# Module User Experience | Gaming

## Onderdeel Unity Practicum 1: Verzamel pickups

## Auteurs

Paul Bergervoet (Gameplay en teamleiding)

Monique Dewanchand (C#)

Lars Tijsma (Unity)

Anton Visser (Gameplay en testen)



Unity Opdracht bij GD1 - Doel: alle pickups verzamelen

In deze opdracht ga je ervoor zorgen dat de speler kan zien dat hij of zij het doel heeft behaald: *alle pickups zijn verzameld*.

Bij dit practicum is er een startpunt, een Unity-project (Unity versie 2021) waarin je het practicum kunt maken. Je kunt ook je eigen Unity-project van Roll-a-Ball gebruiken en daarop voortbouwen.

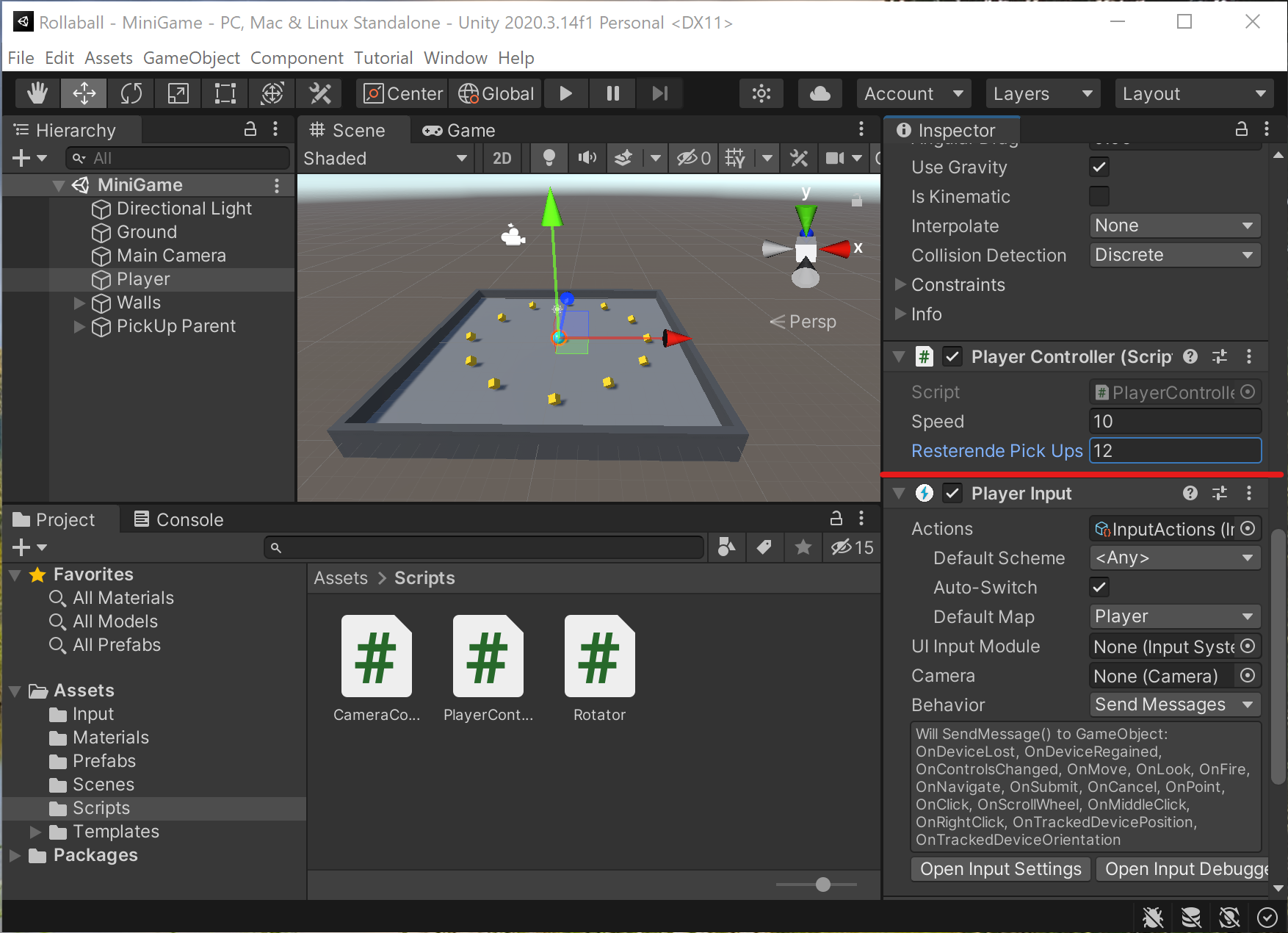
## Variabele maken

* Open het script 'PlayerController'.
* Voeg aan dit script een nieuwe variabele resterendePickUps toe zoals hieronder te zien is.

