Bursa Uludağ Üniversitesi
Bilgisayar Mühendisliği Bölümü
Bilgisayar Grafikleri Dersi Projesi

2B BALIK TUTMA ANİMASYONU

Samed DEMİRALAY 032090035
Taha GÖKPINAR 032090051

1-) Projenin Amacı

Bu projenin amacı OpenGL'in gücünü kullanarak gerçek zamanlı 2B grafiklerin oluşturulması ve etkileşimli deneyimlerin sunulması konusunda becerileri geliştirmeyi amaçlamaktadır. Ayrıca, grafik nesnelerin animasyonlarını ve fiziksel davranışlarını kontrol etme yeteneklerinin de geliştirilmesi hedeflenmektedir. Gerçekleştirilen uygulama, bir deniz sahnesini canlandıracaktır.

2-) Uygulamanın Yerine Getirdiği İşlevler

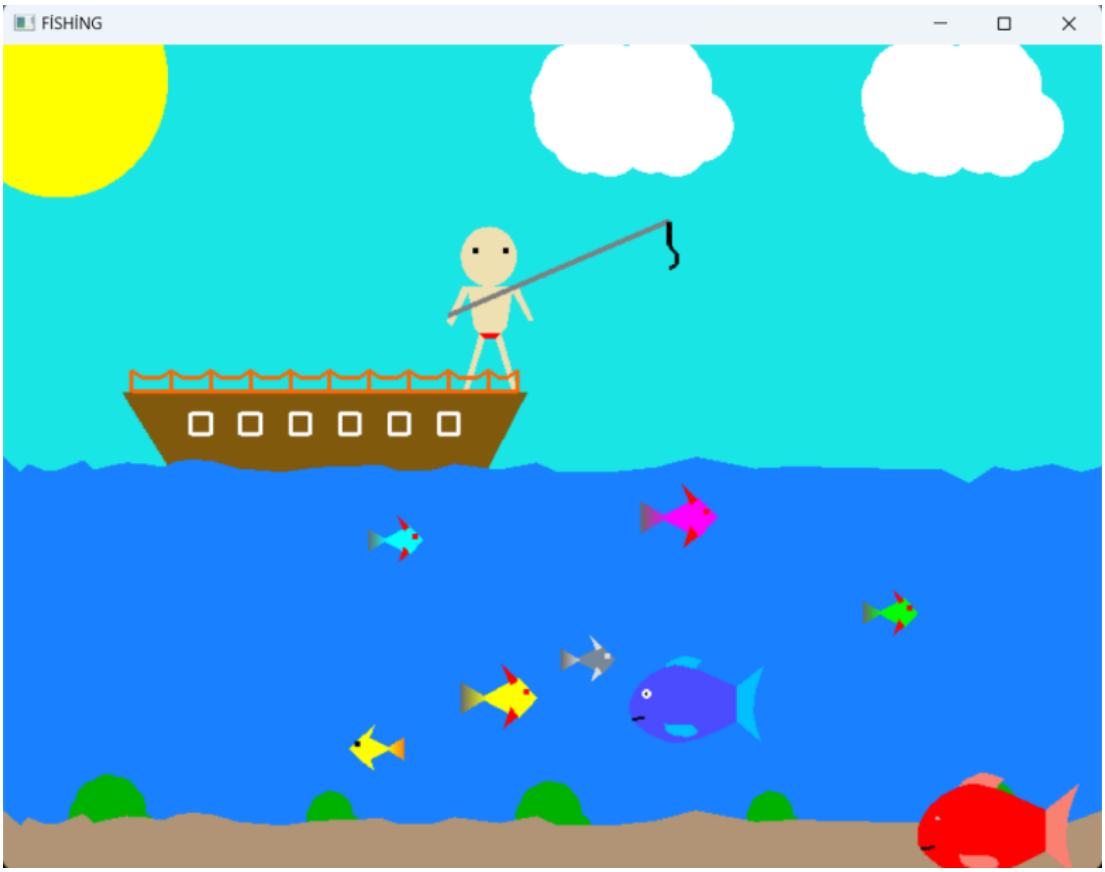
- Olta ve olta hareketleri
- Deniz
- Balık oluşturma
- Gökyüzü
- Gemi/İnsan tasarımı ve geminin hareketi
- Balıkların hareketi

