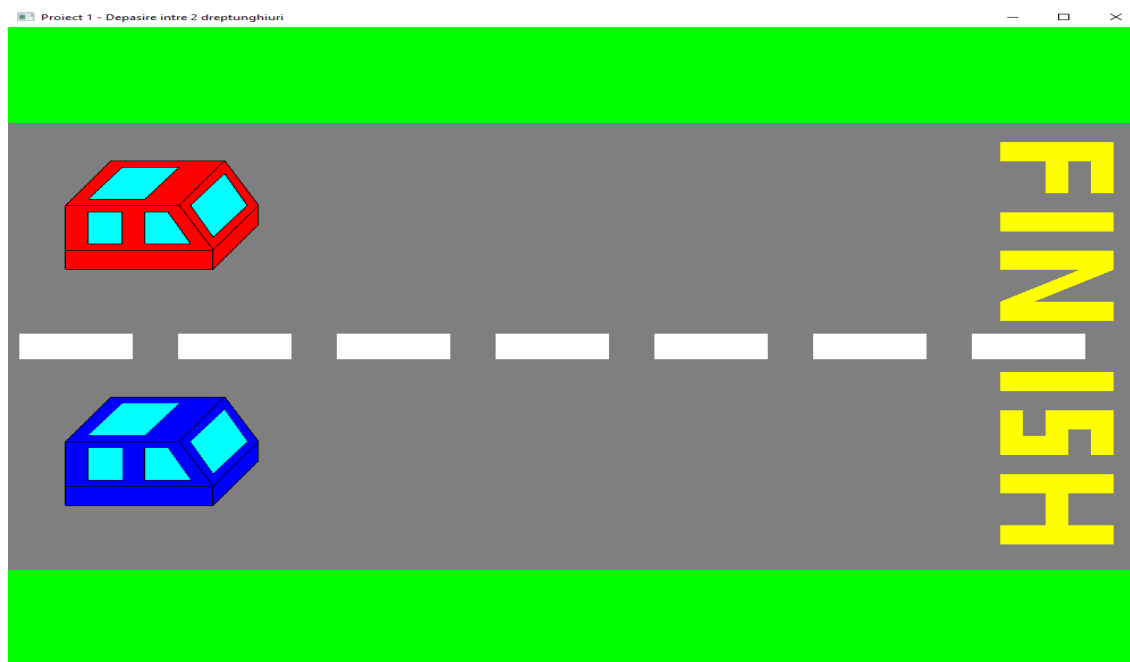


Figura 2.1: Tabla de joc

## 2.2 Cum a fost construită tabla de joc

Pentru a construi tabla de joc am creat un background (un dreptunghi) de culoare verde. Peste acesta am adăugat șoseaua (un dreptunghi de culoare gri). După care, în interiorul

șoselei am creat două dreptunghiuri, unul roșu și unul albastru, pe post de mașini, dreptunghiuri albe pe post de linie punctată (marcaj rutier din legislație) și alte dreptunghiuri de culoare galbenă pentru a scrie mesajul *"FINISH"*.

## 2.3 Cod sursă

```
1 void CreateVBO(void)
2 {
3     // varfurile
4     GLfloat Vertices[] = {
5         // varfuri pentru axe
6         -500.0f, 500.0f, 0.0f, 1.0f,
7         500.0f, 500.0f, 0.0f, 1.0f,
8         500.0f, -500.0f, 0.0f, 1.0f,
9         -500.0f, -500.0f, 0.0f, 1.0f,
10
11         -500.0f, 350.0f, 0.0f, 1.0f,
12         500.0f, 350.0f, 0.0f, 1.0f,
13         500.0f, -350.0f, 0.0f, 1.0f,
14         -500.0f, -350.0f, 0.0f, 1.0f,
15
16         -490.0f, 20.0f, 0.0f, 1.0f,
17         -390.0f, 20.0f, 0.0f, 1.0f,
18         -390.0f, -20.0f, 0.0f, 1.0f,
19         -490.0f, -20.0f, 0.0f, 1.0f,
20
21         -350.0f, 20.0f, 0.0f, 1.0f,
22         -250.0f, 20.0f, 0.0f, 1.0f,
23         -250.0f, -20.0f, 0.0f, 1.0f,
24         -350.0f, -20.0f, 0.0f, 1.0f,
25
26         -210.0f, 20.0f, 0.0f, 1.0f,
27         -110.0f, 20.0f, 0.0f, 1.0f,
28         -110.0f, -20.0f, 0.0f, 1.0f,
29         -210.0f, -20.0f, 0.0f, 1.0f,
30
31         -70.0f, 20.0f, 0.0f, 1.0f,
32         30.0f, 20.0f, 0.0f, 1.0f,
33         30.0f, -20.0f, 0.0f, 1.0f,
34         -70.0f, -20.0f, 0.0f, 1.0f,
```

```

35
36     70.0f, 20.0f, 0.0f, 1.0f,
37     170.0f, 20.0f, 0.0f, 1.0f,
38     170.0f, -20.0f, 0.0f, 1.0f,
39     70.0f, -20.0f, 0.0f, 1.0f,
40
41     210.0f, 20.0f, 0.0f, 1.0f,
42     310.0f, 20.0f, 0.0f, 1.0f,
43     310.0f, -20.0f, 0.0f, 1.0f,
44     210.0f, -20.0f, 0.0f, 1.0f,
45
46     350.0f, 20.0f, 0.0f, 1.0f,
47     450.0f, 20.0f, 0.0f, 1.0f,
48     450.0f, -20.0f, 0.0f, 1.0f,
49     350.0f, -20.0f, 0.0f, 1.0f,
50
51     490.0f, 20.0f, 0.0f, 1.0f,
52     500.0f, 20.0f, 0.0f, 1.0f,
53     500.0f, -20.0f, 0.0f, 1.0f,
54     490.0f, -20.0f, 0.0f, 1.0f,
55
56     -450.0f, 250.0f, 0.0f, 1.0f,
57     -300.0f, 250.0f, 0.0f, 1.0f,
58     -300.0f, 120.0f, 0.0f, 1.0f,
59     -450.0f, 120.0f, 0.0f, 1.0f,
60
61     -450.0f, -250.0f, 0.0f, 1.0f,
62     -300.0f, -250.0f, 0.0f, 1.0f,
63     -300.0f, -120.0f, 0.0f, 1.0f,
64     -450.0f, -120.0f, 0.0f, 1.0f,
65
66     // - WRITING TEXT: "FINISH" - //
67
68     // WORD "F"
69     475.0f, 320.0f, 0.0f, 1.0f,
70     375.0f, 320.0f, 0.0f, 1.0f,
71     375.0f, 290.0f, 0.0f, 1.0f,
72     475.0f, 290.0f, 0.0f, 1.0f,
73
74     475.0f, 290.0f, 0.0f, 1.0f,
75     475.0f, 240.0f, 0.0f, 1.0f,

```

```

76      455.0f, 240.0f, 0.0f, 1.0f,
77      455.0f, 290.0f, 0.0f, 1.0f,
78
79      435.0f, 290.0f, 0.0f, 1.0f,
80      435.0f, 240.0f, 0.0f, 1.0f,
81      415.0f, 240.0f, 0.0f, 1.0f,
82      415.0f, 290.0f, 0.0f, 1.0f,
83
84      // WORD "I"
85      475.0f, 210.0f, 0.0f, 1.0f,
86      375.0f, 210.0f, 0.0f, 1.0f,
87      375.0f, 180.0f, 0.0f, 1.0f,
88      475.0f, 180.0f, 0.0f, 1.0f,
89
90      //WORD "N"
91      475.0f, 150.0f, 0.0f, 1.0f,
92      375.0f, 150.0f, 0.0f, 1.0f,
93      375.0f, 120.0f, 0.0f, 1.0f,
94      475.0f, 120.0f, 0.0f, 1.0f,
95
96      475.0f, 70.0f, 0.0f, 1.0f,
97      375.0f, 70.0f, 0.0f, 1.0f,
98      375.0f, 40.0f, 0.0f, 1.0f,
99      475.0f, 40.0f, 0.0f, 1.0f,
100
101      475.0f, 120.0f, 0.0f, 1.0f,
102      445.0f, 120.0f, 0.0f, 1.0f,
103      375.0f, 70.0f, 0.0f, 1.0f,
104      405.0f, 70.0f, 0.0f, 1.0f,
105
106      // WORD "I"
107      475.0f, -40.0f, 0.0f, 1.0f,
108      375.0f, -40.0f, 0.0f, 1.0f,
109      375.0f, -70.0f, 0.0f, 1.0f,
110      475.0f, -70.0f, 0.0f, 1.0f,
111
112      // WORD "S"
113      475.0f, -100.0f, 0.0f, 1.0f,
114      455.0f, -100.0f, 0.0f, 1.0f,
115      455.0f, -170.0f, 0.0f, 1.0f,
116      475.0f, -170.0f, 0.0f, 1.0f,

```

```

117
118     475.0f, -100.0f, 0.0f, 1.0f,
119     415.0f, -100.0f, 0.0f, 1.0f,
120     415.0f, -130.0f, 0.0f, 1.0f,
121     475.0f, -130.0f, 0.0f, 1.0f,
122
123     415.0f, -100.0f, 0.0f, 1.0f,
124     435.0f, -100.0f, 0.0f, 1.0f,
125     435.0f, -170.0f, 0.0f, 1.0f,
126     415.0f, -170.0f, 0.0f, 1.0f,
127
128     415.0f, -170.0f, 0.0f, 1.0f,
129     415.0f, -140.0f, 0.0f, 1.0f,
130     375.0f, -140.0f, 0.0f, 1.0f,
131     375.0f, -170.0f, 0.0f, 1.0f,
132
133     375.0f, -100.0f, 0.0f, 1.0f,
134     395.0f, -100.0f, 0.0f, 1.0f,
135     395.0f, -170.0f, 0.0f, 1.0f,
136     375.0f, -170.0f, 0.0f, 1.0f,
137
138     // WORD "H"
139     475.0f, -200.0f, 0.0f, 1.0f,
140     375.0f, -200.0f, 0.0f, 1.0f,
141     375.0f, -230.0f, 0.0f, 1.0f,
142     475.0f, -230.0f, 0.0f, 1.0f,
143
144     475.0f, -280.0f, 0.0f, 1.0f,
145     375.0f, -280.0f, 0.0f, 1.0f,
146     375.0f, -310.0f, 0.0f, 1.0f,
147     475.0f, -310.0f, 0.0f, 1.0f,
148
149     435.0f, -310.0f, 0.0f, 1.0f,
150     415.0f, -310.0f, 0.0f, 1.0f,
151     415.0f, -200.0f, 0.0f, 1.0f,
152     435.0f, -200.0f, 0.0f, 1.0f,
153
154     // - WRITING TEXT: "BLUE WIN" - //
155
156     // WORD "B"
157     -300.0f, 700.0f, 0.0f, 1.0f,

```



```

158 -300.0f, 600.0f, 0.0f, 1.0f,
159 -280.0f, 600.f, 0.0f, 1.0f,
160 -280.0f, 700.0f, 0.0f, 1.0f,
161
162 -250.0f, 700.0f, 0.0f, 1.0f,
163 -250.0f, 600.0f, 0.0f, 1.0f,
164 -230.0f, 600.f, 0.0f, 1.0f,
165 -230.0f, 700.0f, 0.0f, 1.0f,
166
167 -300.0f, 700.0f, 0.0f, 1.0f,
168 -300.0f, 680.0f, 0.0f, 1.0f,
169 -230.0f, 680.f, 0.0f, 1.0f,
170 -230.0f, 700.0f, 0.0f, 1.0f,
171
172 -300.0f, 660.0f, 0.0f, 1.0f,
173 -300.0f, 640.0f, 0.0f, 1.0f,
174 -230.0f, 640.f, 0.0f, 1.0f,
175 -230.0f, 660.0f, 0.0f, 1.0f,
176
177 -300.0f, 620.0f, 0.0f, 1.0f,
178 -300.0f, 600.0f, 0.0f, 1.0f,
179 -230.0f, 600.f, 0.0f, 1.0f,
180 -230.0f, 620.0f, 0.0f, 1.0f,
181
182 // WORD "L"
183 -200.0f, 700.0f, 0.0f, 1.0f,
184 -200.0f, 600.0f, 0.0f, 1.0f,
185 -180.0f, 600.f, 0.0f, 1.0f,
186 -180.0f, 700.0f, 0.0f, 1.0f,
187
188 -200.0f, 620.0f, 0.0f, 1.0f,
189 -200.0f, 600.0f, 0.0f, 1.0f,
190 -150.0f, 600.f, 0.0f, 1.0f,
191 -150.0f, 620.0f, 0.0f, 1.0f,
192
193 // WORD "U"
194 -120.0f, 700.0f, 0.0f, 1.0f,
195 -120.0f, 600.0f, 0.0f, 1.0f,
196 -100.0f, 600.f, 0.0f, 1.0f,
197 -100.0f, 700.0f, 0.0f, 1.0f,
198

