BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü

ALGORİTMALAR VE PROGRAMLAMA DERSİ
PROJE ÖDEVİ RAPORU

NAZMİ CİRİM

21360859069

ŞUBE 2

ARALIK 2023

1. GİRİŞ

Proje bir sanal bebek bakımı uygulamasıdır. Sanal bebeğin tokluk, uyku, sevgi, sosyalleşme, sağlık, eğitim, hijyen, tuvalet ve eğlence gibi ihtiyaçları bulunmaktadır. Sanal bebeğin tüm ihtiyaçları belli başlı fonksiyonlarla giderilmektedir. Ayrıca özel olarak bir doping fonksiyonu tanımlanmıştır ve bu fonksiyon kullanıldığı zaman tüm değerler maksimumuna ulaşmaktadır.

2. İHTİYAÇLAR

Sanal bebeğin tokluk, uyku, sevgi, sosyalleşme, sağlık, eğitim, hijyen, tuvalet ve eğlence ihtiyaçları bulunmaktadır.

2.1. Tokluk İhtiyacı

Tokluk ihtiyacını düzenlemek için yemekYeme() fonksiyonu oluşturuldu. yemekYeme() fonksiyonu, kullanıcıdan yiyecek türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.1.1. Tokluk ihtiyacının sut aksiyonu

Kullanıcı "sut" girdisi verirse, bebeğin tokluk seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.1.2. Tokluk ihtiyacının su aksiyonu

Kullanıcı "su" girdisi verirse, bebeğin tokluk seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.1.3. Tokluk ihtiyacının mama aksiyonu

Kullanıcı "mama" girdisi verirse, bebeğin tokluk seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.2. Uyku İhtiyacı

Uyku ihtiyacını düzenlemek için uyu() fonksiyonu oluşturuldu. uyu() fonksiyonu, kullanıcıdan tekrar miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.3. Sevgi İhtiyacı

Sevgi ihtiyacını düzenlemek için sevgiGoster() fonksiyonu oluşturuldu. sevgiGoster() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.3.1. Sevgi ihtiyacının oksa aksiyonu

Kullanıcı "oksa" girdisi verirse, bebeğin sevgi ve sosyalleşme seviyelerini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.3.2. Sevgi ihtiyacının kucakla aksiyonu

Kullanıcı "kucakla" girdisi verirse, bebeğin sevgi ve sosyalleşme seviyelerini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.3.3. Sevgi ihtiyacının op aksiyonu

Kullanıcı "op" girdisi verirse, bebeğin sevgi ve sosyalleşme seviyelerini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.4. Sosyalleşme İhtiyacı

Sevgi ihtiyacını düzenlemek için sosyallesme() fonksiyonu oluşturuldu. sosyallesme() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.4.1. Sosyalleşme ihtiyacının goz_temasi aksiyonu

Kullanıcı "goz_temasi" girdisi verirse, bebeğin sosyalleşme seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.4.2. Sosyalleşme ihtiyacının gulumse aksiyonu

Kullanıcı "gulumse" girdisi verirse, bebeğin sosyalleşme seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.4.3. Sosyalleşme ihtiyacının sarki_soyle aksiyonu

Kullanıcı "sarki_soyle" girdisi verirse, bebeğin sosyalleşme seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.5. Sağlık İhtiyacı

Sağlık ihtiyacını düzenlemek için saglik() fonksiyonu oluşturuldu. saglik() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.5.1. Sağlık ihtiyacının serum aksiyonu

Kullanıcı " serum" girdisi verirse, bebeğin sağlık seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.5.2. Sağlık ihtiyacının surup aksiyonu

Kullanıcı " surup" girdisi verirse, bebeğin sağlık seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.5.3. Sağlık ihtiyacının asi aksiyonu

Kullanıcı " asi" girdisi verirse, bebeğin sağlık seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.6. Eğitim İhtiyacı

Sevgi ihtiyacını düzenlemek için egitim() fonksiyonu oluşturuldu. egitim() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.6.1. Eğitim ihtiyacının kitap_oku aksiyonu

Kullanıcı "kitap_oku" girdisi verirse, bebeğin eğitim seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık

durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.6.2. Eğitim ihtiyacının müzik aksiyonu

Kullanıcı "muzik" girdisi verirse, bebeğin eğitim seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.6.3. Eğitim ihtiyacının egzersiz aksiyonu

Kullanıcı "egzersiz" girdisi verirse, bebeğin eğitim seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.7. Hijyen İhtiyacı

Hijyen ihtiyacını düzenlemek için hijyen() fonksiyonu oluşturuldu. hijyen() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.7.1. Hijyen ihtiyacının tirnak_kes aksiyonu

Kullanıcı "tirnak_kes" girdisi verirse, bebeğin hijyen seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.7.2. Hijyen ihtiyacının dus_al aksiyonu

Kullanıcı "dus_al" girdisi verirse, bebeğin hijyen seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.7.3. Hijyen ihtiyacının altini_degis aksiyonu

Kullanıcı "altini_degis" girdisi verirse, bebeğin hijyen ve tuvalet seviyelerini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.8. Tuvalet İhtiyacı

Tuvalet ihtiyacını düzenlemek için hijyen() fonksiyonu oluşturuldu. hijyen() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir.

2.8.1. Tuvalet ihtiyacının altini_degis aksiyonu

Kullanıcı "altini_degis" girdisi verirse, bebeğin hijyen ve tuvalet seviyelerini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.9. Eğlence İhtiyacı

Eğlence ihtiyacını düzenlemek için eglence() fonksiyonu oluşturuldu. eglence() fonksiyonu, kullanıcıdan eylem türü ve miktarı alır. Alınan bilgilere göre bebek ihtiyaçları güncellenir

2.9.1. Eğlence ihtiyacının cee aksiyonu

Kullanıcı "cee" girdisi verirse, bebeğin eğlence seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.9.2. Eğlence ihtiyacının gidikla aksiyonu

Kullanıcı "gidikla" girdisi verirse, bebeğin eğlence seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

2.9.3. Eğlence ihtiyacının oyuncak aksiyonu

Kullanıcı "oyuncak" girdisi verirse, bebeğin eğlence seviyesini arttırmak için ilgili ihtiyaçlar güncellenir ve bebek durumu kontrol edilir. Bebeğin tokluk seviyesi, uyku durumu ve sağlık durumu gibi çeşitli koşullar kontrol edilir ve uygun mesajlar ekrana yazdırılır. Ayrıca, bebek ölüm durumu kontrol edilerek program sonlandırılır.

3. ÇALIŞMA DETAYLARI

Bu program, sanal bir bebeğin ihtiyaçlarını takip eder. İhtiyaçlar, ihtiyaçlar[ ] dizisinde tutulur ve bu ihtiyaçlar kullanıcı etkileşimiyle veya belirli işlemler sonucunda artar veya azalır.

(Şekil 3.1).

MAX_IHTIYAC sabiti, toplam ihtiyaç sayısını tanımlar ve bu sabit, ihtiyaçlar[ ] dizisinin boyutunu belirler. Bu dizi, bebeğin farklı ihtiyaçlarını tutar. ihtiyac_isimleri[ ] dizisi, ihtiyaçların isimlerini tutar. Bu isimler, her bir ihtiyacın daha okunabilir olmasını sağlar. ihtiyaclariGoster[ ] fonksiyonu, bebeğin mevcut ihtiyaçlarını ekrana yazdırmak için kullanılır. Bu fonksiyon, ihtiyaçlar[ ] dizisini döngüyle gezerek her bir ihtiyacın mevcut değerini ve ismini ekrana yazdırır.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_IHTIYAC 9 // Toplam ihtiyac sayisi

// Sanal bebegin ihtiyaclari ve baslangicc degerleri
int ihtiyaclar[MAX_IHTIYAC] = {5, 5, 5, 5, 5, 5, 5, 5, 5};

// Ihtiyac isimleri
const char *ihtiyac_isimleri[MAX_IHTIYAC] = {
    "Tokluk", "Uyku", "Sevgi", "Sosyallesme", "Saglik", "Egitim", "Hijyen", "Tuvalet", "Eglence"
};

