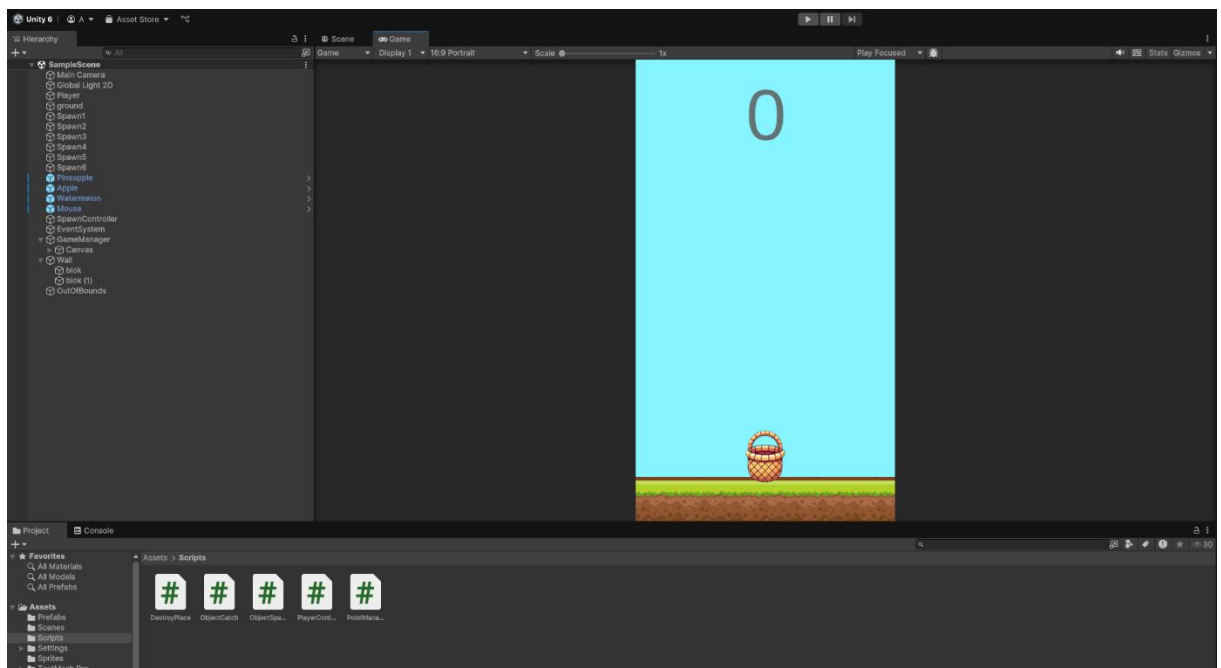
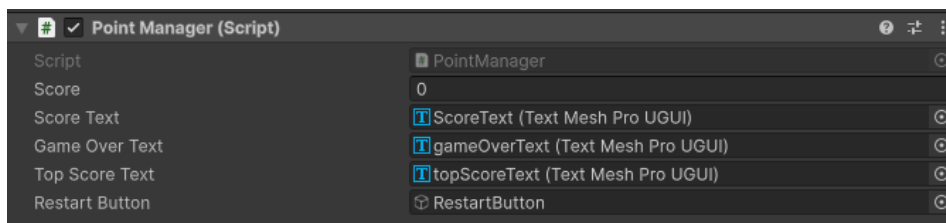
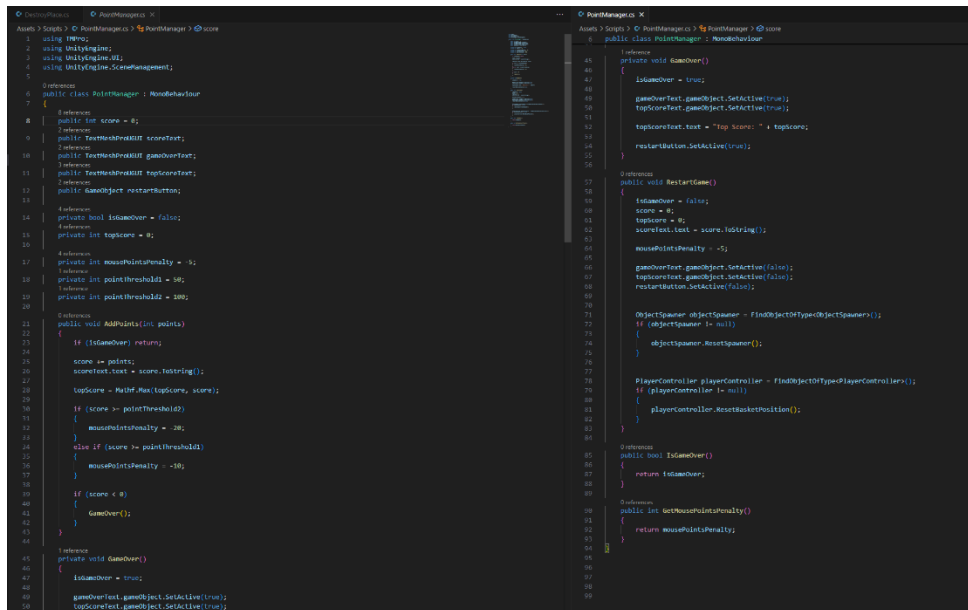