```

```

199 -70.0f, 700.0f, 0.0f, 1.0f,
200 -70.0f, 600.0f, 0.0f, 1.0f,
201 -50.0f, 600.f, 0.0f, 1.0f,
202 -50.0f, 700.0f, 0.0f, 1.0f,
203
204 -120.0f, 620.0f, 0.0f, 1.0f,
205 -50.0f, 620.0f, 0.0f, 1.0f,
206 -50.0f, 600.0f, 0.0f, 1.0f,
207 -120.0f, 600.0f, 0.0f, 1.0f,
208
209 // WORD "E"
210 -20.0f, 700.0f, 0.0f, 1.0f,
211 -20.0f, 600.0f, 0.0f, 1.0f,
212 0.0f, 600.f, 0.0f, 1.0f,
213 0.0f, 700.0f, 0.0f, 1.0f,
214
215 -20.0f, 620.0f, 0.0f, 1.0f,
216 30.0f, 620.0f, 0.0f, 1.0f,
217 30.0f, 600.0f, 0.0f, 1.0f,
218 -20.0f, 600.0f, 0.0f, 1.0f,
219
220 -20.0f, 660.0f, 0.0f, 1.0f,
221 30.0f, 660.0f, 0.0f, 1.0f,
222 30.0f, 640.0f, 0.0f, 1.0f,
223 -20.0f, 640.0f, 0.0f, 1.0f,
224
225 -20.0f, 700.0f, 0.0f, 1.0f,
226 30.0f, 700.0f, 0.0f, 1.0f,
227 30.0f, 680.0f, 0.0f, 1.0f,
228 -20.0f, 680.0f, 0.0f, 1.0f,
229
230 // WORD "W"
231 100.0f, 700.0f, 0.0f, 1.0f,
232 120.0f, 700.0f, 0.0f, 1.0f,
233 140.0f, 600.0f, 0.0f, 1.0f,
234 120.0f, 600.0f, 0.0f, 1.0f,
235
236 140.0f, 600.0f, 0.0f, 1.0f,
237 120.0f, 600.0f, 0.0f, 1.0f,
238 140.0f, 670.0f, 0.0f, 1.0f,
239 160.0f, 670.0f, 0.0f, 1.0f,

```

```

240
241     140.0f, 670.0f, 0.0f, 1.0f,
242     160.0f, 670.0f, 0.0f, 1.0f,
243     180.0f, 600.0f, 0.0f, 1.0f,
244     160.0f, 600.0f, 0.0f, 1.0f,
245
246     180.0f, 600.0f, 0.0f, 1.0f,
247     160.0f, 600.0f, 0.0f, 1.0f,
248     180.0f, 700.0f, 0.0f, 1.0f,
249     200.0f, 700.0f, 0.0f, 1.0f,
250
251     // WORD "I"
252     220.0f, 700.0f, 0.0f, 1.0f,
253     240.0f, 700.0f, 0.0f, 1.0f,
254     240.0f, 600.0f, 0.0f, 1.0f,
255     220.0f, 600.0f, 0.0f, 1.0f,
256
257     // WORD "N"
258     260.0f, 700.0f, 0.0f, 1.0f,
259     280.0f, 700.0f, 0.0f, 1.0f,
260     280.0f, 600.0f, 0.0f, 1.0f,
261     260.0f, 600.0f, 0.0f, 1.0f,
262
263     310.0f, 700.0f, 0.0f, 1.0f,
264     330.0f, 700.0f, 0.0f, 1.0f,
265     330.0f, 600.0f, 0.0f, 1.0f,
266     310.0f, 600.0f, 0.0f, 1.0f,
267
268     280.0f, 700.0f, 0.0f, 1.0f,
269     280.0f, 670.0f, 0.0f, 1.0f,
270     310.0f, 600.0f, 0.0f, 1.0f,
271     310.0f, 630.0f, 0.0f, 1.0f,
272
273     // RED CAR
274     -450.0f, 120.0f, 0.0f, 1.0f,
275     -320.0f, 120.0f, 0.0f, 1.0f,
276     -320.0f, 150.0f, 0.0f, 1.0f,
277     -450.0f, 150.0f, 0.0f, 1.0f,
278     -450.0f, 120.0f, 0.0f, 1.0f,
279
280     -450.0f, 150.0f, 0.0f, 1.0f,

```

281	-320.0f, 150.0f, 0.0f, 1.0f,
282	-350.0f, 220.0f, 0.0f, 1.0f,
283	-450.0f, 220.0f, 0.0f, 1.0f,
284	-450.0f, 150.0f, 0.0f, 1.0f,
285	
286	-320.0f, 120.0f, 0.0f, 1.0f,
287	-320.0f, 150.0f, 0.0f, 1.0f,
288	-280.0f, 220.0f, 0.0f, 1.0f,
289	-280.0f, 190.0f, 0.0f, 1.0f,
290	-320.0f, 120.0f, 0.0f, 1.0f,
291	
292	-350.0f, 220.0f, 0.0f, 1.0f,
293	-320.0f, 150.0f, 0.0f, 1.0f,
294	-280.0f, 220.0f, 0.0f, 1.0f,
295	-310.0f, 290.0f, 0.0f, 1.0f,
296	-350.0f, 220.0f, 0.0f, 1.0f,
297	
298	-310.0f, 290.0f, 0.0f, 1.0f,
299	-350.0f, 220.0f, 0.0f, 1.0f,
300	-450.0f, 220.0f, 0.0f, 1.0f,
301	-410.0f, 290.0f, 0.0f, 1.0f,
302	-310.0f, 290.0f, 0.0f, 1.0f,
303	
304	-310.0f, 290.0f, 0.0f, 1.0f,
305	-350.0f, 220.0f, 0.0f, 1.0f,
306	-450.0f, 220.0f, 0.0f, 1.0f,
307	-410.0f, 290.0f, 0.0f, 1.0f,
308	-310.0f, 290.0f, 0.0f, 1.0f,
309	
310	-430.0f, 230.0f, 0.0f, 1.0f,
311	-400.0f, 280.0f, 0.0f, 1.0f,
312	-350.0f, 280.0f, 0.0f, 1.0f,
313	-380.0f, 230.0f, 0.0f, 1.0f,
314	-430.0f, 230.0f, 0.0f, 1.0f,
315	
316	-320.0f, 170.0f, 0.0f, 1.0f,
317	-340.0f, 220.0f, 0.0f, 1.0f,
318	-310.0f, 270.0f, 0.0f, 1.0f,
319	-290.0f, 220.0f, 0.0f, 1.0f,
320	-320.0f, 170.0f, 0.0f, 1.0f,
321	

```

322     -380.0f, 210.0f, 0.0f, 1.0f,
323     -380.0f, 160.0f, 0.0f, 1.0f,
324     -340.0f, 160.0f, 0.0f, 1.0f,
325     -360.0f, 210.0f, 0.0f, 1.0f,
326     -380.0f, 210.0f, 0.0f, 1.0f,
327
328     -400.0f, 160.0f, 0.0f, 1.0f,
329     -430.0f, 160.0f, 0.0f, 1.0f,
330     -430.0f, 210.0f, 0.0f, 1.0f,
331     -400.0f, 210.0f, 0.0f, 1.0f,
332     -400.0f, 160.0f, 0.0f, 1.0f,
333
334     // BLUE CAR
335     -450.0f, -250.0f, 0.0f, 1.0f,
336     -320.0f, -250.0f, 0.0f, 1.0f,
337     -320.0f, -220.0f, 0.0f, 1.0f,
338     -450.0f, -220.0f, 0.0f, 1.0f,
339     -450.0f, -250.0f, 0.0f, 1.0f,
340
341     -450.0f, -220.0f, 0.0f, 1.0f,
342     -320.0f, -220.0f, 0.0f, 1.0f,
343     -350.0f, -150.0f, 0.0f, 1.0f,
344     -450.0f, -150.0f, 0.0f, 1.0f,
345     -450.0f, -220.0f, 0.0f, 1.0f,
346
347     -320.0f, -250.0f, 0.0f, 1.0f,
348     -320.0f, -220.0f, 0.0f, 1.0f,
349     -280.0f, -150.0f, 0.0f, 1.0f,
350     -280.0f, -180.0f, 0.0f, 1.0f,
351     -320.0f, -250.0f, 0.0f, 1.0f,
352
353     -350.0f, -150.0f, 0.0f, 1.0f,
354     -320.0f, -220.0f, 0.0f, 1.0f,
355     -280.0f, -150.0f, 0.0f, 1.0f,
356     -310.0f, -80.0f, 0.0f, 1.0f,
357     -350.0f, -150.0f, 0.0f, 1.0f,
358
359     -310.0f, -80.0f, 0.0f, 1.0f,
360     -350.0f, -150.0f, 0.0f, 1.0f,
361     -450.0f, -150.0f, 0.0f, 1.0f,
362     -410.0f, -80.0f, 0.0f, 1.0f,

```

```

363     -310.0f, -80.0f, 0.0f, 1.0f,
364
365     -430.0f, -140.0f, 0.0f, 1.0f,
366     -400.0f, -90.0f, 0.0f, 1.0f,
367     -350.0f, -90.0f, 0.0f, 1.0f,
368     -380.0f, -140.0f, 0.0f, 1.0f,
369     -430.0f, -140.0f, 0.0f, 1.0f,
370
371     -320.0f, -200.0f, 0.0f, 1.0f,
372     -340.0f, -150.0f, 0.0f, 1.0f,
373     -310.0f, -100.0f, 0.0f, 1.0f,
374     -290.0f, -150.0f, 0.0f, 1.0f,
375     -320.0f, -200.0f, 0.0f, 1.0f,
376
377     -380.0f, -160.0f, 0.0f, 1.0f,
378     -380.0f, -210.0f, 0.0f, 1.0f,
379     -340.0f, -210.0f, 0.0f, 1.0f,
380     -360.0f, -160.0f, 0.0f, 1.0f,
381     -380.0f, -210.0f, 0.0f, 1.0f,
382
383     -400.0f, -210.0f, 0.0f, 1.0f,
384     -430.0f, -210.0f, 0.0f, 1.0f,
385     -430.0f, -160.0f, 0.0f, 1.0f,
386     -400.0f, -160.0f, 0.0f, 1.0f,
387     -400.0f, -210.0f, 0.0f, 1.0f,
388 };
```

Figura 2.1: Crearea punctelor și a culorilor pentru Tabla de joc

```

1  glDrawArrays(GL_POLYGON, 0, 4);
2  glDrawArrays(GL_POLYGON, 4, 4);
3  glDrawArrays(GL_POLYGON, 8, 4);
4  glDrawArrays(GL_POLYGON, 12, 4);
5  glDrawArrays(GL_POLYGON, 16, 4);
6  glDrawArrays(GL_POLYGON, 20, 4);
7  glDrawArrays(GL_POLYGON, 24, 4);
8  glDrawArrays(GL_POLYGON, 28, 4);
9  glDrawArrays(GL_POLYGON, 32, 4);
10 glDrawArrays(GL_POLYGON, 36, 4);
11 glDrawArrays(GL_POLYGON, 48, 4);
12 glDrawArrays(GL_POLYGON, 52, 4);
13 glDrawArrays(GL_POLYGON, 56, 4);
14 glDrawArrays(GL_POLYGON, 60, 4);
15 glDrawArrays(GL_POLYGON, 64, 4);
16 glDrawArrays(GL_POLYGON, 68, 4);
17 glDrawArrays(GL_POLYGON, 72, 4);
18 glDrawArrays(GL_POLYGON, 76, 4);
19 glDrawArrays(GL_POLYGON, 80, 4);
20 glDrawArrays(GL_POLYGON, 84, 4);
21 glDrawArrays(GL_POLYGON, 88, 4);
22 glDrawArrays(GL_POLYGON, 92, 4);
23 glDrawArrays(GL_POLYGON, 96, 4);
24 glDrawArrays(GL_POLYGON, 100, 4);
25 glDrawArrays(GL_POLYGON, 104, 4);
26 glDrawArrays(GL_POLYGON, 108, 4);

```

Figura 2.2:Cod OpenGL pentru Tabla de joc

```

1  glDrawArrays(GL_POLYGON, 200, 4);
2  glDrawArrays(GL_POLYGON, 205, 4);
3  glDrawArrays(GL_POLYGON, 210, 4);
4  glDrawArrays(GL_POLYGON, 215, 4);
5  glDrawArrays(GL_POLYGON, 220, 4);
6  glDrawArrays(GL_POLYGON, 225, 4);
7  glUniform1i(codColLocation, 4);
8  glDrawArrays(GL_POLYGON, 230, 4);
9  glDrawArrays(GL_POLYGON, 235, 4);
10 glDrawArrays(GL_POLYGON, 240, 4);
11 glDrawArrays(GL_POLYGON, 245, 4);
12 glUniform1i(codColLocation, 3);