| public class PlayerController : MonoBehaviour {  public float speed = 0;  private Rigidbody rb;  private float movementX;  private float movementY;   public int resterendePickUps;   void Start()  {  rb = GetComponent<Rigidbody>();  } |
| --- |

* Open de editor en zet de variabele op 12 zoals hieronder te zien is:

**Tip:** als je zelf een ander aantal pick-ups in de scene hebt staan, gebruik dan het aantal dat jij hebt in plaats van 12.



<<Afbeeldingen/resterendel\_pick-ups\_instellen.png>>

### Variabele gebruiken

Zodra de bal met een pick-up botst, moet resterendePickUps met 1 verlaagd worden.

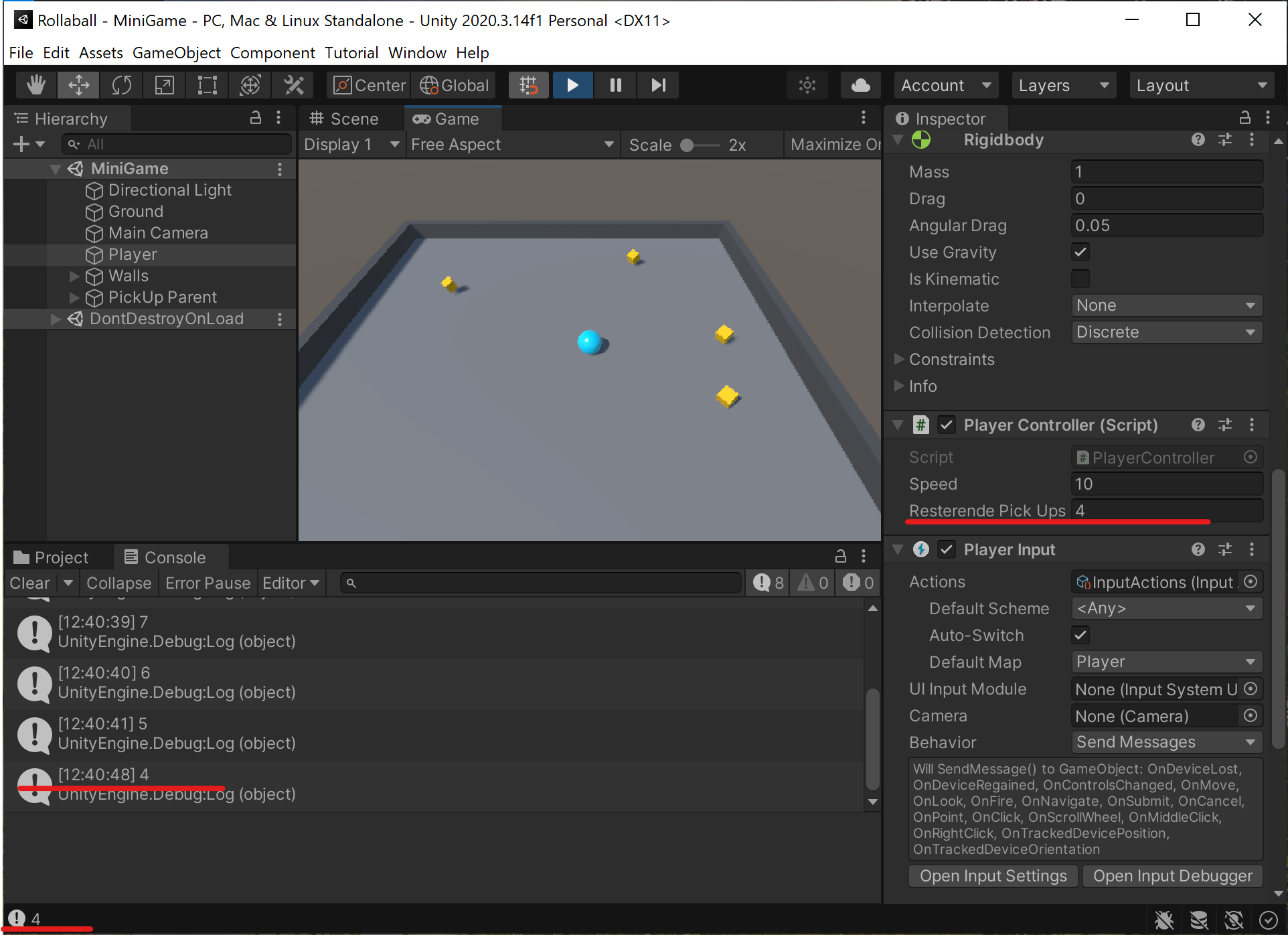
* Plaats de code om resterendePickUps met 1 de verlagen in de methode OnTriggerEnter(Collider other) zoals hieronder te zien is. Bedenk daarbij zelf hoe je deze variabele met 1 kunt verlagen.