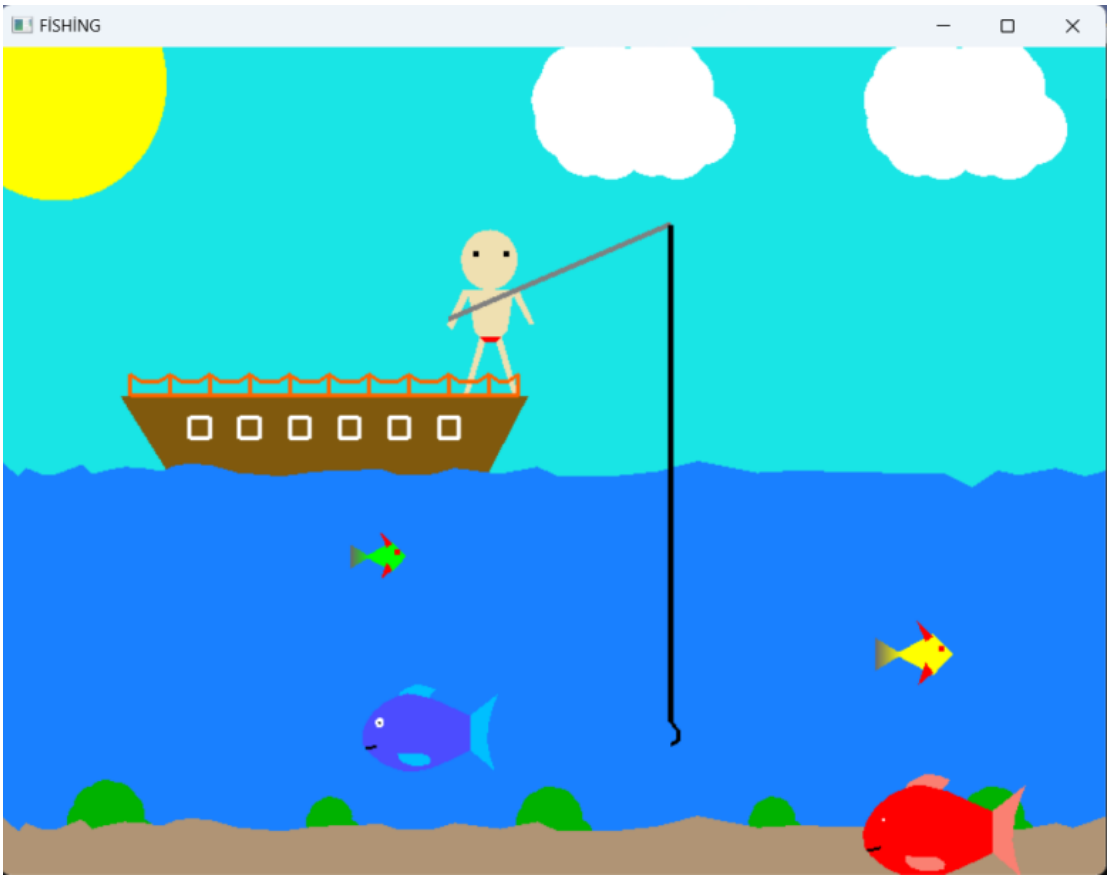
- Klavye işlevleri

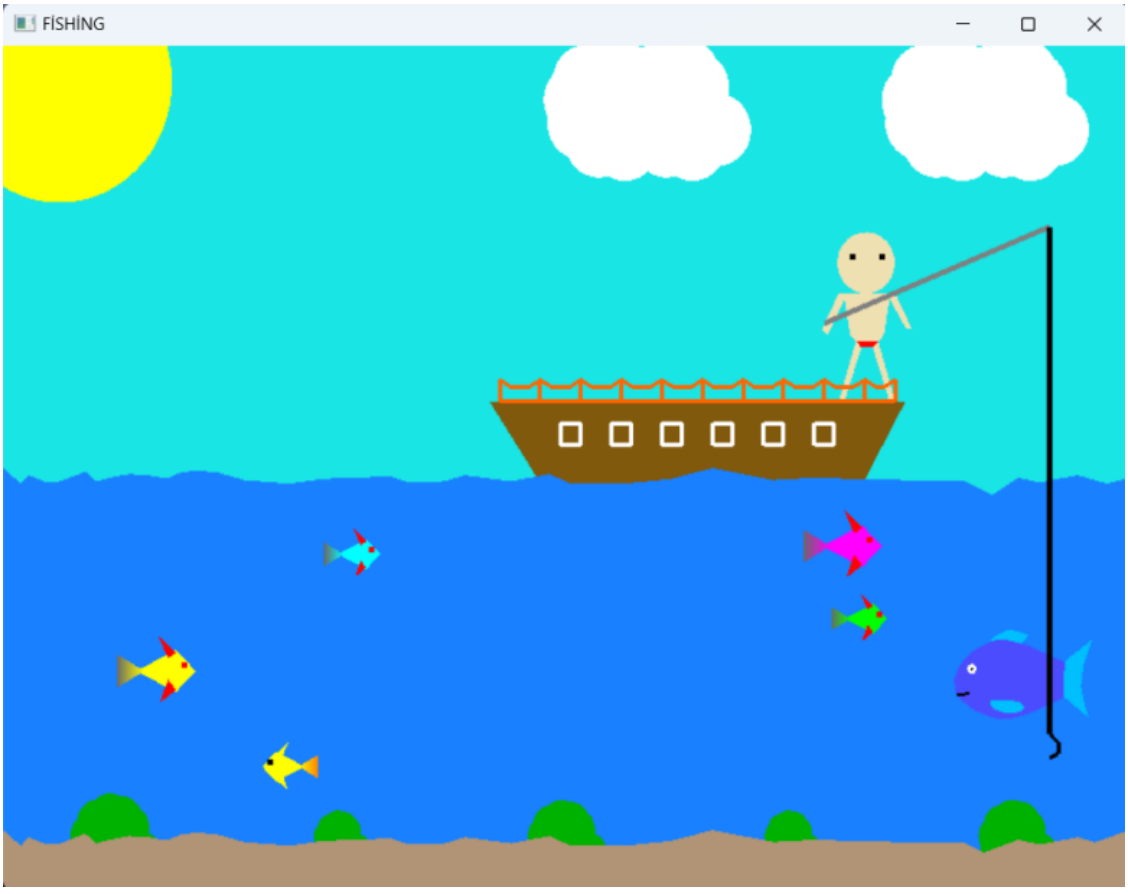
3-) Uygulamanın Özgün Yönleri

- Balıkların tasarımı hazır olarak alındı. Balıkların boyutları, konumları, hareket hızları ve hareket yönleri değiştirildi.
- Gemi tasarımı GeoGebra yazılımından yardım alınarak yapıldı.
- Olta ve insan tasarımı GeoGebra yazılımından yardım alınarak yapıldı.
- Bulutların ve güneşin tasarımı hazır olarak alındı. Bu nesnelerin boyutları, konumları ve şekilleri değiştirildi.
- Deniz tasarımında kum ve yosunlar kodlandı deniz suyu hazır olarak alındı.
- Geminin ve oltanın hareketi klavye işlevleriyle gerçekleştirildi. Geminin üstündeki insanın ve oltanın gemiyle birlikte hareketi geminin hareketine bağlı bir değişkenle gerçekleştirildi.

4-) Demodan ekran görüntüleri




FİSHİNG


FİSHİNG

5-) Hangi Grup Üyesinin Neyi Yaptığı

Proje kapsamında yapılan her şey takım üyelerinin ortak çalışmasıyla yapılmıştır.