```

```
13  glDrawArrays(GL_LINES, 202, 2);
14  glDrawArrays(GL_LINES, 200, 2);
15  glDrawArrays(GL_LINES, 203, 2);
16  glDrawArrays(GL_LINES, 201, 2);
17  glDrawArrays(GL_LINES, 205, 2);
18  glDrawArrays(GL_LINES, 206, 2);
19  glDrawArrays(GL_LINES, 207, 2);
20  glDrawArrays(GL_LINES, 208, 2);
21  glDrawArrays(GL_LINES, 210, 2);
22  glDrawArrays(GL_LINES, 211, 2);
23  glDrawArrays(GL_LINES, 212, 2);
24  glDrawArrays(GL_LINES, 213, 2);
25  glDrawArrays(GL_LINES, 215, 2);
26  glDrawArrays(GL_LINES, 216, 2);
27  glDrawArrays(GL_LINES, 217, 2);
28  glDrawArrays(GL_LINES, 218, 2);
29  glDrawArrays(GL_LINES, 220, 2);
30  glDrawArrays(GL_LINES, 221, 2);
31  glDrawArrays(GL_LINES, 222, 2);
32  glDrawArrays(GL_LINES, 223, 2);
33  glDrawArrays(GL_LINES, 224, 2);
34  glDrawArrays(GL_LINES, 225, 2);
35  glDrawArrays(GL_LINES, 226, 2);
36  glDrawArrays(GL_LINES, 230, 2);
37  glDrawArrays(GL_LINES, 231, 2);
38  glDrawArrays(GL_LINES, 232, 2);
39  glDrawArrays(GL_LINES, 233, 2);
40  glDrawArrays(GL_LINES, 235, 2);
41  glDrawArrays(GL_LINES, 236, 2);
42  glDrawArrays(GL_LINES, 237, 2);
43  glDrawArrays(GL_LINES, 238, 2);
44  glDrawArrays(GL_LINES, 240, 2);
45  glDrawArrays(GL_LINES, 241, 2);
46  glDrawArrays(GL_LINES, 242, 2);
47  glDrawArrays(GL_LINES, 243, 2);
48  glDrawArrays(GL_LINES, 245, 2);
49  glDrawArrays(GL_LINES, 246, 2);
50  glDrawArrays(GL_LINES, 247, 2);
51  glDrawArrays(GL_LINES, 248, 2);
```

Figura 2.3: Cod OpenGL pentru Tabla de joc



```
1  glDrawArrays(GL_POLYGON, 250, 4);
2  glDrawArrays(GL_POLYGON, 255, 4);
3  glDrawArrays(GL_POLYGON, 260, 4);
4  glDrawArrays(GL_POLYGON, 265, 4);
5  glDrawArrays(GL_POLYGON, 270, 4);
6  glUniform1i(codColLocation, 3);
7  glDrawArrays(GL_LINES, 250, 2);
8  glDrawArrays(GL_LINES, 251, 2);
9  glDrawArrays(GL_LINES, 252, 2);
10 glDrawArrays(GL_LINES, 253, 2);
11 glDrawArrays(GL_LINES, 255, 2);
12 glDrawArrays(GL_LINES, 256, 2);
13 glDrawArrays(GL_LINES, 257, 2);
14 glDrawArrays(GL_LINES, 258, 2);
15 glDrawArrays(GL_LINES, 261, 2);
16 glDrawArrays(GL_LINES, 262, 2);
17 glDrawArrays(GL_LINES, 263, 2);
18 glDrawArrays(GL_LINES, 265, 2);
19 glDrawArrays(GL_LINES, 266, 2);
20 glDrawArrays(GL_LINES, 267, 2);
21 glDrawArrays(GL_LINES, 268, 2);
22 glDrawArrays(GL_LINES, 270, 2);
23 glDrawArrays(GL_LINES, 271, 2);
24 glDrawArrays(GL_LINES, 272, 2);
25 glDrawArrays(GL_LINES, 273, 2);
26 glDrawArrays(GL_LINES, 275, 2);
27 glDrawArrays(GL_LINES, 276, 2);
28 glDrawArrays(GL_LINES, 277, 2);
29 glDrawArrays(GL_LINES, 278, 2);
30 glDrawArrays(GL_LINES, 280, 2);
31 glDrawArrays(GL_LINES, 281, 2);
32 glDrawArrays(GL_LINES, 282, 2);
33 glDrawArrays(GL_LINES, 283, 2);
34 glDrawArrays(GL_LINES, 285, 2);
35 glDrawArrays(GL_LINES, 286, 2);
36 glDrawArrays(GL_LINES, 287, 2);
37 glDrawArrays(GL_LINES, 288, 2);
38 glDrawArrays(GL_LINES, 290, 2);
39 glDrawArrays(GL_LINES, 291, 2);
40 glDrawArrays(GL_LINES, 292, 2);
41 glDrawArrays(GL_LINES, 293, 2);
```

```

42 | glUniform1i(codColLocation, 4);
43 | glDrawArrays(GL_POLYGON, 275, 4);
44 | glDrawArrays(GL_POLYGON, 280, 4);
45 | glDrawArrays(GL_POLYGON, 285, 4);
46 | glDrawArrays(GL_POLYGON, 290, 4);

```

Figura 2.4: Cod OpenGL pentru Tabla de joc

```

1 | glDrawArrays(GL_POLYGON, 112, 4);
2 | glDrawArrays(GL_POLYGON, 116, 4);
3 | glDrawArrays(GL_POLYGON, 120, 4);
4 | glDrawArrays(GL_POLYGON, 124, 4);
5 | glDrawArrays(GL_POLYGON, 128, 4);
6 | glDrawArrays(GL_POLYGON, 132, 4);
7 | glDrawArrays(GL_POLYGON, 136, 4);
8 | glDrawArrays(GL_POLYGON, 140, 4);
9 | glDrawArrays(GL_POLYGON, 144, 4);
10 | glDrawArrays(GL_POLYGON, 148, 4);
11 | glDrawArrays(GL_POLYGON, 152, 4);
12 | glDrawArrays(GL_POLYGON, 156, 4);
13 | glDrawArrays(GL_POLYGON, 160, 4);
14 | glDrawArrays(GL_POLYGON, 164, 4);
15 | glDrawArrays(GL_POLYGON, 168, 4);
16 | glDrawArrays(GL_POLYGON, 172, 4);
17 | glDrawArrays(GL_POLYGON, 176, 4);
18 | glDrawArrays(GL_POLYGON, 180, 4);
19 | glDrawArrays(GL_POLYGON, 184, 4);
20 | glDrawArrays(GL_POLYGON, 188, 4);
21 | glDrawArrays(GL_POLYGON, 192, 4);
22 | glDrawArrays(GL_POLYGON, 196, 4);

```

Figura 2.5: Cod OpenGL pentru Tabla de joc

# Adăugarea translațiilor

Această etapă a fost realizată de *Linte Robert Ovidiu*.

## 3.1 Prezentarea Translației 1

Dreptunghiul roșu pleacă cu o viteză inițială mai mică decât a dreptunghiului albastru.

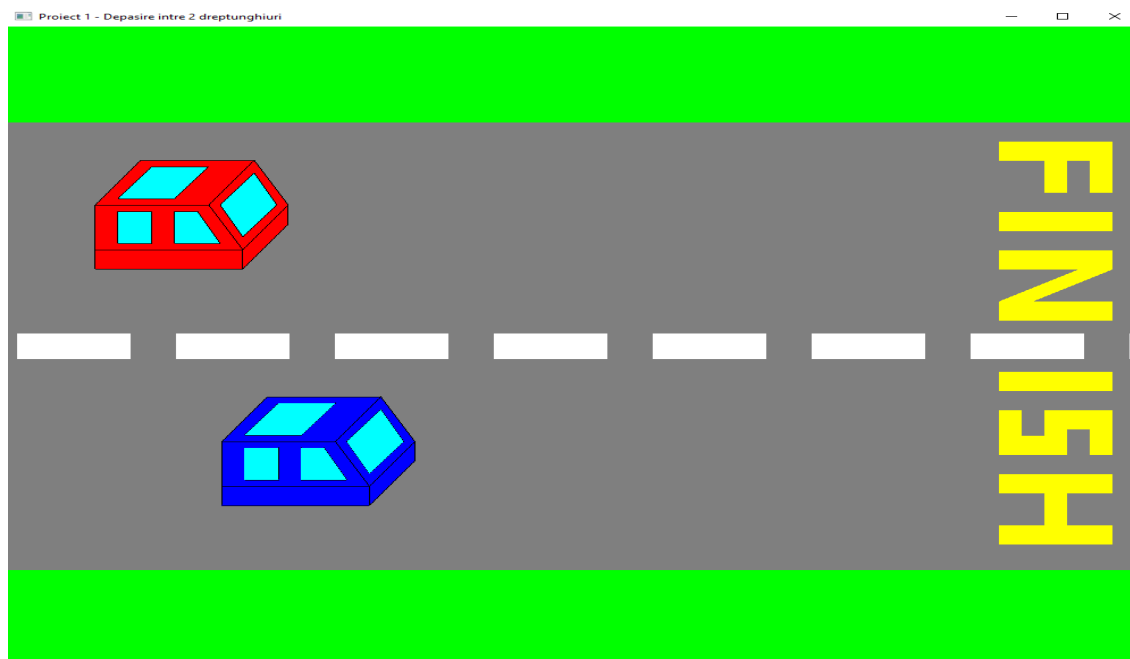


Figura 3.1.1: Deplasarea inițială a dreptunghiurilor

### 3.1.1 Cod sursă

```
1 // Matricea de redimensionare (pentru elementele "fixe")
2 myMatrix = resizeMatrix;
3 // Culoarea
4 codCol = 0;
5 // Transmite variabile uniforme
6 glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
7 ;
8 glUniform1i(codColLocation, codCol);
9
10 //...//
```

```

11 // Matricea pentru dreptunghiul rosu
12 myMatrix = resizeMatrix * matrTransl * matrDepl * matrScale2;
13 // Culoarea
14 codCol = 2;
15 // Transmitere variabile uniforme
16 glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
17 ;
18 glUniform1i(codColLocation, codCol);
19 //...//
20
21 // Matricea pentru dreptunghiul rosu
22 myMatrix = resizeMatrix * matrTransl2 * matrDepl * matrScale2 *
23     matrRot;
24 // Culoarea
25 codCol = 1;
26 // Transmitere variabile uniforme
27 glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
28 ;
29 glUniform1i(codColLocation, codCol);

```

Figura 3.1.2: Cod OpenGL pentru prima translație

```

1 resizeMatrix = glm::ortho(-width, width, -height, height); // scalam
2 , "aducem" scena la "patratul standard" [-1,1]x[-1,1]
3 matrTransl = glm::translate(glm::mat4(1.0f), glm::vec3(i, k, 0.0));
4 // controleaza translatia de-a lungul lui 0x
5 matrTransl3 = glm::translate(glm::mat4(1.0f), glm::vec3(0.0, h, 0.0)
6 );
7 matrDepl = glm::translate(glm::mat4(1.0f), glm::vec3(1.0, 1.0, 0.0))
8 ; // plaseaza patratul rosu
9 matrScale2 = glm::scale(glm::mat4(1.0f), glm::vec3(1.0, 1.0, 0.0));
10 // folosita la desenarea patratului rosu
11 matrTransl2 = glm::translate(glm::mat4(1.0f), glm::vec3(j, l, 0.0));
12 matrRot = glm::rotate(glm::mat4(1.0f), angle, glm::vec3(0.0, 0.0,
13     1.0)); // rotatie folosita la deplasarea patratului rosu

```

Figura 3.1.3: Cod OpenGL pentru prima translație

```

1 void miscad(void)
2 {
3     if (i > -1 && i <= 400 && j <= 750)

```

```

4 {
5     i = i + alpha;
6     alpha = +step;
7 }
8 if (j > -1.0 && j <= 150) {
9     j = j + alpha2;
10    alpha2 = +step2;
11 }

```

Figura 3.1.4: Cod OpenGL pentru prima translație

## 3.2 Prezentarea Translației 2

În momentul în care dreptunghiul albastru, îl depășește total pe cel roșu, începe procesul de depășire (*se aplică o rotație de 0.5 pe dreptunghiul albastru cât și o translație pe diagonală, iar în final se aplică o rotație inversă pentru a-l aduce pe poziția inițială, i.e. paralel cu axa  $Ox$* ).

