

# How to create your own character?

This page explains how characters function in the TopDown Engine and how to build your own.

- [Introduction](#)
- [Base concepts](#)
- [Hierarchy](#)
- [How do I create an Agent ?](#)
  - [Automatic Creation](#)
  - [Copy](#)
  - [Components approach](#)

## Introduction



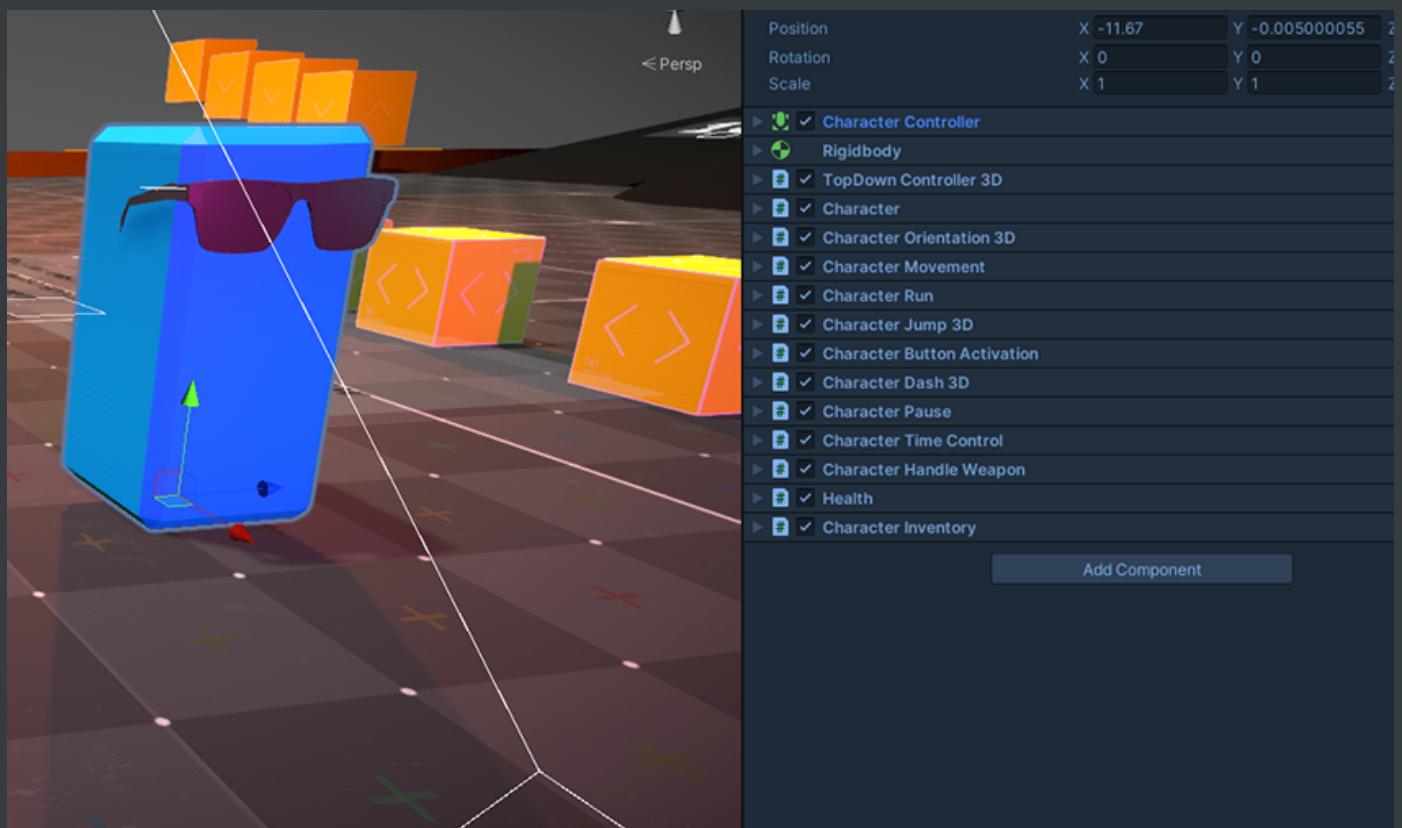
A typical playable character, the Suspenders character from the Loft3D demo

"**Agents**" in the TopDown Engine is a term used to describe any kind of characters, whether they're playable characters, or enemies, NPCs, etc. There are a few core classes that make these agents work, and that you'll need to get familiar with if you want to expand and create more character abilities for example.