6-) Referanslar

- https://freeglut.sourceforge.net/
- OpenGL Tutorial (http://www.opengl-tutorial.org/)
- Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson Education, 2008.
- Computer graphics with OpenGL Book by Donald Hearn and M. Pauline Baker.
- https://github.com/heyiamhemant/Sinking_Ship_OpenGL
- https://github.com/wasimulislam/City-View-OpenGL

```cpp
#include <glut.h>
#include <math.h>

float hookLength = 0.0;
float hookSpeed = 0.5;

GLsizei Width = 800.0;
GLsizei Height = 600.0;

float radius = 100.0;

float fish1X = 5.0f;
float fish1Y = 0.3f;

float fish2X = 0.7f;
float fish2Y = 0.6f;

float fish3X = 0.5f;
float fish3Y = 0.8f;

float fish4X = 1.4f;
float fish4Y = 0.6f;

float fish5X = 2.7f;
float fish5Y = 0.6f;

float fish6X = 2.4f;
float fish6Y = 0.7f;

float fish7X = 0.7f;
float fish7Y = 0.7f;

float fish8X = 2.5f;
float fish8Y = 0.8f;

float ship_x = 5, ship_y = 2;
GLfloat waterlevel = 150.0f;
GLfloat sandlevel = -150.0f;

void sea()
{
    glPushMatrix();
    glColor3f(0.1, 0.5, 1.0);
    glBegin(GL_POLYGON);
    glVertex2f(-200, -550);
    glVertex2f(0, waterlevel);
    glVertex2f(16, waterlevel - 13);
    glVertex2f(23, waterlevel - 6);
    glVertex2f(39, waterlevel - 12);
    glVertex2f(51, waterlevel - 11);
    glVertex2f(73, waterlevel - 3);
    glVertex2f(83, waterlevel - 11);
    glVertex2f(95, waterlevel - 8);
```

```
54      glVertex2f(112, waterlevel - 5);
55      glVertex2f(129, waterlevel - 6);
56      glVertex2f(141, waterlevel - 8);
57      glVertex2f(147, waterlevel - 8);
58      glVertex2f(157, waterlevel - 4);
59      glVertex2f(172, waterlevel - 2);
60      glVertex2f(192, waterlevel - 4);
61      glVertex2f(214, waterlevel - 10);
62      glVertex2f(254, waterlevel - 13);
63      glVertex2f(284, waterlevel - 9);
64      glVertex2f(344, waterlevel - 7);
65      glVertex2f(360, waterlevel - 12);
66      glVertex2f(392, waterlevel - 11);
67      glVertex2f(410, waterlevel - 6);
68      glVertex2f(451, waterlevel - 11);
69      glVertex2f(485, waterlevel - 5);
70      glVertex2f(504, waterlevel - 13);
71      glVertex2f(552, waterlevel - 13);
72      glVertex2f(593, waterlevel - 9);
73      glVertex2f(630, waterlevel);
74      glVertex2f(684, waterlevel - 10);
75      glVertex2f(720, waterlevel - 8);
76      glVertex2f(753, waterlevel - 9);
77      glVertex2f(800, waterlevel - 10);
78      glVertex2f(852, waterlevel - 10);
79      glVertex2f(880, waterlevel - 23);
80      glVertex2f(901, waterlevel - 8);
81      glVertex2f(920, waterlevel - 12);
82      glVertex2f(954, waterlevel - 6);
83      glVertex2f(980, waterlevel - 13);
84      glVertex2f(1004, waterlevel - 7);
85      glVertex2f(1000, 0);
86      glVertex2f(1602, waterlevel - 14);
87      glVertex2f(3000, waterlevel - 200);
88      glEnd();
89
90      glPopMatrix();
91  }
92
93  void sand()
94  {
95      glPushMatrix();
96      glColor3f(0.69f, 0.58f, 0.46f);
97      glBegin(GL_POLYGON);
98      glVertex2f(-200, -550);
99      glVertex2f(0, sandlevel);
100     glVertex2f(16, sandlevel - 13);
101     glVertex2f(23, sandlevel - 6);
102     glVertex2f(39, sandlevel - 12);
103     glVertex2f(51, sandlevel - 11);
104     glVertex2f(73, sandlevel - 3);
105     glVertex2f(83, sandlevel - 11);
106     glVertex2f(95, sandlevel - 8);
```

```
107     glVertex2f(112, sandlevel - 5);
108     glVertex2f(129, sandlevel - 6);
109     glVertex2f(141, sandlevel - 8);
110     glVertex2f(147, sandlevel - 8);
111     glVertex2f(157, sandlevel - 4);
112     glVertex2f(172, sandlevel - 2);
113     glVertex2f(192, sandlevel - 4);
114     glVertex2f(214, sandlevel - 10);
115     glVertex2f(254, sandlevel - 13);
116     glVertex2f(284, sandlevel - 9);
117     glVertex2f(344, sandlevel - 7);
118     glVertex2f(360, sandlevel - 12);
119     glVertex2f(392, sandlevel - 11);
120     glVertex2f(410, sandlevel - 6);
121     glVertex2f(451, sandlevel - 11);
122     glVertex2f(485, sandlevel - 5);
123     glVertex2f(504, sandlevel - 13);
124     glVertex2f(552, sandlevel - 13);
125     glVertex2f(593, sandlevel - 9);
126     glVertex2f(630, sandlevel);
127     glVertex2f(684, sandlevel - 10);
128     glVertex2f(720, sandlevel - 8);
129     glVertex2f(753, sandlevel - 9);
130     glVertex2f(800, sandlevel - 10);
131     glVertex2f(852, sandlevel - 10);
132     glVertex2f(880, sandlevel - 23);
133     glVertex2f(901, sandlevel - 8);
134     glVertex2f(920, sandlevel - 12);
135     glVertex2f(954, sandlevel - 6);
136     glVertex2f(980, sandlevel - 13);
137     glVertex2f(1004, sandlevel - 7);
138     glVertex2f(1000, sandlevel);
139     glVertex2f(1602, sandlevel - 14);
140     glVertex2f(3000, sandlevel - 200);
141     glEnd();
142
143     glPopMatrix();
144 }
145
146 void DrawFish1()
147 {
148     glTranslatef(-200, -300, 0.0);
149     glScalef(250.0f, 250.0f, 0.0f);
150     glTranslatef(fish1X, fish1Y, 0.0);
151     glPushMatrix();
152     glColor3f(0.000, 0.749, 1.000);
153
154     glBegin(GL_POLYGON);
155     glVertex3f(0.428, 0.127, 0.0);
156     glVertex3f(0.517, 0.057, 0.0);
157     glVertex3f(0.499, 0.108, 0.0);
158     glVertex3f(0.488, 0.162, 0.0);
159     glVertex3f(0.486, 0.148, 0.0);
```

```
160     glVertex3f(0.497, 0.233, 0.0);
161     glVertex3f(0.514, 0.282, 0.0);
162     glVertex3f(0.528, 0.318, 0.0);
163     glVertex3f(0.429, 0.245, 0.0);
164     glEnd();
165
166     glColor3f(0.000, 0.749, 1.000);
167     glBegin(GL_POLYGON);
168     glVertex3f(0.160, 0.304, 0.0);
169     glVertex3f(0.177, 0.326, 0.0);
170     glVertex3f(0.193, 0.334, 0.0);
171     glVertex3f(0.221, 0.346, 0.0);
172     glVertex3f(0.224, 0.348, 0.0);
173     glVertex3f(0.244, 0.348, 0.0);
174     glVertex3f(0.265, 0.345, 0.0);
175     glVertex3f(0.303, 0.333, 0.0);
176     glVertex3f(0.276, 0.304, 0.0);
177     glEnd();
178
179
180     glColor3f(0.3, 0.3, 1.000);
181     glBegin(GL_POLYGON);
182     glVertex3f(0.429, 0.243, 0.0);
183     glVertex3f(0.306, 0.276, 0.0);
184     glVertex3f(0.292, 0.301, 0.0);
185     glVertex3f(0.226, 0.316, 0.0);
186     glVertex3f(0.200, 0.319, 0.0);
187     glVertex3f(0.164, 0.309, 0.0);
188     glVertex3f(0.117, 0.288, 0.0);
189     glVertex3f(0.077, 0.256, 0.0);
190     glVertex3f(0.052, 0.222, 0.0);