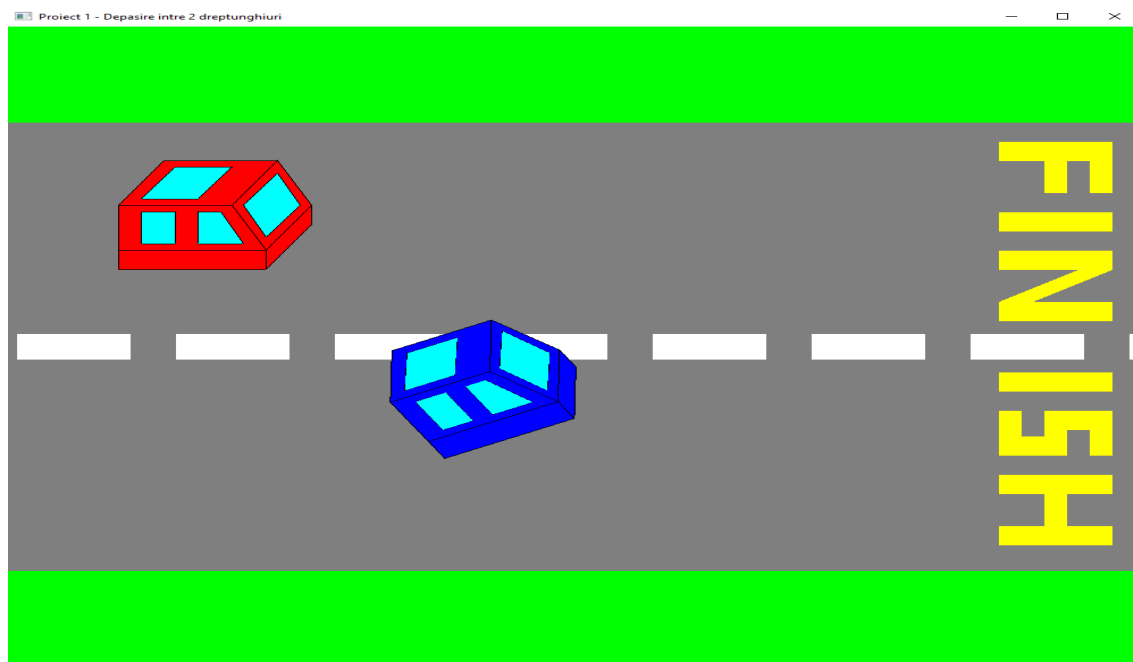


Figura 3.2.1: Depășire dreptunghi roșu

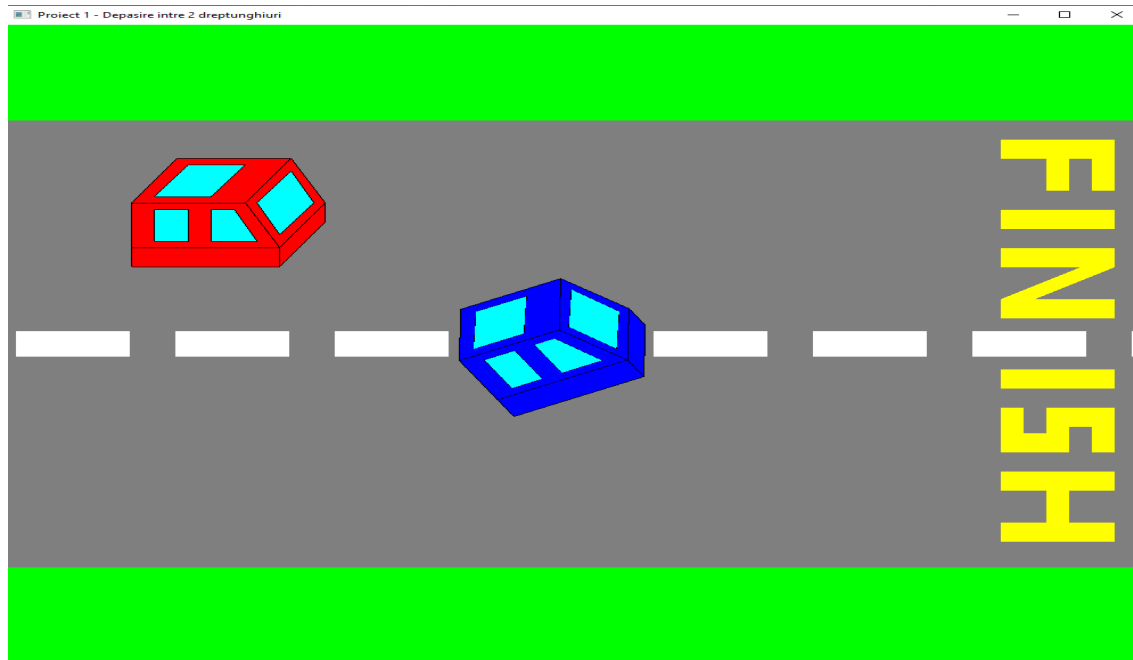


Figura 3.2.2: Depășire dreptunghi roșu

### 3.2.1 Cod sursă

```

1  if (j >= 150 && j <= 250 && angle <= 0.5) {
2      angle += beta;
3      l = l + alpha3;
4      alpha3 = +step3;
5  }
6
7  if (l >= 200 && l <= 370 && angle >= 0.5) {
8      j = j + alpha2;
9      alpha2 = +step2;
10     l = l + alpha3;
11     alpha3 = +step2;
12 }
13 if (l >= 370 && angle <= 0.6 && angle >= 0) {
14     angle -= beta;
15     j = j + alpha2;
16     alpha2 = +step3;
17 }

```

Figura 3.2.3: Cod OpenGL pentru a doua translație

### 3.3 Prezentarea Translației 3

După translația 2, dreptunghiul albastru accelerează până trece linia de *"FINISH"*.

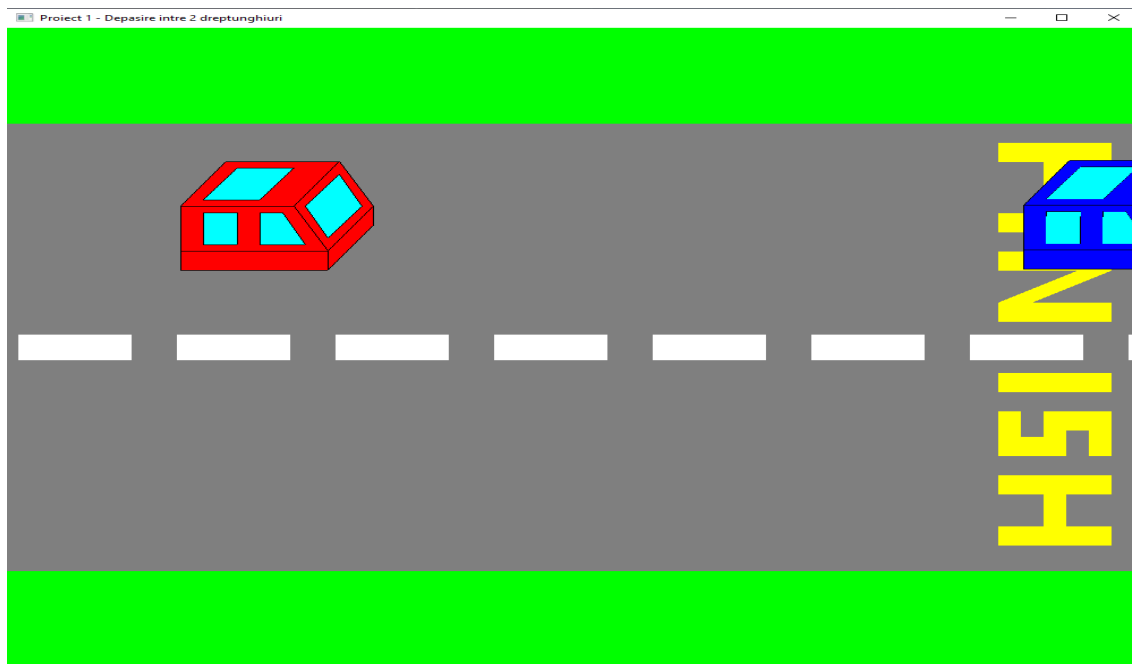


Figura 3.2.1: Accelerare dreptunghi albastru

#### 3.3.1 Cod sursă

```
1  if (l1 >= 370 && j <= 850 && angle <= 0) {  
2      j = j + alpha2;  
3      alpha2 = +step3;  
4  }
```

Figura 3.3.1: Cod OpenGL pentru a treia translație

# Afișarea câștigătorului

## 4.1 Prezentarea înainte de translația finală

Literele sunt inițial puse în afara ecranului (*nu sunt vizibile în tabla de joc*).

### 4.1.1 Cod sursă

```
1 // - WRITING TEXT: "BLUE WIN" - //
2
3 // WORD "B"
4 -300.0f, 700.0f, 0.0f, 1.0f,
5 -300.0f, 600.0f, 0.0f, 1.0f,
6 -280.0f, 600.f, 0.0f, 1.0f,
7 -280.0f, 700.0f, 0.0f, 1.0f,
8
9 -250.0f, 700.0f, 0.0f, 1.0f,
10 -250.0f, 600.0f, 0.0f, 1.0f,
11 -230.0f, 600.f, 0.0f, 1.0f,
12 -230.0f, 700.0f, 0.0f, 1.0f,
13
14 -300.0f, 700.0f, 0.0f, 1.0f,
15 -300.0f, 680.0f, 0.0f, 1.0f,
16 -230.0f, 680.f, 0.0f, 1.0f,
17 -230.0f, 700.0f, 0.0f, 1.0f,
18
19 -300.0f, 660.0f, 0.0f, 1.0f,
20 -300.0f, 640.0f, 0.0f, 1.0f,
21 -230.0f, 640.f, 0.0f, 1.0f,
22 -230.0f, 660.0f, 0.0f, 1.0f,
23
24 -300.0f, 620.0f, 0.0f, 1.0f,
25 -300.0f, 600.0f, 0.0f, 1.0f,
26 -230.0f, 600.f, 0.0f, 1.0f,
27 -230.0f, 620.0f, 0.0f, 1.0f,
28
29 // WORD "L"
30 -200.0f, 700.0f, 0.0f, 1.0f,
31 -200.0f, 600.0f, 0.0f, 1.0f,
32 -180.0f, 600.f, 0.0f, 1.0f,
```



```

33     -180.0f, 700.0f, 0.0f, 1.0f,
34
35     -200.0f, 620.0f, 0.0f, 1.0f,
36     -200.0f, 600.0f, 0.0f, 1.0f,
37     -150.0f, 600.f, 0.0f, 1.0f,
38     -150.0f, 620.0f, 0.0f, 1.0f,
39
40     // WORD "U"
41     -120.0f, 700.0f, 0.0f, 1.0f,
42     -120.0f, 600.0f, 0.0f, 1.0f,
43     -100.0f, 600.f, 0.0f, 1.0f,
44     -100.0f, 700.0f, 0.0f, 1.0f,
45
46     -70.0f, 700.0f, 0.0f, 1.0f,
47     -70.0f, 600.0f, 0.0f, 1.0f,
48     -50.0f, 600.f, 0.0f, 1.0f,
49     -50.0f, 700.0f, 0.0f, 1.0f,
50
51     -120.0f, 620.0f, 0.0f, 1.0f,
52     -50.0f, 620.0f, 0.0f, 1.0f,
53     -50.0f, 600.0f, 0.0f, 1.0f,
54     -120.0f, 600.0f, 0.0f, 1.0f,
55
56     // WORD "E"
57     -20.0f, 700.0f, 0.0f, 1.0f,
58     -20.0f, 600.0f, 0.0f, 1.0f,
59     0.0f, 600.f, 0.0f, 1.0f,
60     0.0f, 700.0f, 0.0f, 1.0f,
61
62     -20.0f, 620.0f, 0.0f, 1.0f,
63     30.0f, 620.0f, 0.0f, 1.0f,
64     30.0f, 600.0f, 0.0f, 1.0f,
65     -20.0f, 600.0f, 0.0f, 1.0f,
66
67     -20.0f, 660.0f, 0.0f, 1.0f,
68     30.0f, 660.0f, 0.0f, 1.0f,
69     30.0f, 640.0f, 0.0f, 1.0f,
70     -20.0f, 640.0f, 0.0f, 1.0f,
71
72     -20.0f, 700.0f, 0.0f, 1.0f,
73     30.0f, 700.0f, 0.0f, 1.0f,

```

```

74      30.0f, 680.0f, 0.0f, 1.0f,
75      -20.0f, 680.0f, 0.0f, 1.0f,
76
77      // WORD "W"
78      100.0f, 700.0f, 0.0f, 1.0f,
79      120.0f, 700.0f, 0.0f, 1.0f,
80      140.0f, 600.0f, 0.0f, 1.0f,
81      120.0f, 600.0f, 0.0f, 1.0f,
82
83      140.0f, 600.0f, 0.0f, 1.0f,
84      120.0f, 600.0f, 0.0f, 1.0f,
85      140.0f, 670.0f, 0.0f, 1.0f,
86      160.0f, 670.0f, 0.0f, 1.0f,
87
88      140.0f, 670.0f, 0.0f, 1.0f,
89      160.0f, 670.0f, 0.0f, 1.0f,
90      180.0f, 600.0f, 0.0f, 1.0f,
91      160.0f, 600.0f, 0.0f, 1.0f,
92
93      180.0f, 600.0f, 0.0f, 1.0f,
94      160.0f, 600.0f, 0.0f, 1.0f,
95      180.0f, 700.0f, 0.0f, 1.0f,
96      200.0f, 700.0f, 0.0f, 1.0f,
97
98      // WORD "I"
99      220.0f, 700.0f, 0.0f, 1.0f,
100     240.0f, 700.0f, 0.0f, 1.0f,
101     240.0f, 600.0f, 0.0f, 1.0f,
102     220.0f, 600.0f, 0.0f, 1.0f,
103
104     // WORD "N"
105     260.0f, 700.0f, 0.0f, 1.0f,
106     280.0f, 700.0f, 0.0f, 1.0f,
107     280.0f, 600.0f, 0.0f, 1.0f,
108     260.0f, 600.0f, 0.0f, 1.0f,
109
110     310.0f, 700.0f, 0.0f, 1.0f,
111     330.0f, 700.0f, 0.0f, 1.0f,
112     330.0f, 600.0f, 0.0f, 1.0f,
113     310.0f, 600.0f, 0.0f, 1.0f,
114

```

```

115     280.0f, 700.0f, 0.0f, 1.0f,
116     280.0f, 670.0f, 0.0f, 1.0f,
117     310.0f, 600.0f, 0.0f, 1.0f,
118     310.0f, 630.0f, 0.0f, 1.0f,

```

Figura 4.1.1: Cod OpenGL pentru literele ascunse

Această etapă a fost realizată de **Popescu Paullo Robertto Karloss**.