// Ihtiyaclari gosteren fonksiyon
void ihtiyaclariGoster(int *ihtiyaclar) {
    printf("\nIhtiyaclar:\n");
    for (int i = 0; i < MAX_IHTIYAC; ++i) {
        printf("%s: %d\n", ihtiyac_isimleri[i], ihtiyaclar[i]);
    }
}
```

Şekil 3.1 : İhtiyaçlar menüsü görseli.

(Şekil 3.2).

yemekYeme( ) fonksiyonu, sanal bir bebeğin farklı yiyeceklerle beslenmesini simüle ediyor. Fonksiyon, kullanıcıdan yiyecek ve miktar girişi alıyor, ardından bu girişe göre bebeğin tokluk seviyesini artırıyor veya diğer ihtiyaçlarını etkileyebiliyor. Kullanıcıdan alınan giriş, önce scanf ile bir karakter dizisine (giris) atanır. Ardından bu giriş, atoi ile tam sayıya dönüştürülür ve sscanf yardımıyla yiyecek miktarı ve adı elde edilir. strcmp fonksiyonu, yemek karakter dizisinin içeriğini kontrol ederek, hangi yiyeceğin seçildiğini belirler ve buna göre işlem yapar.

Bebeğe "sut" içirildiğinde, tokluk seviyesi artar (ihtiyaclar[0]) ve diğer bazı ihtiyaçlar düşer. Tokluk seviyesi 10'u geçerse bebeğin öldüğü simüle edilir. Uykusuzluk veya sağlıksız yaşam durumları kontrol edilir ve gerekli işlemler yapılır.

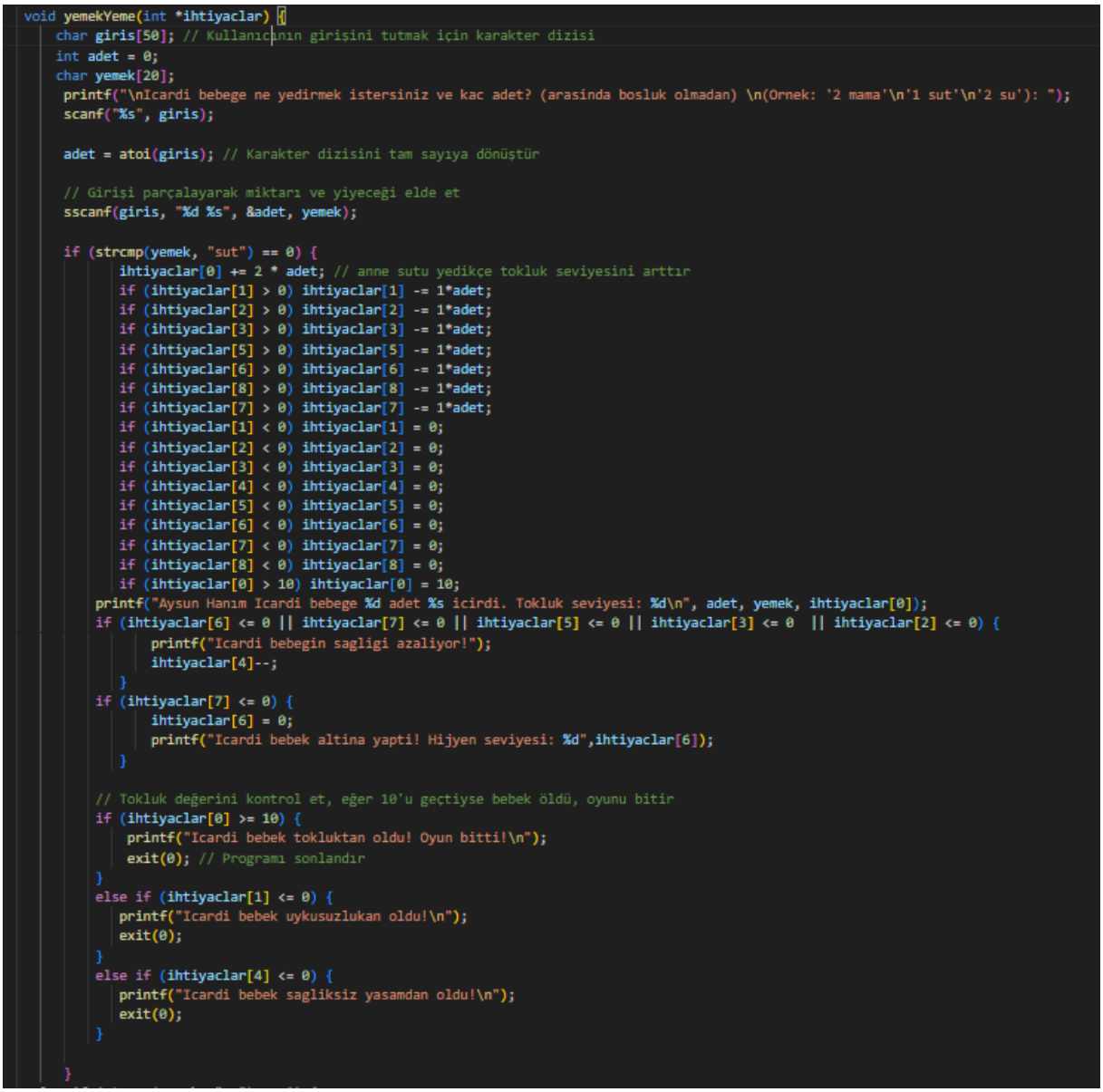
Bebeğe "su" içirildiğinde, tokluk seviyesi artar ve diğer bazı ihtiyaçlar düşer. Su içilmesi durumunda tuvalet ihtiyacı artar ve bu durum kontrol edilir. Diğer sağlık durumları kontrol edilir ve gerekli işlemler yapılır.

Bebeğe "mama" yedirildiğinde, tokluk seviyesi artar ve diğer bazı ihtiyaçlar düşer. Tokluk seviyesi 10'u geçerse bebeğin öldüğü simüle edilir. Uykusuzluk veya sağlıksız yaşam durumları kontrol edilir ve gerekli işlemler yapılır.

Tanımlı olmayan veya desteklenmeyen bir yiyecek girildiğinde, bu durum kullanıcıya bildirilir.

Tüm durumlarda, bebeğin tokluk seviyesi (ihtiyaclar[0]) ve diğer ihtiyaçları kontrol edilir, duruma göre ekrana bilgilendirici mesajlar yazdırılır. Bebeğin belirli ihtiyaçları 0'ın altına düştüğünde veya belirli koşullar sağlanmadığında, sağlık durumu azalır (ihtiyaclar[4]). Belirli şartlar altında bebek farklı sebeplerden ölebilir ve bu durumda oyun sonlandırılır (exit(0)).

Fonksiyonun içinde herhangi bir döngü bulunmamaktadır. Ancak if-else koşulları, farklı yiyecek türlerine göre işlemlerin gerçekleştirilmesini sağlar. Koşul ifadeleri, bebeğin ihtiyaçlarını kontrol eder ve belirli şartlar altında hangi işlemlerin yapılacağını belirler.



```c
void yemekYeme(int *ihtiyaclar) {
    char giris[50]; // Kullanıcının girişini tutmak için karakter dizisi
    int adet = 0;
    char yemek[20];
     printf("\nIcardi bebege ne yedirmek istersiniz ve kac adet? (arasinda bosluk olmadan) \n(Ornek: '2 mama'\n'1 sut'\n'2 su'): ");
     scanf("%s", giris);

     adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

     // Girişi parçalayarak miktarı ve yiyeceği elde et
     sscanf(giris, "%d %s", &adet, yemek);

     if (strcmp(yemek, "sut") == 0) {
            ihtiyaclar[0] += 2 * adet; // anne sutu yedikçe tokluk seviyesini arttır
            if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
            if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
            if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*adet;
            if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
            if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
            if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
            if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
            if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
            if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
            if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
            if (ihtiyaclar[0] > 10) ihtiyaclar[0] = 10;
        printf("Aysun Hanım Icardi bebege %d adet %s icirdi. Tokluk seviyesi: %d\n", adet, yemek, ihtiyaclar[0]);
        if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[3] <= 0  || ihtiyaclar[2] <= 0) {
                printf("Icardi bebegin sagligi azaliyor!");
                ihtiyaclar[4]--;
            }
        if (ihtiyaclar[7] <= 0) {
                ihtiyaclar[6] = 0;
                printf("Icardi bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
            }

        // Tokluk değerini kontrol et, eğer 10'u geçtiyse bebek öldü, oyunu bitir
        if (ihtiyaclar[0] >= 10) {
            printf("Icardi bebek tokluktan oldu! Oyun bitti!\n");
            exit(0); // Programı sonlandır
        }
        else if (ihtiyaclar[1] <= 0) {
            printf("Icardi bebek uykusuzlukan oldu!\n");
            exit(0);
        }
        else if (ihtiyaclar[4] <= 0) {
            printf("Icardi bebek sagliksiz yasamdan oldu!\n");
            exit(0);
        }

