How to Mod Beanulator 3D

a quick and easy guide

written by Nuclear_Diamond

Installation/Getting Started

What you'll need:

- Unity 2020 (2020.3 is what the dev build is on), it's pretty easy to download, just go to https://store.unity.com/download, make an account if you have to and download Unity Personal (it's free)

- A properly installed copy of Beanulator 3D on your computer. If you don't already have Beanulator, it can be downloaded from https://nuclear-diamond.itch.io/beanulator-3d

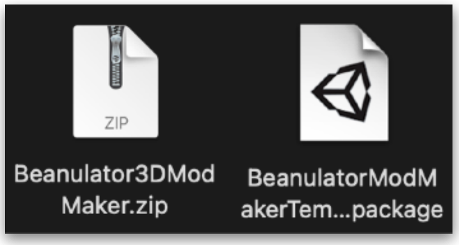
- A copy of "Beanulator3DModMaker.zip" and the .unitypackage file inside of it. It can be found at https://nuclear-diamond.itch.io/beanulator-3d

- Depending on what you're going to be doing, a 3D modeling program is recommended (like Blender, SketchUp, SketchFab... anything as long as it can output .fbx files)

Installing the modding tools to your Unity installation:

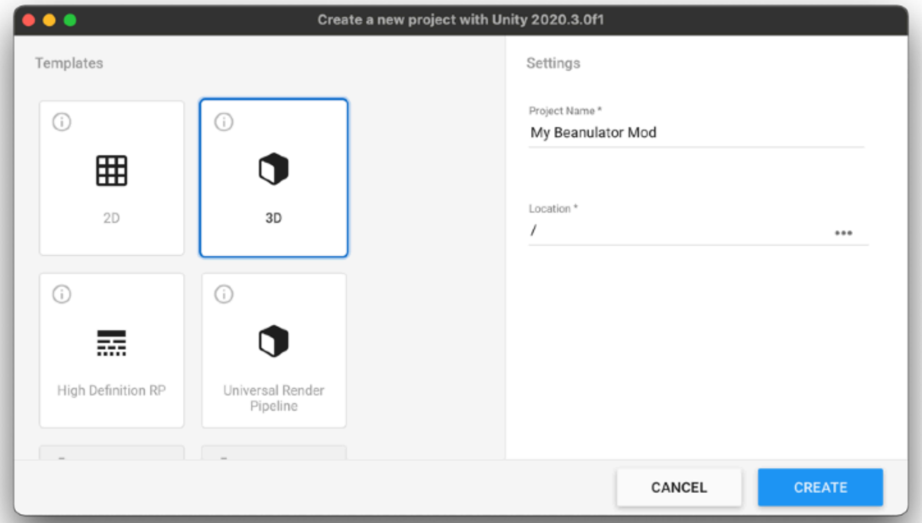
1. Download the latest version of Unity using the Unity Hub or the Unity downloads website. If you don't know how to do that, it's pretty well explained here: https://unity3d.com/get-unity/download

2. Download the file "Beanulator3DModMaker.zip" and unzip it into a folder whose location you'll remember. The file inside is a Unity Asset Package that contains the whole sample modding project.



Important note for Mac users: For some reason the default Mac unzipper unzips the Asset Package as a folder, which is wrong and means Unity can't read the file. Use a third party unzipper (I use The Unarchiver) to properly unzip it as a file.

3. Start up unity, make a new 3D project (not 2D!). The default settings for creating a new project are fine, it'll all be overwritten as soon as you import the Asset Package anyways.


