



**SAKARYA
UYGULAMALI BİLİMLER
ÜNİVERSİTESİ**

2024 - 2025 | GÜZ

ÖDEV

Öğrencinin

Adı Soyadı :Bulut Yüksel.....
Okul Kimliği :23010903088.....
Adı Soyadı :Bekzod Zakirov.....
Okul Kimliği :23010903138.....

Ders bilgileri

Ders Adı :NESNEYE YÖNELİK PROGRAMLAMA.....
Hocanın Adı :DOÇ. DR. ZAFER ALBAYRAK



SnakeGame(Yılan oyunu)

Kullanılan sınıfların UML diagramı

```
+-----+
|  SnakeGame  |
+-----+
| - difficulty: int      |
| - diffinput: int      |
| - width: const int    |
| - height: const int   |
| - gameOver: bool      |
| - score: int           |
| - dir: eDirection     |
| - fruit: pair<int, int> |
| - snake: vector<pair<int, int>> |
+-----+
| + SnakeGame(w: int, h: int) |
| + Difficulty(b: int): int   |
| + Run(): void              |
| - GenerateFruit(): void    |
| - Draw(): void             |
| - Input(): void            |
| - Logic(): void            |
+-----+

+-----+
|  eDirection  |
+-----+
| STOP = 0      |
| LEFT          |
| RIGHT         |
| UP            |
| DOWN          |
+-----+
```

Kodun detaylı açıklaması

```
#include <iostream>
#include <conio.h>
#include <windows.h>
#include <vector>
using namespace std;
```

// Yönleri temsil eden enum yapısı

```
enum eDirection { STOP = 0, LEFT, RIGHT, UP, DOWN };
```

```
class SnakeGame {
```

```
private:
```

```
    int difficulty; // Zorluk seviyesi
    int diffinput; // Kullanıcının girdiği zorluk seviyesi
    const int width; // Haritanın alanının genişliği
    const int height; // Haritanın alanının yüksekliği
    bool gameOver; // Oyunun bitiş durumu
    int score; // Skor
    eDirection dir; // Hareket yönü
    pair<int, int> fruit; // Meyvenin konumu
    vector<pair<int, int>> snake; // Yılanın gövdesi
```

// Rastgele bir meyve konumu oluştur

```
void GenerateFruit() {
    fruit.first = rand() % width;
    fruit.second = rand() % height;
}
```

// Oyun ekranını çiz

```
void Draw() {
    COORD coord = {0, 0};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
```

// Üst duvar

```
for (int i = 0; i < width + 2; i++)
    cout << "#";
cout << endl;
```

// Oyun alanı

```
for (int i = 0; i < height; i++) {
    for (int j = 0; j < width; j++) {
        if (j == 0) cout << "#"; // Sol duvar
        if (make_pair(j, i) == snake[0])
            cout << "O"; // Yılanın başı
        else if (make_pair(j, i) == fruit)
            cout << "F"; // Meyve
        else {
            bool isTail = false;
```

```

        for (int k = 1; k < snake.size(); k++) {
            if (snake[k] == make_pair(j, i)) {
                cout << "o"; // Yılanın gövdesi
                isTail = true;
                break;
            }
        }
        if (!isTail)
            cout << " "; // Boş alan
    }
    if (j == width - 1) cout << "#"; // Sağ duvar
}
cout << endl;
}

// Alt duvar
for (int i = 0; i < width + 2; i++)
    cout << "#";
cout << endl;
cout << "Score: " << score << endl; // Skoru göster
}

// Kullanıcıdan yön girişini al
void Input() {
    if (_kbhit()) {
        switch (_getch()) {
            case 'a': dir = LEFT; break; // Sol
            case 'd': dir = RIGHT; break; // Sağ
            case 'w': dir = UP; break; // Yukarı
            case 's': dir = DOWN; break; // Aşağı
            case 'x': gameOver = true; break; // Oyunu bitir
        }
    }
}

// Oyun mantığı
void Logic() {
    pair<int, int> prev = snake[0]; // Yılanın başı
    pair<int, int> prev2;

    // Yılanın gövdesini oynat
    for (int i = 1; i < snake.size(); i++) {
        prev2 = snake[i];
        snake[i] = prev;
        prev = prev2;
    }

    // Yönler göre yılanın başını oynat

