

# Characta2D

**Character Controller made on custom physics**

Vito Domenico Tagliente

[vitodomenicotagliente@gmail.com](mailto:vitodomenicotagliente@gmail.com)

## Documentazione

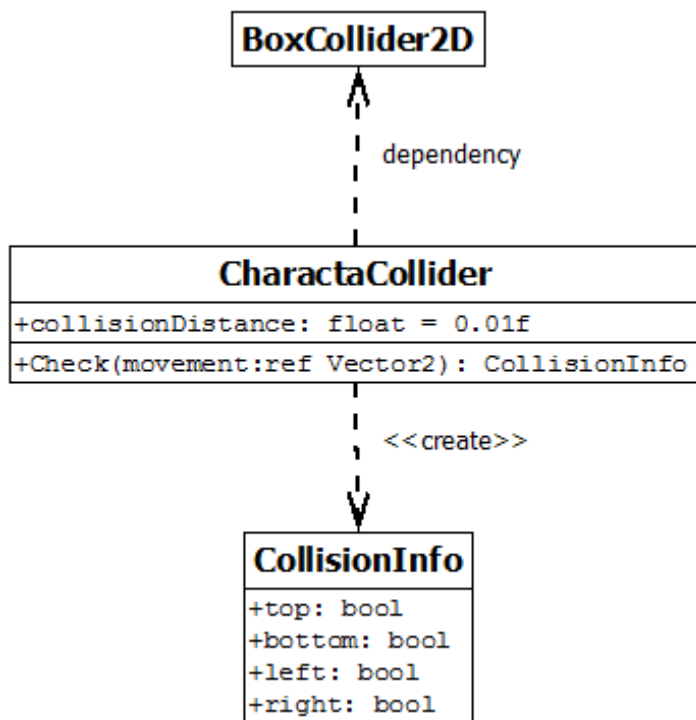
Characta2D è un pacchetto per Unity3D. Trattasi di un insieme di script atti a definire un controllore di personaggio per giochi in due dimensioni.

Perché utilizzare Characta2D?

Se consideriamo lo sviluppo di giochi *Platformer*, in stile Super Mario, ritroviamo che adottando le funzionalità standard della fisica integrata dell'ambiente Unity3D, risulta estremamente complesso definire e gestire in maniera desiderata il movimento del personaggio attraverso i livelli di gioco. Cosa vuol dire? Implementare un salto su muro, per esempio adottando il motore di fisica di default, diventa una impresa. Questo è dovuto al fatto per cui un salto sul muro non è fisicamente reale.

Al contrario, Characta2D implementa un sistema di controllo del personaggio basato su fisica personalizzata. Vuol dire che viene definito un sistema di individuazione delle collisioni e di azione sul movimento dell'oggetto in esame. In questo modo risulta semplice e possibile definire azioni e movimenti adatti al tipo di gioco che si vuole realizzare.