Create a new project with Unity 2020.3.0f1

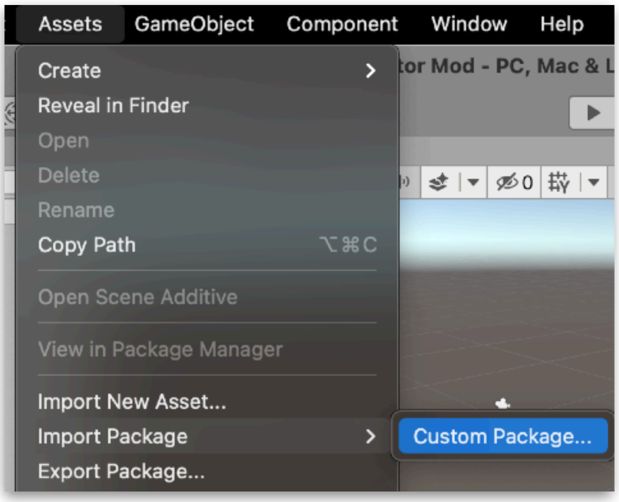
Templates

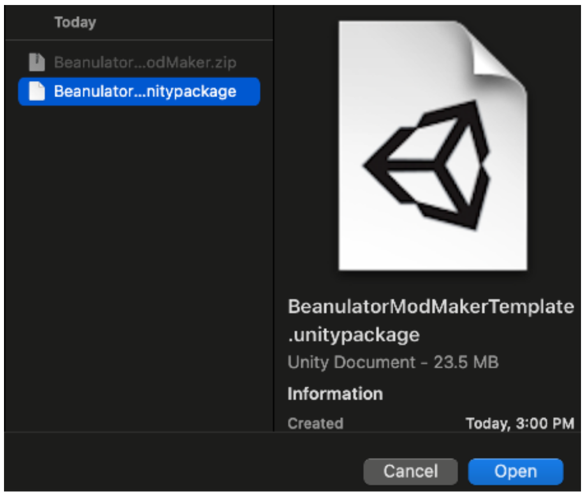
Settings

Project Name *
My Beanulator Mod

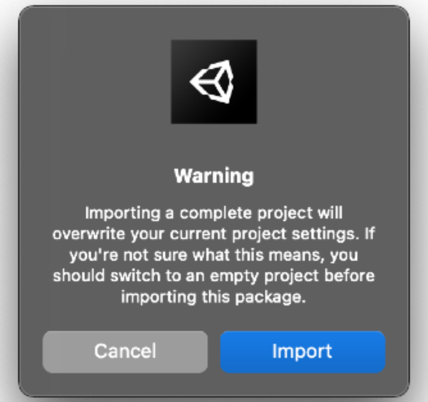
Location *
/

4. Import the Asset Package by going to Assets > Import Package > Custom Package... on your Unity Editor, once it loads in (the button should be at the top of the toolbar)



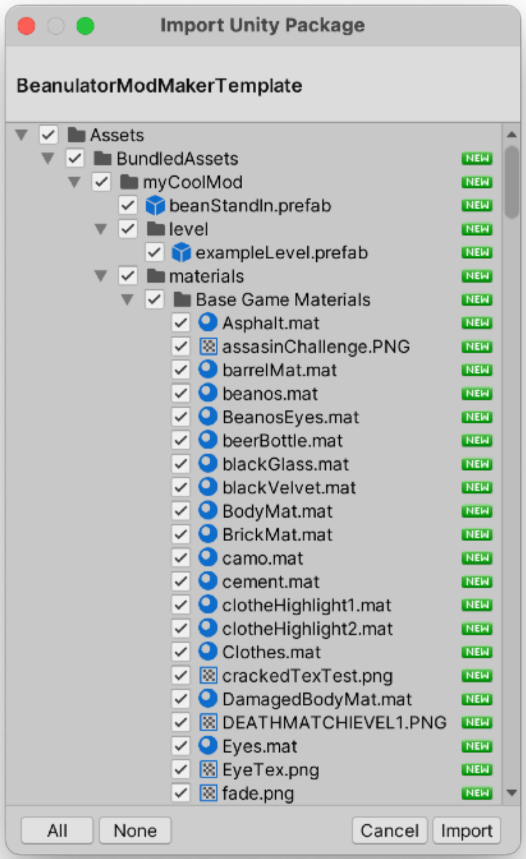


5. You'll be asked whether or not you mind overwriting your project. It should be empty, so click "Import".



Warning

Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package.


Import Unity Package

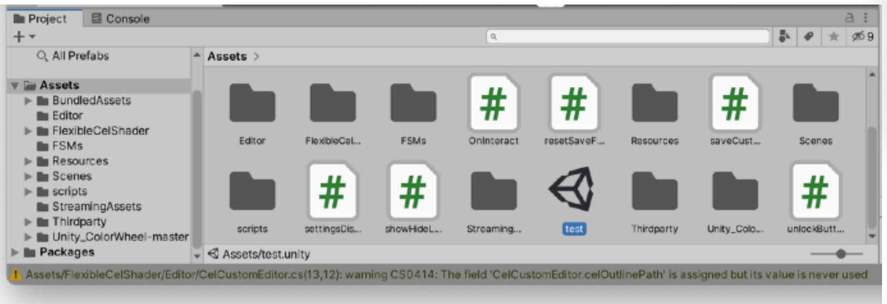
BeanulatorModMakerTemplate

- ✓ Assets NEW
  - ✓ BundledAssets NEW
    - ✓ myCoolMod NEW
      - ✓ beanStandIn.prefab NEW
      - ✓ level NEW
        - ✓ exampleLevel.prefab NEW
      - ✓ materials NEW
        - ✓ Base Game Materials NEW
          - ✓ Asphalt.mat NEW
          - ✓ assasinChallenge.PNG NEW
          - ✓ barrelMat.mat NEW
          - ✓ beanos.mat NEW
          - ✓ BeanosEyes.mat NEW
          - ✓ beerBottle.mat NEW
          - ✓ blackGlass.mat NEW
          - ✓ blackVelvet.mat NEW
          - ✓ BodyMat.mat NEW
          - ✓ BrickMat.mat NEW
          - ✓ camo.mat NEW
          - ✓ cement.mat NEW
          - ✓ clotheHighlight1.mat NEW
          - ✓ clotheHighlight2.mat NEW
          - ✓ Clothes.mat NEW
          - ✓ crackedTexTest.png NEW
          - ✓ DamagedBodyMat.mat NEW
          - ✓ DEATHMATCHIEVEL1.PNG NEW
          - ✓ Eyes.mat NEW
          - ✓ EyeTex.png NEW
          - ✓ fade.png NEW