## 4.2 Prezentare după translație

După ce dreptunghiul albastru reușește să treacă linia de "FINISH", sunt translatate literele în zona de sus a tablei de joc.

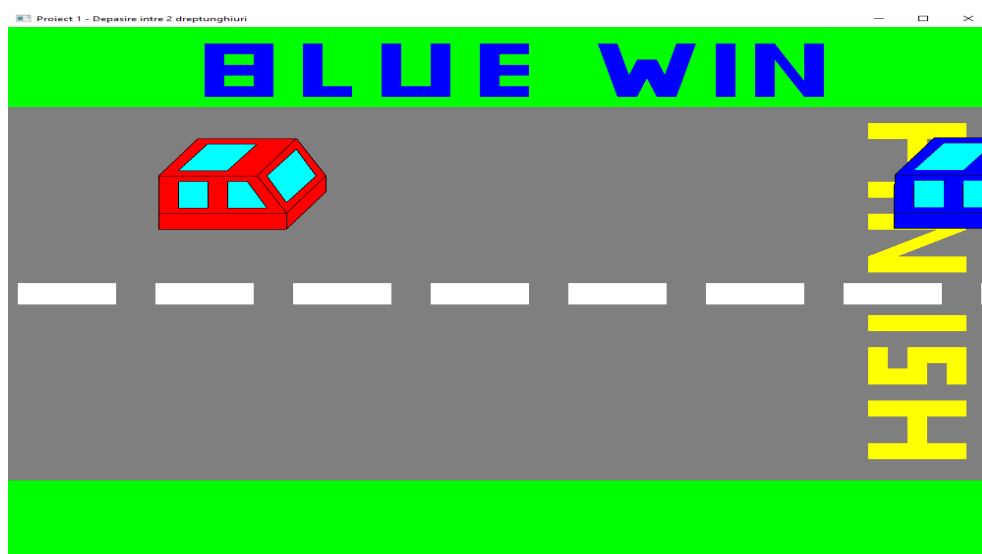


Figura 4.1.1: Afișarea câștigătorului

### 4.2.1 Cod sursă

```

1  if (j >= 750 && h >= -230) {
2      h = h - alpha4;
3      alpha4 = +step3;
4  }

```

Figura 4.1.1: Cod OpenGL pentru translația finală

Această etapă a fost realizată de **Linte Robert Ovidiu**.

# Codul Sursă Complet

Codul îl puteți găsi în fișierul **proiectGrafica.cpp** sau atașat mai jos.

```
1  /* DESCRIERE: DREPTUNGHI CU SATELIT -- varianta cu OpenGL "nou"
2     - utilizeaza diverse transformari si compunerea acestora folosind
        biblioteca glm
3     - functii pentru utilizarea mouse-ului
4  */
5
6  #include <windows.h> // biblioteci care urmeaza sa fie incluse
7  #include <stdlib.h> // necesare pentru citirea shader-elor
8  #include <stdio.h>
9  #include <math.h>
10 #include <iostream>
11 #include <GL/glew.h> // glew apare inainte de freeglut
12 #include <GL/freeglut.h> // nu trebuie uitat freeglut.h
13 #include "loadShaders.h"
14
15 // Din biblioteca glm
16 #include "glm/glm.hpp"
17 #include "glm/gtc/matrix_transform.hpp"
18 #include "glm/gtx/transform.hpp"
19 #include "glm/gtc/type_ptr.hpp"
20
21 using namespace std;
22
23 GLuint
24 VaoId,
25 VboId,
26 ColorBufferId,
27 ProgramId,
28 myMatrixLocation,
29 matrScaleLocation,
30 matrTranslLocation,
31 matrRotlLocation,
32 codColLocation;
33
34 int codCol;
35 float PI = 3.141592, angle = 0;
36 float tx = 0; float ty = 0;
```

```

37 float width = 500, height = 500;
38 float i = 0.0, j = 0.0, h = 0.0, k = 0.0, l = 0.0, alpha4 = 0.0,
    alpha2 = 0.0, alpha3 = 0.0, step2 = 0.5, step3 = 1.5, alpha =
    0.0, step = 0.1, beta = 0.003, ok = 0;
39 glm::mat4
40 myMatrix, resizeMatrix, matrTransl, matrTransl2, matrTransl3,
    matrScale1, matrScale2, matrRot, matrDepl;
41
42 void displayMatrix()
43 {
44     for (int ii = 0; ii < 4; ii++)
45     {
46         for (int jj = 0; jj < 4; jj++)
47             cout << myMatrix[ii][jj] << " ";
48         cout << endl;
49     };
50     cout << "\n";
51 };
52
53 void miscad(void)
54 {
55     if (i > -1 && i <= 400 && j <= 750)
56     {
57         i = i + alpha;
58         alpha = +step;
59     }
60     if (j > -1.0 && j <= 150) {
61         j = j + alpha2;
62         alpha2 = +step2;
63     }
64
65     if (j >= 150 && j <= 250 && angle <= 0.5) {
66         angle += beta;
67         l = l + alpha3;
68         alpha3 = +step3;
69     }
70
71     if (l >= 200 && l <= 370 && angle >= 0.5) {
72         j = j + alpha2;
73         alpha2 = +step2;
74         l = l + alpha3;

```

```

75     alpha3 = +step2;
76 }
77 if (l >= 370 && angle <= 0.6 && angle >= 0) {
78     angle -= beta;
79     j = j + alpha2;
80     alpha2 = +step3;
81 }
82 if (l >= 370 && j <= 850 && angle <= 0) {
83     j = j + alpha2;
84     alpha2 = +step3;
85 }
86 if (j >= 750 && h >= -230) {
87     h = h - alpha4;
88     alpha4 = +step3;
89 }
90
91 glutPostRedisplay();
92 }
93
94
95
96
97 void mouse(int button, int state, int x, int y)
98 {
99     switch (button) {
100     case GLUT_LEFT_BUTTON:
101         if (state == GLUT_DOWN) {
102             alpha = -step;
103             alpha2 = -step2;
104         }
105         glutIdleFunc(miscad);
106         break;
107     case GLUT_RIGHT_BUTTON:
108         if (state == GLUT_DOWN) {
109             alpha = step;
110             alpha2 = step2;
111         }
112         glutIdleFunc(miscad);
113         break;
114     default:
115         break;

```

```

116     }
117 }
118
119 void CreateVBO(void)
120 {
121     // varfurile
122     GLfloat Vertices[] = {
123         // varfuri pentru axe
124         -500.0f, 500.0f, 0.0f, 1.0f,
125         500.0f, 500.0f, 0.0f, 1.0f,
126         500.0f, -500.0f, 0.0f, 1.0f,
127         -500.0f, -500.0f, 0.0f, 1.0f,
128
129         -500.0f, 350.0f, 0.0f, 1.0f,
130         500.0f, 350.0f, 0.0f, 1.0f,
131         500.0f, -350.0f, 0.0f, 1.0f,
132         -500.0f, -350.0f, 0.0f, 1.0f,
133
134         -490.0f, 20.0f, 0.0f, 1.0f,
135         -390.0f, 20.0f, 0.0f, 1.0f,
136         -390.0f, -20.0f, 0.0f, 1.0f,
137         -490.0f, -20.0f, 0.0f, 1.0f,
138
139         -350.0f, 20.0f, 0.0f, 1.0f,
140         -250.0f, 20.0f, 0.0f, 1.0f,
141         -250.0f, -20.0f, 0.0f, 1.0f,
142         -350.0f, -20.0f, 0.0f, 1.0f,
143
144         -210.0f, 20.0f, 0.0f, 1.0f,
145         -110.0f, 20.0f, 0.0f, 1.0f,
146         -110.0f, -20.0f, 0.0f, 1.0f,
147         -210.0f, -20.0f, 0.0f, 1.0f,
148
149         -70.0f, 20.0f, 0.0f, 1.0f,
150         30.0f, 20.0f, 0.0f, 1.0f,
151         30.0f, -20.0f, 0.0f, 1.0f,
152         -70.0f, -20.0f, 0.0f, 1.0f,
153
154         70.0f, 20.0f, 0.0f, 1.0f,
155         170.0f, 20.0f, 0.0f, 1.0f,
156         170.0f, -20.0f, 0.0f, 1.0f,

```

```

157 70.0f, -20.0f, 0.0f, 1.0f,
158
159 210.0f, 20.0f, 0.0f, 1.0f,
160 310.0f, 20.0f, 0.0f, 1.0f,
161 310.0f, -20.0f, 0.0f, 1.0f,
162 210.0f, -20.0f, 0.0f, 1.0f,
163
164 350.0f, 20.0f, 0.0f, 1.0f,
165 450.0f, 20.0f, 0.0f, 1.0f,
166 450.0f, -20.0f, 0.0f, 1.0f,
167 350.0f, -20.0f, 0.0f, 1.0f,
168
169 490.0f, 20.0f, 0.0f, 1.0f,
170 500.0f, 20.0f, 0.0f, 1.0f,
171 500.0f, -20.0f, 0.0f, 1.0f,
172 490.0f, -20.0f, 0.0f, 1.0f,
173
174 -450.0f, 250.0f, 0.0f, 1.0f,
175 -300.0f, 250.0f, 0.0f, 1.0f,
176 -300.0f, 120.0f, 0.0f, 1.0f,
177 -450.0f, 120.0f, 0.0f, 1.0f,
178
179 -450.0f, -250.0f, 0.0f, 1.0f,
180 -300.0f, -250.0f, 0.0f, 1.0f,
181 -300.0f, -120.0f, 0.0f, 1.0f,
182 -450.0f, -120.0f, 0.0f, 1.0f,
183
184 // - WRITING TEXT: "FINISH" - //
185
186 // WORD "F"
187 475.0f, 320.0f, 0.0f, 1.0f,
188 375.0f, 320.0f, 0.0f, 1.0f,
189 375.0f, 290.0f, 0.0f, 1.0f,
190 475.0f, 290.0f, 0.0f, 1.0f,
191
192 475.0f, 290.0f, 0.0f, 1.0f,
193 475.0f, 240.0f, 0.0f, 1.0f,
194 455.0f, 240.0f, 0.0f, 1.0f,
195 455.0f, 290.0f, 0.0f, 1.0f,
196
197 435.0f, 290.0f, 0.0f, 1.0f,

