

GİRİŞ

Projeyi ben Eren Ertürk ve arkadaşım Ahmet Yılmaz birlikte yaptık. Öğrenci numaralarımız sırasıyla 23360859742 ve 22360859044. Yaptığımız görev dağılımını aşağıdaki listede bulabilirsiniz.

Eren Ertürk:

- Menü arayüzü (menu_yazdir, kamp_alani_alt_secenekler, sifahaneye_git_alt_secenekler, hana_git_alt_secenekler, maceraya_atil_alt_secenekler, durumu_goster, savas_becerileri,)
- Fonksiyonlar (hana_git, sifahaneye_git, kampa_git, eglenme_degeri, moral_degeri)
- Raporlama

Ahmet Yılmaz:

- Struct Yapıları
- Main Fonksiyonu
- Fonksiyonlar (uyu, karizma, can, tokluk, su_degeri, hijyen, para, tecrube, guc, ceviklik, dayaniklilik, toplayicilik_becerisi, maceraya_atil, seviye_atla, savas_becerileri, savas, ozanin_savas_secimi)
- Raporlama

TEKNİK DETAYLAR

```
4 typedef struct {
5     char ozan_adi[20];
6     char calgi_adi[20];
7     int para_degeri;
8     int tecrube_degeri;
9     int can_degeri;
10    int tokluk_degeri;
11    int su_degeri;
12    int uyku_degeri;
13    int hijyen_degeri;
14    int eglenme_degeri;
15    int moral_degeri;
16    int toplayicilik_becerisi_degeri;
17
18    int guc_degeri;
19    int ceviklik_degeri;
20    int dayaniklilik_degeri;
21    int karizma_degeri;
22 } Ozan;
23
24 typedef struct {
25     int can_degeri;
26     int guc_degeri;
27     int ceviklik_degeri;
28     int dayaniklilik_degeri;
29 } Haydut;
```

Burada struct yapılarıyla birlikte “Ozan” ve “Haydut” karakterlerinin özelliklerini tanımladık,

bu özellikler mental ve fiziksel durumları tutuyor.

```
31 void menu_yazdir(int* secim);
32 void kamp_alani_alt_secenekler();
33 void sifahaneye_git_alt_secenekler();
34 void hana_git_alt_secenekler();
35 void maceraya_atil_alt_secenekler();
36 void durumu_goster(Ozan* ozan);
37 void uyu(Ozan* ozan, int saat);
38 void karizma(Ozan* ozan, int miktar);
39 void can(Ozan* ozan, int miktar);
40 void tokluk(Ozan* ozan, int miktar);
41 void su_degeri(Ozan* ozan, int miktar);
42 void hijyen(Ozan* ozan, int miktar);
43 void para(Ozan* ozan, int miktar);
44 void tecrube(Ozan* ozan, int miktar);
45 void guc(Ozan* ozan, int miktar);
46 void ceviklik(Ozan* ozan, int miktar);
47 void dayaniklilik(Ozan* ozan, int miktar);
48 void toplayicilik_becerisi(Ozan* ozan, int miktar);
49 void eglence_degeri(Ozan* ozan, int sure);
50 void moral_degeri(Ozan* ozan, int miktar);
51 void kampa_git(Ozan* ozan, int* secim);
52 void sifahaneye_git(Ozan* ozan, int* secim);
53 void hana_git(Ozan* ozan, int* secim);
54 void maceraya_atil(Ozan* ozan, Haydut* haydut, int* secim);
55 void seviye_atla(Ozan* ozan);
56 void savas_becerileri(Ozan* ozan, Haydut* haydut);
57 void savas(Ozan* ozan, Haydut* haydut, int* ozanin_verdigi_hasar, int* ozanin_aldigi_hasar);
58 void ozanin_savas_secimi(Ozan* ozan, Haydut* Haydut);
59
```

Bu kod parçasında ileride kullanacağımız fonksiyonların prototipleri bulunmakta.

```
60 void main() {
61     Haydut* haydut = (Haydut*)malloc(sizeof(Haydut));
62     Ozan* ozan = (Ozan*)malloc(sizeof(Ozan));
63
64     printf("Ozan'in adini giriniz: ");
65     gets(ozan->ozan_adi);
66
67     printf("Ozan'in caldigi calginin adi: ");
68     gets(ozan->calgi_adi);
69
70     ozan->para_degeri = 10;
71     ozan->tecrube_degeri = 50;
72     ozan->can_degeri = 100;
73     ozan->tokluk_degeri = 50;
74     ozan->su_degeri = 50;
75     ozan->hijyen_degeri = 50;
76     ozan->uyku_degeri = 50;
77     ozan->eglence_degeri = 50;
78     ozan->moral_deger = 50;
79     ozan->toplayicilik_becerisi_degeri = 50;
80
81     ozan->guc_degeri = 3;
82     ozan->ceviklik_degeri = 3;
83     ozan->dayaniklilik_degeri = 3;
84     ozan->karizma_degeri = 3;
85
86     srand(time(NULL));
```