6. It'll bring up a popup. Click Import in the bottom right corner of the window.

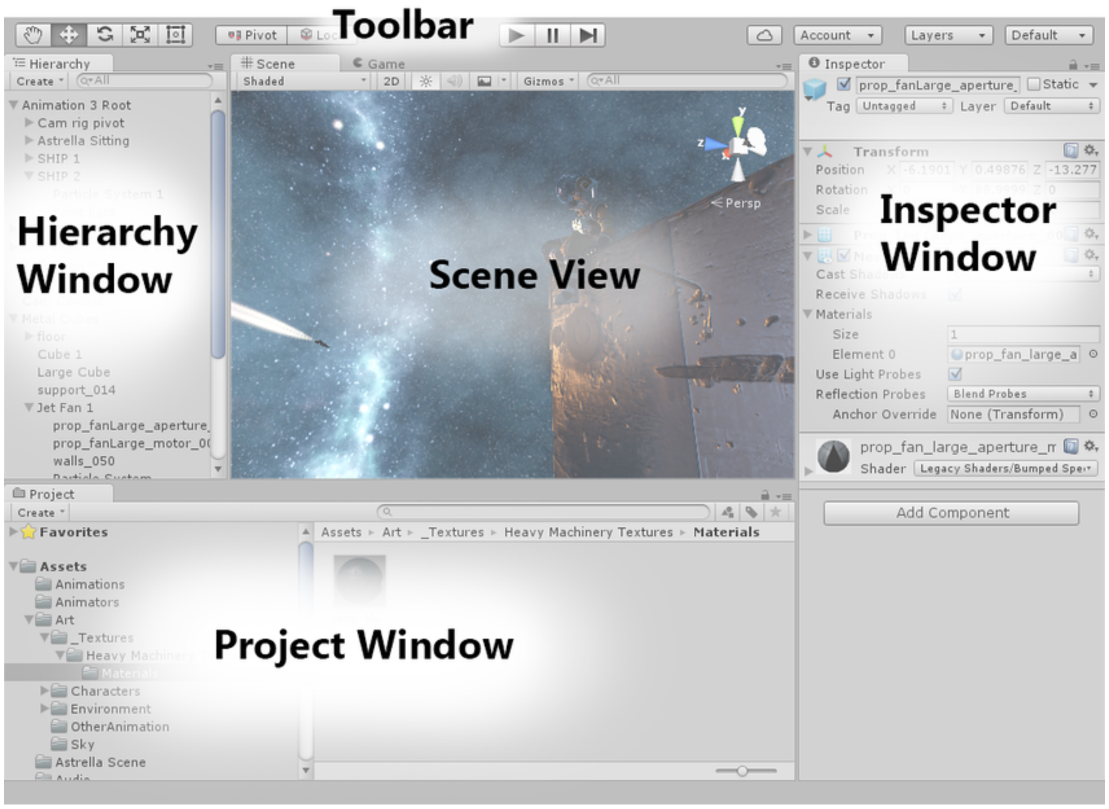
7. It should import. Ignore the warnings, some of the shaders are a little out of date and the new Unity versions really like to complain about stuff like that.

8. Find the scene called "test" by scrolling around in the window called Assets. That's where all the files you can use will be.



```
Assets/FlexibleCelShader/Editor/CelCustomEditor.cs(13,12): warning CS0414: The field 'CelCustomEditor.celOutlinePath' is assigned but its value is never used
```

9. You're done!



Making your first Mod

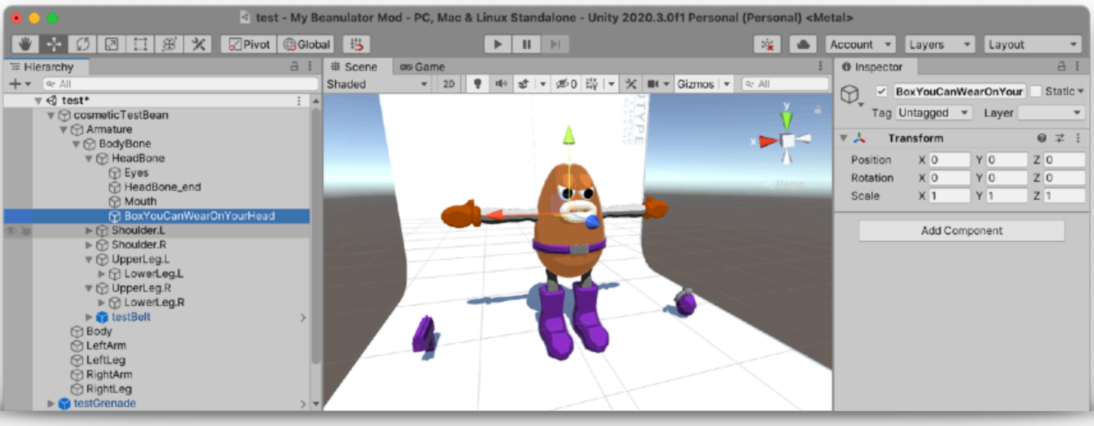
Let's make a mod! I think a quick cosmetic item based on one of the existing ones would be the easiest. Dropdown the dropdown menu on "cosmeticTestBean". Find these bones under the "Armature" dropdown menu.

Cosmetic Slot	Related Bones
Head/Hat	HeadBone
Clothes/Body	BodyBone
Boots	LowerLeg.L, LowerLeg.R

To make a cosmetic item find the related bone(s) for that item (for a hat, the HeadBone, for clothes, the BodyBone, for boots, either of the leg bones; refer to the table above)

Because you're using the stock Test Scene, you'll have to delete one of the items the bean is wearing from the bone. Take note of their structure before you delete it.

