

Characta2D

Character Controller made on custom physics

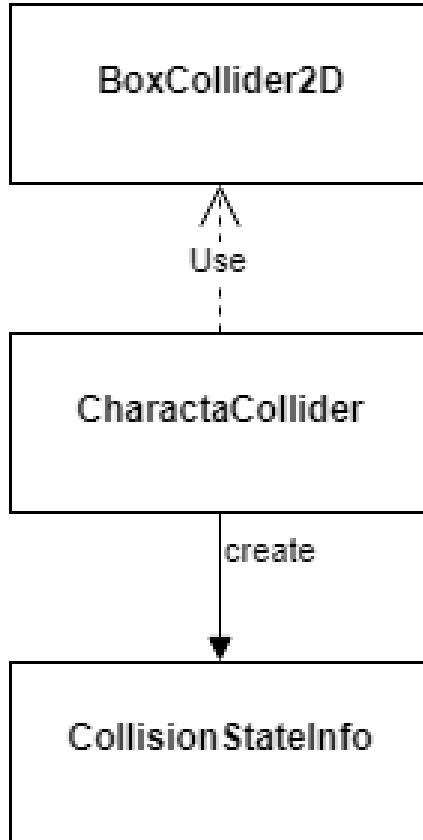
Why custom physics?

- A pure physics-based solution for a platform character is like the hell
- For example, wall jumps (and other stuff) are difficult



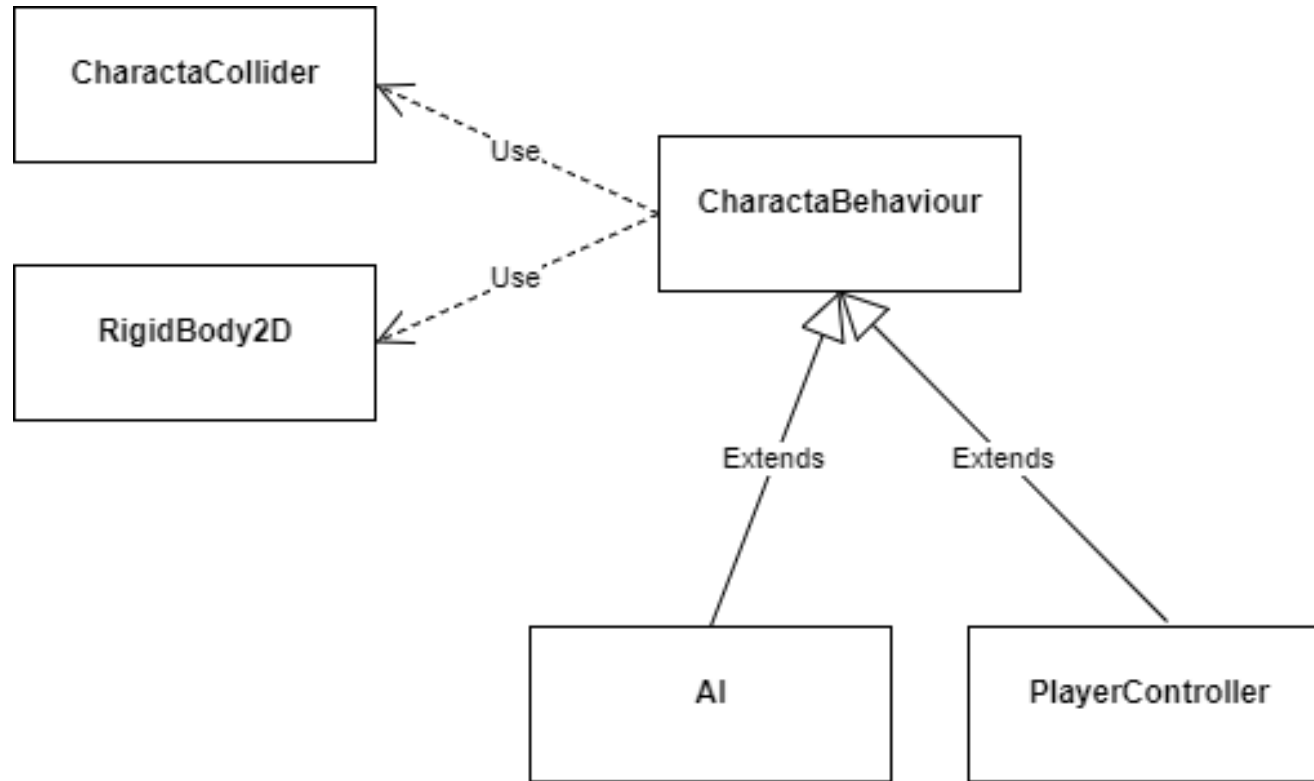
you need to compute collision and decide the movement for yourself, using collider to mostly listen to collision instead of letting the engine decide what's 'great'