Burada haydut ve ozan pointerları oluşturduk ve malloc fonksiyonu ile her ikisi için de bellek ayrılıyor. Kullanıcıdan ozanın ve ozanın çaldığı çalgının isimleri isteniyor. Ozana başlangıç değerleri atanıyor. Srand kullanımıyla birlikte rasgeleliği sağlıyoruz.

```

28     int secim;
29     while (1) {
30         menu_yazdir(&secim);
31         if(secim == 1) {
32             kampa_git(ozan, &secim);
33             durumu_goster(ozan);
34         }
35         else if(secim == 2) {
36             sifahaneye_git(ozan, &secim);
37         }
38         else if(secim == 3) {
39             hana_git(ozan, &secim);
40         }
41         else if(secim == 4) {
42             maceraya_atil(ozan, haydut, &secim);
43         }
44         else if (secim == 5) {
45             durumu_goster(ozan);
46         }
47         else if (secim == 6) {
48             durumu_goster(ozan);
49         }
50         else if (secim == 7) {
51             printf("Cikmak istediginizden emin misiniz?\ncikmak icin tekrar 7'ye basin: ");
52             scanf("%d", &secim);
53             if (secim == 7)
54                 break;
55         }
56         else{
57             printf("\ngecersiz secim yaptiniz yeniden seciniz.\n");
58             menu_yazdir(&secim);
59         }
60         if(ozan->tokluk_degeri == 0) {
61             ozan->can_degeri -= 20;
62         }
63         if(ozan->su_degeri == 0) {
64             ozan->can_degeri -= 15;
65         }
66         if (ozan->uyku_degeri == 0) {
67             ozan->can_degeri -= 10;
68         }
69     }
70     free(haydut);
71     free(ozan);
72 }

```

Burada sonsuz döngüyle birlikte menü arayüzünü çağırıyoruz ve kullanıcıdan seçim yapmasını bekliyoruz. Seçimlerine göre fonksiyonlarımızı çağırıyoruz, kullanıcı menüden çıkmak istediğinde kontrol gerçekleştiriliyor. Geçersiz seçimde hata mesajı alınıyor. Tokluk değeri, su değeri, uyku değerleri sıfırlandığındaki eksilen canlar tanımlanmış. Free fonksiyonuyla bellek temizlenmiş.

```

136 void kampa_git(Ozan* ozan, int* secim) {
137     int uyku_miktari;
138     printf("Kamp alanina gidiyorsunuz...\n");
139     kampa_alani_alt_secenekler();
140     int alt_secim;
141     printf("Kamp alanında ne yapmak istiyorsunuz? ");
142     scanf("%d", &alt_secim);
143     switch (alt_secim) {
144     case 1:
145         tecrube(ozan, 10);
146         karizma(ozan, 10);
147         break;
148     case 2:
149         can(ozan, 10);
150         tecrube(ozan, 10);
151         break;
152     case 3:
153         printf("Ozan ne kadar uyusun: ");
154         scanf("%d", &uyku_miktari);
155         uyu(ozan, uyku_miktari);
156         guc(ozan, 10);
157         ceviklik(ozan, 10);
158         dayaniklilik(ozan, 10);
159         break;
160     case 4:
161         menu_yazdir(secim);
162         break;
163     }
164 }
165