| void OnTriggerEnter(Collider other)  {  if (other.gameObject.CompareTag("PickUp"))  {  other.gameObject.SetActive(false);   *//Plaats hier de code om resterendePickups met 1 te verlagen*  } } |
| --- |

* Controleer of de code werkt door het resterendePickUps naar de console te schrijven met Debug.Log zoals hieronder te zien is.

| void OnTriggerEnter(Collider other)  {  if (other.gameObject.CompareTag("PickUp"))  {  other.gameObject.SetActive(false);    *//Plaats hier de code om resterendePickups met 1 te verlagen*   Debug.Log(resterendePickUps);  } } |
| --- |

* Test het spel en controleer of je de output ziet die met rood aangegeven is in onderstaande screenshot:



<<Afbeeldingen/drie\_pick\_ups\_overgebleven.png>>

### Doel behalen

Het doel voor de speler is het verzamelen van alle pick-ups. In deze opdracht toon je alleen een tekst in de console zodat je (als programmeur) weet dat het doel behaald is en het spel dit op de juiste manier registreert. In een andere opdracht leer je hoe je naar een ander scherm springt als de speler doel heeft bereikt.

* Plaats in de methode OnTriggerEnter code die controleert of resterendePickUps gelijk is aan 0 en in dat geval de tekst "Het doel is behaald" naar de console schrijft.