Right click on the bone you're gonna place the item on **IN THE HIERARCHY WINDOW** (your only choices are the bones listed above) and then right click on the bone and click "Create Empty". The one I created I renamed to "BoxYouCanWearOnYourHead"


test - My Beanulator Mod - PC, Mac & Linux Standalone - Unity 2020.3.0f1 Personal (Personal) <Metal>

```
test*
  cosmeticTestBean
    Armature
      BodyBone
        HeadBone
          Eyes
          HeadBone_end
          Mouth
          BoxYouCanWearOnYourHead
        Shoulder.L
        Shoulder.R
        UpperLeg.L
          LowerLeg.L
        UpperLeg.R
          LowerLeg.R
        testBelt
    Body
    LeftArm
    LeftLeg
    RightArm
    RightLeg
  testGrenade
```

BoxYouCanWearOnYour
Tag Untagged   Layer

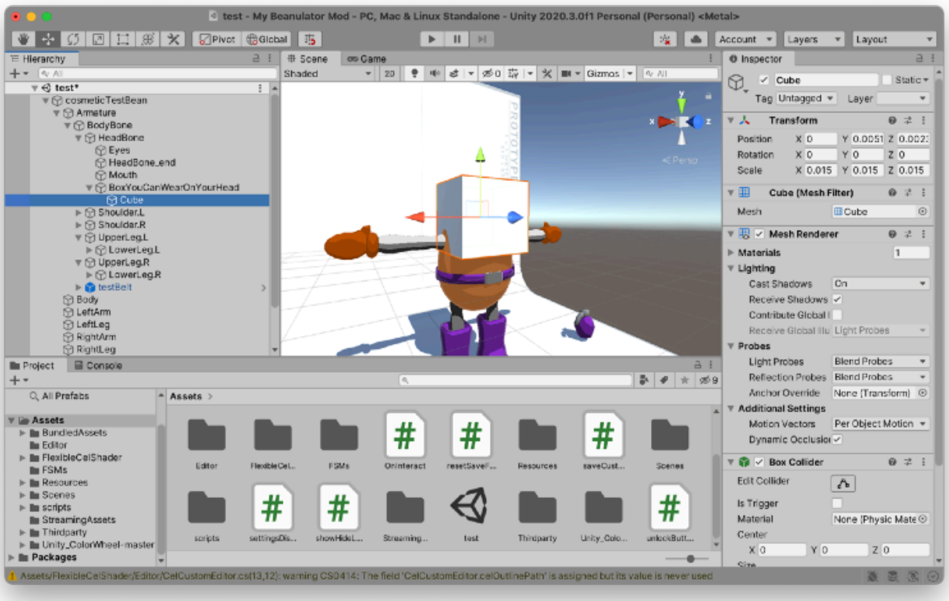
```
Transform
Position   X 0   Y 0   Z 0
Rotation   X 0   Y 0   Z 0
Scale      X 1   Y 1   Z 1
```

Name the empty whatever you want your item name to be. Try to come up with a unique name, otherwise you could cause bugs with other mods. This name won't be displayed in-game, it's just the item's filename. You could make it funny so it's a joke only you will ever laugh at (since no end users will ever see the filename)

This will create an empty GameObject on the bone as a child of the bone. It will mirror the movement of the bone when in-game. Make sure that on the Transform of the object you just created everything is set to 0 (except the scale, all the scale values should be set to 1), because otherwise when loading your cosmetic into the game it won't load in right.

Now, find a 3D model you want and drag it onto the empty **IN THE HIERARCHY WINDOW** *or if you don't have a cool 3D model to use (cringe)* right click on the empty **IN THE HIERARCHY WINDOW** and click "3D" object to create a Primitive on the empty.

Odds are whatever you drag on or the sphere you create will be absolutely massive so mess with the scale values **ON THE OBJECT YOU DRAGGED IN, NOT THE EMPTY, KEEP THE EMPTY SCALED TO 1,1,1** until it's the right scale relative to the bean. The name of whatever you drag onto the empty doesn't really matter since file-wise it's listed as a child of the Empty.



Cube — Tag: Untagged

Transform
Position X 0 Y 0.0051 Z 0.0023
Rotation X 0 Y 0 Z 0
Scale X 0.015 Y 0.015 Z 0.015