```

Burada kamp alanına gidilmek istendiğinde kullanıcıya “Kamp alanına gidiyorsunuz” mesajı iletilir. Kullanıcıdan seçim alınır ve çalgı çalma, nehirde yıkanma, uyuma, menüye dönme işlemlerinden biri gerçekleştirilir.

```

166 void sifahaneye_git(Ozan* ozan, int* secim) {
167     if (*secim == 2) {
168         printf("sifahaneye gidiyorsunuz...");
169         sifahaneye_git_alt_secenekler();
170         int alt_secim;
171         printf("sifahane de ne yapmak istiyorsunuz? ");
172         scanf("%d", &alt_secim);
173         switch (alt_secim) {
174             case 1:
175                 can(ozan, 14);
176                 ceviklik(ozan, 10);
177                 dayaniklilik(ozan, 10);
178                 guc(ozan, 10);
179                 para(ozan, -10);
180                 moral_degeri(ozan, 10);
181                 break;
182             case 2:
183                 can(ozan, 21);
184                 ceviklik(ozan, 10);
185                 dayaniklilik(ozan, 10);
186                 guc(ozan, 10);
187                 para(ozan, -20);
188                 moral_degeri(ozan, 10);
189                 break;
190             case 3:
191                 menu_yazdir(secim);
192                 break;
193         }
194     }
195 }

```

Burada şifahaneye git seçildiğinde “Şifahaneye gidiliyor” mesajı iletiliyor. Kullanıcıdan seçim alınıyor. Switch case yapısıyla okunan veriye göre can, çeviklik, dayanıklılık, para, moral değeri güncellenir ya da ana menüye geri dönülür.

```

197 void hana_git(Ozan* ozan, int* secim) {
198     if (*secim == 3) {
199         printf("hana gidiyorsunuz...");
200         hana_git_alt_secenekler();
201         int alt_secim;
202         printf("han da ne yapmak istiyorsunuz? ");
203         scanf("%d", &alt_secim);
204         switch (alt_secim) {
205             case 1: //Yiyecek satın al ve ye. (-para, can degeri, cevikli
206                 para(ozan, -10);
207                 tokluk(ozan, 10);
208                 can(ozan, 10);
209                 ceviklik(ozan, 10);
210                 dayaniklilik(ozan, 10);
211                 guc(ozan, 10);
212                 break;
213             case 2: //İçecek satın al, iç ve eğlen. (-para, can degeri, c
214                 para(ozan, -10);
215                 can(ozan, 10);
216                 ceviklik(ozan, 10);
217                 dayaniklilik(ozan, 10);
218                 tokluk(ozan, 10);
219                 guc(ozan, 10);
220                 karizma(ozan, 10);
221                 break;
222             case 3: //Enstrüman çal ve şarkı söyle. (karizma, tecrube, su
223                 if (ozan->hijyen_degeri < 20) {
224                     printf("hijyene onem vermediginiz icin handa sarki soyleyemezsiniz.");
225                     break;
226                 }
227                 ozan->para_degeri = (10 + ((ozan->karizma_degeri) * (ozan->hijyen_degeri) / 100));
228                 tecrube(ozan, 20);
229                 karizma(ozan, 10);
230             case 4:
231                 menu_yazdir(secim);
232                 break;
233         }
234     }
235 }