In the meantime, this page aims at presenting the **basic concepts** and allowing you to quickly create your own character (player controlled or AI based).

## Base concepts

The TopDown Engine is designed to **work for both 2D and 3D projects**. The components you'll use to create characters for one setup or the other will be mostly the same, but some will be specific to 2D or 3D. When you encounter a component or an ability, if it doesn't have 2D or 3D in its name, it's safe to assume it'll work for both. Otherwise, you'll want to use components with 2D in their name for 2D only, and components with 3D in their name for 3D only.

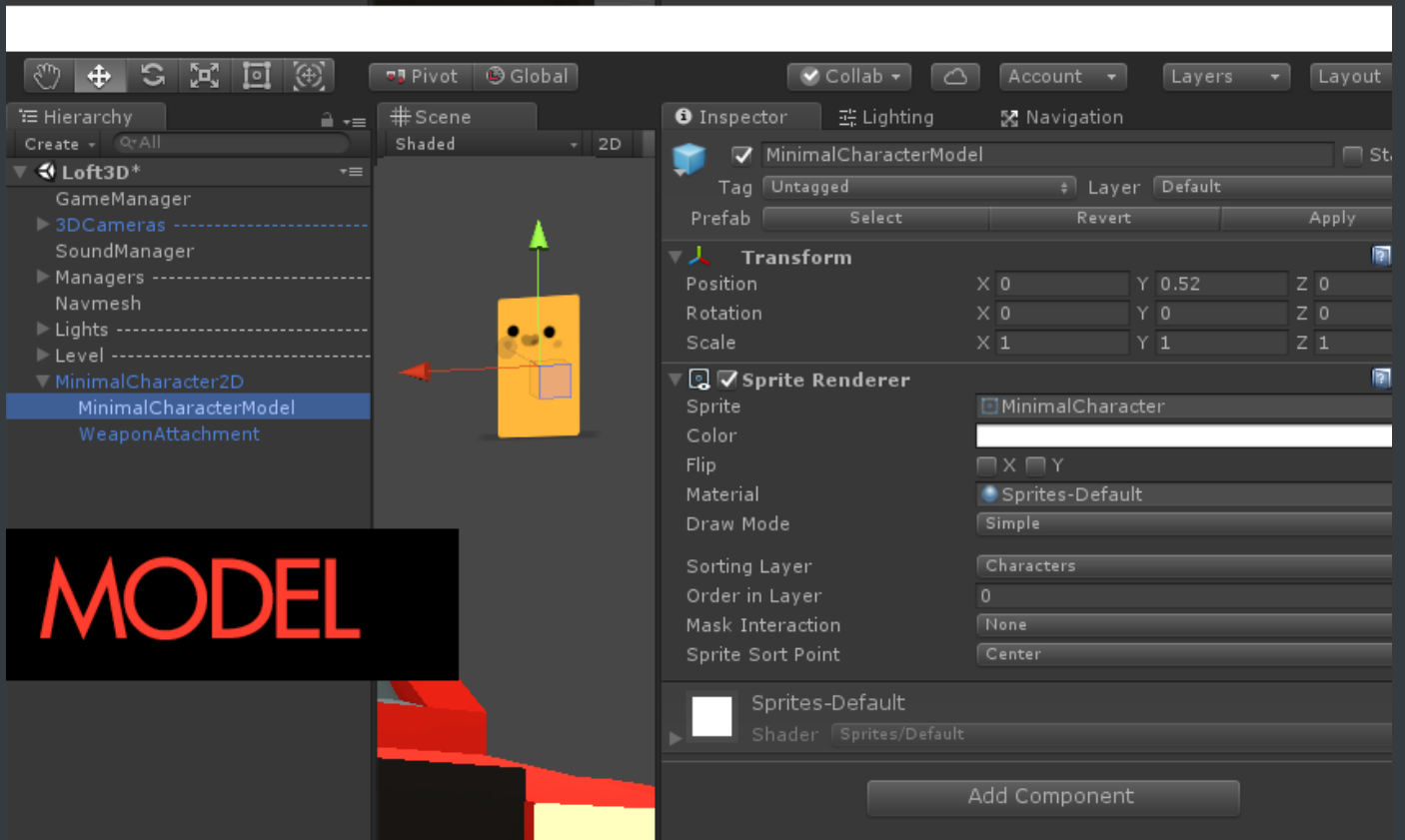
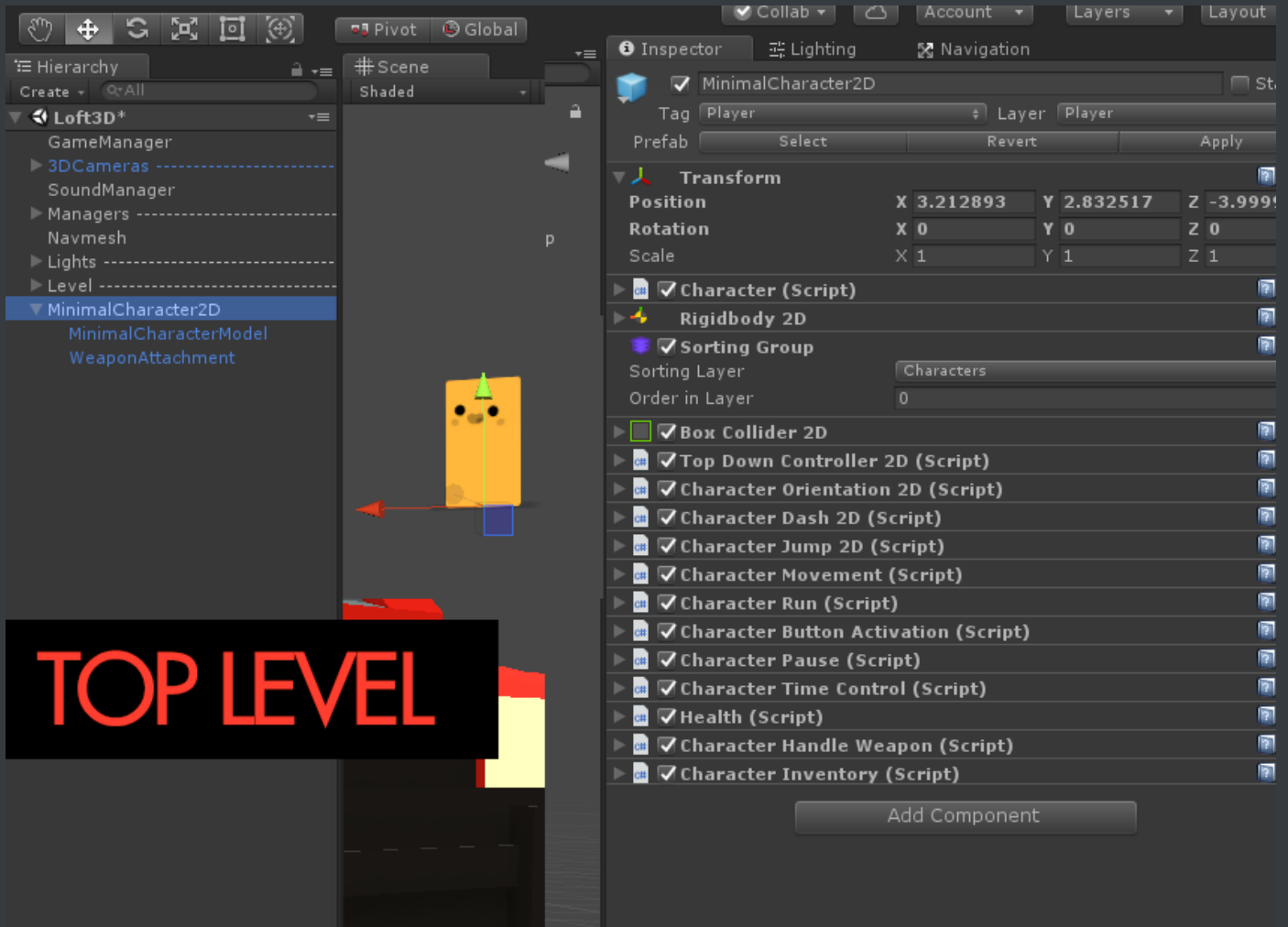