Cube (Mesh Filter)
Mesh: Cube

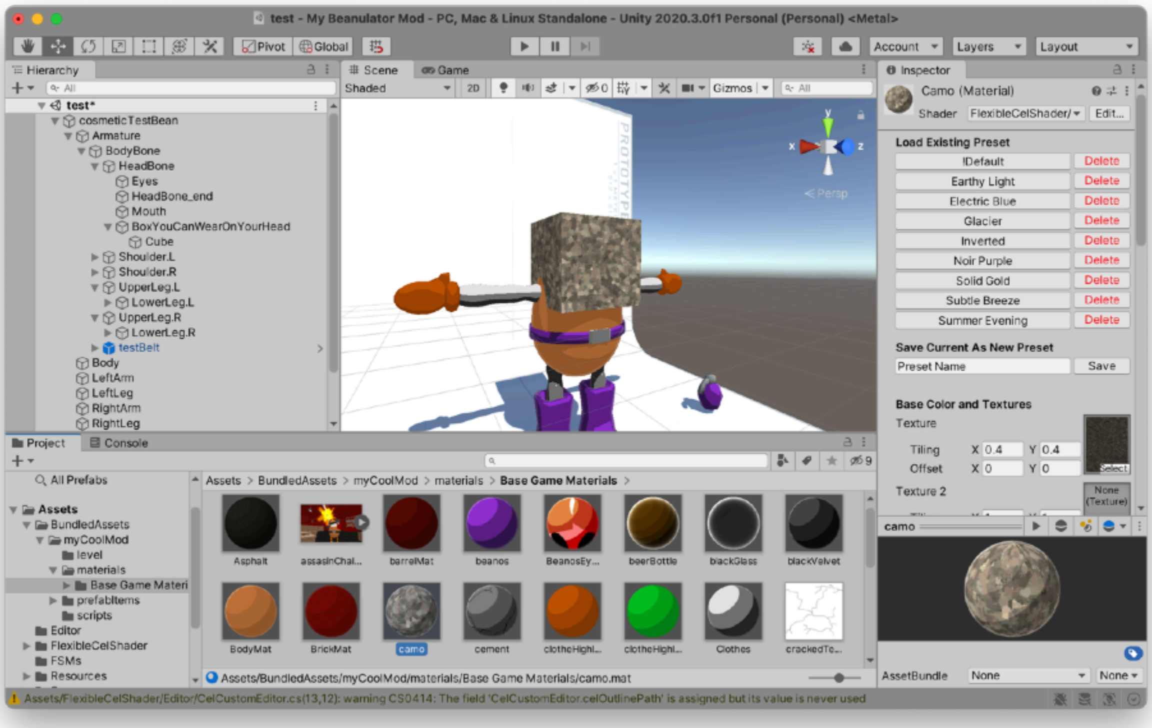
Mesh Renderer
Materials: 1
Lighting
Cast Shadows: On
Receive Shadows: ✓
Contribute Global Illumination:
Receive Global Illumination: Light Probes
Probes
Light Probes: Blend Probes
Reflection Probes: Blend Probes
Anchor Override: None (Transform)
Additional Settings
Motion Vectors: Per Object Motion
Dynamic Occlusion: ✓

Box Collider
Is Trigger:
Material: None (Physic Material)
Center X 0 Y 0 Z 0

```
Assets/FlexibleCelShader/Editor/CelCustomEditor.cs(13,12): warning CS0414: The field 'CelCustomEditor.celOutlinePath' is assigned but its value is never used
```

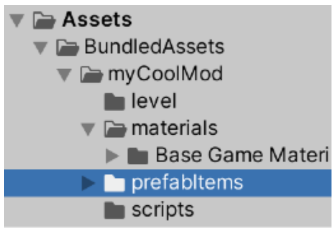
You can mess with the transforms of the object you dragged onto the empty as much as you want as long as you make sure the empty is at zero everything and 1 scale. If your empty gets messed up, you can right click its transform component (on the right side of the screen, in the Inspector Window) and click "Reset".

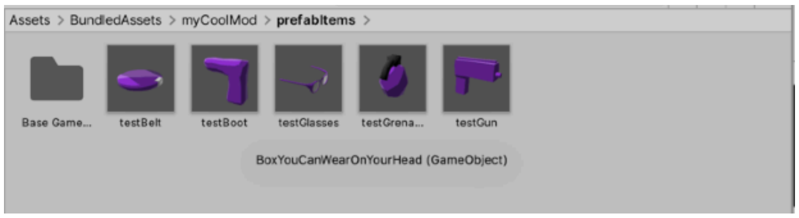
Now that you're done placing the object remember: **YOU CAN'T USE CUSTOM MATERIALS ON THE OBJECT**. (or Beanulator won't be able to import it) So then navigate to Assets > BundledAssets > myCoolMod (the default name) > materials > Base Game Materials. Drag any of the materials that suit your needs onto the cosmetic item.



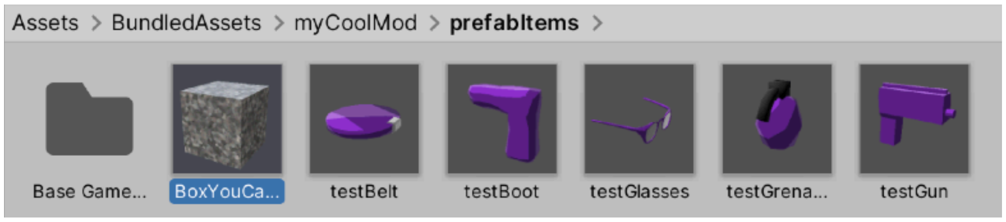
```
Assets/BundledAssets/myCoolMod/materials/Base Game Materials/camo.mat
Assets/FlexibleCelShader/Editor/CelCustomEditor.cs(13,12): warning CS0414: The field 'CelCustomEditor.celOutlinePath' is assigned but its value is never used
```

We're almost done! Navigate to the folder called prefabItems in the Project Window, then drag the empty **FROM THE HIERARCHY WINDOW** into the folder. (as if you were dragging a file around on your desktop)

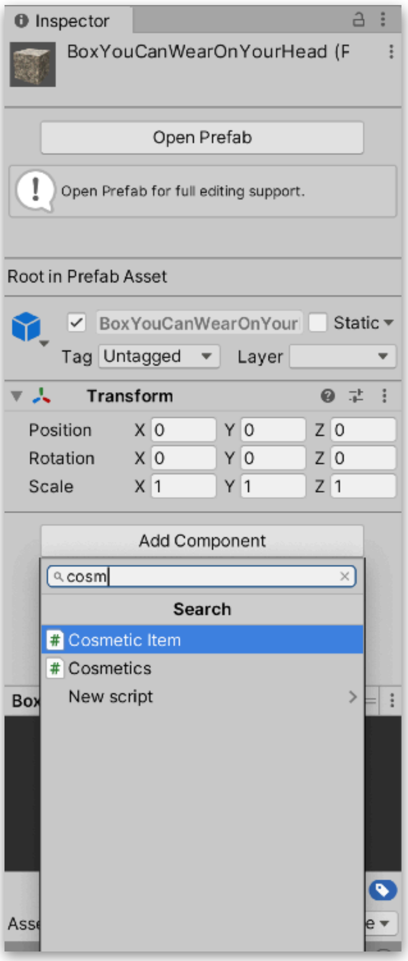



Assets > BundledAssets > myCoolMod > prefabItems >
Base Game...
testBelt
testBoot
testGlasses
testGrena...
testGun
BoxYouCanWearOnYourHead (GameObject)

This will create a Prefab of the item you just created.