### Gestione delle collisioni

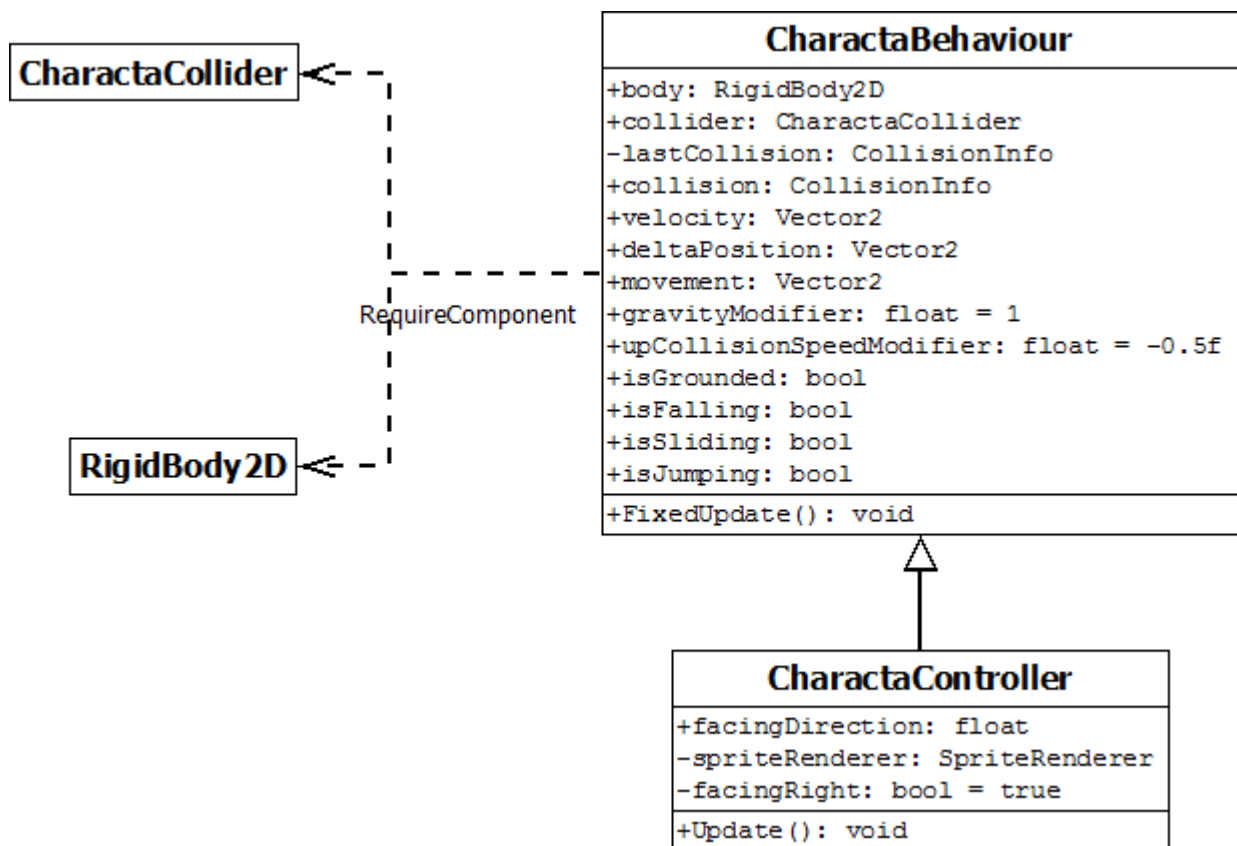


Per gestire le collisioni, è stato definito un collider personalizzato. Da quanto mostrato in figura, tale componente richiede la presenza di un **BoxCollider2D**. Tale scelta permetterà all'ambiente di non stravolgere il comportamento globale del livello di gioco. La fisica personalizzata andrà ad interessare esclusivamente il comportamento dell'oggetto come conseguenza dell'input dell'utente. Il resto (il livello di gioco, i nemici, i proiettili) interagiranno con l'oggetto in esame come definito dalle regole generali del motore di fisica di Unity.

A livello di controllo delle collisioni, il **CharactaCollider** genera un **CollisionInfo**. Trattasi di una struttura dati modellata per

la memorizzazione delle informazioni di collisione attive.

## CharactaController



Trattasi del componente definito per la gestione del personaggio di gioco. Da quanto mostrato in figura, un CharactaController è una specializzazione di un CharactaBehaviour.