```

Burada “hana git” seçeneği seçilirse çalışacak kodlar yazıldı. “Hana git” seçeneği seçildiğinde “Hana gidiliyor” mesajı yazılır. Sonrasında kullanıcının seçimi okunur. Seçimine göre ozan karakterinin değerleri değişir. Ozan, enstrüman çalma seçeneğinde hijyen değeri yeterli değilse handa şarkı söyleyemez. 4. seçenekte menüye geri dönülür.

```

237 void maceraya_atil(Ozan* ozan, Haydut* haydut, int* secim) {
238     double kolay_haydut_ganimeti = 15 + (rand() % 10);
239     double orta_haydut_ganimeti = 30 + (rand() % 20);
240     double zor_haydut_ganimeti = 55 + (rand() % 20);
241     int random_sayi_sifali_bitki = rand() % 100; // bu sayı şifalı bitki bulma şansını
242     int random_sayi_meyve = rand() % 100; // bu sayı meyve bulma şansını kontrol etmek i
243     int random_hayvan_bulma = rand() % 100;
244     double sifali_bitki_bulma_sansi = (ozan->toplayicilik_becerisi_degeri * 4) / 100;
245     if (*secim == 4) {
246         printf("maceraya katiliyorsunuz...");
247         maceraya_atil_alt_secenekler();
248         int alt_secim;
249         printf("nasıl bir maceraya katılmak istiyorsunuz? ");
250         scanf("%d", &alt_secim);
251         switch (alt_secim) {
252             case 1: // Yakın çevreden şifalı bitki
253                 if ((sifali_bitki_bulma_sansi * 100) > random_sayi_sifali_bitki)
254                     ozan->can_degeri += 10;
255
256                 if ((sifali_bitki_bulma_sansi * 100) > random_sayi_meyve)
257                     ozan->tokluk_degeri += 10;
258
259                 if ((sifali_bitki_bulma_sansi * 100 / 2) > random_hayvan_bulma)
260                     ozan->tokluk_degeri += 50;
261
262                 toplayicilik_becerisi(ozan, 10);
263                 can(ozan, 10);
264                 tecrube(ozan, 10);
265                 dayaniklilik(ozan, 10);
266                 ceviklik(ozan, 10);
267                 tokluk(ozan, 10);
268                 uyu(ozan, -10);
269                 guc(ozan, 10);
270                 break;
271             case 2: // Ormanı kesfe çık (kolay).
272                 haydut->guc_degeri = (rand() % 2) + 1;
273                 haydut->dayaniklilik_degeri = (rand() % 2) + 4;
274                 haydut->ceviklik_degeri = (rand() % 2) + 4;
275                 savas_becerileri(ozan, haydut);
276                 ozanin_savas_secimi(ozan, haydut);
277                 tecrube(ozan, 30);
278                 para(ozan, kolay_haydut_ganimeti);
279                 break;
280             case 3: // Kavalıkları kesfe çık (orta
281                 haydut->guc_degeri = (rand() % 2) + 4;
282                 haydut->dayaniklilik_degeri = (rand() % 2) + 4;
283                 haydut->ceviklik_degeri = (rand() % 2) + 4;
284                 savas_becerileri(ozan, haydut);
285                 ozanin_savas_secimi(ozan, haydut);
286                 tecrube(ozan, 60);
287                 para(ozan, orta_haydut_ganimeti);
288                 break;
289             case 4: // Vadiyi kesfe çık (zor). (te
290                 haydut->guc_degeri = (rand() % 3) + 7;
291                 haydut->dayaniklilik_degeri = (rand() % 3) + 7;
292                 haydut->ceviklik_degeri = (rand() % 3) + 7;
293                 savas_becerileri(ozan, haydut);
294                 ozanin_savas_secimi(ozan, haydut);
295                 tecrube(ozan, 90);
296                 para(ozan, zor_haydut_ganimeti);
297                 break;
298             case 5:
299                 menu_yazdir(secim);
300                 break;
301         }
302     }
303 }
304

```

Burada maceraya atılma seçeneğini kodladık. “Maceraya katılıyorsunuz” ve “Nasıl bi maceraya katılmak istiyorsun” metinlerinden sonra kullanıcıdan seçim alınır. Ozan 1. seçeneği seçerse şifalı

bitki bulma şansı olasılığı, meyve bulma şansı, hayvan avlama şansı hesaplanır. Olasılığın sonuçlarına göre değerler güncellenir. 2. seçenek seçilirse ozan orman keşfine çıkar ve haydutla savaşı (haydutun güç, dayanıklılık, çeviklik değerleri rastgele belirlenir.). Savaş sonucunda değerler güncellenir. 3. seçimde kayalık keşfi yapılır, haydutun değerleri rastgele belirlenir, savaş sonrası değerler güncellenir. 4. seçimde vadi keşfi yapılır, haydutun değerleri rastgele belirlenir, savaş sonrası değerler güncellenir. 5. seçimde menüye dönlür.

```

305 void seviye_atla (Ozan* ozan) {
306     if (ozan->tecrube_degeri == 100) {
307         ozan->tecrube_degeri = 3;
308         ozan->dayaniklilik_degeri = 3;
309         ozan->guc_degeri = 3;
310         ozan->karizma_degeri = 3;
311         ozan->toplayicilik_becerisi_degeri = 3;
312     }
313     int secim, miktar, toplam_puan;
314     while (toplam_puan != 0) {
315         printf("-----SEVIYE ATLAMA MENUSU-----\n\n");
316         printf("Seviye atladiniz tebrikler! 5 puan kazandiniz. Puanlariniz:");
317         printf("1- tecrube\n2- dayaniklilik\n3- guc\n4- karizma\n5- toplayicilik\n");
318         scanf("%d", &secim);
319         switch(secim) {
320             case 1:
321                 scanf("%d", &miktar);
322                 if((miktar >= 0) && (miktar <= 5)){
323                     ozan->tecrube_degeri += miktar;
324                     toplam_puan -= miktar;
325                 }
326                 else
327                     printf("gecersiz deger girdiniz.");
328                 break;
329             case 2:
330                 scanf("%d", &miktar);
331                 if((miktar >= 0) && (miktar <= 5)){
332                     ozan->tecrube_degeri += miktar;
333                     toplam_puan -= miktar;
334                 }
335                 else
336                     printf("gecersiz deger girdiniz.");
337                 break;
338             case 3:
339                 scanf("%d", &miktar);
340                 if((miktar >= 0) && (miktar <= 5)){
341                     ozan->tecrube_degeri += miktar;
342                     toplam_puan -= miktar;
343                 }
344                 else
345                     printf("gecersiz deger girdiniz.");
346                 break;
347             case 4:
348                 scanf("%d", &miktar);
349                 if((miktar >= 0) && (miktar <= 5)){
350                     ozan->tecrube_degeri += miktar;
351                     toplam_puan -= miktar;
352                 }
353                 else
354                     printf("gecersiz deger girdiniz.");
355                 break;
356             case 5:
357                 scanf("%d", &miktar);
358                 if((miktar >= 0) && (miktar <= 5)){
359                     ozan->tecrube_degeri += miktar;
360                     toplam_puan -= miktar;
361                 }
362                 else
363                     printf("gecersiz deger girdiniz.");
364                 break;
365             default:
366                 printf("gecersiz islem yaptiniz.");
367                 break;
368         }
369     }
370 }