191     glVertex3f(0.038, 0.187, 0.0);
192     glVertex3f(0.041, 0.144, 0.0);
193     glVertex3f(0.061, 0.119, 0.0);
194     glVertex3f(0.108, 0.083, 0.0);
195     glVertex3f(0.168, 0.060, 0.0);
196     glVertex3f(0.204, 0.053, 0.0);
197     glVertex3f(0.231, 0.054, 0.0);
198     glVertex3f(0.288, 0.067, 0.0);
199     glVertex3f(0.340, 0.087, 0.0);
200     glVertex3f(0.403, 0.115, 0.0);
201     glVertex3f(0.428, 0.127, 0.0);
202     glEnd();
203
204     glPushMatrix();
205     glTranslatef(.1, 0.22, 0.0);
206     glBegin(GL_POLYGON);
207     glColor3f(1.0, 1.0, 1.0);
208
209     for (int i = 0; i < 200; i++)
210     {
211         float pi = 3.1416;
212         float A = (i * 2 * pi) / 50;
```

```
213         float r = 0.016;
214         float x = r * cos(A);
215         float y = r * sin(A);
216         glVertex2f(x, y);
217     }
218     glEnd();
219 
220     glBegin(GL_POLYGON);
221     glColor3f(0.0, 0.0, 0.0);
222     for (int i = 0; i < 200; i++)
223     {
224         float pi = 3.1416;
225         float A = (i * 2 * pi) / 50;
226         float r = 0.005;
227         float x = r * cos(A);
228         float y = r * sin(A);
229         glVertex2f(x, y);
230     }
231     glEnd();
232     glPopMatrix();
233 
234     glColor3f(0.0, 0.0, 0.0);
235     glLineWidth(2.0);
236     glBegin(GL_LINES);
237     glVertex3f(0.048, 0.135, 0.0);
238     glVertex3f(0.077, 0.137, 0.0);
239 
240     glVertex3f(0.077, 0.137, 0.0);
241     glVertex3f(0.091, 0.141, 0.0);
242     glEnd();
243 
244     glColor3f(0.000, 0.749, 1.000);
245     glBegin(GL_POLYGON);
246     glVertex3f(0.165, 0.118, 0.0);
247     glVertex3f(0.169, 0.098, 0.0);
248     glVertex3f(0.182, 0.086, 0.0);
249     glVertex3f(0.213, 0.076, 0.0);
250     glVertex3f(0.253, 0.075, 0.0);
251     glVertex3f(0.281, 0.082, 0.0);
252     glVertex3f(0.286, 0.094, 0.0);
253     glVertex3f(0.281, 0.102, 0.0);
254     glVertex3f(0.267, 0.113, 0.0);
255     glEnd();
256     glPopMatrix();
257 }
258 
259 void DrawFish2()
260 {
261     glTranslatef(1500, -200, 0.0);
262     glScalef(300.0f, 300.0f, 300.0f);
263     glRotatef(180.0, 0.0, 1.0, 0.0);
264     glTranslatef(fish2X, fish2Y, 0.0);
265     glRotatef(180.0, 0.0, 1.0, 0.0);
```

```
266     glPushMatrix();
267     glColor3f(0.980, 0.502, 0.447);
268
269     glBegin(GL_POLYGON);
270     glVertex3f(0.428, 0.127, 0.0);
271     glVertex3f(0.517, 0.057, 0.0);
272     glVertex3f(0.499, 0.108, 0.0);
273     glVertex3f(0.488, 0.162, 0.0);
274     glVertex3f(0.486, 0.148, 0.0);
275     glVertex3f(0.497, 0.233, 0.0);
276     glVertex3f(0.514, 0.282, 0.0);
277     glVertex3f(0.528, 0.318, 0.0);
278     glVertex3f(0.429, 0.245, 0.0);
279     glEnd();
280
281     glColor3f(0.980, 0.502, 0.447);
282     glBegin(GL_POLYGON);
283     glVertex3f(0.160, 0.304, 0.0);
284     glVertex3f(0.177, 0.326, 0.0);
285     glVertex3f(0.193, 0.334, 0.0);
286     glVertex3f(0.221, 0.346, 0.0);
287     glVertex3f(0.224, 0.348, 0.0);
288     glVertex3f(0.244, 0.348, 0.0);
289     glVertex3f(0.265, 0.345, 0.0);
290     glVertex3f(0.303, 0.333, 0.0);
291     glVertex3f(0.276, 0.304, 0.0);
292     glEnd();
293
294
295     glColor3f(1.000, 0.000, 0.000);
296     glBegin(GL_POLYGON);
297     glVertex3f(0.429, 0.243, 0.0);
298     glVertex3f(0.306, 0.276, 0.0);
299     glVertex3f(0.292, 0.301, 0.0);
300     glVertex3f(0.226, 0.316, 0.0);
301     glVertex3f(0.200, 0.319, 0.0);
302     glVertex3f(0.164, 0.309, 0.0);
303     glVertex3f(0.117, 0.288, 0.0);
304     glVertex3f(0.077, 0.256, 0.0);
305     glVertex3f(0.052, 0.222, 0.0);
306     glVertex3f(0.038, 0.187, 0.0);
307     glVertex3f(0.041, 0.144, 0.0);
308     glVertex3f(0.061, 0.119, 0.0);
309     glVertex3f(0.108, 0.083, 0.0);
310     glVertex3f(0.168, 0.060, 0.0);
311     glVertex3f(0.204, 0.053, 0.0);
312     glVertex3f(0.231, 0.054, 0.0);
313     glVertex3f(0.288, 0.067, 0.0);
314     glVertex3f(0.340, 0.087, 0.0);
315     glVertex3f(0.403, 0.115, 0.0);
316     glVertex3f(0.428, 0.127, 0.0);
317     glEnd();
318
```

```
319     glPushMatrix();
320     glTranslatef(.1, 0.22, 0.0);
321     glBegin(GL_POLYGON);
322     glColor3f(1.0, 1.0, 1.0);
323 
324     for (int i = 0; i < 200; i++)
325     {
326         float pi = 3.1416;
327         float A = (i * 2 * pi) / 50;
328         float r = 0.005;
329         float x = r * cos(A);
330         float y = r * sin(A);
331         glVertex2f(x, y);
332     }
333     glEnd();
334     glPopMatrix();
335 
336     glColor3f(0.0, 0.0, 0.0);
337     glLineWidth(2.0);
338     glBegin(GL_LINES);
339     glVertex3f(0.048, 0.135, 0.0);
340     glVertex3f(0.077, 0.137, 0.0);
341 
342     glVertex3f(0.077, 0.137, 0.0);
343     glVertex3f(0.091, 0.141, 0.0);
344     glEnd();
345 
346     glColor3f(0.980, 0.502, 0.447);
347     glBegin(GL_POLYGON);
348     glVertex3f(0.165, 0.118, 0.0);
349     glVertex3f(0.169, 0.098, 0.0);
350     glVertex3f(0.182, 0.086, 0.0);
351     glVertex3f(0.213, 0.076, 0.0);
352     glVertex3f(0.253, 0.075, 0.0);
353     glVertex3f(0.281, 0.082, 0.0);
354     glVertex3f(0.286, 0.094, 0.0);
355     glVertex3f(0.281, 0.102, 0.0);
356     glVertex3f(0.267, 0.113, 0.0);
357     glEnd();
358     glPopMatrix();
359 }
360 
361 void DrawFish3()
362 {
363     glTranslatef(750, -300, 0.0);
364     glScalef(250.0f, 250.0f, 0.0f);
365     glRotatef(0.0, 0.0, 1.0, 0.0);
366     glTranslatef(fish3X, fish3Y, 0.0);
367     glPushMatrix();
368 
369     glColor3f(1.0, 1.0, 0.0);
370     glBegin(GL_POLYGON);
371     glVertex2f(0.7, -0.05);
```

```
372     glVertex2f(0.75, -0.1);
373     glVertex2f(0.85, -0.05);
374     glVertex2f(0.75, 0.0);
375     glEnd();
376
377     glBegin(GL_TRIANGLES);
378     glVertex2f(0.83, -0.05);
379     glColor3f(1.8, 0.5, 0.0);
380     glVertex2f(0.9, -0.09);
381     glVertex2f(0.9, -0.01);
382     glEnd();
383
384     glBegin(GL_TRIANGLES);
385     glColor3f(1.0, 1.0, 0.0);
386     glVertex2f(0.79, -0.125);
387     glVertex2f(0.77, -0.07);
388     glVertex2f(0.75, -0.095);
389     glEnd();
390
391     glBegin(GL_TRIANGLES);
392     glVertex2f(0.795, 0.035);
393     glVertex2f(0.77, -0.02);
394     glVertex2f(0.75, -0.007);
395     glEnd();
396
397     glColor3f(0.0, 0.0, 0.0);