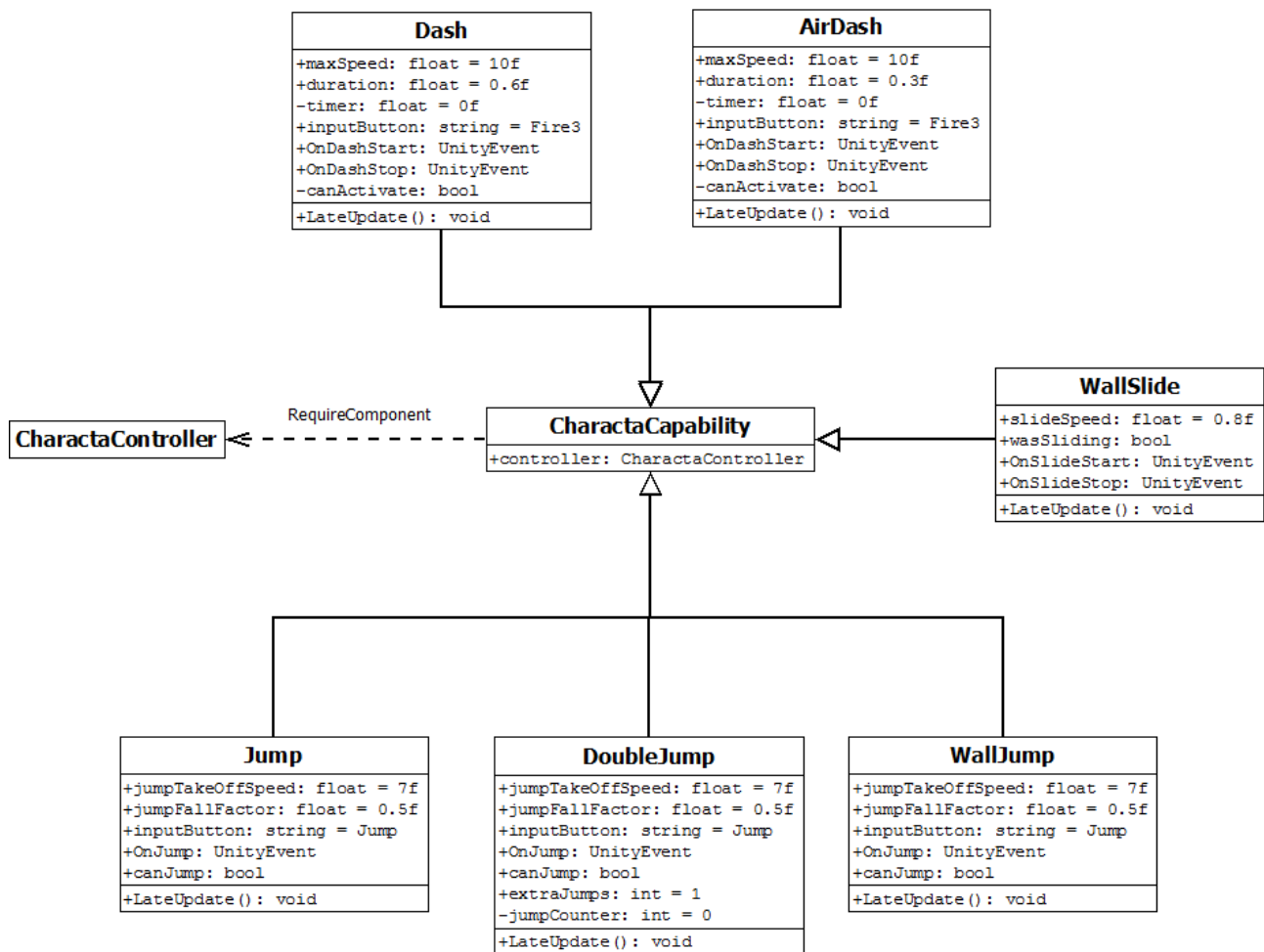
Un CharactaBehaviour definisce i comportamenti fisici dell'oggetto in relazione alle informazioni di collisione individuate dal CharactaCollider.

È da notare la presenza del Rigidbody2D, la cui presenza serve a mantenere la coerenza dell'oggetto con le regole fisiche del livello. In particolare, il corpo rigido viene automaticamente definito come un corpo cinematico. Vuol dire che questo si sposta in relazione alla propria velocità; al contrario, forze e gravità non avranno effetto.

```

Rigidbody2D _body;
public Rigidbody2D body
{
    get
    {
        if (_body == null)
        {
            _body = GetComponent<Rigidbody2D>();
            // Initialize it
            _body.bodyType = RigidbodyType2D.Kinematic;
            _body.simulated = true;
            _body.useFullKinematicContacts = true;
        }
        return _body;
    }
}
  
```

## Modellare le abilità del personaggio



A questo punto viene definito un sistema per la modellazione delle abilità. Vuol dire che integrando o meno questi componenti sull'oggetto, questo sarà in grado di saltare, eseguire doppi salti, salti sui muri, scatti a terra e in aria, etc.

Infinite possibilità di comportamento potranno essere definite agendo sulla velocità dell'oggetto. Come, per esempio, è stato fatto con l'abilità di salto:

```

void LateUpdate()
{
    if (Input.GetButtonDown(inputButton) && canJump)
    {
        controller.velocity.y = jumpTakeOffSpeed;
        OnJump.Invoke();
    }
    else if (Input.GetButtonUp(inputButton))
    {
        if (controller.velocity.y > 0)
        {
            controller.velocity.y *= jumpFallFactor;
        }
    }
}

```