Now click on the Prefab you just created or double click it so that this window shows up in your Inspector (the long window to the right). Then, when this shows up, click "Add Component" and search for "Cosmetic Item" then add the script called "Cosmetic Item" to the Prefab. Also click on the little box at the bottom of the Inspector window which says "AssetBundle" next to it, it will probably say "None" then a popup window will show up, click New (scroll down if you don't see it) and name your mod something unique. This is how Beanulator 3D will refer to your mod internally and this is what the mod's exported filename will be, so write it down.



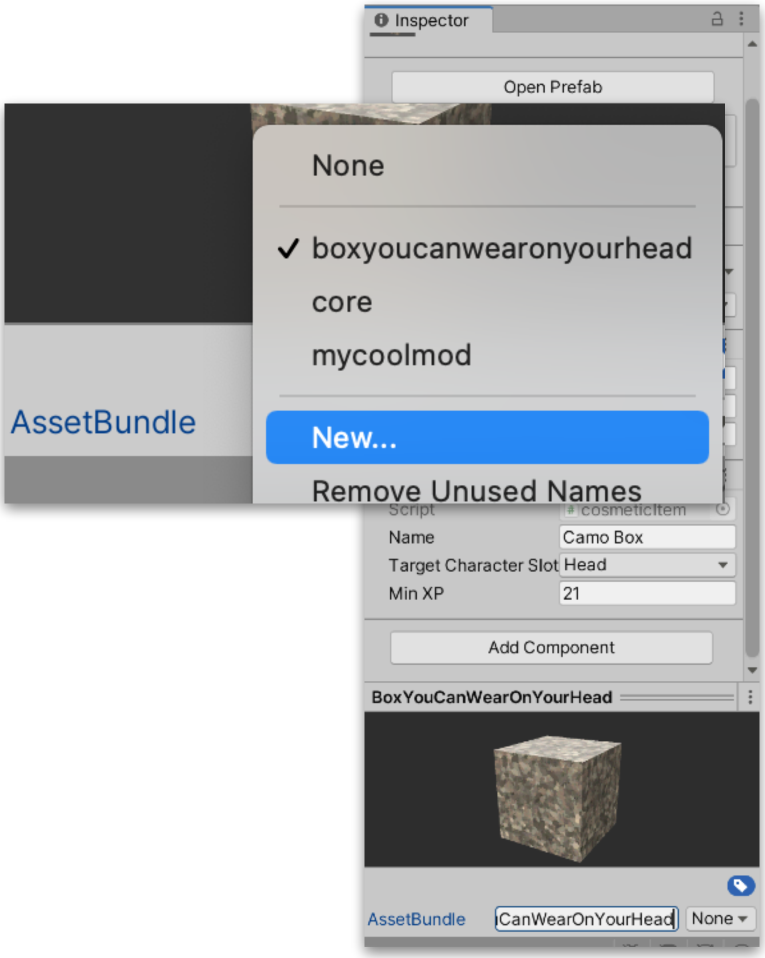
BoxYouCanWearOnYourHead (F

Open Prefab for full editing support.

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Tag Untagged Layer

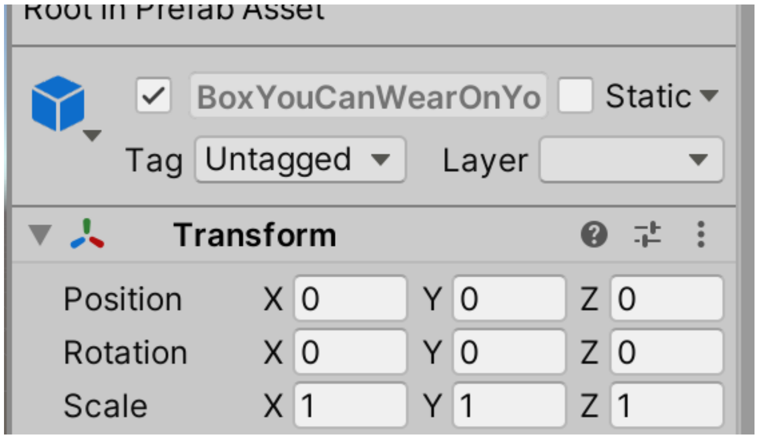
cosm



Script	cosmeticItem
Name	Camo Box
Target Character Slot	Head
Min XP	21

AssetBundle CanWearOnYourHead None

SUPER IMPORTANT STEP I ALWAYS FORGET BUT IS SUPER IMPORTANT:
For your cosmetic mod to show up in game, you have to set its tag to "cosmetic": The left image shows it WRONG, the right image is how it should be set up.