398     glPointSize(4.0);
399     glBegin(GL_POINTS);
400     glVertex2f(0.73, -0.035);
401     glEnd();
402     glPopMatrix();
403 }
404
405 void DrawFish4()
406 {
407     glTranslatef(-500, 100, 0.0);
408     glScalef(250.0f, 250.0f, 0.0f);
409     glTranslatef(fish4X, fish4Y, 0.0);
410     glRotatef(180.0, 0.0, 1.0, 0.0);
411     glPushMatrix();
412
413     glColor3f(0.467, 0.533, 0.600);
414     glBegin(GL_POLYGON);
415     glVertex2f(0.7, -0.05);
416     glVertex2f(0.75, -0.1);
417     glVertex2f(0.85, -0.05);
418     glVertex2f(0.75, 0.0);
419     glEnd();
420
421     glColor3f(0.863, 0.863, 0.863);
422     glBegin(GL_TRIANGLES);
423     glVertex2f(0.83, -0.05);
424     glColor3f(0.412, 0.412, 0.412);
```

```
425     glVertex2f(0.9, -0.09);
426     glVertex2f(0.9, -0.01);
427     glEnd();
428
429     glColor3f(0.863, 0.863, 0.863);
430     glBegin(GL_TRIANGLES);
431     glVertex2f(0.79, -0.125);
432     glVertex2f(0.77, -0.07);
433     glVertex2f(0.75, -0.095);
434     glEnd();
435
436     glColor3f(0.863, 0.863, 0.863);
437     glBegin(GL_TRIANGLES);
438     glVertex2f(0.795, 0.035);
439     glVertex2f(0.77, -0.02);
440     glVertex2f(0.75, -0.007);
441     glEnd();
442
443     glPointSize(4.0);
444     glBegin(GL_POINTS);
445     glVertex2f(0.73, -0.035);
446     glEnd();
447     glPopMatrix();
448 }
449
450 void DrawFish5()
451 {
452     glTranslatef(-900, 50, 0.0);
453     glScalef(350.0f, 350.0f, 0.0f);
454     glTranslatef(fish5X, fish5Y, 0.0);
455     glRotatef(180.0, 0.0, 1.0, 0.0);
456     glPushMatrix();
457
458     glColor3f(1.0, 1.0, 0.0);
459     glBegin(GL_POLYGON);
460     glVertex2f(0.7, -0.05);
461     glVertex2f(0.75, -0.1);
462     glVertex2f(0.85, -0.05);
463     glVertex2f(0.75, 0.0);
464     glEnd();
465
466     glColor3f(1.0, 1.0, 0.0);
467     glBegin(GL_TRIANGLES);
468     glVertex2f(0.83, -0.05);
469     glColor3f(0.412, 0.412, 0.412);
470     glVertex2f(0.9, -0.09);
471     glVertex2f(0.9, -0.01);
472     glEnd();
473
474     glColor3f(1.0, 0.0, 0.0);
475     glBegin(GL_TRIANGLES);
476     glVertex2f(0.79, -0.125);
477     glVertex2f(0.77, -0.07);
```

```
478     glVertex2f(0.75, -0.095);
479     glEnd();
480
481     glColor3f(1.0, 0.0, 0.0);
482     glBegin(GL_TRIANGLES);
483     glVertex2f(0.795, 0.035);
484     glVertex2f(0.77, -0.02);
485     glVertex2f(0.75, -0.007);
486     glEnd();
487
488     glPointSize(4.0);
489     glBegin(GL_POINTS);
490     glVertex2f(0.73, -0.035);
491     glEnd();
492     glPopMatrix();
493 }
494
495 void DrawFish6()
496 {
497     glTranslatef(-700, 50, 0.0);
498     glScalef(350.0f, 350.0f, 0.0f);
499     glTranslatef(fish6X, fish6Y, 0.0);
500     glRotatef(180.0, 0.0, 1.0, 0.0);
501     glPushMatrix();
502
503     glColor3f(1.0, 0.0, 1.0);
504     glBegin(GL_POLYGON);
505
506     glVertex2f(0.7, -0.05);
507     glVertex2f(0.75, -0.1);
508     glVertex2f(0.85, -0.05);
509     glVertex2f(0.75, 0.0);
510     glEnd();
511
512     glColor3f(1.0, 0.0, 1.0);
513     glBegin(GL_TRIANGLES);
514     glVertex2f(0.83, -0.05);
515     glColor3f(0.412, 0.412, 0.412);
516     glVertex2f(0.9, -0.09);
517     glVertex2f(0.9, -0.01);
518     glEnd();
519
520     glColor3f(1.0, 0.0, 0.0);
521     glBegin(GL_TRIANGLES);
522     glVertex2f(0.79, -0.125);
523     glVertex2f(0.77, -0.07);
524     glVertex2f(0.75, -0.095);
525     glEnd();
526
527     glColor3f(1.0, 0.0, 0.0);
528     glBegin(GL_TRIANGLES);
529     glVertex2f(0.795, 0.035);
530     glVertex2f(0.77, -0.02);
```

```
531     glVertex2f(0.75, -0.007);
532     glEnd();
533
534     glPointSize(4.0);
535     glBegin(GL_POINTS);
536     glVertex2f(0.73, -0.035);
537     glEnd();
538     glPopMatrix();
539 }
540
541 void DrawFish7()
542 {
543     glTranslatef(-500, 50, 0.0);
544     glScalef(250.0f, 250.0f, 0.0f);
545     glTranslatef(fish7X, fish7Y, 0.0);
546     glRotatef(180.0, 0.0, 1.0, 0.0);
547     glPushMatrix();
548
549     glColor3f(0.0, 1.0, 1.0);
550     glBegin(GL_POLYGON);
551     glVertex2f(0.7, -0.05);
552     glVertex2f(0.75, -0.1);
553     glVertex2f(0.85, -0.05);
554     glVertex2f(0.75, 0.0);
555     glEnd();
556
557     glColor3f(0.0, 1.0, 1.0);
558     glBegin(GL_TRIANGLES);
559     glVertex2f(0.83, -0.05);
560     glColor3f(0.412, 0.412, 0.412);
561     glVertex2f(0.9, -0.09);
562     glVertex2f(0.9, -0.01);
563     glEnd();
564
565     glColor3f(1.0, 0.0, 0.0);
566     glBegin(GL_TRIANGLES);
567     glVertex2f(0.79, -0.125);
568     glVertex2f(0.77, -0.07);
569     glVertex2f(0.75, -0.095);
570     glEnd();
571
572     glColor3f(1.0, 0.0, 0.0);
573     glBegin(GL_TRIANGLES);
574     glVertex2f(0.795, 0.035);
575     glVertex2f(0.77, -0.02);
576     glVertex2f(0.75, -0.007);
577     glEnd();
578
579     glPointSize(4.0);
580     glBegin(GL_POINTS);
581     glVertex2f(0.73, -0.035);
582     glEnd();
583     glPopMatrix();
```

```
584 }
585
586 void DrawFish8()
587 {
588     glTranslatef(-500, 40, 0.0);
589     glScalef(250.0f, 250.0f, 0.0f);
590     glTranslatef(fish8X, fish8Y, 0.0);
591     glRotatef(180.0, 0.0, 1.0, 0.0);
592     glPushMatrix();
593
594     glColor3f(0.0, 1.0, 0.0);
595     glBegin(GL_POLYGON);
596     glVertex2f(0.7, -0.05);
597     glVertex2f(0.75, -0.1);
598     glVertex2f(0.85, -0.05);
599     glVertex2f(0.75, 0.0);
600     glEnd();
601
602     glColor3f(0.0, 1.0, 0.0);
603     glBegin(GL_TRIANGLES);
604     glVertex2f(0.83, -0.05);
605     glColor3f(0.412, 0.412, 0.412);
606     glVertex2f(0.9, -0.09);
607     glVertex2f(0.9, -0.01);
608     glEnd();
609
610     glColor3f(1.0, 0.0, 0.0);
611     glBegin(GL_TRIANGLES);
612     glVertex2f(0.79, -0.125);
613     glVertex2f(0.77, -0.07);
614     glVertex2f(0.75, -0.095);
615     glEnd();
616
617     glColor3f(1.0, 0.0, 0.0);
618     glBegin(GL_TRIANGLES);
619     glVertex2f(0.795, 0.035);
620     glVertex2f(0.77, -0.02);
621     glVertex2f(0.75, -0.007);
622     glEnd();
623
624     glPointSize(4.0);
625     glBegin(GL_POINTS);
626     glVertex2f(0.73, -0.035);
627     glEnd();
628     glPopMatrix();
629 }
630
631 void sun() {
632     int n=13500, x=50, y=470, r=100;
633     glColor3f(1.0, 1.0, 0.0);
634     double inc = 2 * 3.1416 / (double)n;
635     glBegin(GL_POLYGON);