CharactaCollider



- Requires an Unity's BoxCollider2D: it doesn't break the environment standard physics system
- Handles the collision detection
- Predicts the collision based on the movement

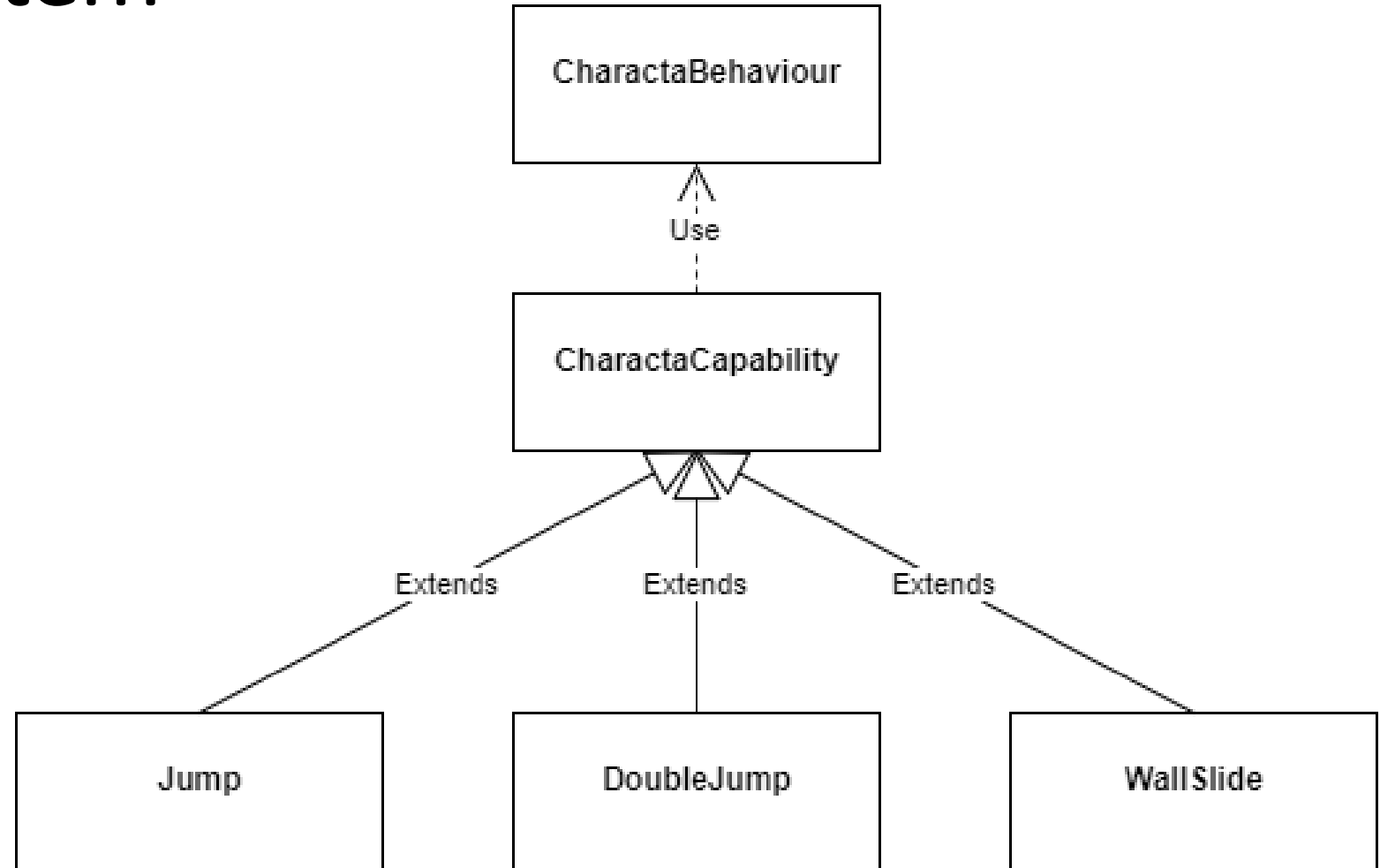
CharactaBehaviour



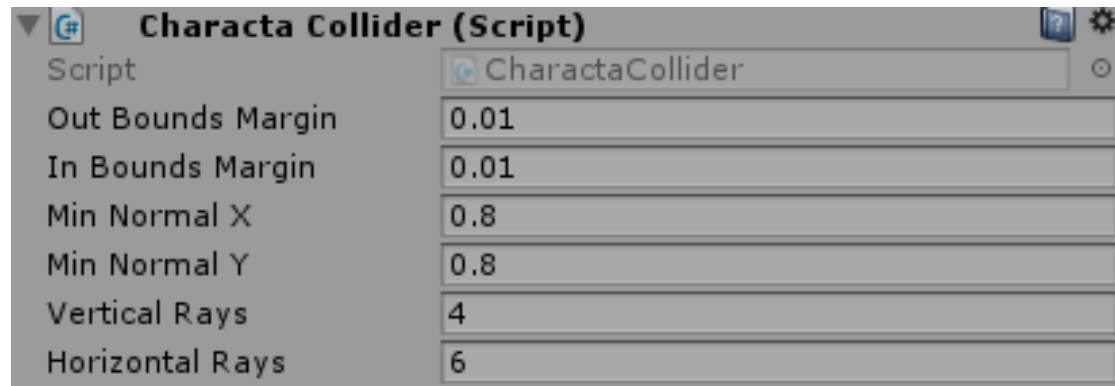
- It's the base class from which define 2D world's characters and objects.
- Requires an Unity Rigidbody2D: it is automatically set to Kinematic body type, which means that the rigid body does still move via its velocity, but the velocity is not affected by forces or gravity.

Capability System

- Define different types of capabilities
- Simply add these components to the CharactaController to enable jumps, wall jumps and so on.
- Create Characta Capabilities by yourself. Customize your package.