Gra mobilna „Fruit Collector”, wykonana przy użyciu środowiska Unity

1. Opis gry: Gra zręcznościowa polegająca na kierowaniu koszykiem w poziomie w celu zbierania spadających owoców i unikaniu myszy. Każdy owoc daje przypisaną do siebie liczbę punktów, a zebranie myszy odejmuje punkty z naszego wyniku. Dodana jest progresja polegająca na utrudnianiu rozgrywki wraz ze wzrostem naszego „score” poprzez zwiększenie liczby ujemnych punktów przez zebranie myszy po osiągnięciu danego „kamienia milowego”. Ostatecznie gra prowadzi do utraty sporej ilości punktów co prowadzi do jej zatrzymania w momencie uzyskania ujemnego „score”. Wtedy gra pokazuje najwyższy wynik jaki udało nam się uzyskać podczas rozgrywki i możemy próbować dalej pobić nasz dotychczasowy rekord naciskając przycisk „try again”. Edycja progresji tej aplikacji jest stosunkowo prosta, przez co można ją odpowiednio skonfigurować po wykonaniu testów sprawdzających trudność samej rozgrywki.
2. Skrypty/kody źródłowe





Point Manager w mojej aplikacji zarządza systemem punktacji gry. Odpowiada między innymi za:

1. Zarządzanie wynikiem:

- Śledzi bieżący wynik gracza (score).
- Wyświetla wynik na ekranie w UI (scoreText).

2. Aktualizacje punktów:

- Zwiększa lub zmniejsza wynik gracza, w zależności od zbieranych obiektów.