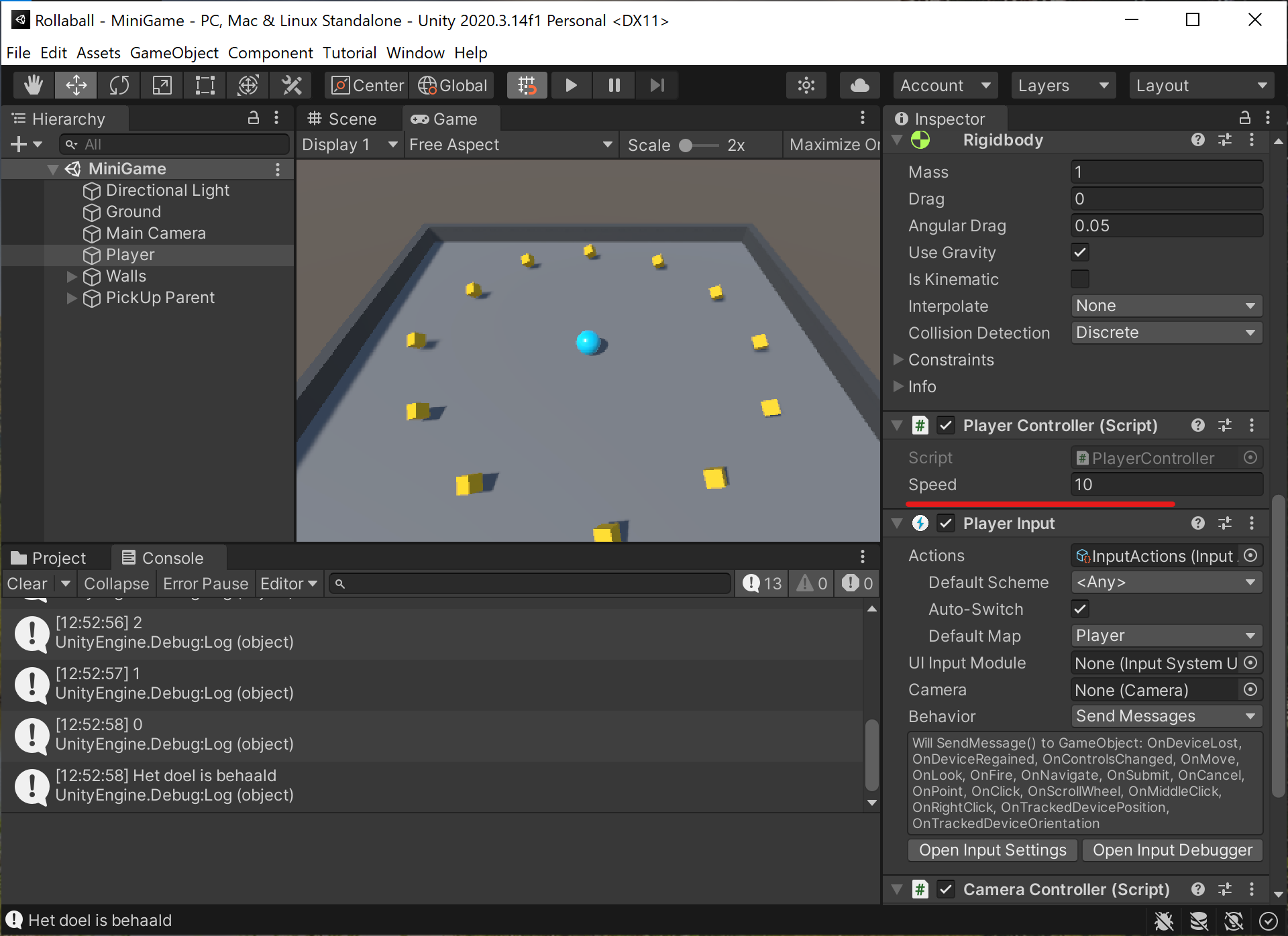
| void OnTriggerEnter(Collider other)  {  if (other.gameObject.CompareTag("PickUp"))  {  other.gameObject.SetActive(false);   *//Plaats hier de code om resterendePickups met 1 te verlagen*   Debug.Log(resterendePickUps);   *//Plaats hier de code waarmee je*   *// - controleert dat alle pickups zijn verzameld*  *// - de tekst naar de console schrijft als dit zo is*  } } |
| --- |

**Tip:** het is handig om zo snel mogelijk en zo vaak mogelijk te testen of de geschreven code werkt zoals je dit hebt bedoeld. Het schrijven van teksten en variabelen is daarbij een goed hulpmiddel.