```


Üstteki kodlarda kullanıcıya seviyesi arttığında kazandığı puanları nereye harcayacağı soruluyor. Switch case yapısıyla artırılacak miktar alınır ve toplam puandan düşülür.

```
377 void kamp_alani_alt_secenekler() {
378     printf("1. Kamp atesinin basinda calgi cal/sarki soyle.\n");
379     printf("2. Nehirde yikan.\n");
380     printf("3. cadirina girip uyu.\n");
381     printf("4. Koy meydanina don.\n");
382 }
383 void sifahaneye_git_alt_secenekler() {
384     printf("1. sifacidan yaralarini sarmasini iste.\n");
385     printf("2. sifacidan merhem yapip surmesini iste.\n");
386     printf("3. Koy meydanina don.\n");
387 }
388 void hana_git_alt_secenekler() {
389     printf("1. Yiyecek satin al ve ye.\n");
390     printf("2. ıcecek satin al, ıc ve eglen.\n");
391     printf("3. Enstruman cal ve sarki soyle.\n");
392     printf("4. Koy meydanina don.\n");
393 }
394 void maceraya_atil_alt_secenekler() {
395     printf("1. Yakın cevreden sifali bitki topla ve avlan.\n");
396     printf("2. Ormani kesfe cik (kolay).\n");
397     printf("3. Kayaliklari kesfe cik (orta).\n");
398     printf("4. Vadiyi kesfe cik (zor).\n");
399     printf("5. Koy meydanina don.\n");
400 }
401 void durumu_goster(Ozan* ozan) {
402     printf("isim: %s\n", ozan->ozan_adi);
403     printf("calgi adi: %s\n", ozan->calgi_adi);
404     printf("para: %d\n", ozan->para_degeri);
405     printf("tecrube: %d\n", ozan->tecrube_degeri);
406     printf("can: %d\n", ozan->can_degeri);
407     printf("tokluk: %d\n", ozan->tokluk_degeri);
408     printf("su: %d\n", ozan->su_degeri);
409     printf("hijyen: %d\n", ozan->hijyen_degeri);
410     printf("uyku: %d\n", ozan->uyku_degeri);
411     printf("eglence: %d\n", ozan->eglence_degeri);
412     printf("moral: %d\n", ozan->moral_deger);
413     printf("guc: %d\n", ozan->guc_degeri);
414     printf("ceviklik: %d\n", ozan->ceviklik_degeri);
415     printf("dayanıklılık: %d\n", ozan->dayanıklılık_degeri);
416     printf("karizma: %d\n", ozan->karizma_degeri);
417     printf("toplayıcılık becerisi: %d\n", ozan->toplayıcılık_becerisi_degeri);
418 }
```

Buradaki kodda menü yapısını fonksiyonlara atadık.

```

420 void uyu(Ozan* ozan, int saat) {
421     if(ozan->uyku_degeri > 100) {
422         printf("uyku seviyesi 100'den büyük olamaz.\n");
423     }
424     else {
425         ozan->uyku_degeri += 10 * saat;
426         ozan->tokluk_degeri -= 4 * saat;
427         ozan->su_degeri -= 3 * saat;
428         if(ozan->uyku_degeri > 100) {
429             ozan->uyku_degeri = 100;
430         }
431     }
432 }