```



```

198 435.0f, 240.0f, 0.0f, 1.0f,
199 415.0f, 240.0f, 0.0f, 1.0f,
200 415.0f, 290.0f, 0.0f, 1.0f,
201
202 // WORD "I"
203 475.0f, 210.0f, 0.0f, 1.0f,
204 375.0f, 210.0f, 0.0f, 1.0f,
205 375.0f, 180.0f, 0.0f, 1.0f,
206 475.0f, 180.0f, 0.0f, 1.0f,
207
208 //WORD "N"
209 475.0f, 150.0f, 0.0f, 1.0f,
210 375.0f, 150.0f, 0.0f, 1.0f,
211 375.0f, 120.0f, 0.0f, 1.0f,
212 475.0f, 120.0f, 0.0f, 1.0f,
213
214 475.0f, 70.0f, 0.0f, 1.0f,
215 375.0f, 70.0f, 0.0f, 1.0f,
216 375.0f, 40.0f, 0.0f, 1.0f,
217 475.0f, 40.0f, 0.0f, 1.0f,
218
219 475.0f, 120.0f, 0.0f, 1.0f,
220 445.0f, 120.0f, 0.0f, 1.0f,
221 375.0f, 70.0f, 0.0f, 1.0f,
222 405.0f, 70.0f, 0.0f, 1.0f,
223
224 // WORD "I"
225 475.0f, -40.0f, 0.0f, 1.0f,
226 375.0f, -40.0f, 0.0f, 1.0f,
227 375.0f, -70.0f, 0.0f, 1.0f,
228 475.0f, -70.0f, 0.0f, 1.0f,
229
230 // WORD "S"
231 475.0f, -100.0f, 0.0f, 1.0f,
232 455.0f, -100.0f, 0.0f, 1.0f,
233 455.0f, -170.0f, 0.0f, 1.0f,
234 475.0f, -170.0f, 0.0f, 1.0f,
235
236 475.0f, -100.0f, 0.0f, 1.0f,
237 415.0f, -100.0f, 0.0f, 1.0f,
238 415.0f, -130.0f, 0.0f, 1.0f,

```

```

239 475.0f, -130.0f, 0.0f, 1.0f,
240
241 415.0f, -100.0f, 0.0f, 1.0f,
242 435.0f, -100.0f, 0.0f, 1.0f,
243 435.0f, -170.0f, 0.0f, 1.0f,
244 415.0f, -170.0f, 0.0f, 1.0f,
245
246 415.0f, -170.0f, 0.0f, 1.0f,
247 415.0f, -140.0f, 0.0f, 1.0f,
248 375.0f, -140.0f, 0.0f, 1.0f,
249 375.0f, -170.0f, 0.0f, 1.0f,
250
251 375.0f, -100.0f, 0.0f, 1.0f,
252 395.0f, -100.0f, 0.0f, 1.0f,
253 395.0f, -170.0f, 0.0f, 1.0f,
254 375.0f, -170.0f, 0.0f, 1.0f,
255
256 // WORD "H"
257 475.0f, -200.0f, 0.0f, 1.0f,
258 375.0f, -200.0f, 0.0f, 1.0f,
259 375.0f, -230.0f, 0.0f, 1.0f,
260 475.0f, -230.0f, 0.0f, 1.0f,
261
262 475.0f, -280.0f, 0.0f, 1.0f,
263 375.0f, -280.0f, 0.0f, 1.0f,
264 375.0f, -310.0f, 0.0f, 1.0f,
265 475.0f, -310.0f, 0.0f, 1.0f,
266
267 435.0f, -310.0f, 0.0f, 1.0f,
268 415.0f, -310.0f, 0.0f, 1.0f,
269 415.0f, -200.0f, 0.0f, 1.0f,
270 435.0f, -200.0f, 0.0f, 1.0f,
271
272 // - WRITING TEXT: "BLUE WIN" - //
273
274 // WORD "B"
275 -300.0f, 700.0f, 0.0f, 1.0f,
276 -300.0f, 600.0f, 0.0f, 1.0f,
277 -280.0f, 600.f, 0.0f, 1.0f,
278 -280.0f, 700.0f, 0.0f, 1.0f,
279

```

```

280 -250.0f, 700.0f, 0.0f, 1.0f,
281 -250.0f, 600.0f, 0.0f, 1.0f,
282 -230.0f, 600.f, 0.0f, 1.0f,
283 -230.0f, 700.0f, 0.0f, 1.0f,
284
285 -300.0f, 700.0f, 0.0f, 1.0f,
286 -300.0f, 680.0f, 0.0f, 1.0f,
287 -230.0f, 680.f, 0.0f, 1.0f,
288 -230.0f, 700.0f, 0.0f, 1.0f,
289
290 -300.0f, 660.0f, 0.0f, 1.0f,
291 -300.0f, 640.0f, 0.0f, 1.0f,
292 -230.0f, 640.f, 0.0f, 1.0f,
293 -230.0f, 660.0f, 0.0f, 1.0f,
294
295 -300.0f, 620.0f, 0.0f, 1.0f,
296 -300.0f, 600.0f, 0.0f, 1.0f,
297 -230.0f, 600.f, 0.0f, 1.0f,
298 -230.0f, 620.0f, 0.0f, 1.0f,
299
300 // WORD "L"
301 -200.0f, 700.0f, 0.0f, 1.0f,
302 -200.0f, 600.0f, 0.0f, 1.0f,
303 -180.0f, 600.f, 0.0f, 1.0f,
304 -180.0f, 700.0f, 0.0f, 1.0f,
305
306 -200.0f, 620.0f, 0.0f, 1.0f,
307 -200.0f, 600.0f, 0.0f, 1.0f,
308 -150.0f, 600.f, 0.0f, 1.0f,
309 -150.0f, 620.0f, 0.0f, 1.0f,
310
311 // WORD "U"
312 -120.0f, 700.0f, 0.0f, 1.0f,
313 -120.0f, 600.0f, 0.0f, 1.0f,
314 -100.0f, 600.f, 0.0f, 1.0f,
315 -100.0f, 700.0f, 0.0f, 1.0f,
316
317 -70.0f, 700.0f, 0.0f, 1.0f,
318 -70.0f, 600.0f, 0.0f, 1.0f,
319 -50.0f, 600.f, 0.0f, 1.0f,
320 -50.0f, 700.0f, 0.0f, 1.0f,

```

```

321
322     -120.0f, 620.0f, 0.0f, 1.0f,
323     -50.0f, 620.0f, 0.0f, 1.0f,
324     -50.0f, 600.0f, 0.0f, 1.0f,
325     -120.0f, 600.0f, 0.0f, 1.0f,
326
327     // WORD "E"
328     -20.0f, 700.0f, 0.0f, 1.0f,
329     -20.0f, 600.0f, 0.0f, 1.0f,
330     0.0f, 600.0f, 0.0f, 1.0f,
331     0.0f, 700.0f, 0.0f, 1.0f,
332
333     -20.0f, 620.0f, 0.0f, 1.0f,
334     30.0f, 620.0f, 0.0f, 1.0f,
335     30.0f, 600.0f, 0.0f, 1.0f,
336     -20.0f, 600.0f, 0.0f, 1.0f,
337
338     -20.0f, 660.0f, 0.0f, 1.0f,
339     30.0f, 660.0f, 0.0f, 1.0f,
340     30.0f, 640.0f, 0.0f, 1.0f,
341     -20.0f, 640.0f, 0.0f, 1.0f,
342
343     -20.0f, 700.0f, 0.0f, 1.0f,
344     30.0f, 700.0f, 0.0f, 1.0f,
345     30.0f, 680.0f, 0.0f, 1.0f,
346     -20.0f, 680.0f, 0.0f, 1.0f,
347
348     // WORD "W"
349     100.0f, 700.0f, 0.0f, 1.0f,
350     120.0f, 700.0f, 0.0f, 1.0f,
351     140.0f, 600.0f, 0.0f, 1.0f,
352     120.0f, 600.0f, 0.0f, 1.0f,
353
354     140.0f, 600.0f, 0.0f, 1.0f,
355     120.0f, 600.0f, 0.0f, 1.0f,
356     140.0f, 670.0f, 0.0f, 1.0f,
357     160.0f, 670.0f, 0.0f, 1.0f,
358
359     140.0f, 670.0f, 0.0f, 1.0f,
360     160.0f, 670.0f, 0.0f, 1.0f,
361     180.0f, 600.0f, 0.0f, 1.0f,

```

```

362     160.0f, 600.0f, 0.0f, 1.0f,
363
364     180.0f, 600.0f, 0.0f, 1.0f,
365     160.0f, 600.0f, 0.0f, 1.0f,
366     180.0f, 700.0f, 0.0f, 1.0f,
367     200.0f, 700.0f, 0.0f, 1.0f,
368
369     // WORD "I"
370     220.0f, 700.0f, 0.0f, 1.0f,
371     240.0f, 700.0f, 0.0f, 1.0f,
372     240.0f, 600.0f, 0.0f, 1.0f,
373     220.0f, 600.0f, 0.0f, 1.0f,
374
375     // WORD "N"
376     260.0f, 700.0f, 0.0f, 1.0f,
377     280.0f, 700.0f, 0.0f, 1.0f,
378     280.0f, 600.0f, 0.0f, 1.0f,
379     260.0f, 600.0f, 0.0f, 1.0f,
380
381     310.0f, 700.0f, 0.0f, 1.0f,
382     330.0f, 700.0f, 0.0f, 1.0f,
383     330.0f, 600.0f, 0.0f, 1.0f,
384     310.0f, 600.0f, 0.0f, 1.0f,
385
386     280.0f, 700.0f, 0.0f, 1.0f,
387     280.0f, 670.0f, 0.0f, 1.0f,
388     310.0f, 600.0f, 0.0f, 1.0f,
389     310.0f, 630.0f, 0.0f, 1.0f,
390
391     // RED CAR
392     -450.0f, 120.0f, 0.0f, 1.0f,
393     -320.0f, 120.0f, 0.0f, 1.0f,
394     -320.0f, 150.0f, 0.0f, 1.0f,
395     -450.0f, 150.0f, 0.0f, 1.0f,
396     -450.0f, 120.0f, 0.0f, 1.0f,
397
398     -450.0f, 150.0f, 0.0f, 1.0f,
399     -320.0f, 150.0f, 0.0f, 1.0f,
400     -350.0f, 220.0f, 0.0f, 1.0f,
401     -450.0f, 220.0f, 0.0f, 1.0f,
402     -450.0f, 150.0f, 0.0f, 1.0f,

```

403	
404	-320.0f, 120.0f, 0.0f, 1.0f,
405	-320.0f, 150.0f, 0.0f, 1.0f,
406	-280.0f, 220.0f, 0.0f, 1.0f,
407	-280.0f, 190.0f, 0.0f, 1.0f,
408	-320.0f, 120.0f, 0.0f, 1.0f,
409	
410	-350.0f, 220.0f, 0.0f, 1.0f,
411	-320.0f, 150.0f, 0.0f, 1.0f,
412	-280.0f, 220.0f, 0.0f, 1.0f,
413	-310.0f, 290.0f, 0.0f, 1.0f,
414	-350.0f, 220.0f, 0.0f, 1.0f,
415	
416	-310.0f, 290.0f, 0.0f, 1.0f,
417	-350.0f, 220.0f, 0.0f, 1.0f,
418	-450.0f, 220.0f, 0.0f, 1.0f,
419	-410.0f, 290.0f, 0.0f, 1.0f,
420	-310.0f, 290.0f, 0.0f, 1.0f,
421	
422	-310.0f, 290.0f, 0.0f, 1.0f,
423	-350.0f, 220.0f, 0.0f, 1.0f,
424	-450.0f, 220.0f, 0.0f, 1.0f,
425	-410.0f, 290.0f, 0.0f, 1.0f,
426	-310.0f, 290.0f, 0.0f, 1.0f,
427	
428	-430.0f, 230.0f, 0.0f, 1.0f,
429	-400.0f, 280.0f, 0.0f, 1.0f,
430	-350.0f, 280.0f, 0.0f, 1.0f,
431	-380.0f, 230.0f, 0.0f, 1.0f,
432	-430.0f, 230.0f, 0.0f, 1.0f,
433	
434	-320.0f, 170.0f, 0.0f, 1.0f,
435	-340.0f, 220.0f, 0.0f, 1.0f,
436	-310.0f, 270.0f, 0.0f, 1.0f,
437	-290.0f, 220.0f, 0.0f, 1.0f,
438	-320.0f, 170.0f, 0.0f, 1.0f,
439	
440	-380.0f, 210.0f, 0.0f, 1.0f,
441	-380.0f, 160.0f, 0.0f, 1.0f,
442	-340.0f, 160.0f, 0.0f, 1.0f,
443	-360.0f, 210.0f, 0.0f, 1.0f,

```

444     -380.0f, 210.0f, 0.0f, 1.0f,
445
446     -400.0f, 160.0f, 0.0f, 1.0f,
447     -430.0f, 160.0f, 0.0f, 1.0f,
448     -430.0f, 210.0f, 0.0f, 1.0f,
449     -400.0f, 210.0f, 0.0f, 1.0f,
450     -400.0f, 160.0f, 0.0f, 1.0f,
451
452     // BLUE CAR
453     -450.0f, -250.0f, 0.0f, 1.0f,
454     -320.0f, -250.0f, 0.0f, 1.0f,
455     -320.0f, -220.0f, 0.0f, 1.0f,
456     -450.0f, -220.0f, 0.0f, 1.0f,
457     -450.0f, -250.0f, 0.0f, 1.0f,
458
459     -450.0f, -220.0f, 0.0f, 1.0f,
460     -320.0f, -220.0f, 0.0f, 1.0f,
461     -350.0f, -150.0f, 0.0f, 1.0f,
462     -450.0f, -150.0f, 0.0f, 1.0f,
463     -450.0f, -220.0f, 0.0f, 1.0f,
464
465     -320.0f, -250.0f, 0.0f, 1.0f,
466     -320.0f, -220.0f, 0.0f, 1.0f,
467     -280.0f, -150.0f, 0.0f, 1.0f,
468     -280.0f, -180.0f, 0.0f, 1.0f,
469     -320.0f, -250.0f, 0.0f, 1.0f,
470
471     -350.0f, -150.0f, 0.0f, 1.0f,
472     -320.0f, -220.0f, 0.0f, 1.0f,
473     -280.0f, -150.0f, 0.0f, 1.0f,
474     -310.0f, -80.0f, 0.0f, 1.0f,
475     -350.0f, -150.0f, 0.0f, 1.0f,
476
477     -310.0f, -80.0f, 0.0f, 1.0f,
478     -350.0f, -150.0f, 0.0f, 1.0f,
479     -450.0f, -150.0f, 0.0f, 1.0f,
480     -410.0f, -80.0f, 0.0f, 1.0f,
481     -310.0f, -80.0f, 0.0f, 1.0f,
482
483     -430.0f, -140.0f, 0.0f, 1.0f,
484     -400.0f, -90.0f, 0.0f, 1.0f,

