Overview (PLEASE READ THIS FIRST!)

If you're reading this, you're probably fixing to add stuff to the BleedingOut skeleton, made by Michelle Lehto. This guide will show you the proper, intended way to add weapons, characters and maps to the game without breaking anything!

The BleedingOut template was designed to be a more user-friendly experience than the official FPS template, while also offering the flexibility needed to properly expand the game. There are a lot of little steps that need to be followed for each item you want to add, but following them precisely will ensure that development goes smoothly, and keeps the project's back-end nicely organized to make future development just as smooth later on.

Before we get to the instructions, it's important to understand a few key points about the overall skeletal structure, so let me quickly go over those first:

- The game is built on a "controller-centric" pattern, which means that each player entity has full autonomy when it comes to processing user input and data. Everything about a player - including their HP, current weapon, KDR, etc. - is handled by the player object, and necessary points are merely tracked and reported by the greater world entity.
- The game's default scene is the "_init" scene, which functions as the launch pad for the rest of the experience. It's mostly used for debugging. If you go into that scene and check its root's custom properties, you'll notice a few debug options:
    - "Skip Menu" (which loads the game straight away on boot instead of showing the opening splash and main menu)
    - "Map Name" (which sets the debug map that loads up when you boot while skipping the menu)
    - "Player Count" (sets the number of players in the map) (WARNING: The game is NOT set up to support more than two players! This is outside the scope of this skeleton, and will need to be added later!)
- Player characters have two different sets of visuals - their body rig, and their arms rigs (named as such in the code). Due to the nature of the provided assets, I've had to make some concessions in the way I have you set up future versions, which is partly why this guide is so detailed and full of scary "don't touch that"s.
- The arms rigs are TINY! I scale them down to hide them inside the player's shape rig, so that they don't clip into walls and whatnot in the environment, but due to Movie Magic™, they appear just the right size in the camera view!

Adding Weapons

What you'll need:

- Weapon model
- Arms rig for that weapon
    - MUST have the following 12 animations, NAMED AS SUCH:
        - idle * SET TO LOOPING
        - ads * SET TO LOOPING
        - moving_idle * SET TO LOOPING
        - moving_ads * SET TO LOOPING
        - fire_idle
        - fire_ads
        - sprint * SET TO LOOPING
        - holster
        - draw
        - reload
        - melee
        - inspect
- A fully rigged weapon model for the body rig, with a matching skin
- An icon for the gun's bullets

Instructions:

- Place the arms file in the project, in "assets/rigs"
- Right-click it and click "New Inherited Scene"
- Duplicate the AnimationPlayer in the new scene, rename the first one "anim_continue" and the second "anim_oneshot"
- Set the process priority of anim_oneshot to 1 (it's under Node > Process in the inspector)
- Save the inherited scene in res://assets/rigs, name it "arms-<NAME_OF_GUN>.tscn" (the arms for the AK74u would be called "arms-AK74u.tscn", for example. Case sensitive.)
- Delete the original arms file (or store it outside the project somewhere, we don't need it anymore)
- In the player scene, add the new arms rig as a child of the camera's "arms" node
- Set the scale of the arms rig to 0.2, and position them in the camera to best fit all their animations (this may take some trial and error)
- Set the new rig to "hidden" in the player scene

==== ==== ==== ====

- In the inspector, create a new Resource. Create a "Weapon".
- Name it EXACTLY the same as the arms rig, without the "arms-" part (the weapon resource for the "arms-AK74u" would be named "AK74u", case-sensitive)
- Drag and drop your desired bullet TSCN into the "Bullet Tscn" field (more info on making bullets is below, but there are a few default ones)

- Set the cooldown as desired, in seconds (make it a tiny bit longer than the weapon’s firing animation for the best effect)
- Set the reload time to the length of the arms’ reload animation, in seconds
- Drag and drop your desired bullet icon from the “textures/bullet_icons” folder into the “Ammo Icon” field
- Save the resource to res://assets/weapons with the EXACT name you gave it

==== ==== ==== ====

- To give the weapon to the player:
    - Go into player.tscn
    - Select the root node, go into the inspector, scroll down to the “Inventory” category
    - Drag and drop the weapon resource into either the “Current Weapon” or “Side Weapon” fields
    -