3. Kończenie gry:

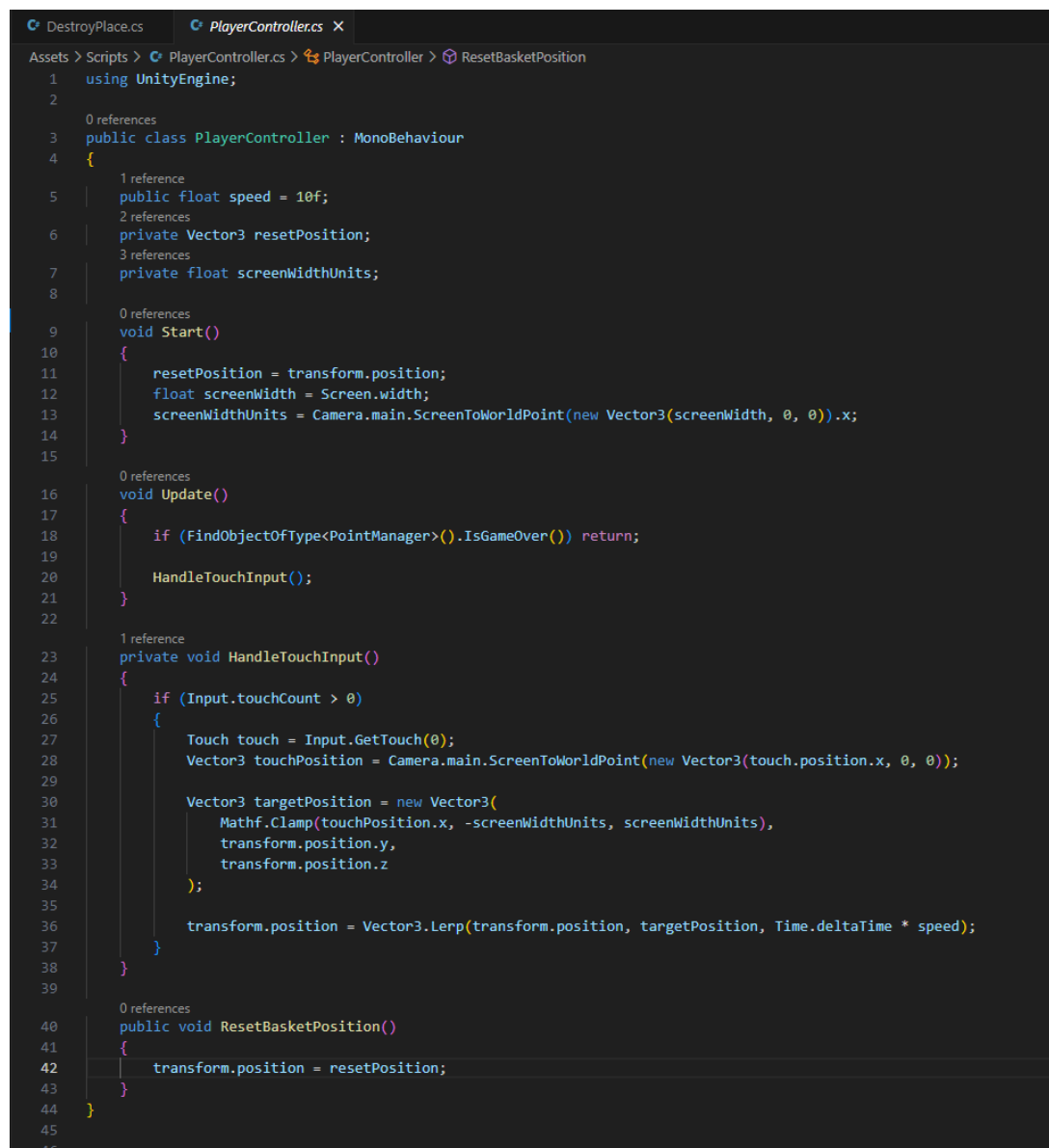
- Kiedy wynik staje się ujemny, uruchamia funkcję GameOver, która kończy grę i wyświetla komunikat o zakończeniu oraz najlepszy wynik osiągnięty w tej rundzie (wyświetlany przez topScoreText).

4. Resetowanie gry:

- Po naciśnięciu przycisku "Try Again", RestartGame resetuje wynik, ukrywa napisy o zakończeniu gry i przywraca początkowy stan gry (np. resetuje pozycję gracza i spawnera obiektów).

5. Zarządzanie wynikami:

- TopScore przechowuje najlepszy wynik danej rundy i aktualizuje się, jeśli gracz osiągnie wyższy wynik podczas rozgrywki.

