

Operacje warunkowe: if, if else, else

Wprowadzenie

Operacje warunkowe pozwalają na kontrolowanie przepływu programu w zależności od spełnienia określonych warunków. Dzięki nim możemy wykonywać różne fragmenty kodu w zależności od sytuacji. W tej dokumentacji dowiesz się, jak działają operacje warunkowe `if`, `if else` i `else` oraz jak ich używać w Unity.

Podstawowe operacje warunkowe

1. Instrukcja `if`

- Wykonuje kod **tylko wtedy**, gdy warunek jest prawdziwy.

Składnia:

```
if (warunek)
{
    // Kod do wykonania, jeśli warunek jest prawdziwy
}
```

Przykład:

```
int zdrowie = 100;
if (zdrowie > 0)
{
    Debug.Log("Gracz żyje!");
}
```

Wyjaśnienie:

- Jeśli `zdrowie` jest większe niż 0, wyświetli się komunikat "Gracz żyje!".
-

2. Instrukcja `if else`

- Wykonuje jeden blok kodu, jeśli warunek jest prawdziwy, lub inny blok kodu, jeśli warunek jest fałszywy.

Składnia:

```
if (warunek)
{
    // Kod do wykonania, jeśli warunek jest prawdziwy
}
else
{
    // Kod do wykonania, jeśli warunek jest fałszywy
}
```

Przykład:

```
int zdrowie = 0;
if (zdrowie > 0)
{
    Debug.Log("Gracz żyje!");
}
else
{
    Debug.Log("Gracz nie żyje!");
}
```

Wyjaśnienie:

- Jeśli `zdrowie` jest większe niż 0, wyświetli się komunikat "Gracz żyje!".
 - W przeciwnym razie wyświetli się komunikat "Gracz nie żyje!".
-

3. Instrukcja `else if`

- Pozwala na sprawdzenie wielu warunków jeden po drugim.

Składnia:

```
if (warunek1)
{
    // Kod do wykonania, jeśli warunek1 jest prawdziwy
}
else if (warunek2)
{
    // Kod do wykonania, jeśli warunek2 jest prawdziwy
}
else
{
    // Kod do wykonania, jeśli żaden z warunków nie jest prawdziwy
}
```

Przykład:

```
int punkty = 75;
if (punkty >= 90)
{
    Debug.Log("Ocena: A");
}
else if (punkty >= 70)
{
    Debug.Log("Ocena: B");
}
else if (punkty >= 50)
{
    Debug.Log("Ocena: C");
}
else
{
    Debug.Log("Ocena: D");
}
```

Wyjaśnienie:

- Program sprawdza kolejne warunki:
 - Jeśli punkty są większe lub równe 90, wyświetli "Ocena: A".
 - Jeśli punkty są większe lub równe 70, wyświetli "Ocena: B".
 - Jeśli punkty są większe lub równe 50, wyświetli "Ocena: C".
 - W przeciwnym razie wyświetli "Ocena: D".

Przykład praktyczny w Unity

Użycie operacji warunkowych w skrypcie:

```
using UnityEngine;

public class WarunkiPrzykład : MonoBehaviour
{
    void Start()
    {
        int zdrowie = 50;
        bool czyGraczMaTarcze = true;

        // Sprawdź, czy gracz żyje
        if (zdrowie > 0)
        {
            Debug.Log("Gracz żyje!");

            // Sprawdź, czy gracz ma tarczę
            if (czyGraczMaTarcze)
            {
                Debug.Log("Gracz ma tarczę!");
            }
            else
            {
                Debug.Log("Gracz nie ma tarczy!");
            }
        }
        else
        {
            Debug.Log("Gracz nie żyje!");
        }
    }
}
```

Wyjaśnienie:

1. Pierwszy warunek (if):
 - Sprawdza, czy zdrowie jest większe niż 0.
 - Jeśli tak, wyświetla "Gracz żyje!".
2. Drugi warunek (if else):
 - Sprawdza, czy gracz ma tarczę (czyGraczMaTarcze).

- Jeśli tak, wyświetla "Gracz ma tarczę!".
- W przeciwnym razie wyświetla "Gracz nie ma tarczy!".

3. Warunek `else` :

- Jeśli `zdrowie` jest mniejsze lub równe 0, wyświetla "Gracz nie żyje!".

Podsumowanie

Operacje warunkowe:

- `if` - wykonuje kod, jeśli warunek jest prawdziwy.
- `if else` - wykonuje jeden blok kodu, jeśli warunek jest prawdziwy, lub inny blok, jeśli jest fałszywy.
- `else if` - pozwala na sprawdzenie wielu warunków jeden po drugim.

Operacje warunkowe są niezwykle przydatne w Unity, np. do kontrolowania zachowania obiektów w grze, sprawdzania stanu gracza czy reagowania na zdarzenia. Zrozumienie ich działania to klucz do tworzenia interaktywnych i dynamicznych gier!