SCG

SMART CHARACTER GENERATOR 2D

Manual

Created by Phun Peeticharoenthum

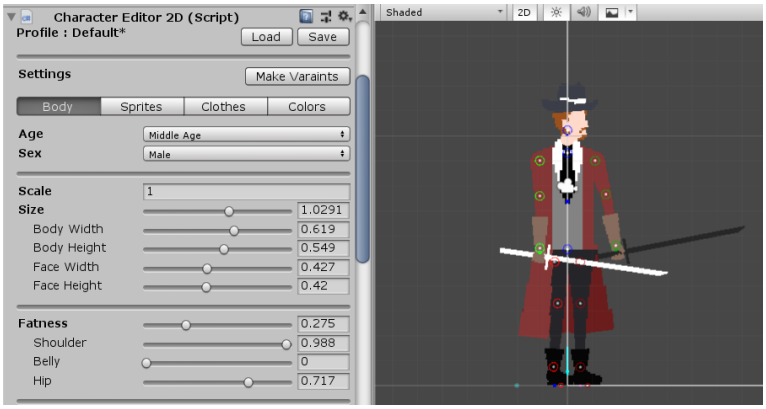
Table of Contents

Introduction

Smart Character Generator 2D (SCG) is a character creation system for Unity that helps you accelerate in your 2D game development by providing you tools to create 2D characters that are highly customizable and easy to animate. It also gives you the power of editing, saving, loading, and even randomly generating characters both in editor and runtime!

Features

1. **Deep Customization:** All parts of the character body can be edited, saved, loaded, and randomly generated both in editor and runtime.



   1.1. **Quick Editing:** Edit your character instantly from the inspector.
      - Quickly alter the body shape to make skinny, tall, or muscular 2D characters.
      - Includes the control of face, hairstyle, breasts, and belly.
      - Change Age & Gender of the character.
      - Fully customize the clothes and colors of the character. The supported clothing options are shirt (or naked), inner shirt/armor, vest, sleeves, inner sleeves, coat, pants, skirt, gloves, long gloves, headwears, eyewear/mask, shoes, and boots.
      - In-hand items customization is also included with controllable finger poses.
      - Built-in Color Palettes help you pick colors faster.

   1.2. **Deep Editing:** Wants to turn all the knobs? You can go deeper and change the sprite, size, and color of every single part via the inspector or code.

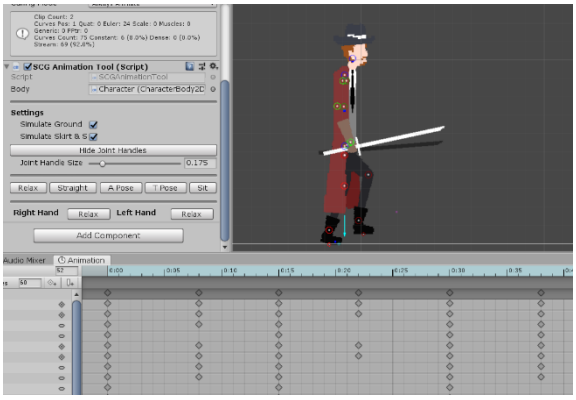
2. Animation Tools

2.1. **Perspective Simulation:** Easily change the angle of the character's perspective to give more depth to your animations.

2.2. **Built-in Joint system:** Animate faster with the assistance of built-in handles, pose shortcuts, rotation limiter, toe joint, foot IK, and finger posing

2.3. **Ground Simulation:** Position the character on the ground level automatically, with this feature, you can create walking animation clips that contain only rotational changes then let Ground Simulation handles all vertical translations of each character. So that walking clip will work on every character even though they have different heights.

2.4. **Skirt & Shirttail Simulation:** Simulate the folding of skirt & shirttail based on leg rotations.



3. **Save System** Save & Load Character Profile using binary files, works in the editor, and build.