An example of the stack of components commonly found on Characters, in this case a 3D character

[This page](#) goes into more details about the mandatory components of a Character, but here's a **brief rundown**. An agent in the TopDown Engine usually has these components :

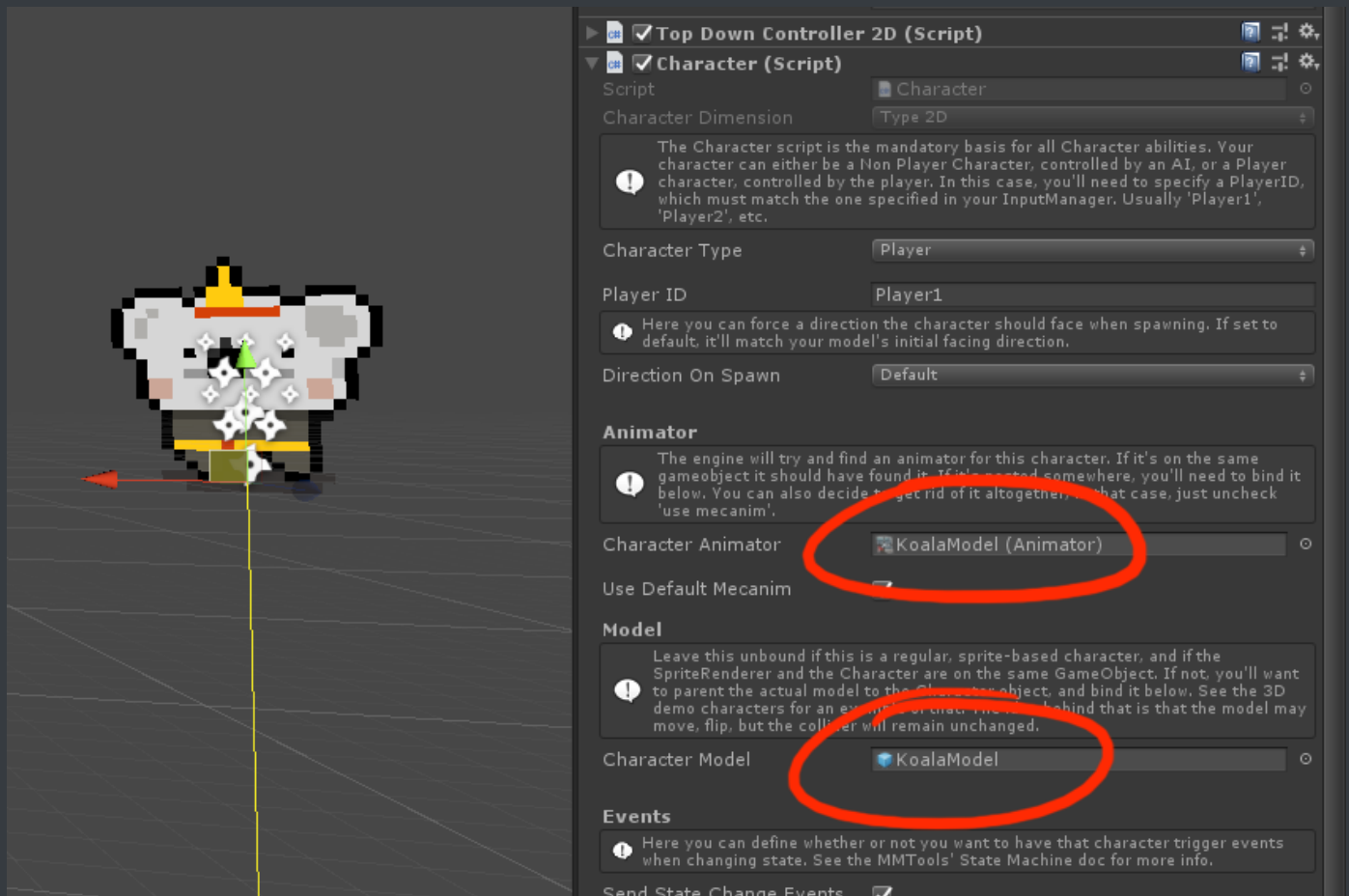
- **A Collider** : BoxCollider2D or CircleCollider2D in 2D, CharacterController in 3D, this collider's size is used to determine collisions and where in the world the agent is.
- **RigidBody2D or 3D** : only used to provide basic interactions with standard physics (completely optional)
- **TopDownController** : responsible for collision detection, basic movement (move left / right), gravity, the TopDownController is your character's motor class. It comes in both 2D and 3D versions, but both share the same methods and logic. Note that the 3D version requires a CharacterController component.
- **Character** : This is the central class that will link all the others. It doesn't do much in itself, but really acts as a central point. That's where you define if the player is an AI or player-controlled, where its model and animator are, stuff like that. It's also the class that will control all character abilities at runtime.
- **Health** : Not mandatory, but in most games your agents will be able to die. The Health component handles damage, health gain/loss, and eventually death.
- **Character Abilities** : So far all the previous components offer lots of possibilities, but don't really "do" anything visible. That's what Character Abilities are for. The asset comes packed with more than 15 abilities, from simple ones like CharacterMovement to more complex ones like weapon handling. They're all optional, and you can pick whichever you want. You can of course also easily create your own abilities to build your very own game.