```

Bu kodda uyu fonksiyonu tanımlandı. Uyu fonksiyonunda ozanın uyku seviyesi artarken tokluk ve su değerleri azalır.

```

433 void karizma(Ozan* ozan, int miktar) {
434     if (ozan->karizma_degeri > 25) {
435         printf("karizma degeri 25'den büyük olamaz.\n");
436     }
437     else {
438         ozan->karizma_degeri += miktar;
439         if (ozan->karizma_degeri > 25) {
440             ozan->karizma_degeri = 25;
441         }
442     }
443 }

```

Bu kodda karizma fonksiyonu tanımlandı. Fonksiyonda karizma değerinin 25ten büyük olup olmadığı kontrol edilmiş ve sonrasında değerde güncelleme yapılmış.

```

444 void can(Ozan* ozan, int miktar) {
445     if (ozan->can_degeri > 100) {
446         printf("saglik degeri 100'den büyük olamaz.\n");
447     }
448     else {
449         ozan->can_degeri += miktar;
450         if (ozan->can_degeri > 100) {
451             ozan->can_degeri = 100;
452         }
453     }
454 }

```

Bu kodda can fonksiyonu yazılmış. Can değeri 100'den büyük mü diye kontrol edilmiş ve can değeri güncellenmiş.

```

455 void tokluk(Ozan* ozan, int miktar) {
456     int secim;
457     printf("acliği nasıl gidermek istersiniz?\n1. ekmek ye (2 altın, +10 tokluk)\n2. yahni ye (10 altın, +50 tokluk)\n3. elma ye (1 altın, +5 t
458     scanf("%d", &secim);
459     switch (secim) {
460         case 1:
461             ozan->para_degeri -= 2 * miktar;
462             ozan->tokluk_degeri += 10 * miktar;
463             break;
464         case 2:
465             ozan->para_degeri -= 10 * miktar;
466             ozan->tokluk_degeri += 50 * miktar;
467             break;
468         case 3:
469             ozan->para_degeri -= 1 * miktar;
470             ozan->tokluk_degeri += 5 * miktar;
471         default:
472             printf("gecersiz işlem yaptınız.");
473     }
474     if (ozan->tokluk_degeri > 100) {
475         printf("acliği değeri 100'den büyük olamaz.\n");
476     }
477     else {
478         ozan->uyku_degeri -= 10 * miktar;
479         if (ozan->tokluk_degeri > 100) {
480             ozan->tokluk_degeri = 100;
481         }
482     }
483 }

```

Bu kodda tokluk fonksiyonu tanımlanmış. Açlığı gidermek için kullanıcıya seçenekler sunuluyor, seçeneklere göre para ve tokluk değeri güncelleniyor. Açlık değerinin 100'den büyük olup olmadığı kontrol ediliyor.

```

484 void su_degeri(Ozan* ozan, int miktar) {
485     if (ozan->su_degeri > 100) {
486         printf("su değeri 100'den büyük olamaz.\n");
487     }
488     else {
489         ozan->su_degeri += miktar;
490         ozan->tokluk_degeri -= 10; // su içmek acıktırır.
491         if (ozan->su_degeri > 100) {
492             ozan->su_degeri = 100;
493         }
494     }
495 }

```

Burada su_degeri fonksiyonu tanımlandı. Su değeri 100'den büyük olup olmadığı kontrol edildikten sonra, su değeri ve tokluk değeri güncellenir.

```

497 void eglenme_degeri(Ozan* ozan, int sure) {
498     if (ozan->eglenme_degeri > 100) {
499         printf("eglenme değeri 100'den büyük olamaz.\n");
500     }
501     else {
502         ozan->eglenme_degeri += sure * 10;
503         ozan->tokluk_degeri -= sure * 5;
504         ozan->uyku_degeri -= sure * 5;
505         if (ozan->eglenme_degeri > 100) {
506             ozan->eglenme_degeri = 100;
507         }
508     }
509 }
510

```

Burada eğlence değeri fonksiyonu tanımlandı. Eğlence değerinin 100'den büyük olup olmadığı sorgulandıktan sonra eğlence değeri, tokluk değeri, uyku değeri güncellenir.

```

511 void moral_degeri(Ozan* ozan, int miktar) {
512     if (ozan->moral_deger > 100) {
513         printf("moral degeri 100'den büyük olamaz.\n");
514     }
515     else {
516         ozan->moral_deger += miktar;
517         if (ozan->moral_deger > 100) {
518             ozan->moral_deger = 100;
519         }
520     }
521 }
522