```

```

485     -350.0f, -90.0f, 0.0f, 1.0f,
486     -380.0f, -140.0f, 0.0f, 1.0f,
487     -430.0f, -140.0f, 0.0f, 1.0f,
488
489     -320.0f, -200.0f, 0.0f, 1.0f,
490     -340.0f, -150.0f, 0.0f, 1.0f,
491     -310.0f, -100.0f, 0.0f, 1.0f,
492     -290.0f, -150.0f, 0.0f, 1.0f,
493     -320.0f, -200.0f, 0.0f, 1.0f,
494
495     -380.0f, -160.0f, 0.0f, 1.0f,
496     -380.0f, -210.0f, 0.0f, 1.0f,
497     -340.0f, -210.0f, 0.0f, 1.0f,
498     -360.0f, -160.0f, 0.0f, 1.0f,
499     -380.0f, -210.0f, 0.0f, 1.0f,
500
501     -400.0f, -210.0f, 0.0f, 1.0f,
502     -430.0f, -210.0f, 0.0f, 1.0f,
503     -430.0f, -160.0f, 0.0f, 1.0f,
504     -400.0f, -160.0f, 0.0f, 1.0f,
505     -400.0f, -210.0f, 0.0f, 1.0f,
506 };
507
508 // culorile varfurilor din colturi
509 GLfloat Colors[] = {
510     0.0f, 1.0f, 0.0f, 1.0f,
511     0.0f, 1.0f, 0.0f, 1.0f,
512     0.0f, 1.0f, 0.0f, 1.0f,
513     0.0f, 1.0f, 0.0f, 1.0f,
514
515     0.5f, 0.5f, 0.5f, 1.0f,
516     0.5f, 0.5f, 0.5f, 1.0f,
517     0.5f, 0.5f, 0.5f, 1.0f,
518     0.5f, 0.5f, 0.5f, 1.0f,
519
520     1.0f, 1.0f, 1.0f, 1.0f,
521     1.0f, 1.0f, 1.0f, 1.0f,
522     1.0f, 1.0f, 1.0f, 1.0f,
523     1.0f, 1.0f, 1.0f, 1.0f,
524
525     1.0f, 1.0f, 1.0f, 1.0f,

```



526	1.0f, 1.0f, 1.0f, 1.0f,
527	1.0f, 1.0f, 1.0f, 1.0f,
528	1.0f, 1.0f, 1.0f, 1.0f,
529	
530	1.0f, 1.0f, 1.0f, 1.0f,
531	1.0f, 1.0f, 1.0f, 1.0f,
532	1.0f, 1.0f, 1.0f, 1.0f,
533	1.0f, 1.0f, 1.0f, 1.0f,
534	
535	1.0f, 1.0f, 1.0f, 1.0f,
536	1.0f, 1.0f, 1.0f, 1.0f,
537	1.0f, 1.0f, 1.0f, 1.0f,
538	1.0f, 1.0f, 1.0f, 1.0f,
539	
540	1.0f, 1.0f, 1.0f, 1.0f,
541	1.0f, 1.0f, 1.0f, 1.0f,
542	1.0f, 1.0f, 1.0f, 1.0f,
543	1.0f, 1.0f, 1.0f, 1.0f,
544	
545	1.0f, 1.0f, 1.0f, 1.0f,
546	1.0f, 1.0f, 1.0f, 1.0f,
547	1.0f, 1.0f, 1.0f, 1.0f,
548	1.0f, 1.0f, 1.0f, 1.0f,
549	
550	1.0f, 1.0f, 1.0f, 1.0f,
551	1.0f, 1.0f, 1.0f, 1.0f,
552	1.0f, 1.0f, 1.0f, 1.0f,
553	1.0f, 1.0f, 1.0f, 1.0f,
554	
555	1.0f, 1.0f, 1.0f, 1.0f,
556	1.0f, 1.0f, 1.0f, 1.0f,
557	1.0f, 1.0f, 1.0f, 1.0f,
558	1.0f, 1.0f, 1.0f, 1.0f,
559	
560	1.0f, 0.0f, 0.0f, 1.0f,
561	1.0f, 0.0f, 0.0f, 1.0f,
562	1.0f, 0.0f, 0.0f, 1.0f,
563	1.0f, 0.0f, 0.0f, 1.0f,
564	
565	0.0f, 0.0f, 1.0f, 1.0f,
566	0.0f, 0.0f, 1.0f, 1.0f,

567	0.0f, 0.0f, 1.0f, 1.0f,
568	0.0f, 0.0f, 1.0f, 1.0f,
569	
570	1.0f, 1.0f, 0.0f, 1.0f,
571	1.0f, 1.0f, 0.0f, 1.0f,
572	1.0f, 1.0f, 0.0f, 1.0f,
573	1.0f, 1.0f, 0.0f, 1.0f,
574	
575	1.0f, 1.0f, 0.0f, 1.0f,
576	1.0f, 1.0f, 0.0f, 1.0f,
577	1.0f, 1.0f, 0.0f, 1.0f,
578	1.0f, 1.0f, 0.0f, 1.0f,
579	
580	1.0f, 1.0f, 0.0f, 1.0f,
581	1.0f, 1.0f, 0.0f, 1.0f,
582	1.0f, 1.0f, 0.0f, 1.0f,
583	1.0f, 1.0f, 0.0f, 1.0f,
584	
585	1.0f, 1.0f, 0.0f, 1.0f,
586	1.0f, 1.0f, 0.0f, 1.0f,
587	1.0f, 1.0f, 0.0f, 1.0f,
588	1.0f, 1.0f, 0.0f, 1.0f,
589	
590	1.0f, 1.0f, 0.0f, 1.0f,
591	1.0f, 1.0f, 0.0f, 1.0f,
592	1.0f, 1.0f, 0.0f, 1.0f,
593	1.0f, 1.0f, 0.0f, 1.0f,
594	
595	1.0f, 1.0f, 0.0f, 1.0f,
596	1.0f, 1.0f, 0.0f, 1.0f,
597	1.0f, 1.0f, 0.0f, 1.0f,
598	1.0f, 1.0f, 0.0f, 1.0f,
599	
600	1.0f, 1.0f, 0.0f, 1.0f,
601	1.0f, 1.0f, 0.0f, 1.0f,
602	1.0f, 1.0f, 0.0f, 1.0f,
603	1.0f, 1.0f, 0.0f, 1.0f,
604	
605	1.0f, 1.0f, 0.0f, 1.0f,
606	1.0f, 1.0f, 0.0f, 1.0f,
607	1.0f, 1.0f, 0.0f, 1.0f,

608	1.0f, 1.0f, 0.0f, 1.0f,
609	
610	1.0f, 1.0f, 0.0f, 1.0f,
611	1.0f, 1.0f, 0.0f, 1.0f,
612	1.0f, 1.0f, 0.0f, 1.0f,
613	1.0f, 1.0f, 0.0f, 1.0f,
614	
615	1.0f, 1.0f, 0.0f, 1.0f,
616	1.0f, 1.0f, 0.0f, 1.0f,
617	1.0f, 1.0f, 0.0f, 1.0f,
618	1.0f, 1.0f, 0.0f, 1.0f,
619	
620	1.0f, 1.0f, 0.0f, 1.0f,
621	1.0f, 1.0f, 0.0f, 1.0f,
622	1.0f, 1.0f, 0.0f, 1.0f,
623	1.0f, 1.0f, 0.0f, 1.0f,
624	
625	1.0f, 1.0f, 0.0f, 1.0f,
626	1.0f, 1.0f, 0.0f, 1.0f,
627	1.0f, 1.0f, 0.0f, 1.0f,
628	1.0f, 1.0f, 0.0f, 1.0f,
629	
630	1.0f, 1.0f, 0.0f, 1.0f,
631	1.0f, 1.0f, 0.0f, 1.0f,
632	1.0f, 1.0f, 0.0f, 1.0f,
633	1.0f, 1.0f, 0.0f, 1.0f,
634	
635	1.0f, 1.0f, 0.0f, 1.0f,
636	1.0f, 1.0f, 0.0f, 1.0f,
637	1.0f, 1.0f, 0.0f, 1.0f,
638	1.0f, 1.0f, 0.0f, 1.0f,
639	
640	1.0f, 1.0f, 0.0f, 1.0f,
641	1.0f, 1.0f, 0.0f, 1.0f,
642	1.0f, 1.0f, 0.0f, 1.0f,
643	1.0f, 1.0f, 0.0f, 1.0f,
644	
645	1.0f, 1.0f, 0.0f, 1.0f,
646	1.0f, 1.0f, 0.0f, 1.0f,
647	1.0f, 1.0f, 0.0f, 1.0f,
648	1.0f, 1.0f, 0.0f, 1.0f,

```

649     };
650
651     // se creeaza un buffer nou
652     glGenBuffers(1, &VboId);
653     // este setat ca buffer curent
654     glBindBuffer(GL_ARRAY_BUFFER, VboId);
655     // punctele sunt "copiate" in bufferul curent
656     glBufferData(GL_ARRAY_BUFFER, sizeof(Vertices), Vertices,
        GL_STATIC_DRAW);
657
658     // se creeaza / se leaga un VAO (Vertex Array Object) - util cand
        se utilizeaza mai multe VBO
659     glGenVertexArrays(1, &VaoId);
660     glBindVertexArray(VaoId);
661     // se activeaza lucrul cu attribute; atributul 0 = pozitie
662     glEnableVertexAttribArray(0);
663     glVertexAttribPointer(0, 4, GL_FLOAT, GL_FALSE, 0, 0);
664
665     // un nou buffer, pentru culoare
666     glGenBuffers(1, &ColorBufferId);
667     glBindBuffer(GL_ARRAY_BUFFER, ColorBufferId);
668     glBufferData(GL_ARRAY_BUFFER, sizeof(Colors), Colors,
        GL_STATIC_DRAW);
669     // atributul 1 = culoare
670     glEnableVertexAttribArray(1);
671     glVertexAttribPointer(1, 4, GL_FLOAT, GL_FALSE, 0, 0);
672 }
673 void DestroyVBO(void)
674 {
675     glDisableVertexAttribArray(1);
676     glDisableVertexAttribArray(0);
677     glBindBuffer(GL_ARRAY_BUFFER, 0);
678     glDeleteBuffers(1, &ColorBufferId);
679     glDeleteBuffers(1, &VboId);
680     glBindVertexArray(0);
681     glDeleteVertexArrays(1, &VaoId);
682 }
683
684 void CreateShaders(void)
685 {
686     ProgramId = LoadShaders("03_02_Shader.vert", "03_02_Shader.frag");