4. **Random Character Generator** Generate random characters, you can also create patterns and rules to generate the specific kind of character with **Variants system.**

Getting Started

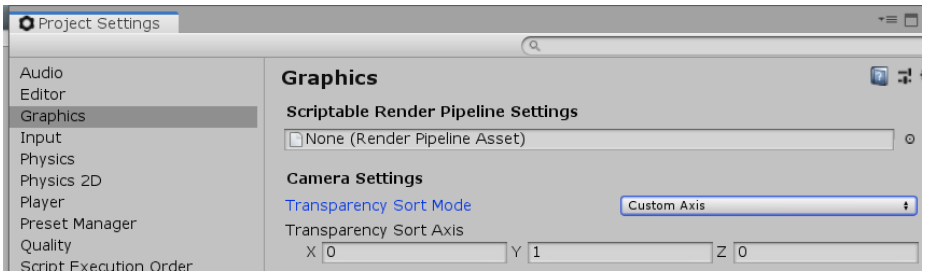
Smart Character Generator 2D asset requires at least a 2018.4.23f1 (LTS) version of Unity. Here are some guidelines that you should know before start working with the asset.

1. Guidelines

1) Sorting Characters by the Y value

Each character in the scene is sorted by the Sorting Group component. With Transparency Sort Mode set to Custom Axis, the character with a lower Y position will get drawn in front of characters with a higher Y position. For this to work, however, you have to change the **Transparency Sort Mode** in **Camera Settings.** The first time you imported Smart Character Generator 2D, there will be a dialog box that asks if you want to do that automatically. You can change the setting manually at **Edit > Project Settings... > Graphics**:

- **Transparency Sort Mode**: Custom Axis
- **Transparency Sort Axis:** 0, 1, 0



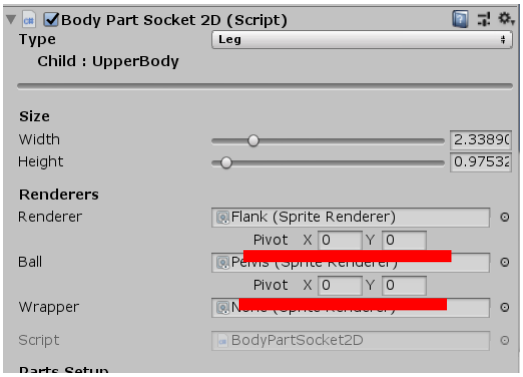
2) Resources Folder

- When saving or loading character appearance, make sure all sprites on every body part are located inside **Assets > Resources**. This makes saving and loading sprites work in both Editor and Build.
- Character profiles that are saved within Editor are also stored inside Resources as binary files.
- For character profiles that are saved in build, the binary files will be created and stored at **Application.dataPath** instead.

3) Handling Sprites

If you want to use your sprite for the character, please follow these guidelines:

- Make sure that the pivot of all the sprites used on the character is **at the center** of the sprites**.** If you want to modify the pivot, you can adjust them later through the **Body Part 2D** Inspector window.



- Sprites that are used for arm, leg, and body should not have empty spaces around them.
- Sprites that are used for face, hair, eyewear, and headwear should have the same size and snap together perfectly when putting them on top of each other.
- If you are using Photoshop, you can start by drawing a face sprite and position it at the center of the image, then leaving space around it for drawing hair, eyewear, and headwear on the separate layers. For importing into Unity, I recommend that you export each layer as a separate file. In Photoshop, go to **File > Export > Layer to Files...** instead of using Unity's PSD Importer. With this method, the space around your drawings won't get cut out.
- To save and load sprites on the characters properly, please avoid using the duplicated name on sprites even if they are stored at different locations.
- If you rename or move the sprites, all existing character save files might not be able to load the sprites since the path to those sprites has changed. You might have to manually load and save them again after modifying the name/location of the sprites. If loading doesn't work, wait for a minute, refresh the project window, and try again.

2. Example Scene

You can try out all the features of Smart Character Generator 2D through the example scene, located in **Assets > SCG > Scenes > Example.unity**. Simply open the scene and hit Play.