```

Burada moral değeri fonksiyonu tanımlanmış. Kontrolten sonra değerler güncellenir.

```

523 void hijyen(Ozan* ozan, int miktar) {
524     if (ozan->hijyen_degeri > 100) {
525         printf("hijyen degeri 100'den büyük olamaz.\n");
526     }
527     else {
528         ozan->hijyen_degeri += miktar;
529         if (ozan->hijyen_degeri > 100) {
530             ozan->hijyen_degeri = 100;
531         }
532     }
533 }
534 void para(Ozan* ozan, int miktar) {
535     ozan->para_degeri += miktar;
536 }
537 void tecrube(Ozan* ozan, int miktar) {
538     if (ozan->tecrube_degeri > 100) {
539         printf("tecrube degeri 100'den büyük olamaz.\n");
540     }
541     else {
542         ozan->tecrube_degeri += miktar;
543         if (ozan->tecrube_degeri > 100) {
544             ozan->tecrube_degeri = 100;
545         }
546     }
547     if (ozan->tecrube_degeri == 100) {
548         seviye_atla(ozan);
549     }
550 }

```

Burada para, hijyen, tecrübe fonksiyonları tanımlanıyor. Kontrolten sonra değerler güncelleniyor.

```

551 void guc(Ozan* ozan, int miktar) {
552     if (ozan->guc_degeri > 25) {
553         printf("guc degeri 25'den büyük olamaz.\n");
554     }
555     else {
556         ozan->guc_degeri += miktar;
557         if (ozan->guc_degeri > 25) {
558             ozan->guc_degeri = 25;
559         }
560     }
561 }
562 void ceviklik(Ozan* ozan, int miktar) {
563     if (ozan->ceviklik_degeri > 25) {
564         printf("ceviklik degeri 25'den büyük olamaz.\n");
565     }
566     else {
567         ozan->ceviklik_degeri += miktar;
568         if (ozan->ceviklik_degeri > 25) {
569             ozan->ceviklik_degeri = 25;
570         }
571     }
572 }
573 void dayaniklilik(Ozan* ozan, int miktar) {
574     if (ozan->dayaniklilik_degeri > 25) {
575         printf("dayaniklilik degeri 25'den büyük olamaz.\n");
576     }
577     else {
578         ozan->dayaniklilik_degeri += miktar;
579         if (ozan->dayaniklilik_degeri > 25) {
580             ozan->dayaniklilik_degeri = 25;
581         }
582     }
583 }

```

Burada güç, çeviklik, dayanıklılık fonksiyonları tanımlandı. Kontrollerden sonra değerler güncellenir.

```

573 void dayaniklilik(Ozan* ozan, int miktar) {
574     if (ozan->dayaniklilik_degeri > 25) {
575         printf("dayaniklilik degeri 25'den büyük olamaz.\n");
576     }
577     else {
578         ozan->dayaniklilik_degeri += miktar;
579         if (ozan->dayaniklilik_degeri > 25) {
580             ozan->dayaniklilik_degeri = 25;
581         }
582     }
583 }

```

Dayanıklılık fonksiyonu tanımlandı. Kontrolde sonra değerler güncellendi.

```

596 void savas_becerileri(Ozan* ozan, Haydut* haydut) {
597     printf("%s'in savas becerileri\n", ozan->ozan_adi);
598     printf("guc degeri: %d\ndayaniklilik degeri: %d\nceviklik degeri: %d\n", ozan->guc_degeri, ozan->dayaniklilik_degeri, ozan->ceviklik_degeri);
599     printf("haydutun savas becerileri\n");
600     printf("guc degeri: %d\ndayaniklilik degeri: %d\nceviklik degeri: %d", haydut->guc_degeri, haydut->dayaniklilik_degeri, haydut->ceviklik_degeri);
601 }