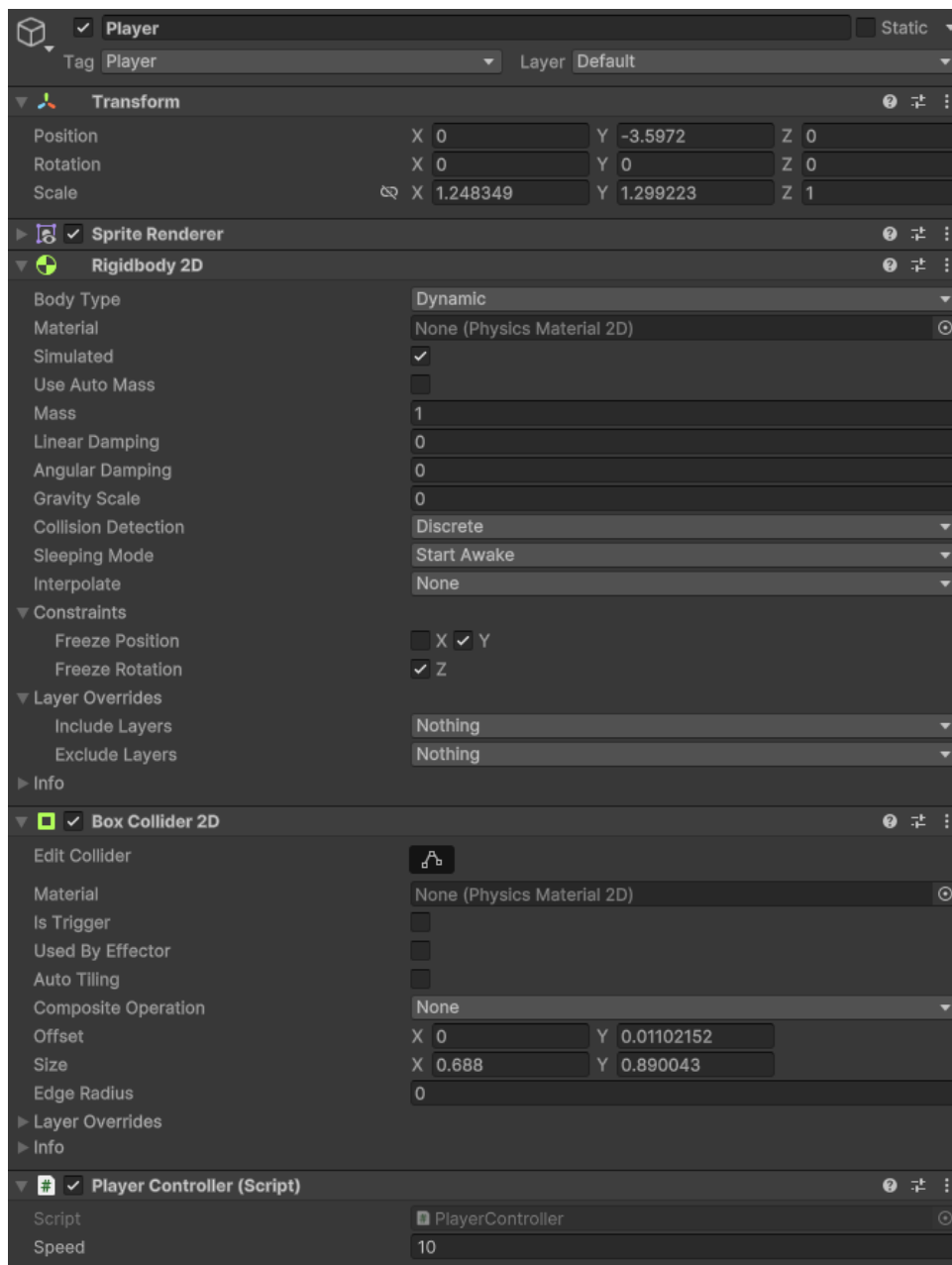


```

Assets > Scripts > PlayerController.cs > PlayerController > ResetBasketPosition
1  using UnityEngine;
2
3  0 references
4  public class PlayerController : MonoBehaviour
5  {
6      1 reference
7      public float speed = 10f;
8      2 references
9      private Vector3 resetPosition;
10     3 references
11     private float screenWidthUnits;
12
13     0 references
14     void Start()
15     {
16         resetPosition = transform.position;
17         float screenWidth = Screen.width;
18         screenWidthUnits = Camera.main.ScreenToWorldPoint(new Vector3(screenWidth, 0, 0)).x;
19     }
20
21     0 references
22     void Update()
23     {
24         if (FindObjectOfType<PointManager>().IsGameOver()) return;
25         HandleTouchInput();
26     }
27
28     1 reference
29     private void HandleTouchInput()
30     {
31         if (Input.touchCount > 0)
32         {
33             Touch touch = Input.GetTouch(0);
34             Vector3 touchPosition = Camera.main.ScreenToWorldPoint(new Vector3(touch.position.x, 0, 0));
35
36             Vector3 targetPosition = new Vector3(
37                 Mathf.Clamp(touchPosition.x, -screenWidthUnits, screenWidthUnits),
38                 transform.position.y,
39                 transform.position.z
40             );
41
42             transform.position = Vector3.Lerp(transform.position, targetPosition, Time.deltaTime * speed);
43         }
44     }
45
46     0 references
47     public void ResetBasketPosition()
48     {
49         transform.position = resetPosition;
50     }
51 }