### Variabele private maken

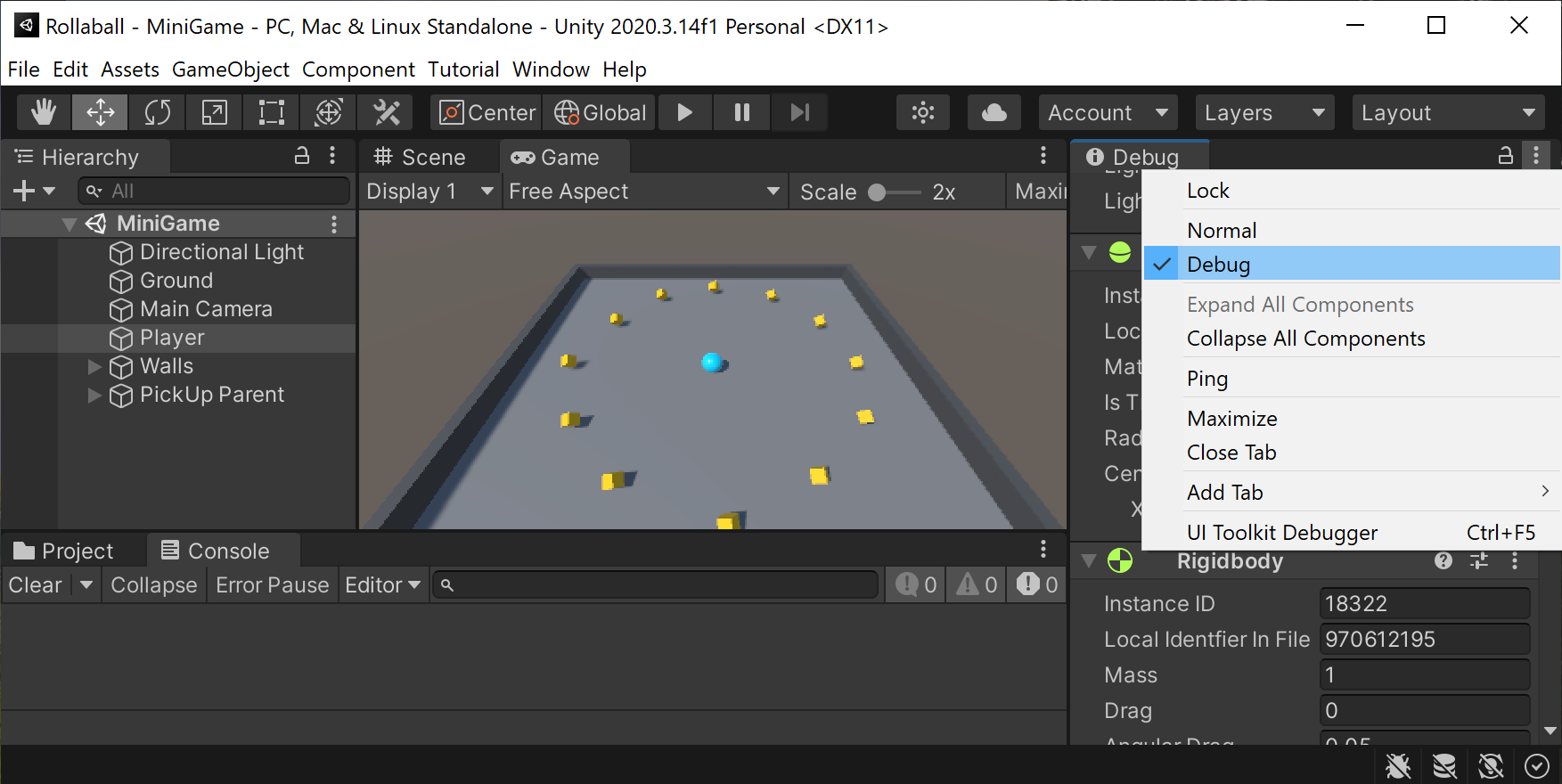
In Unity maak je een variabele public als je als game designer niet zeker bent over de geschikte waarde die de variabele moet hebben. Doordat je de waarde van de variabel in de editor kunt aanpassen kun je snel verschillende waarden proberen. De waarde van resterendePickUps staat echter vast en is gelijk aan het aantal pick ups aan het begin van het spel. Deze variabele kan daarom beter niet public zijn, maar private.