    }
```

Şekil 3.2 : yemekYeme fonksiyonu görseli.

(Şekil 3.3).

sevgiGoster( ) fonksiyonu, sanal bir bebeğe farklı sevgi gösterileri uygulayarak sevgi seviyesini artırıyor. Fonksiyon, kullanıcıdan sevgi gösterisi ve gösteri sayısını alıyor, ardından bu girişe göre bebeğin sevgi seviyesini ve diğer ihtiyaçlarını etkileyebiliyor. Kullanıcıdan alınan giriş, önce scanf ile bir karakter dizisine (giris) atanır. Ardından bu giriş, atoi ile tam sayıya dönüştürülür ve sscanf yardımıyla eylem sayısı ve eylem adı elde edilir. strcmp fonksiyonu, eylem karakter dizisinin içeriğini kontrol ederek, hangi sevgi gösterisinin seçildiğini belirler ve buna göre işlem yapar.

Bebeğe "okşama" yapıldığında, sevgi seviyesi artar (ihtiyaclar[2] ve ihtiyaclar[3]) ve diğer bazı ihtiyaçlar düşer. Sevgi seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır (ihtiyaclar[4]).

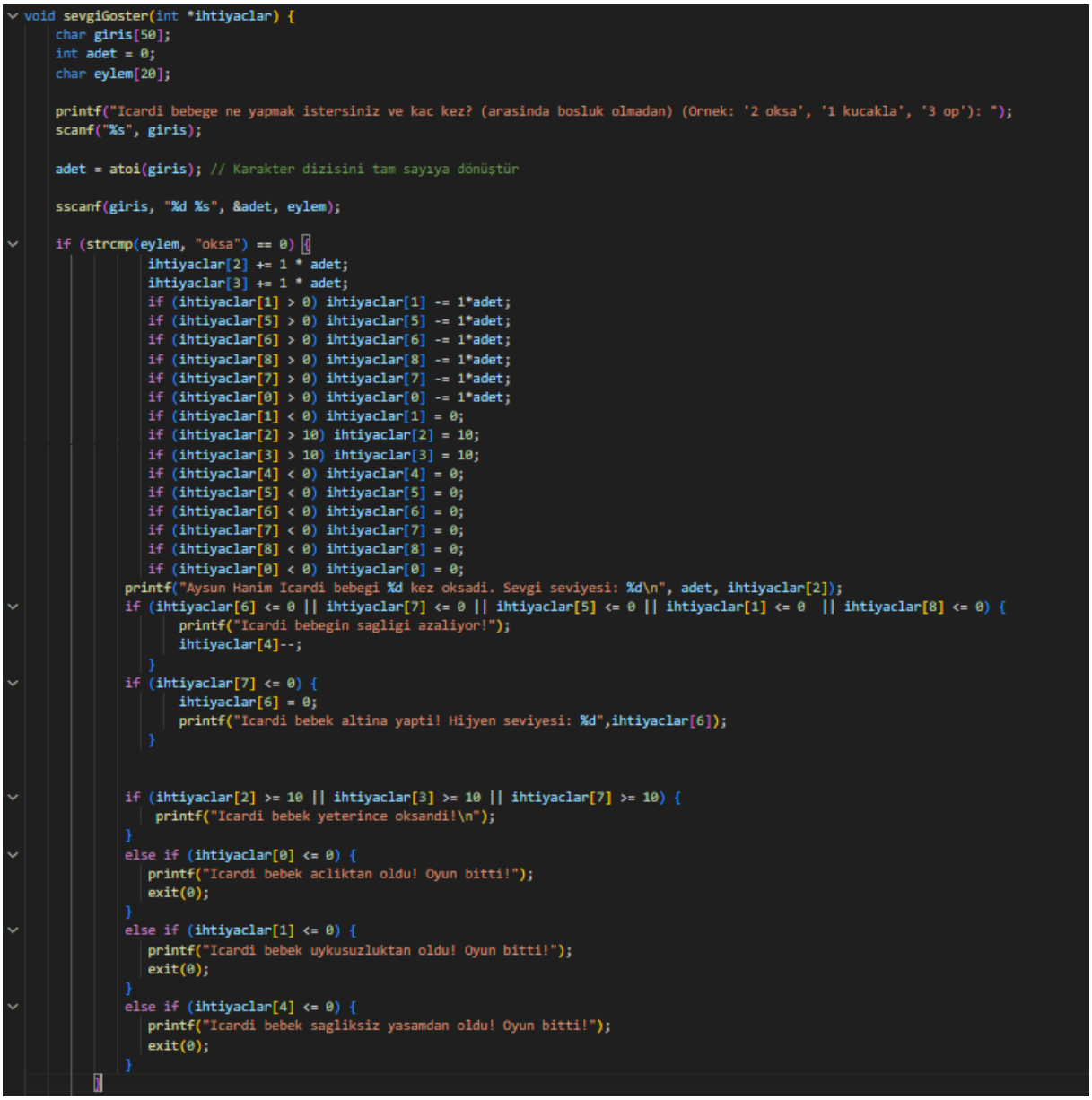
Bebeğe "öpme" yapıldığında, sevgi seviyesi artar ve diğer bazı ihtiyaçlar düşer. Sevgi seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Bebeği "kucaklama" eylemi gerçekleştirildiğinde, sevgi seviyesi artar ve diğer bazı ihtiyaçlar düşer. Sevgi seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Tanımlı olmayan veya desteklenmeyen bir eylem girildiğinde, bu durum kullanıcıya bildirilir.

Tüm durumlarda, bebeğin sevgi seviyesi ve diğer ihtiyaçları kontrol edilir, duruma göre ekrana bilgilendirici mesajlar yazdırılır. Bebeğin belirli ihtiyaçları 0'ın altına düştüğünde veya belirli koşullar sağlanmadığında, sağlık durumu azalır (ihtiyaclar[4]). Belirli şartlar altında bebek farklı sebeplerden ölebilir ve bu durumda oyun sonlandırılır (exit(0)).

Bu fonksiyonun içinde herhangi bir döngü bulunmamaktadır. Ancak if-else koşulları, farklı sevgi gösterisi türlerine göre işlemlerin gerçekleştirilmesini sağlar. Koşul ifadeleri, bebeğin ihtiyaçlarını kontrol eder ve belirli şartlar altında hangi işlemlerin yapılacağını belirler.



```c
void sevgiGoster(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Icardi bebege ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 oksa', '1 kucakla', '3 op'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

    if (strcmp(eylem, "oksa") == 0) {
            ihtiyaclar[2] += 1 * adet;
            ihtiyaclar[3] += 1 * adet;
            if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
            if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
            if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
            if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
            if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
            if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] > 10) ihtiyaclar[2] = 10;
            if (ihtiyaclar[3] > 10) ihtiyaclar[3] = 10;
            if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
            if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
            if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
        printf("Aysun Hanim Icardi bebegi %d kez oksadi. Sevgi seviyesi: %d\n", adet, ihtiyaclar[2]);
        if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[1] <= 0  || ihtiyaclar[8] <= 0) {
            printf("Icardi bebegin sagligi azaliyor!");
            ihtiyaclar[4]--;
        }
        if (ihtiyaclar[7] <= 0) {
            ihtiyaclar[6] = 0;
            printf("Icardi bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
        }


        if (ihtiyaclar[2] >= 10 || ihtiyaclar[3] >= 10 || ihtiyaclar[7] >= 10) {
         printf("Icardi bebek yeterince oksandi!\n");
        }
        else if (ihtiyaclar[0] <= 0) {
           printf("Icardi bebek acliktan oldu! Oyun bitti!");
           exit(0);
        }
        else if (ihtiyaclar[1] <= 0) {
           printf("Icardi bebek uykusuzluktan oldu! Oyun bitti!");
           exit(0);
        }
        else if (ihtiyaclar[4] <= 0) {
           printf("Icardi bebek sagliksiz yasamdan oldu! Oyun bitti!");
           exit(0);
        }
    }
```

Şekil 3.3 : sevgiGoster fonksiyonu görseli.

(Şekil 3.4).

hijyen( ) fonksiyonu, kullanıcıdan alınan girişlere bağlı olarak sanal bir bebek üzerinde hijyenle ilgili eylemler gerçekleştirir ve bebeğin hijyen seviyesini günceller. Fonksiyon, kullanıcıdan alınan eylem ve eylem sayısına göre bebeğin hijyen seviyesini artırır veya düşürür. Kullanıcıdan alınan giriş, önce scanf ile bir karakter dizisine (giris) atanır. Ardından bu giriş, atoi ile tam sayıya dönüştürülür ve sscanf yardımıyla eylem sayısı ve eylem adı elde edilir.

strcmp fonksiyonu, eylem karakter dizisinin içeriğini kontrol ederek, hangi hijyen eyleminin seçildiğini belirler ve buna göre işlem yapar.

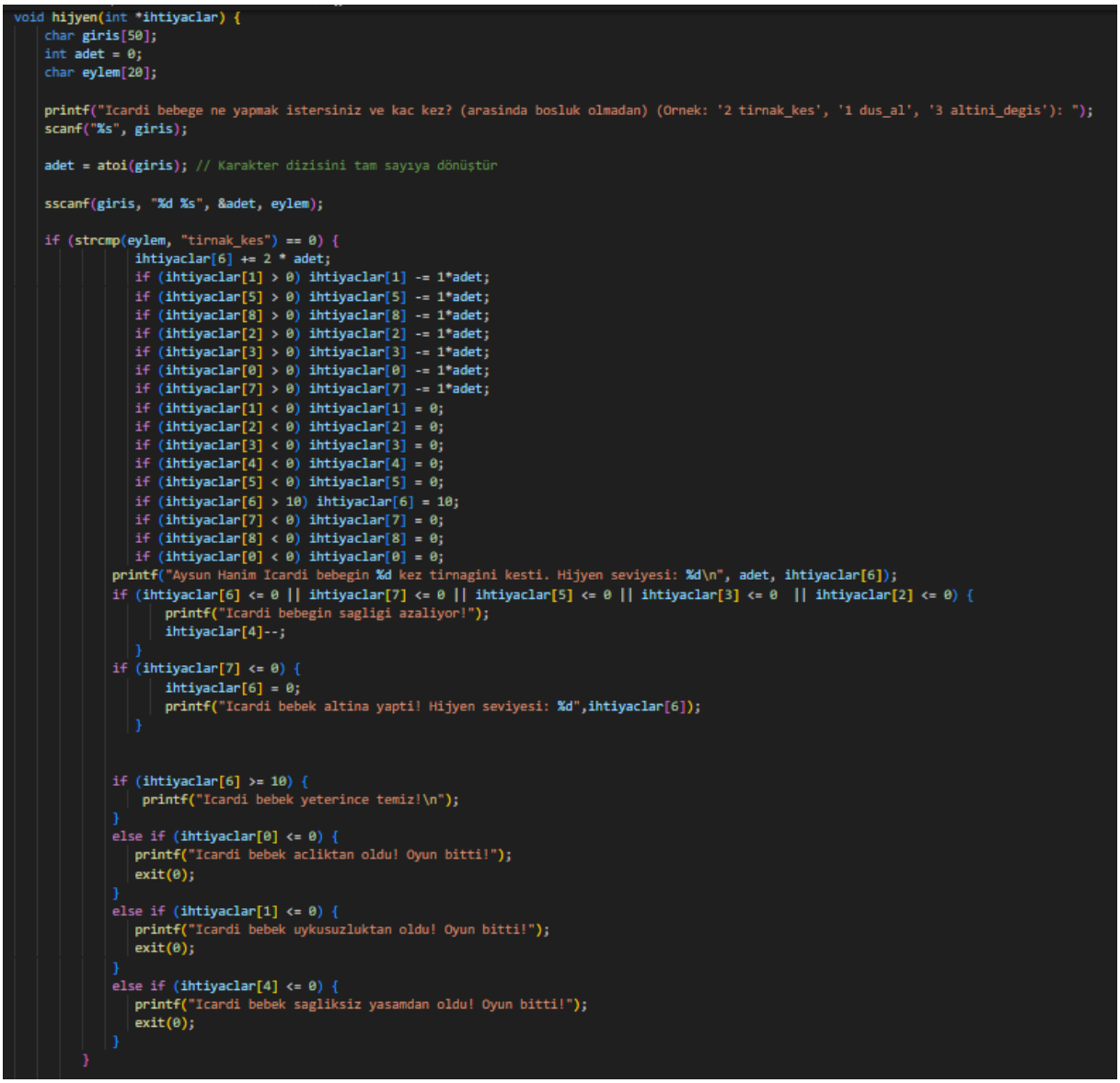
Bebeğin tırnaklarının kesilmesi, hijyen seviyesini artırır (ihtiyaclar[6]) ve diğer bazı ihtiyaçları düşürür. Hijyen seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır (ihtiyaclar[4]).

Bebeğin duş alması, hijyen seviyesini daha fazla artırır (ihtiyaclar[6]) ve diğer bazı ihtiyaçları düşürür. Hijyen seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Bebeğin altının değiştirilmesi, hijyen seviyesini daha fazla artırır (ihtiyaclar[6]) ve idrar/pislik seviyesini azaltır (ihtiyaclar[7]). Hijyen seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Tüm durumlarda, bebeğin hijyen seviyesi ve diğer ihtiyaçları kontrol edilir, duruma göre ekrana bilgilendirici mesajlar yazdırılır. Bebeğin belirli ihtiyaçları 0'ın altına düştüğünde veya belirli koşullar sağlanmadığında, sağlık durumu azalır (ihtiyaclar[4]). Belirli şartlar altında bebek farklı sebeplerden ölebilir ve bu durumda oyun sonlandırılır (exit(0)).

Bu fonksiyonun içinde herhangi bir döngü bulunmamaktadır. Ancak if-else koşulları, farklı hijyen eylemi türlerine göre işlemlerin gerçekleştirilmesini sağlar. Koşul ifadeleri, bebeğin ihtiyaçlarını kontrol eder ve belirli şartlar altında hangi işlemlerin yapılacağını belirler.