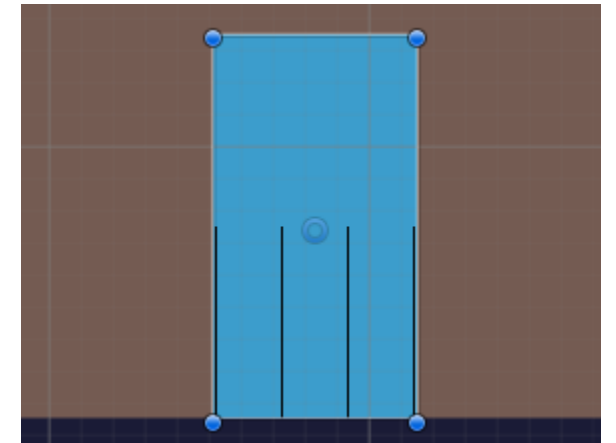


Collision Detection

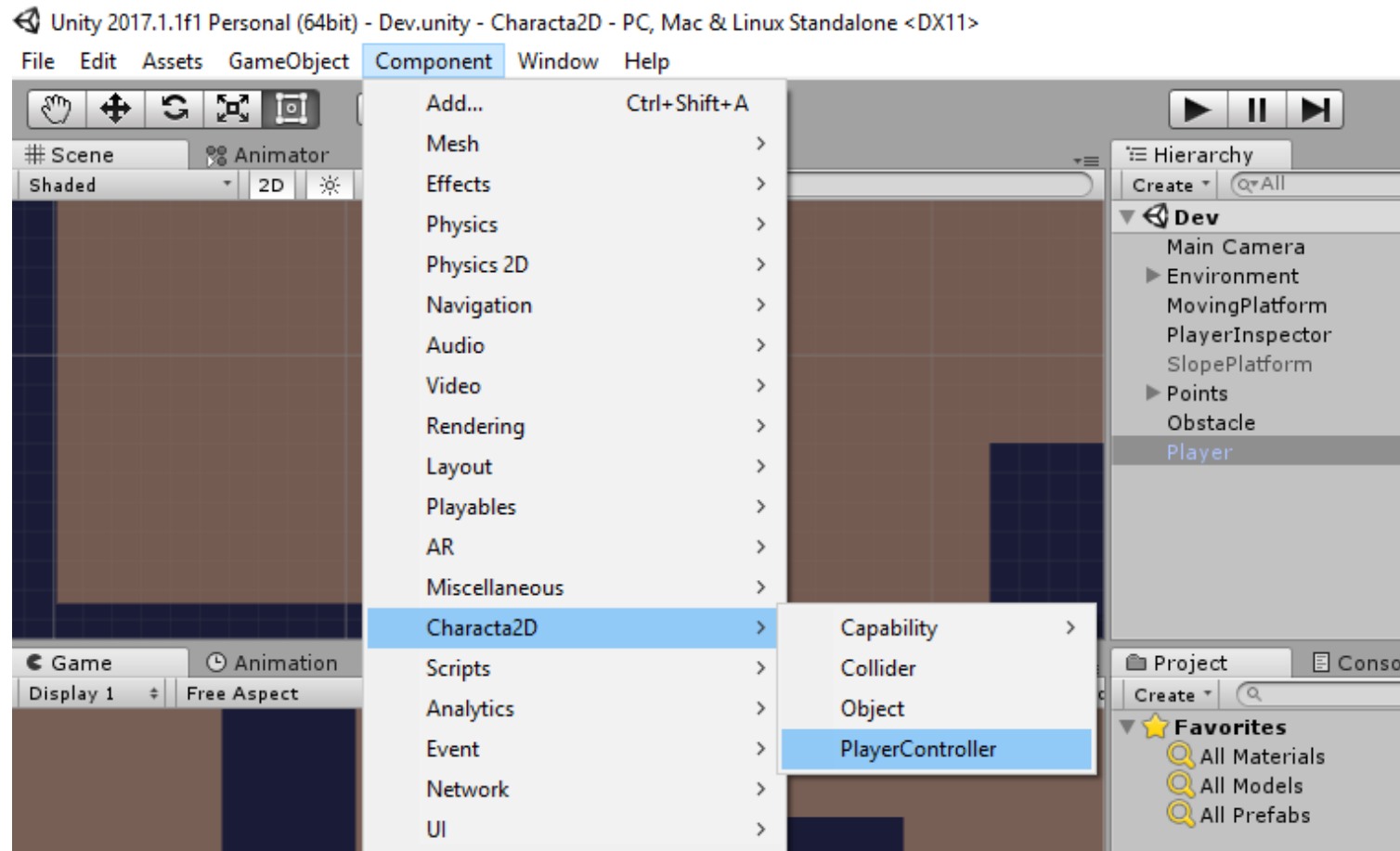


The Characta collision detection is based on raycasts. It is possible to set the number of horizontal and vertical rays (min 3) that should be used by the CharactaCollider. Rays will be used to check the collision state of the current object.

The OutBoundsMargin attribute is used to define at which distance, starting from the BoxCollider2D bounds, a collision can be valued as true. The InBoundsMargin attribute is used to define the margin inside the BoxCollider2D bounds in which the rays should start. As you can see in picture below, the vertical rays aren't on the BoxCollider2D bounds border, but they start at 0.01f distance inside.



How to setup a player



Simply add the PlayerController on a sprite. All the dependencies will be automatically resolved. Next set and customize all the parameters, starting from the collider to the player controller. Finally, customize the player behaviour adding some capabilities.

Characta2D

Credits: Vito Domenico Tagliente
vitodomenicotagliente@gmail.com