* Verander de declaratie van de variabe resterendePickUps van public in private.
* Controleer dat de variabele niet meer in de editor te zien is:

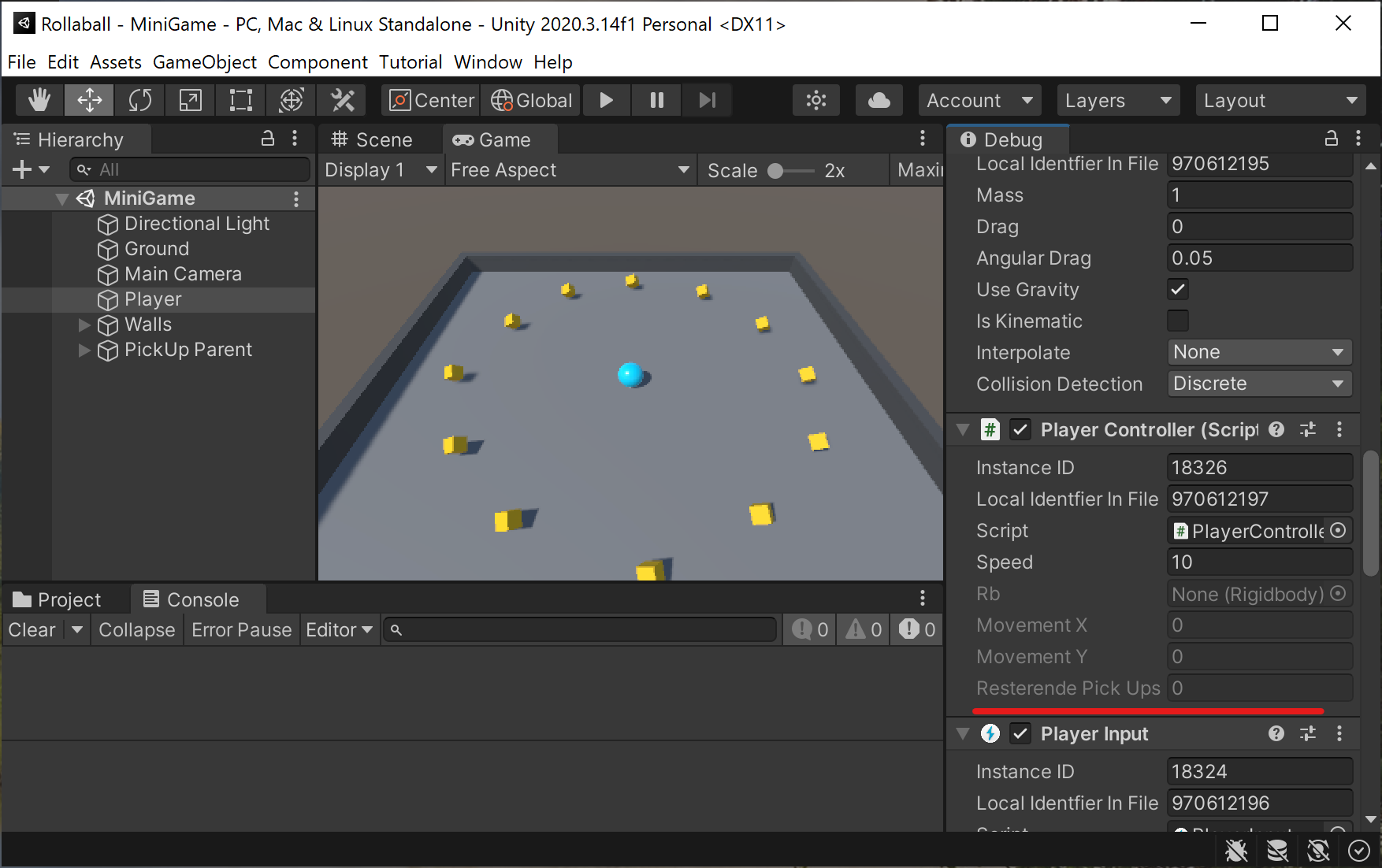


Hoewel de variabele nu beter beschermd is tegen onbedoelde wijzigingen, is het onhandig dat je de waarde niet meer kunt zien in de editor. Je kunt de variabele weer zichtbaar maken door het Inspector-venster in debug-modus te zetten.

* Klik op de drie puntjes van het Inspector-window en selecteer de optie “Debug” zoals hieronder te zien is



Je ziet nu van alle componenten alle verborgen variabelen in grijs. Als het goed is zie je dat de variabele resterendePickUps op 0 staat en dat je dit niet kunt aanpassen in de Unity-editor:



Het spel werkt nu niet meer goed omdat resterendePickUps op 0 wordt geïnitialiseerd en zodra de speler een pick-up pakt, wordt resterendePickUps negatief. Dit kun je oplossen door de variabele in een handige methode te initialiseren met de juiste waarde.

* Zet de waarde van resterendePickUps op 12 (of je eigen aantal) in de methode Start:

| void Start() {  rb = GetComponent<Rigidbody>();   *//Zorg hier dat resterendePickUps de juiste start-waarde krijgt* } |
| --- |

* Test het spel en controleer dat alles weer goed werkt.

### Automatisch alle pickups initialiseren

Wanneer je als game designer het aantal pick-ups verandert, moet je goed opletten dat je resterendePickUps met het goede getal blijft initialiseren. Dat is omslachtig en foutgevoelig. Unity biedt functionaliteit om automatisch het aantal pickups op te halen.