```

PlayerController to skrypt, który zarządza ruchem koszyka (gracza) w grze. Skrypt odpowiada za obsługę dotyku na urządzeniach mobilnych, umożliwiając płynne poruszanie koszykiem w lewo i prawo. Skrypt również obsługuje resetowanie koszyka do początkowej pozycji, gdy gra się kończy lub gracz wciśnie przycisk "Try Again".



```
Assets > Scripts > ObjectSpawner.cs > ObjectSpawner > ResetSpawner
1 using UnityEngine;
2
3 0 references
4 public class ObjectSpawner : MonoBehaviour
5 {
6     2 references
7     public GameObject[] objectsToSpawn;
8     2 references
9     public float spawnInterval = 2f;
10    3 references
11    private bool isGameOver = false;
12
13    0 references
14    void Start()
15    {
16        InvokeRepeating("SpawnObject", 1f, spawnInterval);
17    }
18
19    0 references
20    void Update()
21    {
22        CheckGameOver();
23    }
24
25    0 references
26    void SpawnObject()
27    {
28        if (isGameOver) return;
29
30        int index = Random.Range(0, objectsToSpawn.Length);
31        Vector3 spawnPosition = new Vector3(Random.Range(-3f, 3f), 5f, 0);
32
33        Instantiate(objectsToSpawn[index], spawnPosition, Quaternion.identity);
34    }
35
36    1 reference
37    void CheckGameOver()
38    {
39        PointManager pointManager = FindObjectOfType<PointManager>();
40        if (pointManager != null && pointManager.IsGameOver())
41        {
42            isGameOver = true;
43            CancelInvoke("SpawnObject");
44        }
45    }
46
47    0 references
48    public void ResetSpawner()
49    {
50        isGameOver = false;
51        InvokeRepeating("SpawnObject", 1f, spawnInterval);
52    }
53 }
```

Object Spawner (Script)