1) In runtime, you can modify the gender, hairstyle, color, and body shape of the character, using UI on the screen. Or just select the **Character** GameObject and fully customize it through **Character Editor 2D** in the Inspector window which works in both edit mode and runtime.
2) **ExampleModifier.cs** is an example script that receives your modification through UIs and modifies the **CharacterBody2D** component accordingly. You can learn to modify your character through code by taking a look at the useful methods **of CharacterBody2D** class demonstrated in **ExampleModifier.cs**
3) At the bottom right corner, you can try out:
    - **Perspective Simulation** by toggling **Override Perspective** and use the slider to adjust the angle of the character.
    - Triggering Walk/Idle Animation.
    - **Saving, Loading**, and **Randomly** Generating the character.
4) **Character Generator 2D** is demonstrated by continually generating **Character AIs** that attached with **ExampleWalkAI.cs.** These characters will later get recycled back to the **Character Pool** and get reactivated again according to the concept of **Object Pooling**. In this example scene, the Character Pool is already generated before entering play mode to help save resources and performance**.** But you can change the settings of Character Generator 2D from Inspector window to instantiate characters at runtime as well if you like.

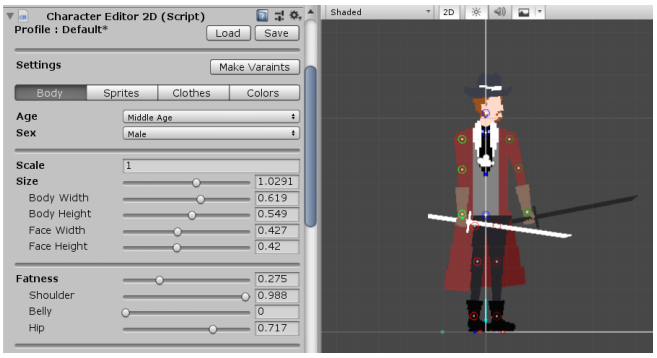
Main Components

There are 3 main components on the **Character prefab**:

- **Character Editor 2D** for customizing the character.
- **SCG Animation Tool** located at the child transform, used for configuring Built-in Handles and simulation settings. It also has shortcut buttons for body poses.
- **Character Body 2D** handles all body parts of the character and keep them intact.

Character Customization

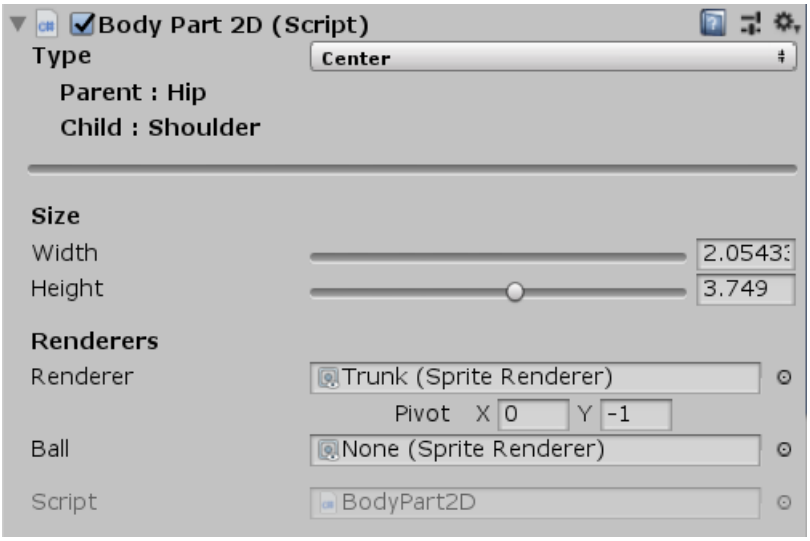
1) Character Editor 2D Inspector window



Character Editor 2D can be attached to Character Body 2D to fully customize the character through the Inspector window.