```c
void hijyen(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Icardi bebege ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 tirnak_kes', '1 dus_al', '3 altini_degis'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

    if (strcmp(eylem, "tirnak_kes") == 0) {
                ihtiyaclar[6] += 2 * adet;
                if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
                if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
                if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
                if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
                if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*adet;
                if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
                if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
                if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
                if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
                if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
                if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
                if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
                if (ihtiyaclar[6] > 10) ihtiyaclar[6] = 10;
                if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
                if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
                if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
            printf("Aysun Hanim Icardi bebegin %d kez tirnagini kesti. Hijyen seviyesi: %d\n", adet, ihtiyaclar[6]);
            if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[3] <= 0  || ihtiyaclar[2] <= 0) {
                    printf("Icardi bebegin sagligi azaliyor!");
                    ihtiyaclar[4]--;
                }
            if (ihtiyaclar[7] <= 0) {
                    ihtiyaclar[6] = 0;
                    printf("Icardi bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
                }


            if (ihtiyaclar[6] >= 10) {
                printf("Icardi bebek yeterince temiz!\n");
            }
            else if (ihtiyaclar[0] <= 0) {
                printf("Icardi bebek acliktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[1] <= 0) {
                printf("Icardi bebek uykusuzluktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[4] <= 0) {
                printf("Icardi bebek sagliksiz yasamdan oldu! Oyun bitti!");
                exit(0);
            }
        }
```

Şekil 3.4 : hijyen fonksiyonu görseli.

(Şekil 3.5).

egitim( ) fonksiyonu, kullanıcının seçtiği eylemlere bağlı olarak sanal bir bebek üzerinde eğitimle ilgili eylemler gerçekleştirir ve bebeğin eğitim seviyesini günceller. Fonksiyon, kullanıcıdan alınan eylem ve eylem sayısına göre bebeğin eğitim seviyesini artırır veya düşürür. Kullanıcıdan alınan giriş, önce scanf ile bir karakter dizisine (giris) atanır. Ardından bu giriş, atoi ile tam sayıya dönüştürülür ve sscanf yardımıyla eylem sayısı ve eylem adı elde edilir. strcmp fonksiyonu, eylem karakter dizisinin içeriğini kontrol ederek, hangi eğitim eyleminin seçildiğini belirler ve buna göre işlem yapar.

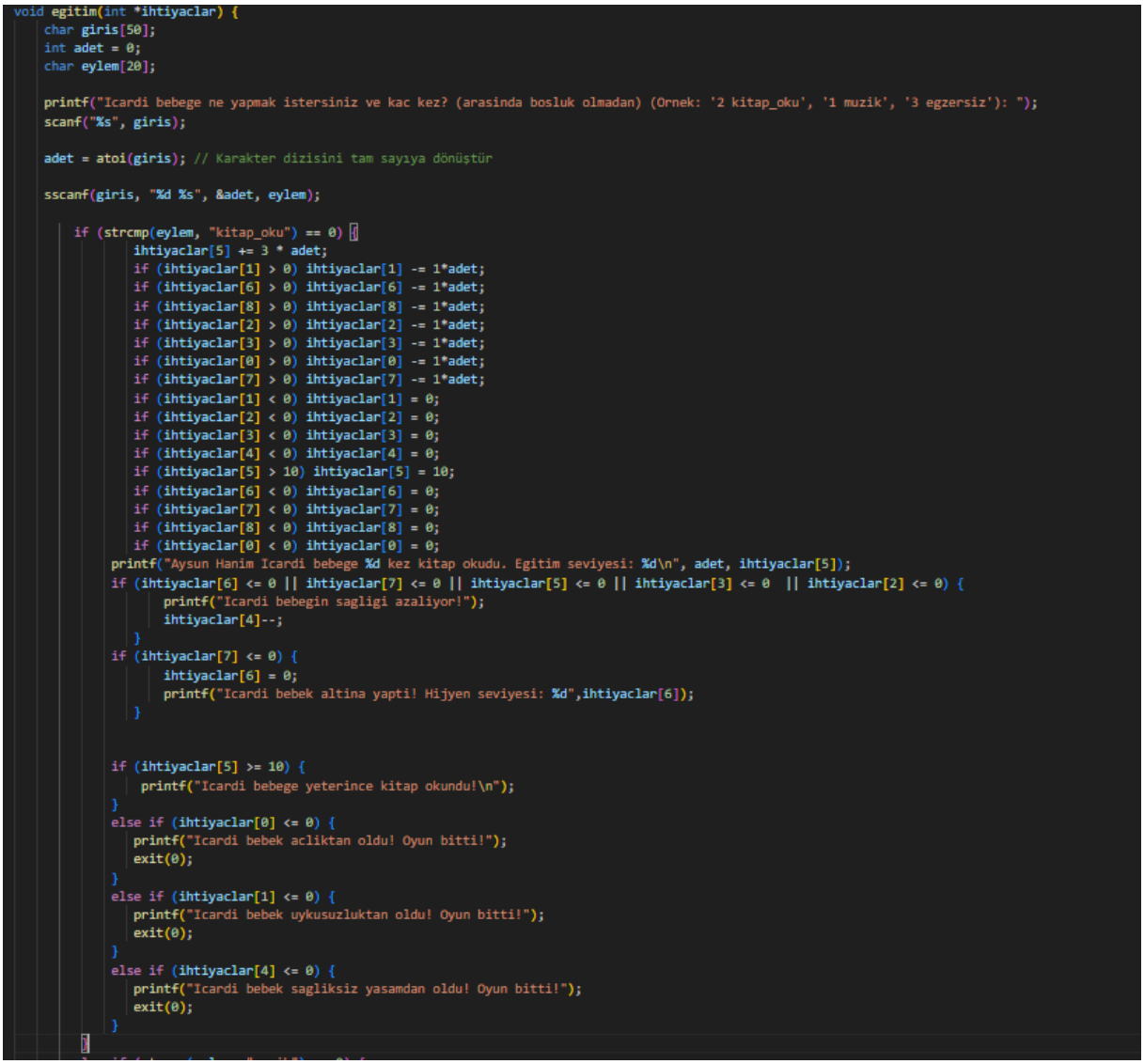
Bebeğe kitap okunması, eğitim seviyesini artırır (ihtiyaclar[5]) ve diğer bazı ihtiyaçları düşürür. Eğitim seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır (ihtiyaclar[4]).

Bebeğin müzik dinlemesi, eğitim seviyesini artırır (ihtiyaclar[5]) ve eğlence seviyesini artırır (ihtiyaclar[8]). Eğitim veya eğlence seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Bebeğin egzersiz yapması, eğitim seviyesini artırır (ihtiyaclar[5]) ve eğlence seviyesini artırır (ihtiyaclar[8]) ve sevgi seviyesini artırır (ihtiyaclar[2]). Eğitim, eğlence veya sevgi seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Tüm durumlarda, bebeğin eğitim seviyesi ve diğer ihtiyaçları kontrol edilir, duruma göre ekrana bilgilendirici mesajlar yazdırılır. Bebeğin belirli ihtiyaçları 0'ın altına düştüğünde veya belirli koşullar sağlanmadığında, sağlık durumu azalır (ihtiyaclar[4]). Belirli şartlar altında bebek farklı sebeplerden ölebilir ve bu durumda oyun sonlandırılır (exit(0)).

Bu fonksiyonun içinde herhangi bir döngü bulunmamaktadır. Ancak if-else koşulları, farklı eğitim eylemi türlerine göre işlemlerin gerçekleştirilmesini sağlar. Koşul ifadeleri, bebeğin ihtiyaçlarını kontrol eder ve belirli şartlar altında hangi işlemlerin yapılacağını belirler.