Script

ObjectSpawner

Objects To Spawn

4

Element 0

Pineapple

Element 1

Apple

Element 2

Watermelon

Element 3

Mouse

+ -

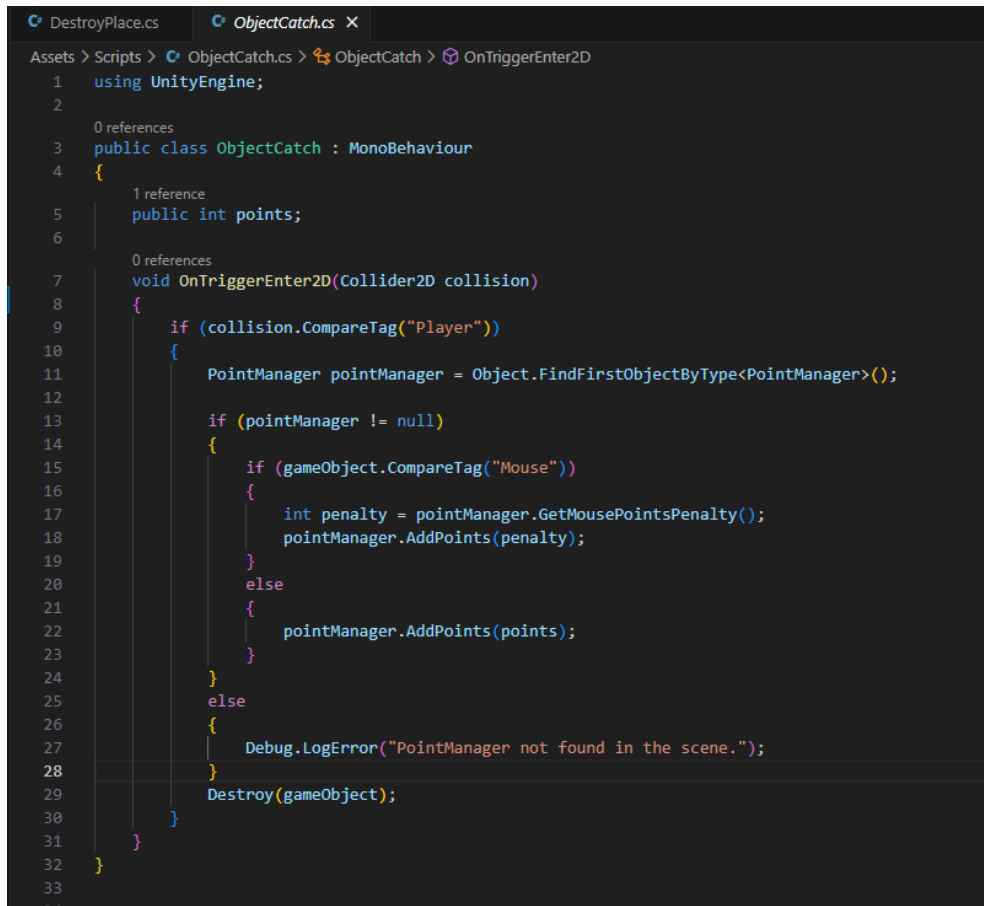
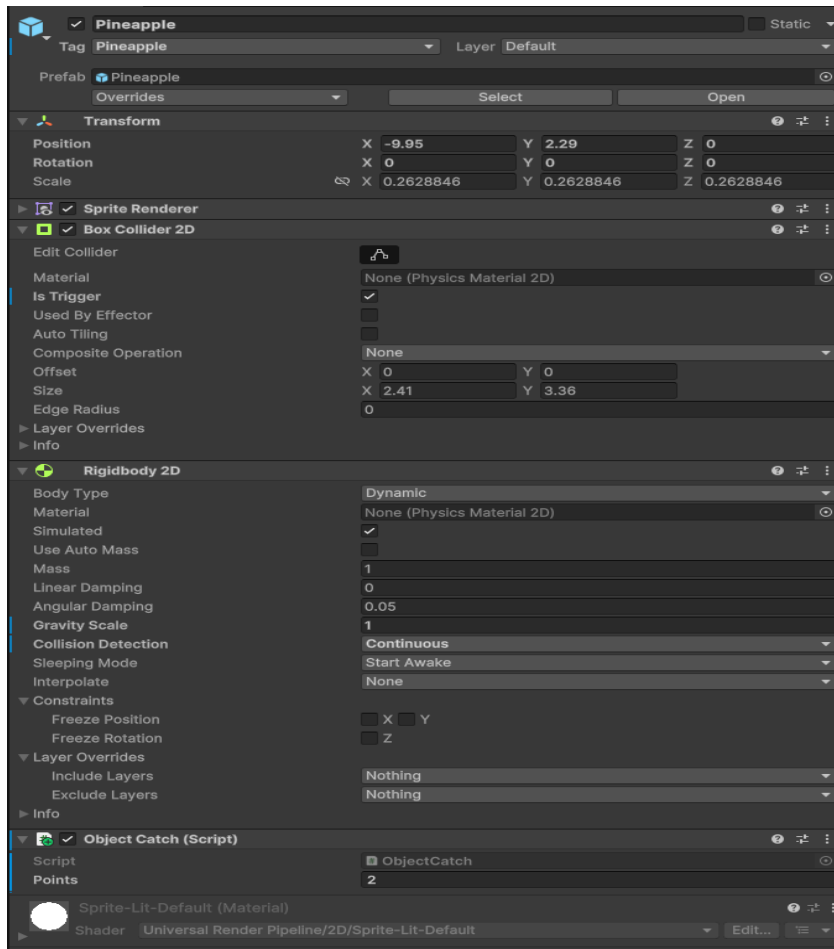
Spawn Interval