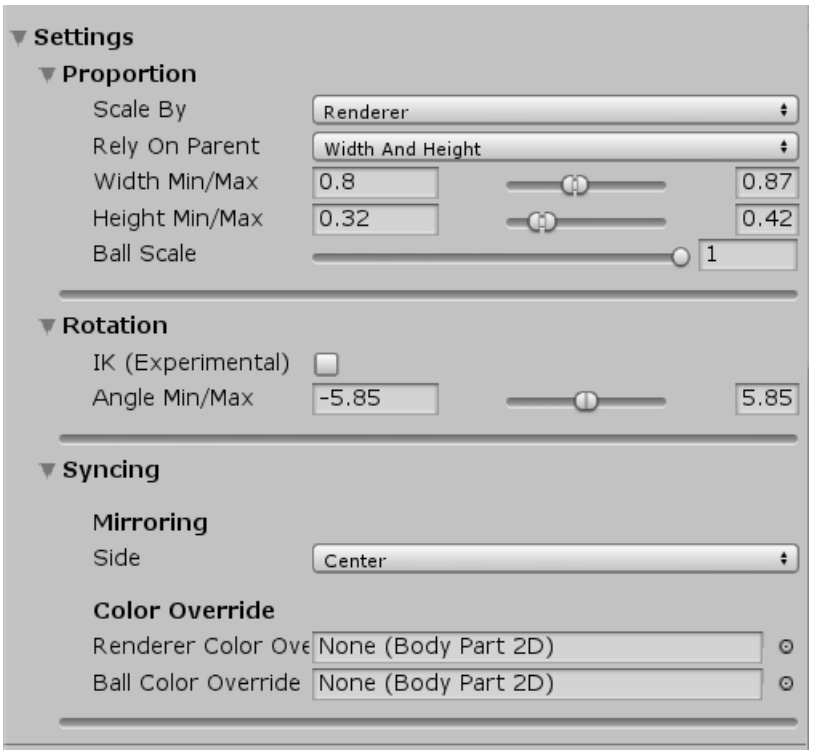
- **Save & Load Section:** You can save and load character as a 'profile' while in Editor using the Save & Load button.
- **Body Tab:** Allows you to modify gender, age, scale, size, and overall body shape. Press **Enable Quick Edit** to quickly alter the body shape as you wish.
- **Sprite Tab:** Allows you to assign sprites for face, headwear, hair, eyewear, mustache, body, and in-hand items.
- **Clothes Tab:** Allows you to design the character's clothes. Supported clothing options include shirt (or naked), inner shirt/armor, vest, sleeves, inner sleeves, coat, pants, skirt, gloves, long gloves, shoes, and boots.
- **Colors Tab:** Allows you to pick colors for each part of the character. You can use **Color Palette** to save favorite colors for skin, hair, and clothes. All palettes can be modified via **Window > SCG > Color Palette.**

2) Body Part 2D Inspector window



- if you want to edit sizes of each body parts directly, you can do so by selecting on the GameObject that has **Body Part 2D, Body Part Socket 2D, Hand Part 2D,** or **Foot Part 2D** attached on it to modify their size separately, note that Quick Edit Mode on the Body Tab will be disabled and will no longer control overall body shape if you start modifying body parts directly.
- Parent part and child part of a Body Part 2D is displayed at the top of the Body Part 2D Inspector window. It also tells you if the part has a buddy part.
- Each body part contains up to 3 renderers: main renderer, ball renderer, and wrapper renderer for elbows and knees. Except for the head part that also has additional renderers for hair, headwear, eyewear, and mustache.
- For **Hand Part 2D,** you can modify finger posing via the Hand Part 2D Inspector window.
- For **Foot Part 2D**, there is an additional toe renderer that separates the foot into 2 parts allowing tiptoe posing.

- On each body part, you can also modify their settings through the Inspector window. The settings include:
    - **Proportion:** Decide if the size of that body part will be controlled by its parent part or not and how. You can also limit the size of that body part here.
    - **Rotation:** Limit the rotation of that body part while working in Edit Mode.
    - **Syncing:** Decide if that body part should sync its appearance with other body parts or not. This is usually for syncing the appearance of arms and legs. In the Color Override section, you can decide if the main renderer and ball renderer should inherit their color from other body parts or not.