```

```

switch (dir) {
    case LEFT: snake[0].first--; break;
    case RIGHT: snake[0].first++; break;
    case UP: snake[0].second--; break;
    case DOWN: snake[0].second++; break;
    default: break;
}

// Yılanın duvarlardan geçmesini sağla (ekranın diğer tarafından çıkar)
if (snake[0].first >= width) snake[0].first = 0;
else if (snake[0].first < 0) snake[0].first = width - 1;

if (snake[0].second >= height) snake[0].second = 0;
else if (snake[0].second < 0) snake[0].second = height - 1;

// Yılanın kendine çarptı
for (int i = 1; i < snake.size(); i++)
    if (snake[0] == snake[i])
        gameOver = true;

// Meyve yendi
if (snake[0] == fruit) {
    score += 10; // Skor artır
    snake.push_back({0, 0}); // Yılanın boyunu uzat
    GenerateFruit(); // Yeni bir meyve oluştur
}
}

public:
// Oyun sınıfı kurucu fonksiyonu
SnakeGame(int w, int h) : width(w), height(h), gameOver(false), score(0), dir(STOP) {
    snake.push_back({width / 2, height / 2}); // Yılanın başlangıç konumu
    GenerateFruit(); // İlk meyveyi oluştur
}

// Zorluğa göre bekleme süresini dön
int Difficulty(int b) {
    if (b == 1) return 250; // Kolay
    else if (b == 2) return 175; // Orta
    else if (b == 3) return 100; // Zor
    return 250; // Varsayılan
}

// Oyunu çalıştır
void Run() {
    // Kullanıcıdan zorluk seviyesini al
    while (true) {
        cout << "Select difficulty (1~3): ";

```

```

        cin >> diffinput;
        if (diffinput >= 1 && diffinput <= 3) break;
        cout << "Unknown parameter given. Please try again." << endl;
        system("cls");
    }

    int delay = Difficulty(diffinput); // Seçilen zorluğa göre gecikme süresi

    while (!gameOver) {
        Draw(); // Ekranı yaz
        Input(); // Girdi al
        Logic(); // Oyun mantığını işle
        Sleep(delay); // Oyunun timerı
    }
    cout << "Game Over!" << endl; // Oyun bitti mesajını yaz
}

};

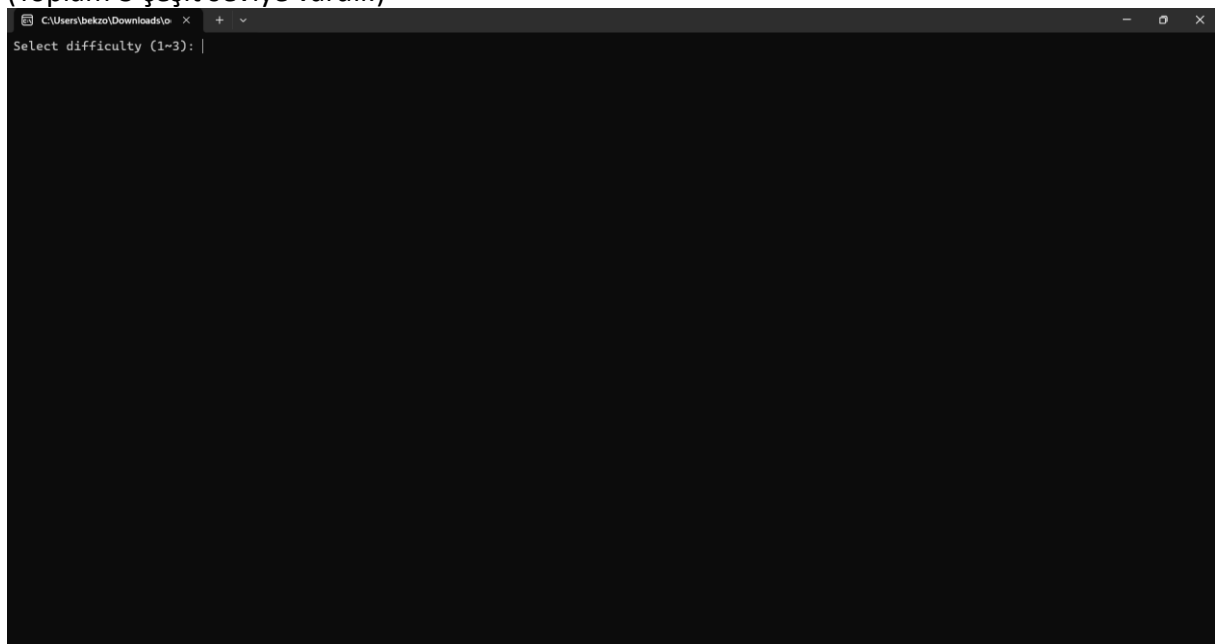
int main() {
    SnakeGame game(20, 20); // Oyun alanını belirle
    game.Run(); // Oyunu çalıştır
    system("cls"); // Konsolu temizle
    cout << "Game Over"; // Oyun bitişi mesajı
    getch(); // Getchar
    return 0;
}

```

Oyun

Oyunu açtığındaki hali

Oyun başlamadan önce zorluluk seviyesini seçmenizi istiyecek.
(Toplam 3 çeşit seviye vardır.)



Kurallar

- Sonrasında oyun sizin hareket etmenizi bekleyecektir.Sizin burdaki amacınız meyveleri toplamaktır onları (**F**) harfı ile belirttik.(Bir meyve 10 puana eşittir).
- Sizin duvardan geçebilme kabiliyetiniz vardır.
- Sizin tek düşmanınız kendinizdir, kendinize çarpıştığınızda oyun bitiyor.

```
C:\Users\bekzo\Downloads>python 0.py
#####): 1
#
#
#
#
#
#
#
# F
#
# O
#
#
#
#
#
#
#
#
#
#####
Score: 8
```

Github: <https://github.com/BulutYuksel41/oopproject>