## Hierarchy



The recommended hierarchy for Characters

However you create your character, one thing that is really important is to understand how to structure it. You'll want to **separate** the **logic** (the Rigidbody, TopDownController, Character abilities, etc) from the **visuals** (your model / sprite / Spine setup, etc). You can have many more layers (maybe your animator sits on its own layer, etc), but the recommended setup is the one on the image above : a top level with all the logic, and the model nested inside it.

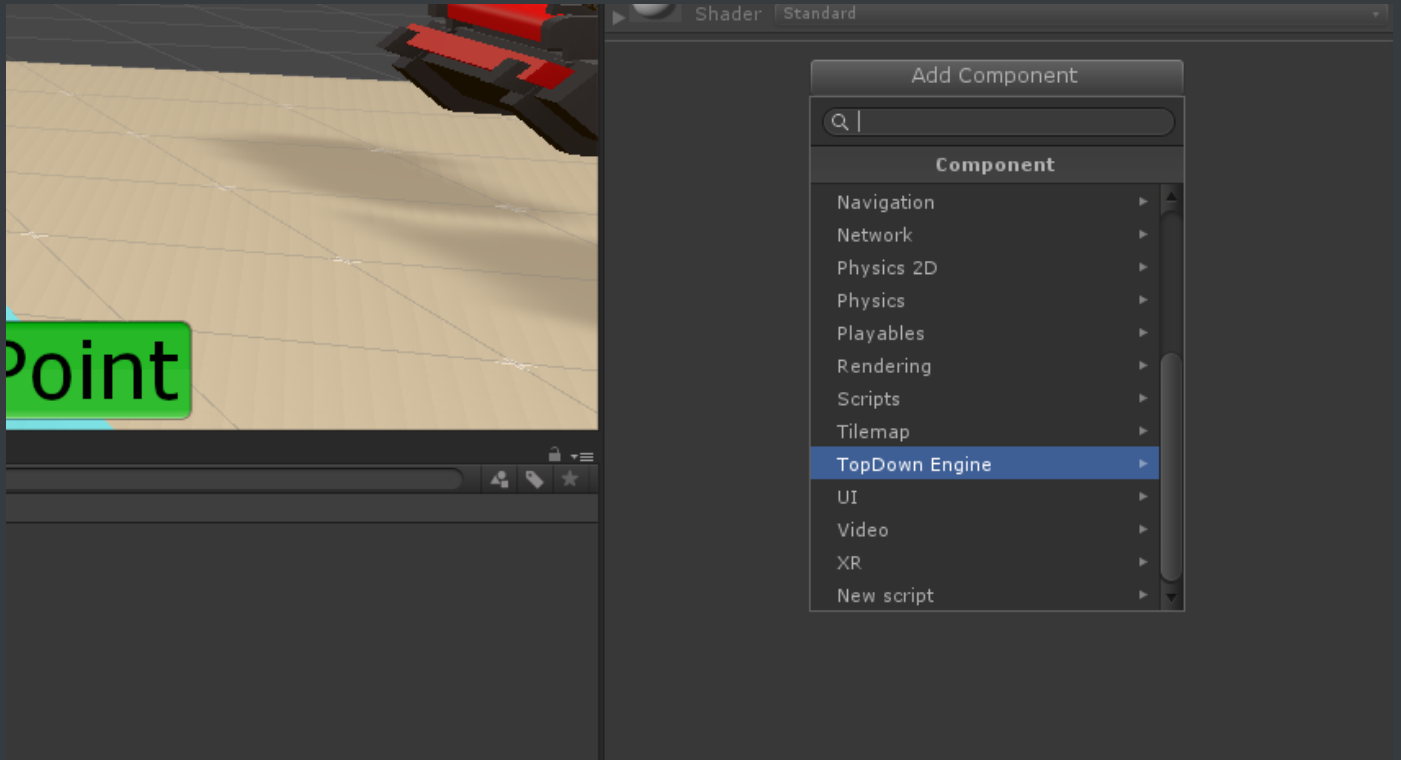


Letting the Character component know where the Animator and Model are

Make sure you link the Model to the Character class, same for the animator, from the inspector. Some other classes (Orientation abilities, Health) may also require you to tell them, via their inspector, where the Model or Animator is.

## How do I create an Agent ?

There are **many ways** you can create a playable or AI character in the TopDown Engine. Here we'll cover **the 3 recommended ones**. Note that if you prefer doing differently, as long as it works for you, it's all fine.