0.5

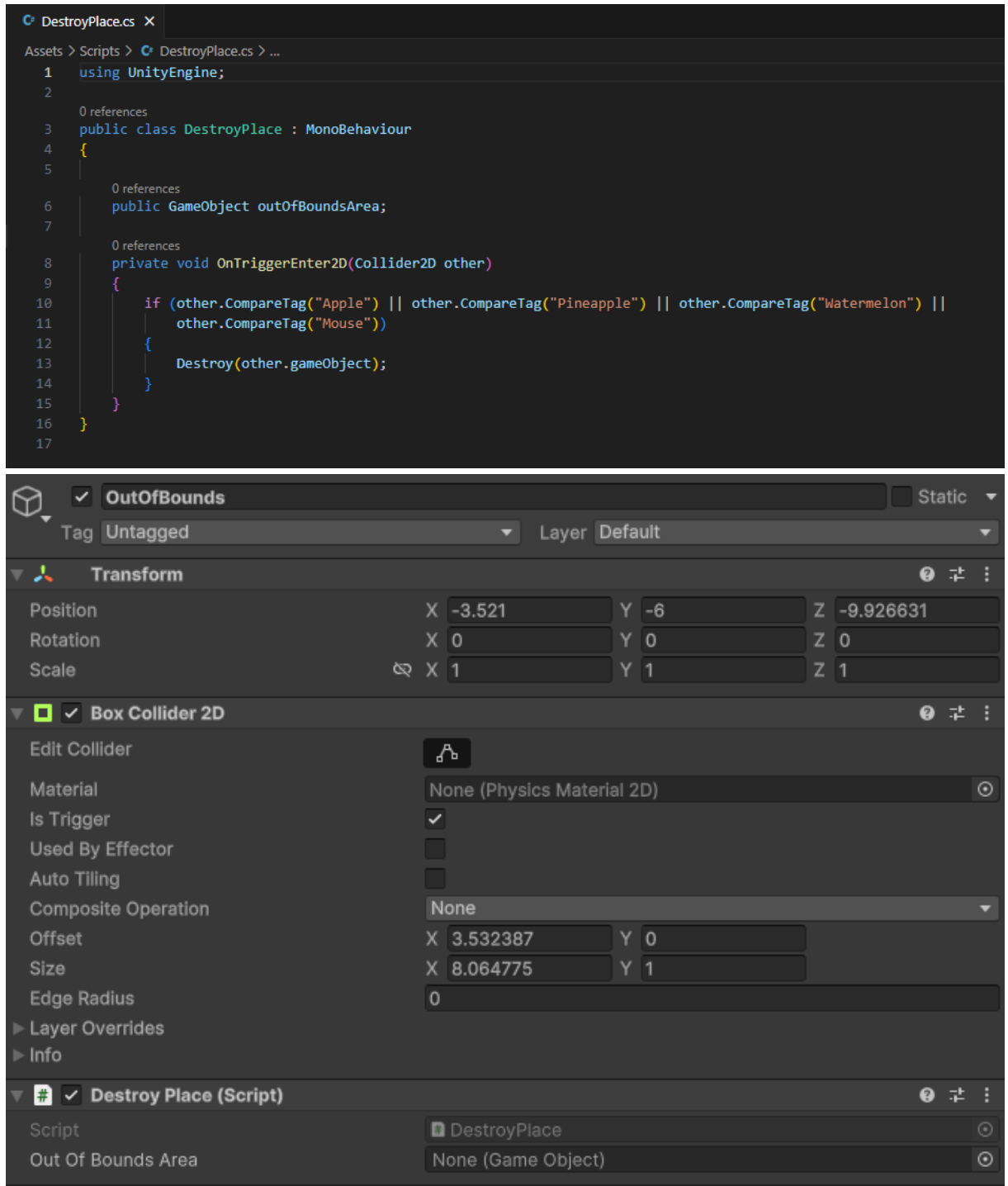
Add Component

ObjectSpawner jest odpowiedzialny za spawnowanie obiektów w grze. Zarządza on tworzeniem obiektów w określonych odstępach czasu, w losowych pozycjach na ekranie. Skrypt umożliwia dynamiczne generowanie elementów gry, które gracz może zbierać lub unikać. Odpowiada on między innymi za:

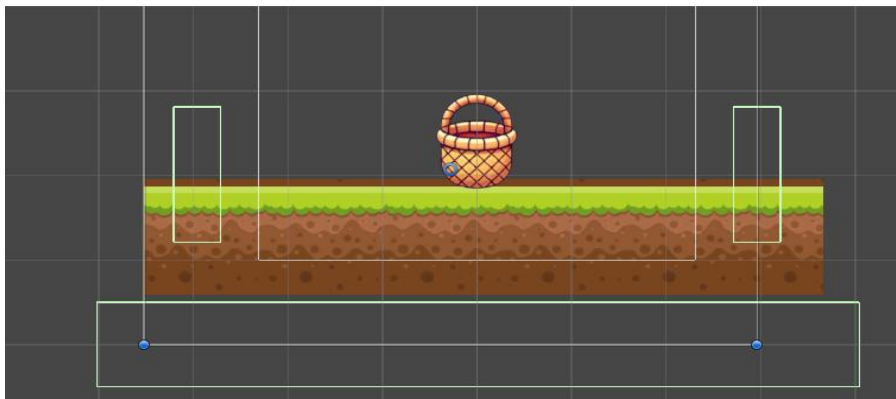
- Losowe spawnowanie obiektów: Co określony czas, skrypt wybiera losowy obiekt z tablicy prefabrykowanych elementów i umieszcza go w losowej pozycji na ekranie.
- Kontrole interwału spawnów: Czas między kolejnymi spawnowaniami jest kontrolowany przez zmienną spawnInterval.
- Resetowanie spawnowania: Po zakończeniu rundy lub restarcie gry, skrypt umożliwia resetowanie procesu spawnowania obiektów.