636     for (double theta = 0.0; theta <= 2 * 3.1416; theta += inc) {
```

```
637             glVertex2f(r * cos(theta) + x, r * sin(theta) + y);
638         }
639         glEnd();
640     }
641
642     void ship()
643     {
644         glScalef(18, 18, 0);
645         glColor3f(0.5, 0.5, 0.5);
646
647         glColor3f(0.5f, 0.35f, 0.05f);
648         glBegin(GL_POLYGON);
649         glVertex2f(ship_x + 1, ship_y + 5.5);
650         glVertex2f(ship_x + 4, ship_y + 1);
651         glVertex2f(ship_x + 19, ship_y + 1);
652         glVertex2f(ship_x + 21.5, ship_y + 5.5);
653         glEnd();
654
655         glColor3f(1.0, 0.4, 0.0);
656         glBegin(GL_LINE_LOOP);
657         glVertex2f(ship_x + 1.5, ship_y + 5.5);
658         glVertex2f(ship_x + 1.5, ship_y + 6.5);
659         glVertex2f(ship_x + 2, ship_y + 6.2);
660         glVertex2f(ship_x + 3, ship_y + 6.2);
661         glVertex2f(ship_x + 3.5, ship_y + 6.5);
662         glVertex2f(ship_x + 3.5, ship_y + 5.5);
663         glEnd();
664
665         glBegin(GL_LINE_LOOP);
666         glVertex2f(ship_x + 3.5, ship_y + 5.5);
667         glVertex2f(ship_x + 3.5, ship_y + 6.5);
668         glVertex2f(ship_x + 4, ship_y + 6.2);
669         glVertex2f(ship_x + 5, ship_y + 6.2);
670         glVertex2f(ship_x + 5.5, ship_y + 6.5);
671         glVertex2f(ship_x + 5.5, ship_y + 5.5);
672         glEnd();
673
674         glBegin(GL_LINE_LOOP);
675         glVertex2f(ship_x + 5.5, ship_y + 5.5);
676         glVertex2f(ship_x + 5.5, ship_y + 6.5);
677         glVertex2f(ship_x + 6, ship_y + 6.2);
678         glVertex2f(ship_x + 7, ship_y + 6.2);
679         glVertex2f(ship_x + 7.5, ship_y + 6.5);
680         glVertex2f(ship_x + 7.5, ship_y + 5.5);
681         glEnd();
682
683         glBegin(GL_LINE_LOOP);
684         glVertex2f(ship_x + 7.5, ship_y + 5.5);
685         glVertex2f(ship_x + 7.5, ship_y + 6.5);
686         glVertex2f(ship_x + 8, ship_y + 6.2);
687         glVertex2f(ship_x + 9, ship_y + 6.2);
688         glVertex2f(ship_x + 9.5, ship_y + 6.5);
689         glVertex2f(ship_x + 9.5, ship_y + 5.5);
```

```
690     glEnd();
691
692     glBegin(GL_LINE_LOOP);
693     glVertex2f(ship_x + 9.5, ship_y + 5.5);
694     glVertex2f(ship_x + 9.5, ship_y + 6.5);
695     glVertex2f(ship_x + 10, ship_y + 6.2);
696     glVertex2f(ship_x + 11, ship_y + 6.2);
697     glVertex2f(ship_x + 11.5, ship_y + 6.5);
698     glVertex2f(ship_x + 11.5, ship_y + 5.5);
699     glEnd();
700
701     glBegin(GL_LINE_LOOP);
702     glVertex2f(ship_x + 11.5, ship_y + 5.5);
703     glVertex2f(ship_x + 11.5, ship_y + 6.5);
704     glVertex2f(ship_x + 12, ship_y + 6.2);
705     glVertex2f(ship_x + 13, ship_y + 6.2);
706     glVertex2f(ship_x + 13.5, ship_y + 6.5);
707     glVertex2f(ship_x + 13.5, ship_y + 5.5);
708     glEnd();
709
710     glBegin(GL_LINE_LOOP);
711     glVertex2f(ship_x + 13.5, ship_y + 5.5);
712     glVertex2f(ship_x + 13.5, ship_y + 6.5);
713     glVertex2f(ship_x + 14, ship_y + 6.2);
714     glVertex2f(ship_x + 15, ship_y + 6.2);
715     glVertex2f(ship_x + 15.5, ship_y + 6.5);
716     glVertex2f(ship_x + 15.5, ship_y + 5.5);
717     glEnd();
718
719     glBegin(GL_LINE_LOOP);
720     glVertex2f(ship_x + 15.5, ship_y + 5.5);
721     glVertex2f(ship_x + 15.5, ship_y + 6.5);
722     glVertex2f(ship_x + 16, ship_y + 6.2);
723     glVertex2f(ship_x + 17, ship_y + 6.2);
724     glVertex2f(ship_x + 17.5, ship_y + 6.5);
725     glVertex2f(ship_x + 17.5, ship_y + 5.5);
726     glEnd();
727
728     glBegin(GL_LINE_LOOP);
729     glVertex2f(ship_x + 17.5, ship_y + 5.5);
730     glVertex2f(ship_x + 17.5, ship_y + 6.5);
731     glVertex2f(ship_x + 18, ship_y + 6.2);
732     glVertex2f(ship_x + 19, ship_y + 6.2);
733     glVertex2f(ship_x + 19.5, ship_y + 6.5);
734     glVertex2f(ship_x + 19.5, ship_y + 5.5);
735     glEnd();
736
737     glBegin(GL_LINE_LOOP);
738     glVertex2f(ship_x + 19.5, ship_y + 5.5);
739     glVertex2f(ship_x + 19.5, ship_y + 6.5);
740     glVertex2f(ship_x + 20, ship_y + 6.2);
741     glVertex2f(ship_x + 20.5, ship_y + 6.2);
742     glVertex2f(ship_x + 21, ship_y + 6.5);
```

```
743     glVertex2f(ship_x + 21, ship_y + 5.5);
744     glEnd();
745
746     glColor3f(1.0, 1.0, 1.0);
747     glBegin(GL_LINE_LOOP);
748     glVertex2f(ship_x + 4.5, ship_y + 3.5);
749     glVertex2f(ship_x + 4.5, ship_y + 4.5);
750     glVertex2f(ship_x + 5.5, ship_y + 4.5);
751     glVertex2f(ship_x + 5.5, ship_y + 3.5);
752     glEnd();
753
754     glBegin(GL_LINE_LOOP);
755     glVertex2f(ship_x + 7, ship_y + 3.5);
756     glVertex2f(ship_x + 7, ship_y + 4.5);
757     glVertex2f(ship_x + 8, ship_y + 4.5);
758     glVertex2f(ship_x + 8, ship_y + 3.5);
759     glEnd();
760
761     glBegin(GL_LINE_LOOP);
762     glVertex2f(ship_x + 9.5, ship_y + 3.5);
763     glVertex2f(ship_x + 9.5, ship_y + 4.5);
764     glVertex2f(ship_x + 10.5, ship_y + 4.5);
765     glVertex2f(ship_x + 10.5, ship_y + 3.5);
766     glEnd();
767
768     glBegin(GL_LINE_LOOP);
769     glVertex2f(ship_x + 12, ship_y + 3.5);
770     glVertex2f(ship_x + 12, ship_y + 4.5);
771     glVertex2f(ship_x + 13, ship_y + 4.5);
772     glVertex2f(ship_x + 13, ship_y + 3.5);
773     glEnd();
774
775
776     glBegin(GL_LINE_LOOP);
777     glVertex2f(ship_x + 14.5, ship_y + 3.5);
778     glVertex2f(ship_x + 14.5, ship_y + 4.5);
779     glVertex2f(ship_x + 15.5, ship_y + 4.5);
780     glVertex2f(ship_x + 15.5, ship_y + 3.5);
781     glEnd();
782
783     glBegin(GL_LINE_LOOP);
784     glVertex2f(ship_x + 17, ship_y + 3.5);
785     glVertex2f(ship_x + 17, ship_y + 4.5);
786     glVertex2f(ship_x + 18, ship_y + 4.5);
787     glVertex2f(ship_x + 18, ship_y + 3.5);
788     glEnd();
789
790     glFlush();
791 }
792
793 void circle(int n, int x, int y, int r) {
794     glColor3f(0.937, 0.878, 0.694);
795     glScalef(0.85, 0.85, 0.0);
```

```
796     glTranslatef(173.5f, 206.5f, 0.0);
797     double inc = 2 * 3.1416 / (double)n;
798     glBegin(GL_POLYGON);
799     for (double theta = 0.0; theta <= 2 * 3.1416; theta += inc) {
800         glVertex2f(r * cos(theta) + x + (ship_x * 21.2111), r * sin
              (theta) + y);
801     }
802     glEnd();
803 }
804
805 void man() {
806