3) Modifying character through code

If you want to modify the settings that appear on Character Editor 2D Inspector window through code, you can do so by accessing Character Body 2D directly**.** Here are useful functions that are available:

CharacterBody2D	
Save & Load Character	
bool Load (string path)	Load character profile from the path provided. Then return the result.
Save (string path)	Save character profile to the path provided.
Character Generation	
CharacterBody2D Generate (SCGVariants variants)	Randomly generate this CharacterBody2D based on given SCGVariants. Then return the result.
static SCGVariants CreateVariants (CharacterBody2D body)	Create SCGVariants based on the CharacterBody2D provided.
SCGVariants CreateVaraints ()	Create SCGVariants based on this CharacterBody2D. Then return the result.
Recycle (bool changeAppearance)	Recycle and send this CharacterBody2D back to the Character Pool if it exists.
Rendering	
Repaint ()	Force this CharacterBody2D to repaint itself.
Sort ()	Force this CharacterBody2D to sort itself based on Sorting Orders.
ChangeOrder (SpriteRenderer target, int i)	Manually change the sorting order of target SpriteRenderer to the given order.
Overall Shape	
SetAge (Age age)	Modify the age of the character.
SetSex (Sex sex)	Modify the gender of the character.
• SetSize (float size) • SetScale (float scale)	Modify the overall body shape of the character.

• SetWidth (float width) • SetHeight (float height) • SetFaceWidth (float faceWidth) • SetFaceHeight (float faceHeight) • SetFatness (float fatness) • SetBellyHeight (float bellyHeight) • SetBreastsHeight • (float breastsHeight) • SetHipWidth (float hipWidth) • SetShoulderHeight (float shoulderHeight)	
Clothes & Appearance	
• SetGloveColor (Color gloveColor) • SetDifferentSleeve (bool differentSleeve) • SetEyewearColor (Color eyewearColor) • SetEyewearStyle (Sprite eyewearStyle) • SetFemaleBody (Sprite femaleBody) • SetFemaleFace (Sprite femaleFace) • SetMaleBody (Sprite maleBody) • SetMaleFace (Sprite maleFace) • SetGrayHairColor (Color grayHairColor) • SetHairColor (Color hairColor) • SetHairStyle (Sprite hairStyle) • SetHatColor (Color hatColor) • SetHatStyle (Sprite hatStyle)	Modify the clothes and colors of the character.

• SetInnerBodyColor (Color innerBodyColor) • SetInnerBodyStyle (Sprite innerBodyStyle) • SetInnerSleeve (bool innerSleeve) • SetInnerSleeveColor • (Color innerSleeveColor) • SetLeftInner (Sprite leftInner) • SetLeftInnerColor (Color leftInnerColor) • SetMustageStyle (Sprite mustageStyle) SetPantColor (Color pantColor) • SetRightInner (Sprite rightInner) • SetRightInnerColor (Color rightInnerColor) • SetShirtColor (Color shirtColor) • SetShoesColor (Color shoesColor) • SetShoesStyle (Sprite shoesStyle) • SetSkinColor (Color skinColor) • SetSleeveColor (Color sleeveColor) • SetToeStyle (Sprite toeStyle) • SetWearCoat (bool wearCoat) • SetWearShirt (bool wearShirt) • SetWearSkirt (bool wearSkirt) • SetArmCover (int armCover) • SetGloves (int gloves) • SetGlovesWrap (float glovesWrap) • SetLegCover (int legCover) • SetShoesWrap (float shoesWrap)	

Character Posing	
• APose () • TPose () • Sit () • Relax () • StraightenBody () • SetTurn (int i) • SwitchTurn ()	Change the character's pose.