De methode GameObject.FindGameObjectsWithTag("Pick Up") geeft een lijst terug met alle Game Objecten die de Tag "Pick Up" hebben.

* Vervang de initialisatie van resterendePickUps met een aanroep van bovenstaande methode zoals hieronder te zien is:

| void Start() {  rb = GetComponent<Rigidbody>();   resterendePickUps = GameObject.FindGameObjectsWithTag("Pick Up"); } |
| --- |

* Test het spel. Als het goed is zie je nu een foutmelding. Wat betekent deze foutmelding?
* Los deze foutmelding op door achter bovenstaande Lenght te zetten:

| void Start() {  rb = GetComponent<Rigidbody>();   resterendePickUps =  GameObject.FindGameObjectsWithTag("PickUp").Length; } |
| --- |

* Voeg een aantal pickups toe, of verwijder er een paar en test of het spel nog steeds correct werkt.

# Optionele uitbreidingen

## Tekst aan de speler laten zien

De teksten die je naar de console schrijft zijn bedoeld voor jou als programmeur en niet voor de speler van het spel. In video’s 7.1 tot en met 7.4 van de Unity Roll-a-Ball tutorial leer je hoe je tekst op het speelscherm kan zetten zodat de speler dit kan zien.

* Volg de video’s 7.1 tot en met 7.4 van de Roll-a-Ball speellijst: <https://youtu.be/yxt1DYerz8M>. Pas de instructies zo toe dat ze op jouw versie van het spel werken.

## Uitdaging: nog dynamischer

Je kunt nu een willekeurig aantal pick ups in het level plaatsen zonder dat je de code daarop hoeft aan te passen. De code werkt echter niet wanneer je het spel zo zou maken dat de pick-ups ook kunnen verschijnen tijdens het spelen van het spel.

* Probeer de code zo te maken, dat het spel nog steeds zou werken als er af en toe pick ups bij zou komen tijdens het spelen van het spel.