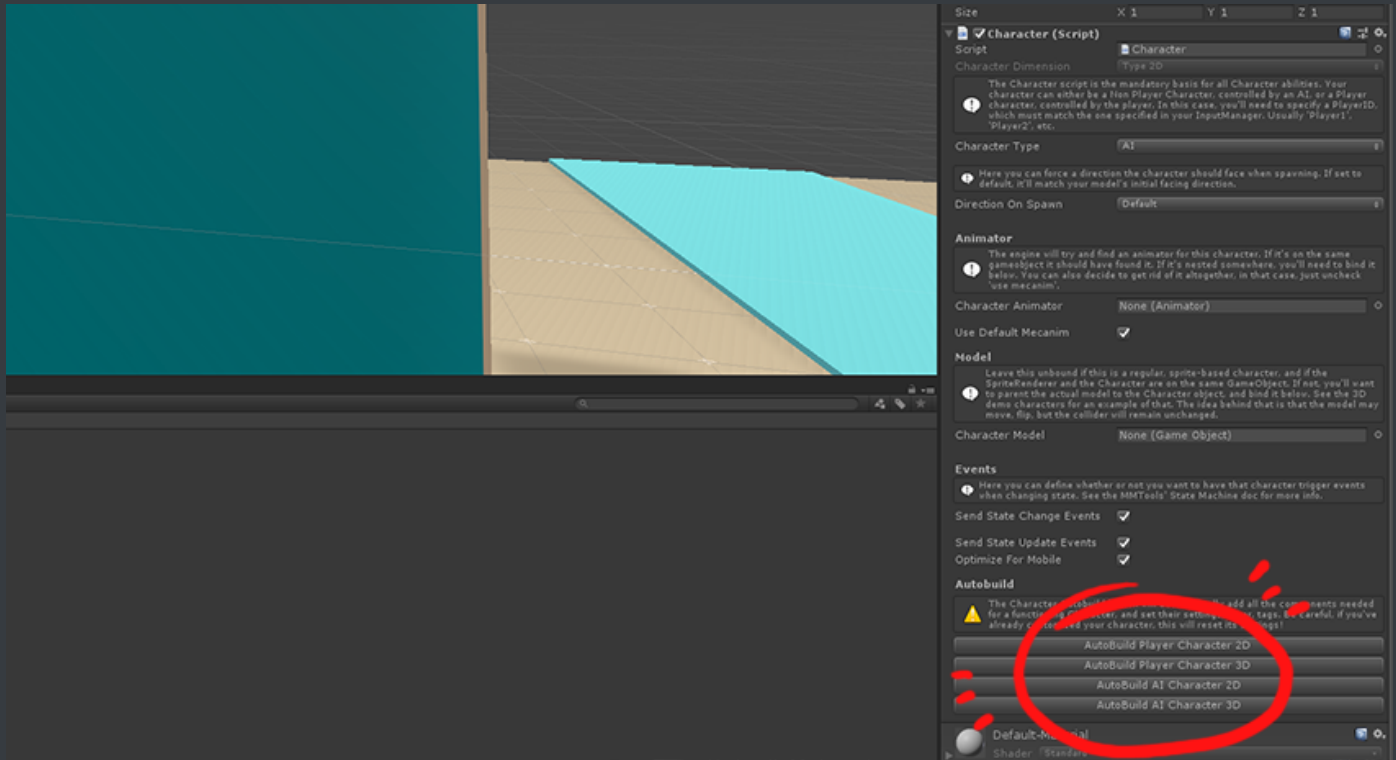


You can use the AddComponent menu to add most TopDown Engine's components (not just Character ones)

## Automatic Creation

The "**Autobuild Character**" feature allows you to create a new character in a few seconds. Note that after that initial setup you'll still have to setup animations and all that.

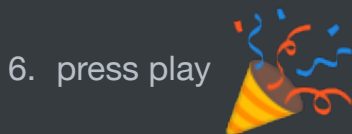
Here's how to proceed :



The Autobuild buttons

### For 3D :

1. open the MinimalScene3D demo scene (or any scene that meets the minimal requirements)
2. create an *empty* game object, name it "Test", position at 0,0.5,0
3. under it, add a Cube, center it, scale it to 1,2,1, remove its BoxCollider
4. on Test, add a **Character** comp, drag the Cube in its *CharacterModel* slot
5. on the Character inspector, press **Autobuild Player Character 3D**



### For 2D :

1. open the Minimal2D demo scene (or any scene that meets the minimal requirements)
2. create an *empty* game object, name it "Test", position it in the scene
3. under it, add a game object, center it, name it "Model", add a sprite renderer to it and set its sprite
4. on Test, add a **Character** comp, drag the sprite in its *CharacterModel* slot
5. on the Character inspector, press **Autobuild Player Character 2D**