Every body parts are also can be modified through code directly. Here are useful functions that are available:

BodyPart2D	
BodyPart2D Appearance	
• SetHeight (float height) • SetWidth (float width) • SetColor (Color color) • SetPivot (Vector2 pivot) • SetRendererSprite (Sprite sprite) • SetBallPivot (Vector2 pivot) • SetBallSprite (Sprite sprite)	Modify the appearance of the BodyPart2D
BodyPart2D Posing	
Straighten ()	Straighten the BodyPart2D and child parts.
StraightenSelf ()	Straighten the BodyPart2D only.
• SetSide (int i) • SwitchSide ()	Set the local scale of the BodyPart2D.

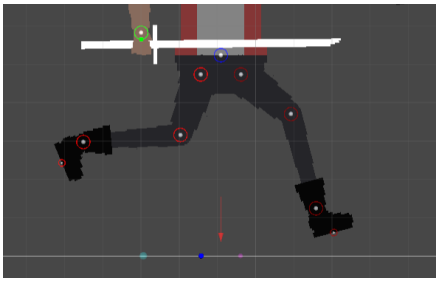
HandPart2D	
HandPart2D Appearance	
• SetColor (Color glovesColor, Color skinColor, int glovesWrap) • SetItemOffset (Vector2 itemOffset) • SetItemSize (Vector2 itemSize)	Modify the appearance of the HandPart2D
HandPart2D Posing	
• Hold () • Straight () • SetPose (int i, int pose)	Change hand and finger posing.

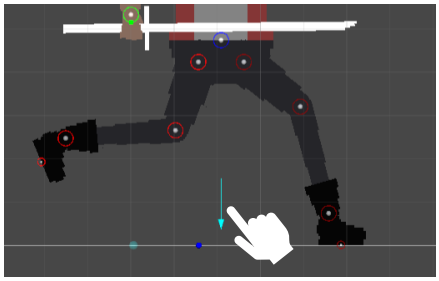
Animation Tools

1) Built-in Joint system

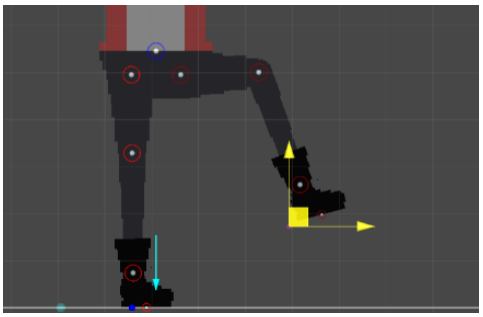
When selecting Character Body 2D, SCG Animation Tool, or any body parts. The built-in handles will be displayed in Scene View.

- Quickly rotate any body parts using **Rotation Handles**. You can click on the white dot at the center to reset the rotation.
- Switch head, hand, and body side using **Switch Side Buttons.**
- Click on **Ground Simulation Button** to enable/disable Ground Simulation or modify **CharacterBody2D. simulateGround** through code.

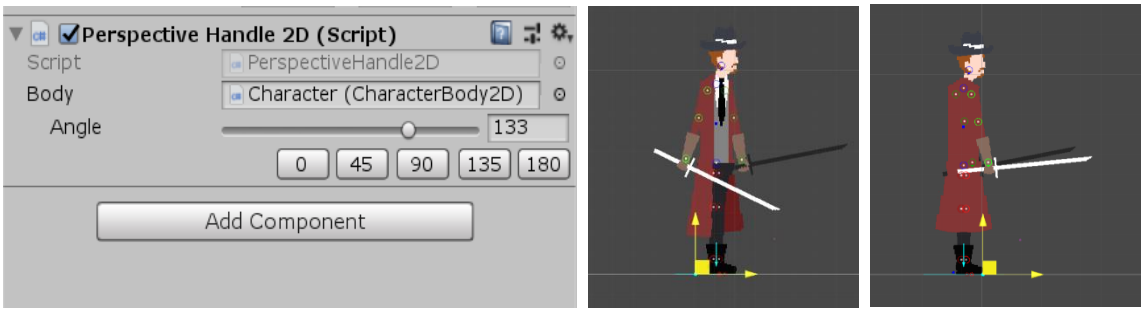




- Click on **Foot IK Shortcut** to use **Foot IK Handle.**



- You can adjust the body angle of the character using **Perspective Handle** inside the **Character prefab** or click on **Perspective Handle Shortcut.**