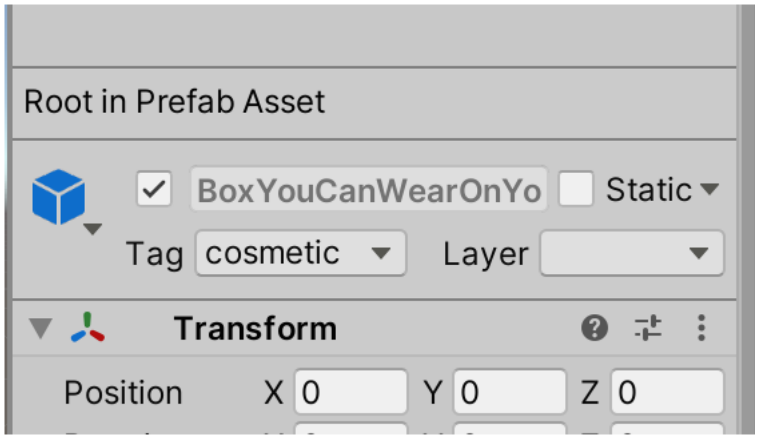
Root in Prefab Asset

BoxYouCanWearOnYo Static

Tag Untagged Layer

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1



Root in Prefab Asset

BoxYouCanWearOnYo Static

Tag cosmetic Layer

Transform

Position	X 0	Y 0	Z 0

You're almost done!

Checklist for a succesfull export (cosmetic):

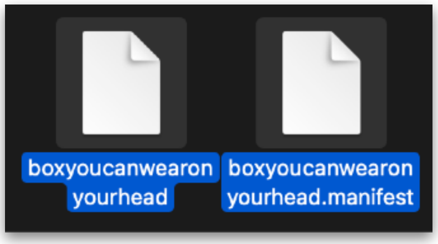
- Your prefab is in the prefabItems folder
- Your prefab has a cool and unique name
- Your prefab is tagged with the "cosmetic" tag
- Your prefab's transforms are all set to their default values
- Your prefab's AssetBundle property is set to whatever you want to call your mod (NOT "core" or "Beanulator 3D")
- Your prefab has a Cosmetic Item script and that script's name is set to whatever you want your object to display as in-game (pick a short name so it's easy to read) and the min XP value is set to something sane
- Your prefab's "target character slot" is set to the slot that corresponds to the bone you originally built the cosmetic on
- Ideally, remove all colliders from the GameObject because they could interfere with hit detection or physics simulation.
- All materials on the object are built-in, otherwise they'll show up pink in-game

EXPORTING MODS

Click "Assets" in the toolbar (on mac it's on the top of the screen, on Windows it's on the top of the window) then go down to the button that says "Export Mod to Beanulator Mod Folder". You must have Beanulator installed on your computer for this to work.
If you don't know where your Beanulator folder is, click "Show Persistent Path", which is also in the Assets menu, right under "Export Mod to Beanulator Mod Folder".

Click the "Export Mod to Beanulator Mod Folder" button. If everything is set up right, you will then be able to start up Beanulator and have your mod show up in-game. To confirm it exported correctly, go to Beanulator's settings menu then click "mods" in the top left to show your mods list. If your mod name shows up in there, your good to go. If it didn't, check the log in the Unity mod development project. (click the bar at the bottom of the Editor window that shows an icon to the left and text to the right). If the export was successful and your mod doesn't show up in-game, double check the "**Checklist for a successful export (cosmetic)**"

upon succesful export it will send you to a folder. There will be two files in that folder: [yourmodname] and [yourmodname].manifest



Note for Mac Users:
Auto-export seems to be broken on mac, when you hit Export it will send you to a folder. In this folder you will find the two files, [yourmodname] and [yourmodname].manifest, which you will then drag to the folder

/Users/[username]/Library/ApplicationSupport/
com.DecaffeinatedStudios.Beanulator3D/
On Mac, go to the finder, toolbar, select Go, then select Go To Folder, or search for "Go to folder" in the Help menu, then paste the path above with [username] replaced with your username.

Notes/Best Practice

- Your prefab is in the prefabItems folder
- Your prefab has a cool and unique name
- Your prefab is tagged with the correct tag ("cosmetics" for a cosmetic, "Weapon" for a weapon)
- Your prefab's transforms are all set to their default values
- Your prefab's AssetBundle property is set to whatever you want to call your mod (NOT "core" or "Beanulator 3D")
- Your prefab has the appropriate Cosmetic Item (for a cosmetic) or Gun Object for a weapon, or "OnInteract" and "EXPLODEWITHDELAY" in the case of a grenade. In the case of a generic interactive item, it's kind of a case-by-case situation.
- If Your prefab is a cosmetic it's "target character slot" is set to the slot that corresponds to the bone you originally built the cosmetic on
- For cosmetics, Ideally, remove all colliders from the GameObject because they could interfere with hit detection or physics simulation. For guns and grenades, make sure only one Collider is attached per prefab.
- All materials on the object are built-in, otherwise they'll show up pink in-game
- All relevant fields in the attached scripts are filled out (some fields, like Bullet GO and Muzzle Flash in the Gun Object script, don't need to be filled out because they get filled out automatically by the game engine in-game. If you do fill them out it could cause bugs.
- If you don't care about uploading your own 3D models, the stock 3D models are in the Resources/models folder.

Everything you can do to make mods has to be in the prefabItems folder.

