

Czym są zmienne typu Reference Type i notacja z kropką w Unity?

Wprowadzenie

W programowaniu zmienne to narzędzia, które pozwalają nam przechowywać i zarządzać danymi. W Unity często używamy zmiennych typu **Reference Type** (typy referencyjne), które są szczególnie ważne, gdy pracujemy z obiektami w grze, takimi jak postacie, przeciwnicy czy interfejsy. W tej dokumentacji dowiesz się, czym są Reference Types, jak działają oraz co oznacza tajemnicza **notacja z kropką (.)**, która często pojawia się w kodzie.

Czym są Reference Types?

Reference Types to zmienne, które nie przechowują bezpośrednio wartości (jak liczby czy tekst), ale **wskazują** na miejsce w pamięci, gdzie znajduje się obiekt. Można to porównać do adresu domu – zmienna przechowuje "adres" obiektu, a nie sam obiekt.

Przykład:

Wyobraź sobie, że masz książkę. Zamiast nosić ją ze sobą, zapisujesz jej lokalizację w bibliotece (np. "półka 3, rząd 5"). Gdy chcesz przeczytać książkę, idziesz pod ten adres. W programowaniu zmienna typu Reference Type działa podobnie – przechowuje "adres" obiektu, a nie sam obiekt.

Notacja z kropką (.)

Notacja z kropką to sposób, w jaki odwołujemy się do **właściwości** lub **metod** obiektu. Kropka (.) oznacza "idź do" lub "użyj". Można to porównać do otwierania szuflad w biurku – każda szuflada zawiera coś innego, a kropka pomaga nam znaleźć to, czego potrzebujemy.

Przykład:

Jeśli masz obiekt `gracz`, który ma właściwość `zdrowie`, możesz odwołać się do niej w ten sposób:

```
gracz.zdrowie = 100;
```

Tutaj kropka (.) oznacza: "Weź obiekt `gracz` i użyj jego właściwości `zdrowie`".

Jak to działa w Unity?

W Unity większość obiektów (np. postacie, kamery, światła) to `GameObjects`, które mają komponenty. Komponenty to dodatkowe funkcjonalności, które można dodać do obiektu, np. fizykę (`Rigidbody`), kolizje (`Collider`) czy skrypty. Aby uzyskać dostęp do tych komponentów, używamy notacji z kropką.

Przykład:

1. **Transform** - komponent, który przechowuje pozycję, rotację i skalę obiektu

```
transform.position = new Vector3(0, 10, 0); // transform to zmienna która jest już dla nas zainicjalizowana
```

- `transform` – to obiekt (komponent) przechowujący informacje o pozycji.
- `position` – to właściwość, która mówi, gdzie obiekt się znajduje.
- Kropka (`.`) oznacza: "Weź komponent `transform` i użyj

2. **Rigidbody2D albo Rigidbody** – komponent, który dodaje fizykę do obiektu.

```
Rigidbody rb = GetComponent<Rigidbody>(); // musimy sami "wyciągnąć" referencję do Rigidbody  
rb.useGravity = true;
```

- `GetComponent<Rigidbody>()` – pobiera komponent `Rigidbody` z obiektu.
- `rb.useGravity` – ustawia właściwość `useGravity` na `true`.
- Kropka (`.`) oznacza: "Weź komponent `Rigidbody` i użyj jego właściwości `useGravity`".

Przykład praktyczny w Unity

Użycie notacji z kropką w skrypcie:

```
public class PlayerController : MonoBehaviour  
{  
    // Zmienna przechowująca komponent Rigidbody  
    private Rigidbody rb;  
  
    void Start()  
    {  
        // Pobierz komponent Rigidbody z obiektu  
        rb = GetComponent<Rigidbody>();  
  
        // Ustaw grawitację za pomocą notacji z kropką  
        rb.useGravity = true;  
  
        // Ustaw pozycję gracza za pomocą notacji z kropką  
        transform.position = new Vector3(0, 5, 0);  
    }  
  
    void Update()  
    {  
        // Poruszaj graczem za pomocą fizyki  
        float moveX = Input.GetAxis("Horizontal") * 10f * Time.deltaTime;  
        float moveZ = Input.GetAxis("Vertical") * 10f * Time.deltaTime;  
        rb.velocity = new Vector3(moveX, 0, moveZ);  
    }  
}
```

Wyjaśnienie

1. `rb = GetComponent<Rigidbody>();` :
 - Pobieramy komponent `Rigidbody` z obiektu, do którego podpięty jest skrypt.

- `GetComponent<T>()` to metoda, która szuka komponentu danego typu (`T`).
- 2. `rb.useGravity = true; :`
 - Używamy notacji z kropką, aby ustawić właściwość `useGravity` na `true`.
 - Kropka (`.`) oznacza: "Weź komponent `Rigidbody` i ustaw jego właściwość `
- 3. `transform.position = new Vector3(0, 5, 0); :`
 - Używamy notacji z kropką, aby ustawić pozycję obiektu.
 - Kropka (`.`) oznacza: "Weź komponent `Transform` i ustaw jego właściwość `position`".

Podsumowanie

1. Reference Types:
 - Przechowują "adres" obiektu w pamięci, a nie sam obiekt.
 - Zmiany w jednej zmiennej wpływają na inne zmienne wskazujące na ten sam obiekt.
2. Notacja z kropką (`.`):
 - Służy do odwoływania się do właściwości lub metod obiektu.
 - Można ją porównać do otwierania szuflad w biurku – każda kropka prowadzi do kolejnej "szuflady".
3. Komponenty w Unity:
 - To dodatkowe funkcjonalności, które można dodać do obiektów gry.
 - Aby uzyskać dostęp do komponentów, używamy notacji z kropką.