```c
void egitim(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Icardi bebege ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 kitap_oku', '1 muzik', '3 egzersiz'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

        if (strcmp(eylem, "kitap_oku") == 0) {
            ihtiyaclar[5] += 3 * adet;
            if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
            if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
            if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
            if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
            if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*adet;
            if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
            if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
            if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
            if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
            if (ihtiyaclar[5] > 10) ihtiyaclar[5] = 10;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
            if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
        printf("Aysun Hanim Icardi bebege %d kez kitap okudu. Egitim seviyesi: %d\n", adet, ihtiyaclar[5]);
        if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[3] <= 0  || ihtiyaclar[2] <= 0) {
            printf("Icardi bebegin sagligi azaliyor!");
            ihtiyaclar[4]--;
        }
        if (ihtiyaclar[7] <= 0) {
            ihtiyaclar[6] = 0;
            printf("Icardi bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
        }


        if (ihtiyaclar[5] >= 10) {
         printf("Icardi bebege yeterince kitap okundu!\n");
        }
        else if (ihtiyaclar[0] <= 0) {
           printf("Icardi bebek acliktan oldu! Oyun bitti!");
           exit(0);
        }
        else if (ihtiyaclar[1] <= 0) {
           printf("Icardi bebek uykusuzluktan oldu! Oyun bitti!");
           exit(0);
        }
        else if (ihtiyaclar[4] <= 0) {
           printf("Icardi bebek sagliksiz yasamdan oldu! Oyun bitti!");
           exit(0);
        }
    }
```

Şekil 3.5 : eğitim fonksiyonu görseli.

(Şekil 3.6).

eglence( ) fonksiyonu, kullanıcının seçtiği eylemlere bağlı olarak sanal bir bebek üzerinde eğlenceyle ilgili eylemler gerçekleştirir ve bebeğin eğlence seviyesini günceller. Fonksiyon, kullanıcıdan alınan eylem ve eylem sayısına göre bebeğin eğlence seviyesini artırır veya düşürür. Kullanıcıdan alınan giriş, önce scanf ile bir karakter dizisine (giris) atanır. Ardından bu giriş, atoi ile tam sayıya dönüştürülür ve sscanf yardımıyla eylem sayısı ve eylem adı elde edilir. strcmp fonksiyonu, eylem karakter dizisinin içeriğini kontrol ederek, hangi eğlence eyleminin seçildiğini belirler ve buna göre işlem yapar.

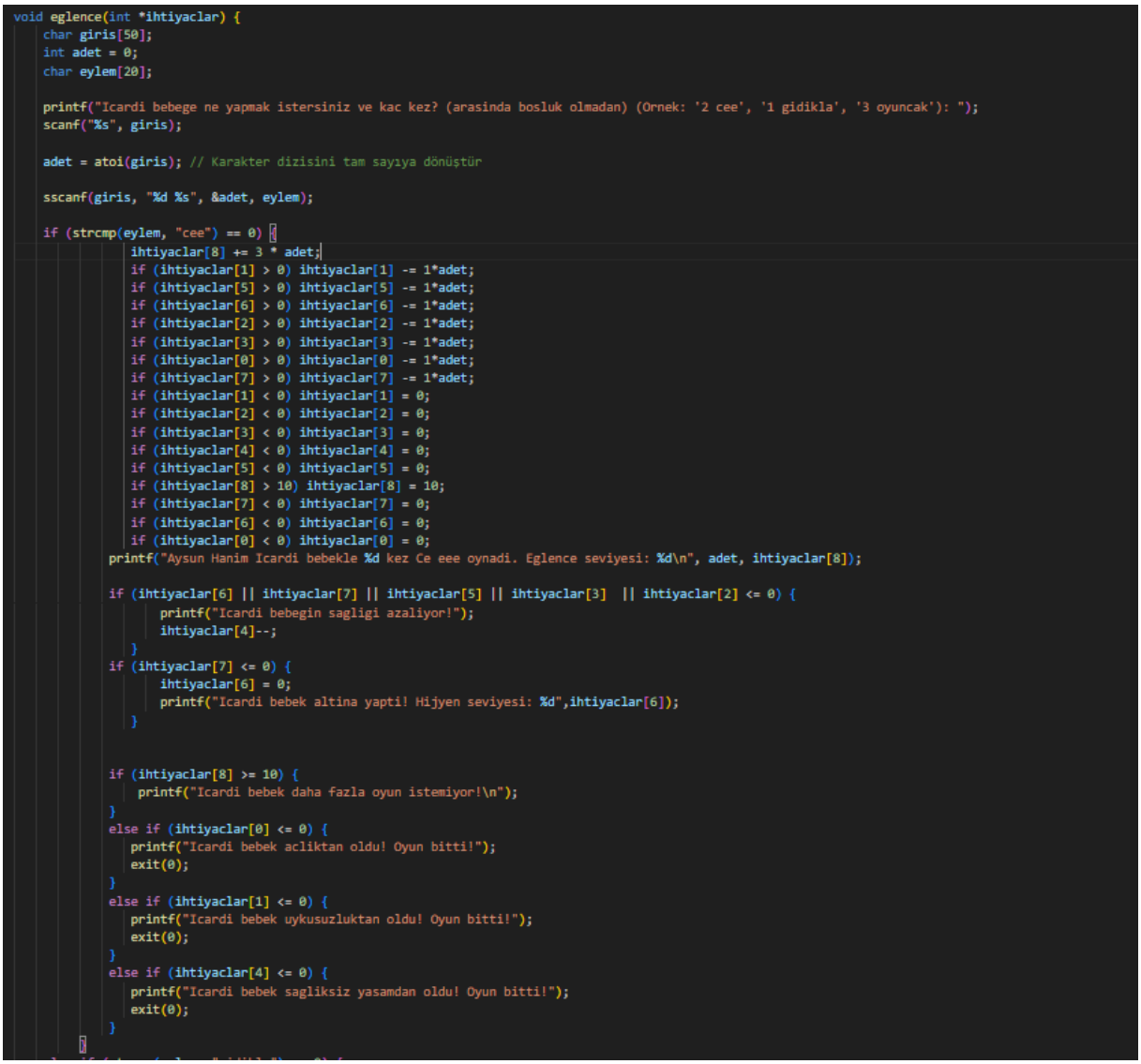
Bebeğin cee eylemi yapması, eğlence seviyesini artırır (ihtiyaclar[8]) ve diğer bazı ihtiyaçları düşürür. Eğlence seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır (ihtiyaclar[4]).

Bebeğin gidiklama eylemi yapması, eğlence seviyesini artırır (ihtiyaclar[8]) ve diğer bazı ihtiyaçları düşürür. Eğlence seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Bebeğin oyuncakla oynaması, eğlence seviyesini artırır (ihtiyaclar[8]) ve diğer bazı ihtiyaçları düşürür. Eğlence seviyesi maksimum değer olan 10'u geçerse, bebeğin bu eylemi yeterli sayıda aldığı varsayılır. Diğer ihtiyaçlar ve sağlık durumları kontrol edilir, eksiklik varsa sağlık durumu azalır.

Tüm durumlarda, bebeğin eğlence seviyesi ve diğer ihtiyaçları kontrol edilir, duruma göre ekrana bilgilendirici mesajlar yazdırılır. Bebeğin belirli ihtiyaçları 0'ın altına düştüğünde veya belirli koşullar sağlanmadığında, sağlık durumu azalır (ihtiyaclar[4]). Belirli şartlar altında bebek farklı sebeplerden ölebilir ve bu durumda oyun sonlandırılır (exit(0)).

Bu fonksiyonun içinde herhangi bir döngü bulunmamaktadır. Ancak if-else koşulları, farklı eğlence eylemi türlerine göre işlemlerin gerçekleştirilmesini sağlar. Koşul ifadeleri, bebeğin ihtiyaçlarını kontrol eder ve belirli şartlar altında hangi işlemlerin yapılacağını belirler.



```c
void eglence(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Icardi bebege ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 cee', '1 gidikla', '3 oyuncak'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

    if (strcmp(eylem, "cee") == 0) {
            ihtiyaclar[8] += 3 * adet;
            if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
            if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
            if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
            if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
            if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*adet;
            if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
            if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
            if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
            if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
            if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
            if (ihtiyaclar[8] > 10) ihtiyaclar[8] = 10;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
        printf("Aysun Hanim Icardi bebekle %d kez Ce eee oynadi. Eglence seviyesi: %d\n", adet, ihtiyaclar[8]);

        if (ihtiyaclar[6] || ihtiyaclar[7] || ihtiyaclar[5] || ihtiyaclar[3]  || ihtiyaclar[2] <= 0) {
            printf("Icardi bebegin sagligi azaliyor!");
            ihtiyaclar[4]--;
        }
        if (ihtiyaclar[7] <= 0) {
            ihtiyaclar[6] = 0;
            printf("Icardi bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
        }