807     glTranslatef(23.0, 170.0, 0.0);
808     glColor3f(1.0f, 1.0f, 1.0f);
809     glScalef(10.0, 10.0, 0.0);
810
811     // basen bölgesi
812     glBegin(GL_POLYGON);
813     glColor3f(1.0, 0.0, 0.0);
814     glVertex2f(32.5f + (ship_x * 1.8), 8.0f);
815     glVertex2f(32.0f + (ship_x * 1.8), 8.5f);
816     glVertex2f(34.0f + (ship_x * 1.8), 8.5f);
817     glVertex2f(33.5f + (ship_x * 1.8), 8.0f);
818     glEnd();
819
820     // gövde
821     glBegin(GL_POLYGON);
822     glColor3f(0.937, 0.878, 0.694);
823     glVertex2f(32.0f + (ship_x * 1.8), 8.5f);
824     glVertex2f(31.5f + (ship_x * 1.8), 9.0f);
825     glVertex2f(31.0f + (ship_x * 1.8), 12.0f);
826     glVertex2f(31.5f + (ship_x * 1.8), 12.5f);
827     glVertex2f(34.5f + (ship_x * 1.8), 12.5f);
828     glVertex2f(35.0f + (ship_x * 1.8), 12.0f);
829     glVertex2f(34.5f + (ship_x * 1.8), 9.0f);
830     glVertex2f(34.0f + (ship_x * 1.8), 8.5f);
831     glEnd();
832
833     //sol omuz
834     glBegin(GL_TRIANGLES);
835     glVertex2f(31.5f + (ship_x * 1.8), 12.5f);
836     glVertex2f(31.0f + (ship_x * 1.8), 12.0f);
837     glVertex2f(30.5f + (ship_x * 1.8), 12.5f);
838     glEnd();
839
840     //sağ omuz
841     glBegin(GL_TRIANGLES);
842     glVertex2f(34.5f + (ship_x * 1.8), 12.5f);
843     glVertex2f(35.0f + (ship_x * 1.8), 12.0f);
844     glVertex2f(35.5f + (ship_x * 1.8), 12.5f);
845     glEnd();
846
847     //sol kol
```

```
848     glBegin(GL_POLYGON);
849     glColor3f(0.937, 0.878, 0.694);
850     glVertex2f(31.0f + (ship_x * 1.8), 12.0f);
851     glVertex2f(30.5f + (ship_x * 1.8), 12.5f);
852     glVertex2f(29.0f + (ship_x * 1.8), 9.5f);
853     glVertex2f(29.5f + (ship_x * 1.8), 9.0f);
854     glEnd();
855
856     //sağ kol
857     glBegin(GL_POLYGON);
858     glColor3f(0.937, 0.878, 0.694);
859     glVertex2f(35.0f + (ship_x * 1.8), 12.0f);
860     glVertex2f(35.5f + (ship_x * 1.8), 12.5f);
861     glVertex2f(37.0f + (ship_x * 1.8), 9.5f);
862     glVertex2f(36.5f + (ship_x * 1.8), 9.5f);
863     glEnd();
864
865     // Sol bacak
866     glBegin(GL_POLYGON);
867     glColor3f(0.937, 0.878, 0.694);
868     glVertex2f(32.5f + (ship_x * 1.8), 8.0f);
869     glVertex2f(32.0f + (ship_x * 1.8), 8.5f);
870     glVertex2f(30.5f + (ship_x * 1.8), 3.5f);
871     glVertex2f(31.0f + (ship_x * 1.8), 3.5f);
872     glEnd();
873
874     // Sağ bacak
875     glBegin(GL_POLYGON);
876     glColor3f(0.937, 0.878, 0.694);
877     glVertex2f(33.5f + (ship_x * 1.8), 8.0f);
878     glVertex2f(34.0f + (ship_x * 1.8), 8.5f);
879     glVertex2f(35.5f + (ship_x * 1.8), 3.5f);
880     glVertex2f(35.0f + (ship_x * 1.8), 3.5f);
881     glEnd();
882
883     //sağ göz
884     glPointSize(4.0);
885     glColor3f(0.0, 0.0, 0.0);
886     glBegin(GL_POINTS);
887     glVertex2f(31.7f + (ship_x * 1.8), 15.5f);
888     glEnd();
889
890     //sol göz
891     glPointSize(4.0);
892     glColor3f(0.0, 0.0, 0.0);
893     glBegin(GL_POINTS);
894     glVertex2f(34.4f + (ship_x * 1.8), 15.5f);
895     glEnd();
896 }
897
898 void update(int value) {
899     if (fish1X < -3)
900     {
```

```
901         fish1X = 5.0f;
902     }
903     fish1X -= 0.01f;
904     fish1Y = sin(fish1X * 2) / 8 + 0.8;
905     if (fish1X < 0)
906     {
907         fish1Y* (-1);
908     }
909     if (fish2X > 7)
910     {
911         fish2X = 0;
912     }
913     fish2X += 0.008;
914     fish2Y = sin(fish2X) / 10;
915     if (fish2X < 10)
916     {
917         fish2Y *= (-1);
918     }
919     if (fish3X < -10)
920     {
921         fish3X = 1;
922     }
923     fish3X -= 0.015;
924     fish3Y = sin(fish3X + 1.0) / 7 + 1.0;
925     if (fish4X > 9)
926     {
927         fish4X = -1;
928     }
929     fish4X += 0.018;
930     fish4Y = sin(fish4X) / 7 - 0.3;
931     if (fish5X > 7)
932     {
933         fish5X = -0.9;
934     }
935     fish5X += 0.010;
936     fish5Y = sin(fish5X) / 4;
937     if (fish6X > 7)
938     {
939         fish6X = -1;
940     }
941     fish6X += 0.011;
942     fish6Y = sin(fish6X + 0.5) / 5;
943     if (fish6X < 90)
944     {
945         fish6Y *= (-1);
946     }
947
948     if (fish7X > 8)
949     {
950         fish7X = -1;
951     }
952     fish7X += 0.018;
953     fish7Y = sin(fish7X + 0.5)/6;
```

```
954
955     if (fish7X < 10)
956     {
957         fish7Y *= (-1);
958     }
959
960     if (fish8X > 8)
961     {
962         fish8X = -1;
963     }
964     fish8X += 0.018;
965     fish8Y = sin(fish8X + 0.5) / 6;
966
967     if (fish8X < 10)
968     {
969         fish8Y *= (-1);
970     }
971
972     glutPostRedisplay();
973     glutTimerFunc(25, update, 0);
974 }
975
976 void hook() {
977     glLineWidth(4.0);
978     glTranslatef(-745.0, 150.0, 0.0);
979     glColor3f(0.5f, 0.5f, 0.5f);
980     glScalef(40.0, 40.0, 0.0);
981     glBegin(GL_LINES);
982     glVertex2f(26.5f + (ship_x / 2.22), 3.0f);
983     glVertex2f(31.5f + (ship_x / 2.22), 5.0f);
984     glEnd();
985
986     glColor3f(0.0f, 0.0f, 0.0f);
987     glLineWidth(3.0);
988     glBegin(GL_LINES);
989     glVertex2f(31.5f + (ship_x / 2.22), 5.0f);
990     glVertex2f(31.5f + (ship_x / 2.22), 4.5f + hookLength);
991     glEnd();
992
993     glColor3f(0.0f, 0.0f, 0.0f);
994     glLineWidth(3.0);
995     glBegin(GL_LINE_STRIP);
996     glVertex2f(31.5f + (ship_x / 2.22), 4.5f + hookLength);
997     glVertex2f(31.59537444407249f + (ship_x / 2.22), 4.3931998091281f +
            hookLength);
998     glVertex2f(31.6901701872822f + (ship_x / 2.22), 4.2916329378167f +
            hookLength);
999     glVertex2f(31.6901701872822 + (ship_x / 2.22), 4.088499195194f +
          hookLength);
1000    glVertex2f(31.5993960713431f + (ship_x / 2.22), 4.0391662215094f +
            hookLength);
1001    glVertex2f(31.5f + (ship_x / 2.22), 4.0f + hookLength);
1002    glEnd();
```

```
1003
1004 }
1005
1006 void drawCloud(float x, float y, float radius)
1007 {
1008     const int numCircles = 10;
1009     const float angleIncrement = 2.0f * 3.1416 / numCircles;
1010     const float minRadius = radius * 0.5f;
1011     const float maxRadius = radius * 1.5f;
1012
1013     glColor3f(1.0f, 1.0f, 1.0f);
1014
1015     for (int i = 0; i < numCircles; ++i)
1016     {
1017         const float angle = i * angleIncrement;