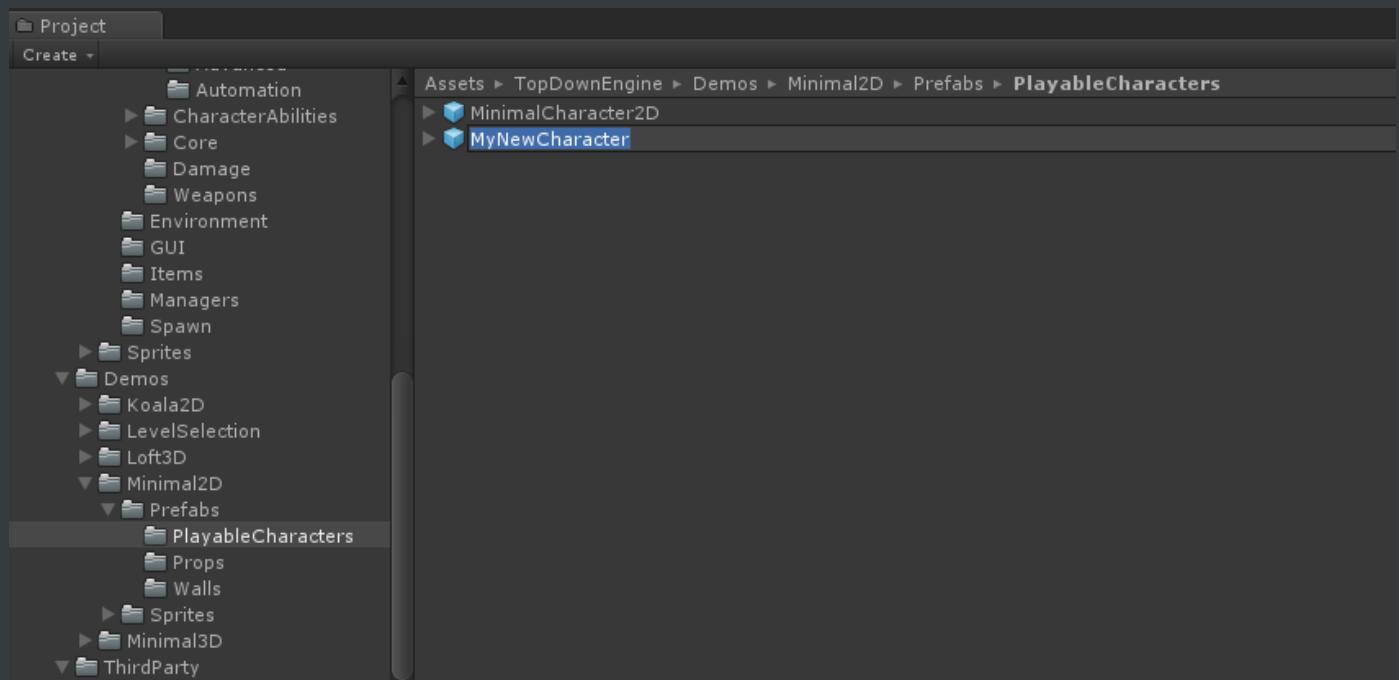
## 6. press play



Of course it'd be the same thing for AIs, except you'd pick Autobuild AI Character 2D/3D. If you went for an AI character, it's ready to get a Brain and some Actions and Decisions. You can now fine tune the various settings, **remove the abilities you're not interested in** for this character, add animations, etc. Or you can leave it like that and **start prototyping** the rest of your game and levels.

## Copy

Another fast way to create an agent is to find one you like in the demos, and **create yours from that**. The process for that is quite simple :



### Copying an existing prefab

1. **Find an agent you like** in one of the demos.
2. **Locate its prefab** (select the Agent in Scene View, and in its inspector, at the very top right under the prefab's name and tag there's a Select button)
3. **Duplicate the prefab** (cmd + D)
4. **Rename it** to whatever you want your Character to be called
5. **Make the changes you want.** Maybe you'll just want to replace some settings, maybe you'll want to change the sprite and animations. It's up to you from there.



## Components approach

You can also **create a Character from scratch**. It's longer but why not?

1. Start with an **empty gameobject**. Ideally you'll want to separate the Character part from the visual part. The best possible hierarchy has the TopDownController/Collider/Character/Abilities on the top level, and then nests the visual parts (sprite, model, etc).
2. At the top of the inspector, **set the tag to Player** if it's a player character, or to anything you prefer if it's not. Same thing for the layer.
3. On your top level object, add a **Collider** (either 2D or 3D, we recommend a Capsule for 3D characters, a box for 2D ones). Adjust its size to match your sprite/model dimensions. Then add a Rigidbody2D or Rigidbody component.
4. Add a **TopDownController** (2D or 3D) component. Set the various settings (see the class documentation if you need help with that), and make sure to set the various collision masks (platforms, one ways, etc). Add a **CharacterController** component too if you're going for 3D.
5. Add a **Character** component. Check the various settings to make sure they're ok with you.
6. Add the **Character Abilities** you want (it's best to use the AddComponent button at the bottom of the inspector for that, and navigate there)
7. Optionnally, add a **Health** component, a **HealthBar** component, etc.