        if (ihtiyaclar[8] >= 10) {
            printf("Icardi bebek daha fazla oyun istemiyor!\n");
        }
        else if (ihtiyaclar[0] <= 0) {
            printf("Icardi bebek acliktan oldu! Oyun bitti!");
            exit(0);
        }
        else if (ihtiyaclar[1] <= 0) {
            printf("Icardi bebek uykusuzluktan oldu! Oyun bitti!");
            exit(0);
        }
        else if (ihtiyaclar[4] <= 0) {
            printf("Icardi bebek sagliksiz yasamdan oldu! Oyun bitti!");
            exit(0);
        }
    }
```

Şekil 3.6 : eğlence fonksiyonu görseli.

(Şekil 3.7).

uyu( ) fonksiyonu, bebeğin uyku ihtiyacını karşılar ve belirli şartlara göre bebeğin durumunu günceller. Eğer tekrar (uyku tekrarı) değeri 0 veya daha küçükse, uyku tamamlanmış olarak kabul edilir ve fonksiyon işlemi sonlandırır. Uyku ihtiyacı olmayan bir bebeğin durumu kontrol edilir ve gerekli mesajlar yazdırılır.

Bebeğin uyku ihtiyacı (ihtiyaclar[1]) maksimum değer olan 10'a ulaşmamışsa, bebeğin uyku ihtiyacı karşılanır.

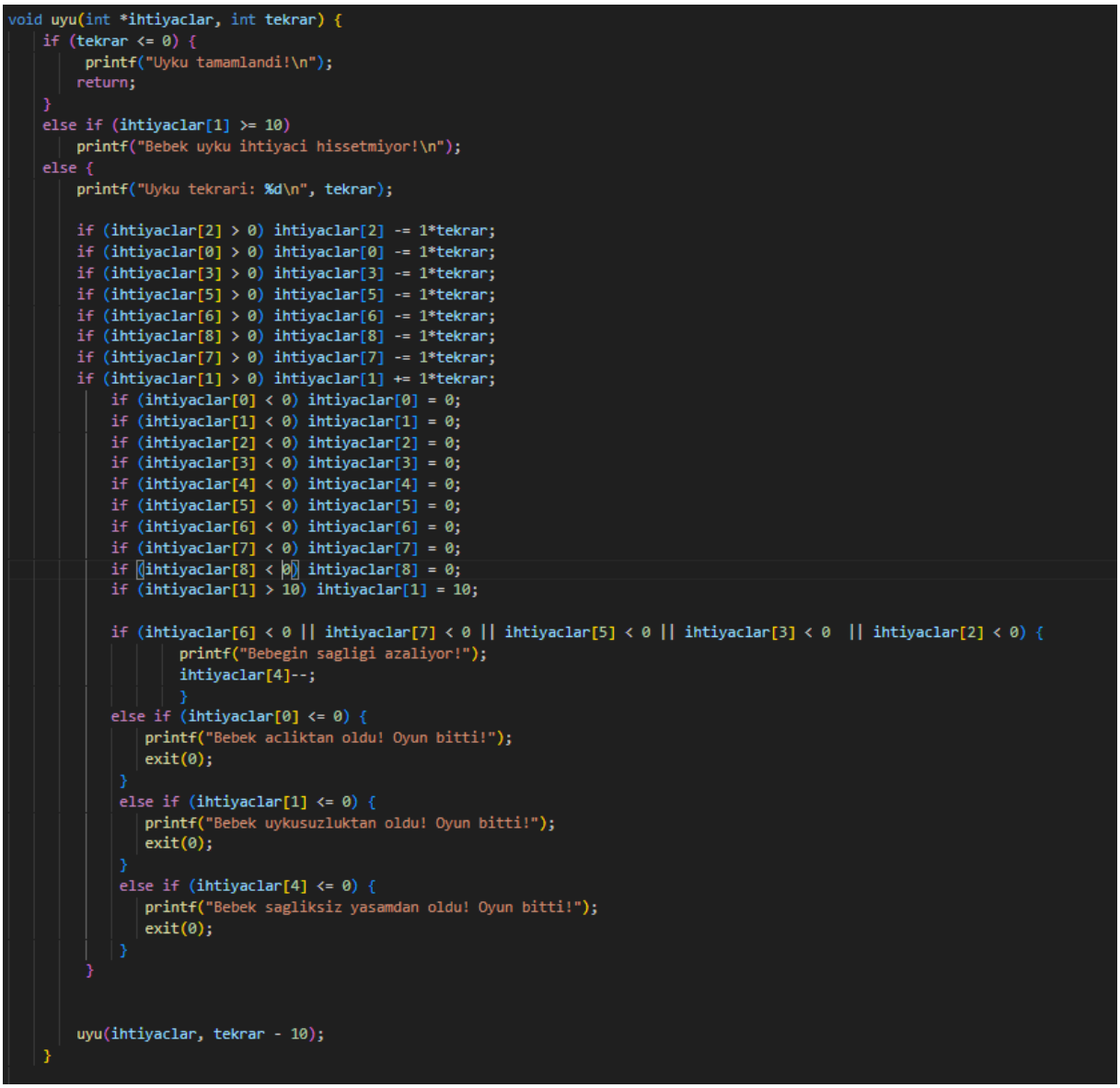
Uyku tekrarı (tekrar) sayısı kontrol edilir ve buna bağlı olarak bebeğin ihtiyaçları güncellenir.

Bebeğin uyku ihtiyacını azaltmak için bazı ihtiyaçlar (örneğin, enerji seviyesi ihtiyaclar[2], açlık seviyesi ihtiyaclar[0] vb.) düşürülür.

Uyku ihtiyacı arttırılır (ihtiyaclar[1]), fakat bu değer maksimum değeri aşarsa 10 olarak sabitlenir.

Bebeğin diğer ihtiyaçları 0'ın altına düşerse veya belirli koşullar sağlanmazsa (if-else blokları ile kontrol edilir): Bebeğin sağlık durumu azalır (ihtiyaclar[4]). Bebek açlık, uykusuzluk veya sağlıksız yaşam nedeniyle ölebilir ve oyun sonlandırılır (exit(0)).

Bu fonksiyon, kendini tekrar çağırarak (uyu(ihtiyaclar, tekrar - 10)) belirli bir süre uyutma işlemini gerçekleştirir. Fonksiyon, bir döngü içermiyor ancak if-else koşullarıyla belirli durumlar kontrol edilir ve işlemler gerçekleştirilir. if-else blokları, bebeğin uyku durumunu, uyku ihtiyacını, ihtiyaçlarını ve sağlık durumunu kontrol eder.



```c
void uyu(int *ihtiyaclar, int tekrar) {
    if (tekrar <= 0) {
         printf("Uyku tamamlandi!\n");
        return;
    }
    else if (ihtiyaclar[1] >= 10)
        printf("Bebek uyku ihtiyaci hissetmiyor!\n");
    else {
        printf("Uyku tekrari: %d\n", tekrar);

        if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*tekrar;
        if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*tekrar;
        if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*tekrar;
        if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*tekrar;
        if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*tekrar;
        if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*tekrar;
        if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*tekrar;
        if (ihtiyaclar[1] > 0) ihtiyaclar[1] += 1*tekrar;
            if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
            if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
            if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
            if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
            if (ihtiyaclar[1] > 10) ihtiyaclar[1] = 10;

            if (ihtiyaclar[6] < 0 || ihtiyaclar[7] < 0 || ihtiyaclar[5] < 0 || ihtiyaclar[3] < 0  || ihtiyaclar[2] < 0) {
                    printf("Bebegin sagligi azaliyor!");
                    ihtiyaclar[4]--;
                    }
            else if (ihtiyaclar[0] <= 0) {
                printf("Bebek acliktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[1] <= 0) {
                printf("Bebek uykusuzluktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[4] <= 0) {
                printf("Bebek sagliksiz yasamdan oldu! Oyun bitti!");
                exit(0);
            }
        }