If you want to control the perspective value through code, here are some options:

- Call **SetValue (float value)** on **PerspectiveHandle.**
- Call **SetPerspective (float perspective)** on **CharacterBody2D.** note that this will override perspective changes on the animation clips.
- Call **AutoPerspective()** on **CharacterBody2D** to reset the value from SetValue() method and let the animation clips take control of the perspective changes.
- Set **simulatePerspective** on CharacterBody2D to enable/disable the Perspective Simulation on each character.

2) **SCG Animation Tool Inspector window**

- Access the SCG Animation Tool Inspector window to configuring Built-in Handles and simulation settings.
- You can turn off ground simulation on each character if you want.
- It also has shortcut buttons for body poses.

3) **Animation Tips**

- When animating a character, you can simply hit Record on the Animator and use **Rotation Handles** and **Foot IK Handle** to create new poses for the animation clip.
- **Ground Simulation** will position the character on the ground level automatically, with this feature, you can create walking animation clips that contain only rotational changes then let Ground Simulation handles all vertical translations of each character. So that walking clip will work on every character even though they have different heights.
- When rotating leg parts, Ground Simulation will be disabled automatically to prevent it from fighting over your adjustment.
- You can animate **Perspective Handle** to record perspective changes.
- Note that when moving legs using Foot IK Handle, the Animator **won't be recording the rotational changes.** You might have to tweak Rotation Handles a little bit to make the Animator record the changes.
- You can animate Foot IK Handle by recording its position changes. But this will let Foot IK Handle take control of the whole leg and **override its rotational changes.** So, if you want to record the rotational changes of the upper leg, knee, and foot separately, you should remove all records of the Foot IK Handle position.

Character Generator

Character Generator 2D Inspector window

With **Character Generator 2D** component, you can easily generate random characters based on **SCGVariants** which is like a blueprint for creating characters.

- First, choose the **Body Blueprint** that you want to Instantiate and put it in the Setup section.
- Create at least one **SCGVariants** to use for the randomization, its role is to limit the **Possibilities** of character generation based on what your configuration.
- You can create SCGVariants from scratch or generate it from the Character Editor 2D component, simply hit Make Variants on the character that you want to use as a prototype. To generate SCGVariants from CharacterBody2D through code, you can call **Load (CharacterBody2D body)** on **SCGVariants**.
- You can press Generate Button to instantly generate a random character in Edit Mode.
- If you put more than one SCGVariants in the Character Generator 2D component, it will generate the character based on randomly picked SCGVariants from the list.
- **Auto Spawning** lets you continually generate characters at runtime.
- **Spawn Pooling** lets you prepare the **Character Pool** to use in Auto Spawning to save resources and performance at runtime.
- To Recycle spawned characters from the pool through code, call **Recycle ()** method on target **CharacterBody2D.**

Character Generator 2D Methods

CharacterGenerator2D	
Character Pool Modification	
• AddToPool () • ClearPool () • CreatePool ()	Modify the Character Pool.
Character Generation	
• CharacterBody2D Generate () • CharacterBody2D Generate (CharacterBody2D target) • static CharacterBody2D Generate (SCGVariants blueprint, CharacterBody2D bodyBlueprint) • static CharacterBody2D Generate (SCGVariants blueprint, CharacterBody2D bodyBlueprint, Vector3 position) • CharacterBody2D Generate (Vector3 position)	Methods for randomly generating characters.

Support Contact

If you encounter bugs or have any suggestions on how to improve Smart Character Generator 2D you can submit them at:

Issue Tracker: https://github.com/phanphantz/SmartCharacterGenerator2D

Or feel free to contact me directly at:

E-mail: phun.peeticharoenthum@gmail.com

Thanks,

Phun Peeticharoenthum