```

Buradaki kodda ozan ve haydutun savaş becerileri ekrana yazdırılıyor.

```

603 void savas(Ozan* ozan, Haydut* haydut, int* ozanin_verdigi_hasar, int* ozanin_aldigi_hasar) {
604     while((ozan->can_degeri > 0) || (haydut->can_degeri > 0)) {
605         if(ozan->ceviklik_degeri > haydut->ceviklik_degeri) {
606             *ozanin_verdigi_hasar = ozan->guc_degeri * 4;
607         } else if(ozan->ceviklik_degeri < haydut->ceviklik_degeri) {
608             *ozanin_aldigi_hasar = (haydut->guc_degeri * 4) - (haydut->guc_degeri * 4 * 4 * ozan->dayaniklilik_degeri / 100 - 1);
609             ozan->can_degeri -= *ozanin_aldigi_hasar;
610         }
611     }
612 }

```

Burada savas fonksiyonu yazıldı. Savaş simülasyonunu gerçekleştirmek için kullanılıyor.

```

614 void ozanin_savas_secimi(Ozan* ozan, Haydut* haydut) {
615     int savasma_secimi;
616     int ozanin_aldigi_hasar;
617     int ozanin_verdigi_hasar;
618     int ozanin_kacma ihtimali;
619     int random_ozanin_kacma ihtimali; // bu sayı ozanin kacabilme ihtimali için random oluşturulacaktır.
620     printf("1-savas\n2-kac\n");
621     scanf("%d", &savasma_secimi);
622     switch(savasma_secimi) {
623         case 1:
624             savas(ozan, haydut, &ozanin_verdigi_hasar, &ozanin_aldigi_hasar);
625             break;
626         case 2:
627             ozanin_kacma ihtimali = 4 * ozan->ceviklik_degeri / 100;
628             random_ozanin_kacma ihtimali = rand() % 100;
629             if((ozanin_kacma ihtimali * 100) > random_ozanin_kacma ihtimali) {
630                 printf("ozan basariyl kacti");
631             } else {
632                 savas(ozan, haydut, &ozanin_verdigi_hasar, &ozanin_aldigi_hasar);
633             }
634             break;
635     }
636     if(ozan->can_degeri == 0 || ozan->can_degeri < 0) {
637         printf("%s yenildi! Savas bitti.", ozan->ozan_adi);
638     }
639     if(haydut->can_degeri == 0 || haydut->can_degeri < 0) {
640         printf("haydut yenildi! Savas bitti");
641     }
642 }

```

Burada ozanin_savas_secimi fonksiyonu yazıldı. Kaçma ihtimali yüzdesel olarak hesaplandı ve rand() ile karşılaştırıldı. Srand(time(NULL)) ile rastgele sayılar her çalıştırmada farklı üretildi. Savaş veya kaç seçeneği sunan bir döngü yapısı oluşturuldu.

SENARYO

Oyunumuz bir ozanın köy meydanından başlayarak çeşitli yerlerde hayatta kalmaya çalışmasını ve maceralara atılmasını konu alıyor. Oyuncu, ozanın fiziksel, zihinsel ve sosyal ihtiyaçlarını dengede tutarak maceralar yapar, yeteneklerini geliştirir ve haydutlarla savaşıarak ödüller kazanır .

EKSİKLİKLER VE GELİŞTİRMELER

1- Scanf ile giriş alırken tür ve aralık doğrulamaları eklenebilirdi.

Örnek:

```
int secim;
printf("Bir seçenek giriniz: ");
if (scanf("%d", &secim) != 1) {
    printf("Geçersiz giriş! Lütfen bir sayı giriniz.\n");
    while (getchar() != '\n'); // Hatalı girişleri temizle.
}
```

2- Can, tokluk, moral vb. gibi değerleri belirli aralıklarda tutabilirdik.

Örnek:

```
void tokluk(Ozan* ozan, int miktar) {
    ozan->tokluk_degeri += miktar;
    if (ozan->tokluk_degeri > 100) {
        ozan->tokluk_degeri = 100; // Maksimum.
    } else if (ozan->tokluk_degeri < 0) {
        ozan->tokluk_degeri = 0; // Minimum.
    }
}
```

3- Oyunumuza bir kaydetme seçeneği ve geri yükleme seçeneği eklenebilirdi.

SONUÇ

Projemizde fazladan moral seviyesi, eğlence miktarı, su ihtiyacı gibi özellikler ekledik.

KAYNAKÇA

Pointers in C. Tutorialspoint. (n.d.-a).

https://www.tutorialspoint.com/cprogramming/c_pointers.htm

Pointers to structures in C. Tutorialspoint. (n.d.-b).

https://www.tutorialspoint.com/cprogramming/c_pointers_to_structures.htm

Structures in C. Tutorialspoint. (n.d.-c).

https://www.tutorialspoint.com/cprogramming/c_structures.htm