        uyu(ihtiyaclar, tekrar - 10);
    }
```

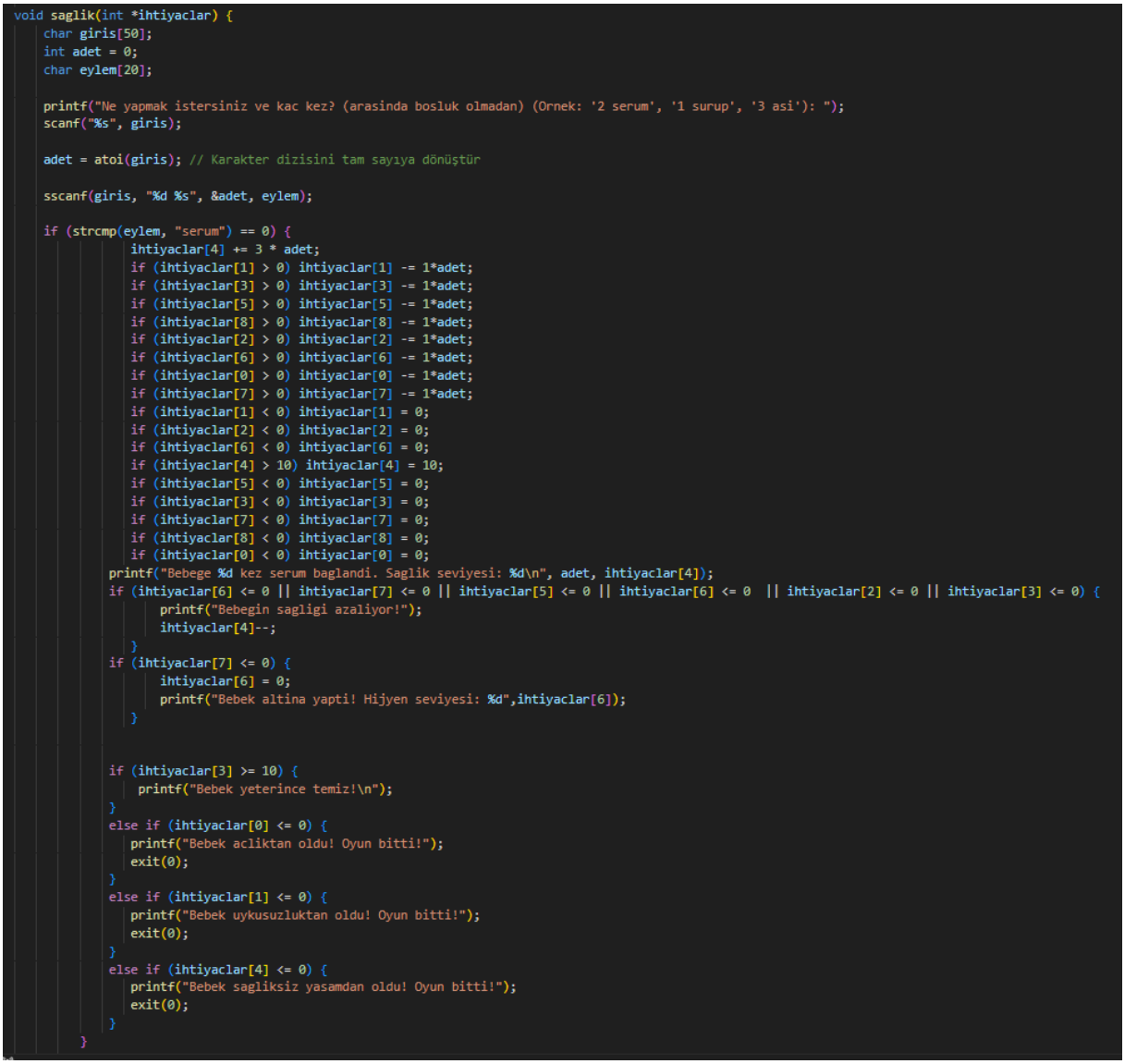
Şekil 3.7 : uyu fonksiyonu görseli.

(Şekil 3.8).

saglik( ) fonksiyonu, kullanıcıdan eylem ve adet bilgisini alır ve bu eyleme göre bebeğin sağlık durumunu günceller. Eylem türüne göre (serum, surup, asi) belirli işlemler gerçekleştirilir. Bebeğin sağlık parametreleri, eyleme bağlı olarak güncellenir. Bebeğin belirli sağlık durumları kontrol edilir ve duruma göre mesajlar yazdırılır. Bebek sağlık değerleri belli bir seviyenin altına düşerse, sağlık durumu azalır ve buna bağlı olarak sağlık değerleri düşürülür.

scanf ile kullanıcıdan alınan giriş, işlem yapılacak eylem ve eylemin kaç kez gerçekleştirileceği bilgisini içerir. Aldığı eyleme göre if-else yapılarıyla kontrol edilir ve ilgili eylem işlemleri yapılır (serum, surup, asi). Eylemler sırasında belirli sağlık parametreleri (örneğin, tokluk seviyesi ihtiyaclar[3], hijyen seviyesi ihtiyaclar[6] gibi) kontrol edilir ve bu parametrelerin değerleri güncellenir. Eğer belirli sağlık durumları belli bir seviyenin altına düşerse (if koşullarıyla kontrol edilir): Bebeğin sağlık seviyesi azalır (ihtiyaclar[4]). Bebek altına yapabilir ve hijyen seviyesi azalabilir. Bebeğin sağlık durumu, belirli koşullar altında kontrol edilir ve eğer belirli bir sağlık değerinin üstünde veya altında ise buna göre mesajlar yazdırılır. Belirli koşullar sağlanırsa (else if bloklarıyla kontrol edilir): Bebek açlık, uykusuzluk veya sağlıksız yaşam nedeniyle ölebilir ve oyun sonlandırılır (exit(0)).

Bu fonksiyon, if-else yapıları kullanarak eylemleri belirler ve bu eylemlere göre işlemleri gerçekleştirir. Kullanıcının girdisi olan eyleme (serum, surup, asi) göre farklı işlemler yapılır. Her eylem için aynı parametreler kontrol edilir ve bu parametrelerin değerleri belirli koşullara göre güncellenir.



```c
void saglik(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 serum', '1 surup', '3 asi'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

    if (strcmp(eylem, "serum") == 0) {
            ihtiyaclar[4] += 3 * adet;
            if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
            if (ihtiyaclar[3] > 0) ihtiyaclar[3] -= 1*adet;
            if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
            if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
            if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
            if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
            if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
            if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
            if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
            if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
            if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
            if (ihtiyaclar[4] > 10) ihtiyaclar[4] = 10;
            if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
            if (ihtiyaclar[3] < 0) ihtiyaclar[3] = 0;
            if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
            if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
            if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
        printf("Bebege %d kez serum baglandi. Saglik seviyesi: %d\n", adet, ihtiyaclar[4]);
        if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[6] <= 0  || ihtiyaclar[2] <= 0 || ihtiyaclar[3] <= 0) {
            printf("Bebegin sagligi azaliyor!");
            ihtiyaclar[4]--;
        }
        if (ihtiyaclar[7] <= 0) {
            ihtiyaclar[6] = 0;
            printf("Bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
        }


        if (ihtiyaclar[3] >= 10) {
            printf("Bebek yeterince temiz!\n");
        }
        else if (ihtiyaclar[0] <= 0) {
            printf("Bebek acliktan oldu! Oyun bitti!");
            exit(0);
        }
        else if (ihtiyaclar[1] <= 0) {
            printf("Bebek uykusuzluktan oldu! Oyun bitti!");
            exit(0);
        }
        else if (ihtiyaclar[4] <= 0) {
            printf("Bebek sagliksiz yasamdan oldu! Oyun bitti!");
            exit(0);
        }
    }
```

Şekil 3.8 : saglik fonksiyonu görseli.

(Şekil 3.9).

sosyalleşme( ) fonksiyonu, kullanıcıdan eylem ve adet bilgisini alır ve bu eyleme göre bebeğin sosyalleşme seviyesini günceller. Eylem türüne göre (goz_temasi, gulumse, sarki_soyle) belirli işlemler gerçekleştirilir. Bebeğin sosyal durumu, yapılan eylemlerle ve bu eylemlere bağlı olarak güncellenir. Bebeğin belirli sağlık parametreleri kontrol edilir ve duruma göre mesajlar yazdırılır.

scanf ile kullanıcıdan alınan giriş, işlem yapılacak eylem ve eylemin kaç kez gerçekleştirileceği bilgisini içerir. Aldığı eyleme göre if-else yapılarıyla kontrol edilir ve ilgili eylem işlemleri yapılır (goz_temasi, gulumse, sarki_soyle). Eylemler sırasında belirli sağlık parametreleri (örneğin, sosyalleşme seviyesi ihtiyaclar[3]) kontrol edilir ve bu parametrelerin değerleri güncellenir. Eğer belirli sağlık durumları belli bir seviyenin altına düşerse (if koşullarıyla kontrol edilir): Bebeğin sağlık seviyesi azalır (ihtiyaclar[4]). Bebek altına yapabilir ve hijyen seviyesi azalabilir. Bebeğin sağlık durumu, belirli koşullar altında kontrol edilir ve eğer belirli bir sağlık değerinin üstünde veya altında ise buna göre mesajlar yazdırılır. Belirli koşullar sağlanırsa (else if bloklarıyla kontrol edilir): Bebek açlık, uykusuzluk veya sağlıksız yaşam nedeniyle ölebilir ve oyun sonlandırılır (exit(0)).

Fonksiyon, if-else yapıları kullanarak eylemleri belirler ve bu eylemlere göre işlemleri gerçekleştirir. Kullanıcının girdisi olan eyleme (goz_temasi, gulumse, sarki_soyle) göre farklı işlemler yapılır. Her eylem için aynı parametreler kontrol edilir ve bu parametrelerin değerleri belirli koşullara göre güncellenir.

```c
void sosyallesme(int *ihtiyaclar) {
    char giris[50];
    int adet = 0;
    char eylem[20];

    printf("Ne yapmak istersiniz ve kac kez? (arasinda bosluk olmadan) (Ornek: '2 goz_temasi', '1 gulumse', '3 sarki_soyle'): ");
    scanf("%s", giris);

    adet = atoi(giris); // Karakter dizisini tam sayıya dönüştür

    sscanf(giris, "%d %s", &adet, eylem);

    if (strcmp(eylem, "goz_temasi") == 0) {
                ihtiyaclar[3] += 3 * adet;
                if (ihtiyaclar[1] > 0) ihtiyaclar[1] -= 1*adet;
                if (ihtiyaclar[5] > 0) ihtiyaclar[5] -= 1*adet;
                if (ihtiyaclar[8] > 0) ihtiyaclar[8] -= 1*adet;
                if (ihtiyaclar[2] > 0) ihtiyaclar[2] -= 1*adet;
                if (ihtiyaclar[6] > 0) ihtiyaclar[6] -= 1*adet;
                if (ihtiyaclar[0] > 0) ihtiyaclar[0] -= 1*adet;
                if (ihtiyaclar[7] > 0) ihtiyaclar[7] -= 1*adet;
                if (ihtiyaclar[1] < 0) ihtiyaclar[1] = 0;
                if (ihtiyaclar[2] < 0) ihtiyaclar[2] = 0;
                if (ihtiyaclar[6] < 0) ihtiyaclar[6] = 0;
                if (ihtiyaclar[4] < 0) ihtiyaclar[4] = 0;
                if (ihtiyaclar[5] < 0) ihtiyaclar[5] = 0;
                if (ihtiyaclar[3] > 10) ihtiyaclar[3] = 10;
                if (ihtiyaclar[7] < 0) ihtiyaclar[7] = 0;
                if (ihtiyaclar[8] < 0) ihtiyaclar[8] = 0;
                if (ihtiyaclar[0] < 0) ihtiyaclar[0] = 0;
            printf("Bebekle %d kez goz temasi kuruldu. Sosyallesme seviyesi: %d\n", adet, ihtiyaclar[3]);
            if (ihtiyaclar[6] <= 0 || ihtiyaclar[7] <= 0 || ihtiyaclar[5] <= 0 || ihtiyaclar[6] <= 0  || ihtiyaclar[2] <= 0) {
                    printf("Bebegin sagligi azaliyor!");
                    ihtiyaclar[4]--;
                }
            if (ihtiyaclar[7] <= 0) {
                    ihtiyaclar[6] = 0;
                    printf("Bebek altina yapti! Hijyen seviyesi: %d",ihtiyaclar[6]);
                }