Stuff you can't export to custom mods: (CANNOT)
Custom materials (stuff that isn't included in the project)
Custom scripts (stuff that isn't included in the project)
Most particle systems (they show up super buggy)
custom textures (stuff that isn't included in the project)
custom sounds (stuff that isn't included in the project)

STUFF YOU CAN EXPORT:
Custom images **set as sprites** placed in the prefabitems folder. Level icons, for example.
Your own models (make sure to set their asset bundle to your mod's asset bundle so they export correctly)
Most physics joints
Every script included in the mod maker
Every material included the mod maker
The built in models (under the resources folder)
If you want to export your own image, you can draw it out in Blender as a 3D model with different parts of the mesh having different materials, then set those materials to the built in materials.

Custom Cosmetics

See the above tutorial. It covers cosmetics pretty well.

Weapon modding

For weapons, duplicate the existing testGun prefab, then double click it to open the editor window. Change the model of the gun or whatever, mess with the stats, change the name.
Available gunshot sounds: BeanettaShot, lmg_fire01, UziShot, suppressedShot, punch, kick, DeathFlash ,
Available bullet types: shotgunBullet, smallBullet.
You can find the gunshot sounds in Resources/Audio/sfx/
You can find the bullets in Resources/prefabs/
You cannot modify the Resources folder.

Bullets will spawn at whatever you drag into BulletSource (must be a child of the gun object) Make sure the blue arrow is aiming towards the direction you want to aim.

Structure of a weapon in Beanulator:
Base empty (all transforms set to default). This object has the collider and rigidbody and scripts attached. If you're unsure, check out how the testGun is set up. Make sure you set up the collider yourself, with care, otherwise the gun will fall through terrain or snap off arms.
Structure:
Base Empty
L The gun's model
L The bullet source (can be a child of either the base empty or the gun's model, i'd recommend the Base Empty even though in the testGun example it's set to the gun's model)

L indicates 'Child of object'. In this case both the gun's model and the bullet source are children of the base gun empty.

Custom Level Design

Check out how the exampleLevel is set up. This will give you an idea of the overall structure of levels in beanulator. Things to note:
Levels in Beanulator are actually Prefabs, not scenes.

SUPER IMPORTANT: The base gameobject (empty) of the level has to have the tag "levelController". If you don't set this, other objects in the game don't know what to talk to when stuff like winning the level or the player dying happens, and can cause tons of awful bugs.

Patrol routes/Waypoints: Collections of empties. The empties must have the tag "Waypoint" so that AI's will wander from waypoint to waypoint. If a level has no waypoints AI won't even work and will crash the game (since the AI's will be stuck in an infinite loop of finding waypoints to navigate to)

MapBase: I'd recommend storing all the level stuff that isn't important under an Empty whose transforms are zeroed out (again, position, rotation zeroed out, scale set to 1). This makes the hierarchy window less full of crap and less distracting.

BeanStandin: These are spawners for beans. They hold an "ARC Instantiator" script. **MAKE SURE THEY AREN'T SET TO 'Is Player' unless it is the player**. Otherwise weird bugs will happen.
Primary is the color of the arms, **Secondary** is the color of the gloves and boots.
HatC, ClotheC,BootC currently don't do anything. I don't plan on making them do anything. I don't even know why they are there.

Recommended settings for an average goon:

Health: <15
Toughness: <6~7
Strength: 750-2000 (significantly less than 2000 means they'll likely have terrible aim and may not be able to stand up)

Recommended settings for a tough enemy:

Health: 20~30
Toughness: 7~10
Strength: 2000

Leave a good gun near the AI so it'll pick it up

Recommended settings for a really tough enemy:

Ambidextrous set to True
Health: 30~50
Toughness: 10~15
Strength: 2000

Leave a good gun near the AI so it'll pick it up

Give it good cosmetics

Recommended settings for a level boss:

Ambidextrous set to True
2000-3000 **strength** (don't go above that or the bean's limbs will probably snap off because of the sheer strength of the muscles, instantly killing the supposedly tough boss)
Toughness 15+ means almost nothing will make the boss ragdoll
More than 100 **Health**
Set the hat, boots and clothes to something cool. The current mod making file doesn't have most of the stock in game cosmetics so you should make your own cool cosmetics for your boss.

Unused slots:
HatC, ClotheC,BootC,Cosm slot, Target Bean Base, Selected Item.

REMEMBER TO SET Is Player to FALSE for AI's

Properties you can set in the Level Data script

Level name: The display name of your level

Level Blurb: The text that displays under the icon of your level. Write backstory here, or prod the player in the right direction. Idk what the character limit is but just don't write too much or it won't show up in-game. Just test your game to confirm the blurb fits.

Level Type:
Deathmatch: Kill all enemy beans spawned in at beginning of round to win (don't use a spawner!!)

Assasin: Kill a target bean to win the level. Set the target bean's spawner's property "Is Level Target" to True.

Steal: Steal a briefcase (or any rigidbody set with a tag of "Weapon"). Set the target drop point (where the briefcase needs to be taken to) by dragging a reference the target point into the "place here to win" property. Place a reference to the briefcase into the "Win Object" slot

Sandbox: No level objective. Player will have to quit out manually.

Level Preview: A Sprite that will show up when the level is highlighted. I'd recommend a screenshot of the most interesting part of your level. Roughly 4:3 aspect ratio but it really doesn't matter.

Other stuff

For an example on how to make grenades and other similar items (using the OnInteract script) check how the testGrenade is set up.