1018         const float xOffset = (minRadius + (maxRadius - minRadius) *
              (i / float(numCircles))) * cos(angle);
1019         const float yOffset = (minRadius + (maxRadius - minRadius) *
              (i / float(numCircles))) * sin(angle);
1020         const float cloudX = x + xOffset;
1021         const float cloudY = y + yOffset;
1022
1023         glBegin(GL_TRIANGLE_FAN);
1024         glVertex2f(cloudX, cloudY);
1025
1026         const int numSegments = 100;
1027         const float angleIncrementSegment = 2.0f * 3.1416 /
              numSegments;
1028
1029         for (int j = 0; j <= numSegments; ++j)
1030         {
1031             const float segmentAngle = j * angleIncrementSegment;
1032             const float segmentX = cloudX + radius * cos
                  (segmentAngle);
1033             const float segmentY = cloudY + radius * sin
                  (segmentAngle);
1034             glVertex2f(segmentX, segmentY);
1035         }
1036
1037         glEnd();
1038     }
1039 }
1040
1041 void seaweed(float x, float y, float radius)
1042 {
1043     const int numCircles = 10;
1044     const float angleIncrement = 2.0f * 3.1416 / numCircles;
1045     const float minRadius = radius * 0.5f;
1046     const float maxRadius = radius * 1.5f;
1047
1048     glColor3f(0.0f, 0.7f, 0.0f);
1049
1050     for (int i = 0; i < numCircles; ++i)
```

```
1051      {
1052          const float angle = i * angleIncrement;
1053          const float xOffset = (minRadius + (maxRadius - minRadius) *
              (i / float(numCircles))) * cos(angle);
1054          const float yOffset = (minRadius + (maxRadius - minRadius) *
              (i / float(numCircles))) * sin(angle);
1055          const float cloudX = x + xOffset;
1056          const float cloudY = y + yOffset;
1057
1058          glBegin(GL_TRIANGLE_FAN);
1059          glVertex2f(cloudX, cloudY);
1060
1061          const int numSegments = 100;
1062          const float angleIncrementSegment = 2.0f * 3.1416 /
              numSegments;
1063
1064          for (int j = 0; j <= numSegments; ++j)
1065          {
1066              const float segmentAngle = j * angleIncrementSegment;
1067              const float segmentX = cloudX + radius * cos
                  (segmentAngle);
1068              const float segmentY = cloudY + radius * sin
                  (segmentAngle);
1069              glVertex2f(segmentX, segmentY);
1070          }
1071
1072          glEnd();
1073      }
1074  }
1075
1076  void display(void)
1077  {
1078      glClear(GL_COLOR_BUFFER_BIT);
1079      glMatrixMode(GL_MODELVIEW);
1080      glLoadIdentity();
1081
1082      glPushMatrix();
1083      circle(25, 240, 170, 30);
1084      glPopMatrix();
1085
1086      glPushMatrix();
1087      man();
1088      glPopMatrix();
1089
1090      glPushMatrix();
1091      glTranslatef(0, 70, 0.0);
1092      ship();
1093      glPopMatrix();
1094
1095      glPushMatrix();
1096      sea();
1097      glPopMatrix();
1098
```

```
1099      glPushMatrix();
1100      seaweed(900.0f, -155.0f, 17.0);
1101      seaweed(700.0f, -155.0f, 12.0);
1102      seaweed(500.0f, -155.0f, 17.0);
1103      seaweed(300.0f, -155.0f, 12.0);
1104      seaweed(100.0f, -155.0f, 20.0);
1105      glPopMatrix();
1106
1107      glPushMatrix();
1108      sand();
1109      glPopMatrix();
1110
1111      glPushMatrix();
1112      sun();
1113      glPopMatrix();
1114
1115      glPushMatrix();
1116      DrawFish8();
1117      glPopMatrix();
1118
1119      glPushMatrix();
1120      DrawFish2();
1121      glPopMatrix();
1122
1123      glPushMatrix();
1124      DrawFish5();
1125      glPopMatrix();
1126
1127      glPushMatrix();
1128      DrawFish6();
1129      glPopMatrix();
1130
1131      glPushMatrix();
1132      DrawFish7();
1133      glPopMatrix();
1134
1135      glPushMatrix();
1136      DrawFish4();
1137      glPopMatrix();
1138
1139      glPushMatrix();
1140      DrawFish3();
1141      glPopMatrix();
1142
1143      glPushMatrix();
1144      DrawFish1();
1145      glPopMatrix();
1146
1147      glPushMatrix();
1148      hook();
1149      glPopMatrix();
1150
1151      glPushMatrix();
```

```
1152      drawCloud(900.0f, 455.0f, 30.0); // Bulutu çizmek için
            koordinatları ve yarıçap
1153      drawCloud(840.0f, 455.0f, 30.0);
1154      drawCloud(600.0f, 455.0f, 30.0); // Bulutu çizmek için
            koordinatları ve yarıçap
1155      drawCloud(540.0f, 455.0f, 30.0);
1156      glPopMatrix();
1157
1158      glFlush();
1159      glutSwapBuffers();
1160  }
1161
1162  void key(int key, int x, int y)
1163  {
1164      switch (key)
1165      {
1166      case GLUT_KEY_LEFT:
1167          if (ship_x <= -3.0)
1168          {
1169              ship_x -= 0;
1170          }
1171          else
1172          {
1173              ship_x -= 2.0;
1174          }
1175          glutPostRedisplay();
1176          break;
1177      case GLUT_KEY_RIGHT:
1178          if (ship_x >= 35)
1179          {
1180              ship_x += 0;
1181          }
1182          else
1183          {
1184              ship_x += 2.0;
1185          }
1186          glutPostRedisplay();
1187          break;
1188      case GLUT_KEY_DOWN:
1189          if(hookLength>-12.5)
1190              hookLength -= hookSpeed;
1191          break;
1192      case GLUT_KEY_UP:
1193              if(hookLength<=-0.5)
1194              hookLength += hookSpeed;
1195          break;
1196          glutPostRedisplay();
1197      default:
1198          break;
1199      }
1200  }
1201
1202  void init(void)
```

```
1203 {
1204     glClearColor(0.1f, 0.9f, 0.9f, 0.1f);
1205     glColor3f(1.0, 0.0, 0.0);
1206     glPointSize(1.0);
1207     glMatrixMode(GL_PROJECTION);
1208     glLoadIdentity();
1209     gluOrtho2D(0.0, 999.0, -200.0, 500.0);
1210 }
1211
1212 int main(int argc, char** argv)
1213 {
1214     glutInit(&argc, argv);
1215     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
1216     glutInitWindowSize(800, 600);
1217     glutInitWindowPosition(200, 100);
1218     glutCreateWindow("FİSHİNG");
1219     init();
1220     glutDisplayFunc(display);
1221     glutSpecialFunc(key);
1222     glutTimerFunc(25, update, 0);
1223     glutMainLoop();
1224     return 0;
1225 }
```