            if (ihtiyaclar[3] >= 10) {
                printf("Bebek yeterince temiz!\n");
            }
            else if (ihtiyaclar[0] <= 0) {
                printf("Bebek acliktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[1] <= 0) {
                printf("Bebek uykusuzluktan oldu! Oyun bitti!");
                exit(0);
            }
            else if (ihtiyaclar[4] <= 0) {
                printf("Bebek sagliksiz yasamdan oldu! Oyun bitti!");
                exit(0);
            }
        }
```

Şekil 3.9 : sosyalleşme fonksiyonu görseli.

(Şekil 3.10).

doping( ) fonksiyonu, oyunun kuralları içerisinde sınırlı bir sayıda (üç kez) çağrılabilecek olan bir doping işlemini simüle eder. Bu fonksiyon, bebeğin ihtiyaçlarını maksimum seviyeye çıkarır ve bu işlem sırasında kaç kez çağrıldığını izler.

doping( ) fonksiyonu, bebeğin ihtiyaçlarını (beslenme, uyku, sağlık vb.) maksimuma (10) çıkarır. Fonksiyon, bu işlemi üç kez gerçekleştirebilir. Her doping işlemi yapıldığında,

kullanım sayısı bir artar ve toplam kullanım sayısı üçü geçtiğinde kural ihlali mesajı verilir ve oyun sonlandırılır. Kullanım sayısı izlenir ve bu sayı üç kez aşıldığında kural ihlali mesajı verilir, oyun sonlandırılır.

if-else yapısı, doping işleminin kaç kez gerçekleştirilebileceğini kontrol eder. Döngü ile bebeğin ihtiyaçlarının tutulduğu dizi üzerinde dolaşarak ihtiyaçları maksimuma çıkarır (ihtiyaclar[i] = 10). Her doping işlemi yapıldığında, dopingKullanmaSayisi bir artırılır ve bu sayı üçü geçtiğinde kural ihlali mesajı verilir, oyun sonlandırılır.

Fonksiyon, if-else yapısı ile kullanım sayısını kontrol eder ve üç kezden az olması durumunda doping işlemini gerçekleştirir. for döngüsü, bebeğin ihtiyaçlarını maksimum seviyeye çıkarmak için ihtiyaçları dolaşır ve eğer ihtiyaçlar 10'dan küçükse, ihtiyaçlar 10'a çıkarılır. İşlem sonunda if-else yapılarıyla kontrol edilir: Eğer kullanım sayısı üçü geçmişse, kural ihlali mesajı verilir ve oyun sonlandırılır. Aksi takdirde, kullanım sayısı bir artırılır ve işlem devam eder.

```c
// Global değişken: doping fonksiyonunun kaç kere çağrıldığını tutmak için
int dopingKullanmaSayisi = 0;

void doping(int *ihtiyaclar) {
    // Doping fonksiyonu 3 kez çağrılabilir
    if (dopingKullanmaSayisi < 3) {
        // İhtiyaç değerlerini maksimuma (10) çıkar
        for (int i = 0; i < 9; ++i) {
            if (ihtiyaclar[i] < 10) {
                ihtiyaclar[i] = 10;
            }
        }
        // Kullanım sayısını artır
        dopingKullanmaSayisi++;
        printf("Doping yapildi! Degerler maksimuma cikarildi.\n");

        // Eğer kullanım sayısı 3'ü geçtiyse oyunu bitir
        if (dopingKullanmaSayisi >= 3) {
            printf("Kural ihlali! Oyun bitti!\n");
            exit(0); // Programı sonlandır
        }
    } else {
        printf("Kural ihlali! Doping fonksiyonu 3 kezden fazla kullanildi.\nOyun bitti!\n");
        exit(0); // Programı sonlandır
    }
}
```

Şekil 3.10 : doping fonksiyonu görseli.

(Şekil 3.11).

main( ) fonksiyonu, kullanıcı arayüzü sağlar ve kullanıcının seçtiği işlemlere göre bebek simülasyonunun ihtiyaçlarını güncelleyen işlevleri çağırır. Kullanıcının seçtiği işleme göre ilgili fonksiyonları çağırarak bebek simülasyonunun durumunu günceller ve ekrana yansıtır.

while döngüsü, kullanıcı çıkış yapana kadar seçenekler sunar ve kullanıcının seçimine göre ilgili işlevi gerçekleştirir. switch-case yapısı, kullanıcının seçtiği işleme göre ilgili fonksiyonu çağırır. Kullanıcı çıkış yapmadığı sürece sürekli olarak seçenekler sunulur ve kullanıcı seçim yapmaya devam eder.

printf fonksiyonları, kullanıcıya seçenekleri ve bilgilendirici mesajları gösterir. scanf fonksiyonları, kullanıcının seçimlerini ve bazı işlemler için gerekli verileri alır. switch-case yapısı, kullanıcının seçimine göre ilgili fonksiyonları çağırır.

while (1) döngüsü, kullanıcı çıkış yapana kadar programın çalışmasını sağlar. switch-case yapısı, kullanıcının seçtiği işleme göre ilgili fonksiyonları çağırır. Her işlem yapıldıktan sonra ihtiyaclariGoster( ) fonksiyonu çağrılarak bebek simülasyonunun güncellenmiş ihtiyaçları ekrana yazdırılır.

4. TEKNİK EKSİKLİKLER

Bazı yerlerde tekrarlayan kod parçaları var, bu parçalar birleştirilebilir veya fonksiyonlar aracılığıyla daha verimli hale getirilebilir.

4.1. Kritik Eksiklikler

Program, hata durumlarına karşı yeterince dirençli değil ve hata durumlarını yönetme yeteneği yok, beklenmedik durumlar programın çalışmasını durdurabilir veya yanlış sonuçlar üretebilir.

4.2. Diğer Eksiklikler

İhtiyaçları izlemek için daha karmaşık veri yapıları (örneğin, ağaçlar, bağlı listeler) kullanılabilir, böylece daha fazla detay eklenip daha karmaşık ilişkiler oluşturulabilir.

5. YARATICILIK AŞAMASI

```c
// Global değişken: doping fonksiyonunun kaç kere çağrıldığını tutmak için
int dopingKullanmaSayisi = 0;

void doping(int *ihtiyaclar) {
    // Doping fonksiyonu 3 kez çağrılabilir
    if (dopingKullanmaSayisi < 3) {
        // İhtiyaç değerlerini maksimuma (10) çıkar
        for (int i = 0; i < 9; ++i) {
            if (ihtiyaclar[i] < 10) {
                ihtiyaclar[i] = 10;
            }
        }
        // Kullanım sayısını artır
        dopingKullanmaSayisi++;
        printf("Doping yapildi! Degerler maksimuma cikarildi.\n");

        // Eğer kullanım sayısı 3'ü geçtiyse oyunu bitir
        if (dopingKullanmaSayisi >= 3) {
            printf("Kural ihlali! Oyun bitti!\n");
            exit(0); // Programı sonlandır
        }
    } else {
        printf("Kural ihlali! Doping fonksiyonu 3 kezden fazla kullanildi.\nOyun bitti!\n");
        exit(0); // Programı sonlandır
    }
}
```

doping fonksiyonu, bir dizi ihtiyaç değerini maksimuma (10) çıkarmak için tasarlanmıştır. Fonksiyon, ihtiyaç değerlerini kontrol eder ve mevcut değer 10'dan küçükse maksimuma çıkarır. Ayrıca, bu fonksiyon, toplamda 3 kez çağrılabilir. Her çağrıldığında, çağrıldığı sayı bir değişkende (dopingKullanmaSayisi) saklanır.

dopingKullanmaSayisi: Global bir değişken olarak tanımlanmıştır. Bu değişken, doping fonksiyonunun kaç kez çağrıldığını takip eder.

doping Fonksiyonu: Fonksiyon, parametre olarak bir tamsayı dizisi (ihtiyaclar) alır ve bu dizi üzerinde değişiklik yapar.

Kullanım Kontrolü: Fonksiyon, öncelikle doping fonksiyonunun daha önce kaç kez çağrıldığını kontrol eder. Bu, dopingKullanmaSayisi değişkenini izleyerek yapılır.

İhtiyaç Değerlerinin Kontrolü ve Güncellenmesi: Döngü, dizideki her değeri kontrol eder. Eğer ihtiyaç değeri 10'dan küçükse, değeri 10 yapar.

Kullanım Sınırı: Eğer fonksiyon toplamda 3 kez çağrılmışsa ve bir dördüncü kez çağrılırsa, bir kural ihlali olduğunu belirterek programı sonlandırır (exit(0) ile).

Fonksiyon Çağrılarına Geri Bildirim: Her bir çağrıda, ekrana "Doping yapıldı! Değerler maksimuma çıkarıldı." gibi bir geri bildirim yazdırır.

Kullanım Sınırlarına Ulaşılması Durumunda Oyunun Bitirilmesi: Eğer fonksiyon toplamda 3 kez çağrılmışsa ve bir dördüncü kez çağrılırsa, bir kural ihlali olduğunu ve oyunun bittiğini belirterek programı sonlandırır (exit(0) ile).

Bu fonksiyon, bir dizi ihtiyaç değerini 10'a çıkarırken, aynı zamanda belirli bir sınırlama getirerek oyundaki kural ihlallerini engellemeyi amaçlamaktadır. Özellikle, bu fonksiyonun bir oyun içinde belirli bir kontenjan dahilinde kullanılabileceği önemli bir kuralı uygulamaktadır.

KAYNAKÇA

Tutorialspoint , C library function – atoi( ) , www.tutorialspoint.com/c_standard_library/c_function_atoi.htm

Tutorialspoint , C library function – sscanf( ), https://www.tutorialspoint.com/c_standard_library/c_function_sscanf.htm

Tutorialspoint , C library function - exit(), https://www.tutorialspoint.com/c_standard_library/c_function_exit.htm

GeeksforGeeks, strcmp( ) in C, https://www.geeksforgeeks.org/strcmp-in-c/