```

```

687     glUseProgram(ProgramId);
688 }
689 void DestroyShaders(void)
690 {
691     glDeleteProgram(ProgramId);
692 }
693
694 void Initialize(void)
695 {
696     glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // culoarea de fond a
        ecranului
697     CreateVBO();
698     CreateShaders();
699     codColLocation = glGetUniformLocation(ProgramId, "codCuloare");
700     myMatrixLocation = glGetUniformLocation(ProgramId, "myMatrix");
701 }
702 void RenderFunction(void)
703 {
704     glClear(GL_COLOR_BUFFER_BIT);
705
706     // TO DO: schimbati transformarile (de exemplu deplasarea are loc
        pe axa Oy sau pe o alta dreapta)
707     resizeMatrix = glm::ortho(-width, width, -height, height); //
        scalam, "aducem" scena la "patratul standard" [-1,1]x[-1,1]
708     matrTransl = glm::translate(glm::mat4(1.0f), glm::vec3(i, k, 0.0))
        ; // controleaza translatia de-a lungul lui Ox
709     matrTransl3 = glm::translate(glm::mat4(1.0f), glm::vec3(0.0, h,
        0.0));
710     matrDepl = glm::translate(glm::mat4(1.0f), glm::vec3(1.0, 1.0,
        0.0)); // plaseaza patratul rosu
711     matrScale2 = glm::scale(glm::mat4(1.0f), glm::vec3(1.0, 1.0, 0.0))
        ; // folosita la desenarea patratului rosu
712     matrTransl2 = glm::translate(glm::mat4(1.0f), glm::vec3(j, 1, 0.0)
        );
713     matrRot = glm::rotate(glm::mat4(1.0f), angle, glm::vec3(0.0, 0.0,
        1.0)); // rotatie folosita la deplasarea patratului rosu
714
715     // Matricea de redimensionare (pentru elementele "fixe")
716     myMatrix = resizeMatrix;
717     // Culoarea
718     codCol = 0;

```

```

719 // Transmite variabile uniforme
720 glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
    ;
721 glUniform1i(codColLocation, codCol);
722
723 glDrawArrays(GL_POLYGON, 0, 4);
724 glDrawArrays(GL_POLYGON, 4, 4);
725 glDrawArrays(GL_POLYGON, 8, 4);
726 glDrawArrays(GL_POLYGON, 12, 4);
727 glDrawArrays(GL_POLYGON, 16, 4);
728 glDrawArrays(GL_POLYGON, 20, 4);
729 glDrawArrays(GL_POLYGON, 24, 4);
730 glDrawArrays(GL_POLYGON, 28, 4);
731 glDrawArrays(GL_POLYGON, 32, 4);
732 glDrawArrays(GL_POLYGON, 36, 4);
733 glDrawArrays(GL_POLYGON, 40, 4);
734 glDrawArrays(GL_POLYGON, 44, 4);
735 glDrawArrays(GL_POLYGON, 48, 4);
736 glDrawArrays(GL_POLYGON, 52, 4);
737 glDrawArrays(GL_POLYGON, 56, 4);
738 glDrawArrays(GL_POLYGON, 60, 4);
739 glDrawArrays(GL_POLYGON, 64, 4);
740 glDrawArrays(GL_POLYGON, 68, 4);
741 glDrawArrays(GL_POLYGON, 72, 4);
742 glDrawArrays(GL_POLYGON, 76, 4);
743 glDrawArrays(GL_POLYGON, 80, 4);
744 glDrawArrays(GL_POLYGON, 84, 4);
745 glDrawArrays(GL_POLYGON, 88, 4);
746 glDrawArrays(GL_POLYGON, 92, 4);
747 glDrawArrays(GL_POLYGON, 96, 4);
748 glDrawArrays(GL_POLYGON, 100, 4);
749 glDrawArrays(GL_POLYGON, 104, 4);
750 glDrawArrays(GL_POLYGON, 108, 4);
751
752 // Matricea pentru dreptunghiul rosu
753 myMatrix = resizeMatrix * matrTransl * matrDepl * matrScale2;
754 // Culoarea
755 codCol = 2;
756 // Transmite variabile uniforme
757 glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
    ;
758 glUniform1i(codColLocation, codCol);
759 // Apelare DrawArrays

```

```
758     glDrawArrays(GL_POLYGON, 200, 4);
759     glDrawArrays(GL_POLYGON, 205, 4);
760     glDrawArrays(GL_POLYGON, 210, 4);
761     glDrawArrays(GL_POLYGON, 215, 4);
762     glDrawArrays(GL_POLYGON, 220, 4);
763     glDrawArrays(GL_POLYGON, 225, 4);
764     glUniform1i(codColLocation, 4);
765     glDrawArrays(GL_POLYGON, 230, 4);
766     glDrawArrays(GL_POLYGON, 235, 4);
767     glDrawArrays(GL_POLYGON, 240, 4);
768     glDrawArrays(GL_POLYGON, 245, 4);
769     glUniform1i(codColLocation, 3);
770     glDrawArrays(GL_LINES, 202, 2);
771     glDrawArrays(GL_LINES, 200, 2);
772     glDrawArrays(GL_LINES, 203, 2);
773     glDrawArrays(GL_LINES, 201, 2);
774     glDrawArrays(GL_LINES, 205, 2);
775     glDrawArrays(GL_LINES, 206, 2);
776     glDrawArrays(GL_LINES, 207, 2);
777     glDrawArrays(GL_LINES, 208, 2);
778     glDrawArrays(GL_LINES, 210, 2);
779     glDrawArrays(GL_LINES, 211, 2);
780     glDrawArrays(GL_LINES, 212, 2);
781     glDrawArrays(GL_LINES, 213, 2);
782     glDrawArrays(GL_LINES, 215, 2);
783     glDrawArrays(GL_LINES, 216, 2);
784     glDrawArrays(GL_LINES, 217, 2);
785     glDrawArrays(GL_LINES, 218, 2);
786     glDrawArrays(GL_LINES, 220, 2);
787     glDrawArrays(GL_LINES, 221, 2);
788     glDrawArrays(GL_LINES, 222, 2);
789     glDrawArrays(GL_LINES, 223, 2);
790     glDrawArrays(GL_LINES, 224, 2);
791     glDrawArrays(GL_LINES, 225, 2);
792     glDrawArrays(GL_LINES, 226, 2);
793     glDrawArrays(GL_LINES, 230, 2);
794     glDrawArrays(GL_LINES, 231, 2);
795     glDrawArrays(GL_LINES, 232, 2);
796     glDrawArrays(GL_LINES, 233, 2);
797     glDrawArrays(GL_LINES, 235, 2);
798     glDrawArrays(GL_LINES, 236, 2);
```

```

799     glDrawArrays(GL_LINES, 237, 2);
800     glDrawArrays(GL_LINES, 238, 2);
801     glDrawArrays(GL_LINES, 240, 2);
802     glDrawArrays(GL_LINES, 241, 2);
803     glDrawArrays(GL_LINES, 242, 2);
804     glDrawArrays(GL_LINES, 243, 2);
805     glDrawArrays(GL_LINES, 245, 2);
806     glDrawArrays(GL_LINES, 246, 2);
807     glDrawArrays(GL_LINES, 247, 2);
808     glDrawArrays(GL_LINES, 248, 2);
809
810     // Matricea pentru dreptunghiul rosu
811     myMatrix = resizeMatrix * matrTransl2 * matrDepl * matrScale2 *
            matrRot;
812     // Culoarea
813     codCol = 1;
814     // Transmite variabile uniforme
815     glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
            ;
816     glUniform1i(codColLocation, codCol);
817     // Apelare DrawArrays
818     glDrawArrays(GL_POLYGON, 250, 4);
819     glDrawArrays(GL_POLYGON, 255, 4);
820     glDrawArrays(GL_POLYGON, 260, 4);
821     glDrawArrays(GL_POLYGON, 265, 4);
822     glDrawArrays(GL_POLYGON, 270, 4);
823     glUniform1i(codColLocation, 3);
824     glDrawArrays(GL_LINES, 250, 2);
825     glDrawArrays(GL_LINES, 251, 2);
826     glDrawArrays(GL_LINES, 252, 2);
827     glDrawArrays(GL_LINES, 253, 2);
828     glDrawArrays(GL_LINES, 255, 2);
829     glDrawArrays(GL_LINES, 256, 2);
830     glDrawArrays(GL_LINES, 257, 2);
831     glDrawArrays(GL_LINES, 258, 2);
832     glDrawArrays(GL_LINES, 261, 2);
833     glDrawArrays(GL_LINES, 262, 2);
834     glDrawArrays(GL_LINES, 263, 2);
835     glDrawArrays(GL_LINES, 265, 2);
836     glDrawArrays(GL_LINES, 266, 2);
837     glDrawArrays(GL_LINES, 267, 2);

```



```

838     glDrawArrays(GL_LINES, 268, 2);
839     glDrawArrays(GL_LINES, 270, 2);
840     glDrawArrays(GL_LINES, 271, 2);
841     glDrawArrays(GL_LINES, 272, 2);
842     glDrawArrays(GL_LINES, 273, 2);
843     glDrawArrays(GL_LINES, 275, 2);
844     glDrawArrays(GL_LINES, 276, 2);
845     glDrawArrays(GL_LINES, 277, 2);
846     glDrawArrays(GL_LINES, 278, 2);
847     glDrawArrays(GL_LINES, 280, 2);
848     glDrawArrays(GL_LINES, 281, 2);
849     glDrawArrays(GL_LINES, 282, 2);
850     glDrawArrays(GL_LINES, 283, 2);
851     glDrawArrays(GL_LINES, 285, 2);
852     glDrawArrays(GL_LINES, 286, 2);
853     glDrawArrays(GL_LINES, 287, 2);
854     glDrawArrays(GL_LINES, 288, 2);
855     glDrawArrays(GL_LINES, 290, 2);
856     glDrawArrays(GL_LINES, 291, 2);
857     glDrawArrays(GL_LINES, 292, 2);
858     glDrawArrays(GL_LINES, 293, 2);
859     glUniform1i(codColLocation, 4);
860     glDrawArrays(GL_POLYGON, 275, 4);
861     glDrawArrays(GL_POLYGON, 280, 4);
862     glDrawArrays(GL_POLYGON, 285, 4);
863     glDrawArrays(GL_POLYGON, 290, 4);
864
865     myMatrix = resizeMatrix * matrTransl3;
866     glUniformMatrix4fv(myMatrixLocation, 1, GL_FALSE, &myMatrix[0][0])
        ;
867     glUniform1i(codColLocation, codCol);
868     glDrawArrays(GL_POLYGON, 112, 4);
869     glDrawArrays(GL_POLYGON, 116, 4);
870     glDrawArrays(GL_POLYGON, 120, 4);
871     glDrawArrays(GL_POLYGON, 124, 4);
872     glDrawArrays(GL_POLYGON, 128, 4);
873     glDrawArrays(GL_POLYGON, 132, 4);
874     glDrawArrays(GL_POLYGON, 136, 4);
875     glDrawArrays(GL_POLYGON, 140, 4);
876     glDrawArrays(GL_POLYGON, 144, 4);
877     glDrawArrays(GL_POLYGON, 148, 4);

```

```

878     glDrawArrays(GL_POLYGON, 152, 4);
879     glDrawArrays(GL_POLYGON, 156, 4);
880     glDrawArrays(GL_POLYGON, 160, 4);
881     glDrawArrays(GL_POLYGON, 164, 4);
882     glDrawArrays(GL_POLYGON, 168, 4);
883     glDrawArrays(GL_POLYGON, 172, 4);
884     glDrawArrays(GL_POLYGON, 176, 4);
885     glDrawArrays(GL_POLYGON, 180, 4);
886     glDrawArrays(GL_POLYGON, 184, 4);
887     glDrawArrays(GL_POLYGON, 188, 4);
888     glDrawArrays(GL_POLYGON, 192, 4);
889     glDrawArrays(GL_POLYGON, 196, 4);
890     glutSwapBuffers();
891     glFlush();
892 }
893 void Cleanup(void)
894 {
895     DestroyShaders();
896     DestroyVBO();
897 }
898
899 int main(int argc, char* argv[])
900 {
901     glutInit(&argc, argv);
902     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
903     glutInitWindowPosition(100, 100);
904     glutInitWindowSize(1000, 1000);
905     glutCreateWindow("Proiect 1 - Depasire intre 2 dreptunghiuri");
906     glewInit();
907     Initialize();
908     glutDisplayFunc(RenderFunction);
909     glutMouseFunc(mouse);
910     glutCloseFunc(Cleanup);
911     glutMainLoop();
912 }

```

# Codul pentru Shader

Codul îl puteți găsi în fișierul **03\_02\_Shader.frag** sau atașat mai jos.

```
1 // Shader-ul de fragment / Fragment shader
2 #version 330
3
4 in vec4 ex_Color;
5 uniform int codCuloare;
6
7 out vec4 out_Color;
8
9 void main(void)
10 {
11     switch (codCuloare)
12     {
13     case 0:
14         out_Color = ex_Color;
15         break;
16     case 1:
17         out_Color=vec4 (0.0, 0.0, 1.0, 0.0);
18         break;
19     case 2:
20         out_Color=vec4 (1.0, 0.0, 0.0, 0.0);
21         break;
22     case 3:
23         out_Color=vec4(0.0, 0.0, 0.0, 0.0);
24         break;
25     case 4:
26         out_Color=vec4(0.0, 1.0, 1.0, 1.0);
27         break;
28     default:
29         break;
30     };
31 }
```

# Referințe

- Fișierul *03\_02\_animatie\_new.cpp* din *Laborator 3*.
- *Materialele din Curs*.