ObjectCatch odpowiada za obsługę obiektów, które gracz może złapać. Skrypt wykrywa kolizje pomiędzy obiektami a graczem i w odpowiedzi przyznaje punkty za złapany obiekt lub odejmuje punkty w przypadku złapania myszy. Po zebraniu obiektu, jest on usuwany ze sceny, co pozwala na płynne kontynuowanie rozgrywki.



Jednakże nie wszystkie obiekty złapiemy, przez co ich kopie mogłyby się nadpisywać co by spowodowało długoterminowo problemy z płynnością rozgrywki. Żeby tego uniknąć stworzyłem skrypt DestroyPlace, który jest wpisany w obiekt poniżej ekranu. W momencie kiedy owoc lub mysz spadnie poza ekran, wykona kolizję z OutOfBounds co spowoduje usunięcie obiektu ze sceny, dzięki czemu zachowamy płynność rozgrywki.



Planowane modyfikacje:

- Wprowadzenie nowego obiektu który po zebraniu kończy rozgrywkę.
- Dodanie muzyki w tle oraz efektów dźwiękowych.
- Zwiększenie systemu progresji(po uzyskaniu określonych kamieni milowych):
 - Przyspieszenie prędkości spadania obiektów.
 - Zmiana tła.
 - Zmiana muzyki w zależności od tła.