

Etra's Starter Assets

Unity Hub Documentation v1.3.0

Patreon: <https://www.patreon.com/Games4Nongamers>

Please watch the following videos for a breakdown on asset features:

1) Overview: <https://youtu.be/5Yb4oaFpZ04>

2) Modding: <https://youtu.be/q26z6KW1geI>

Etra's Starter Assets is an expanded combination of the Unity First and Third Person Starter Assets. A new Ability System and Character Creator lets you quickly build new character controllers with various abilities.

The Etra's Starter Assets package is compatible with Unity 2020.3 and 2021.3. It also the Standard And Universal Unity Render Pipelines.

Just like the original packages, the Etra's Starter Assets package requires the Input System and Cinemachine packages to work. The packages will automatically install when you import the Etra's Starter Assets package into your project.

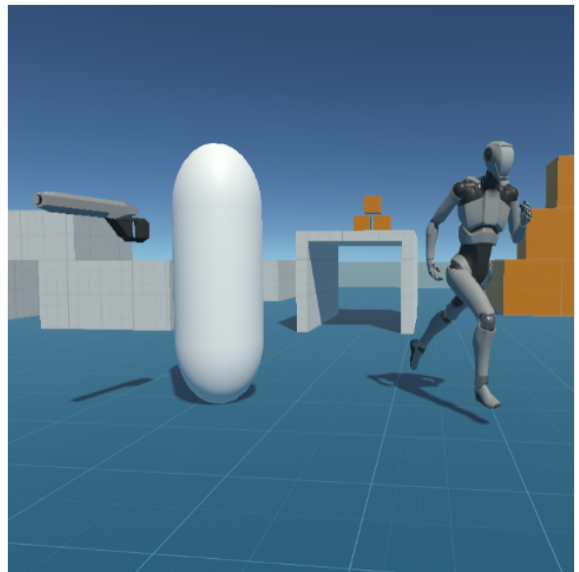


Table of Contents:

- Package Dependencies
- Set up the Starter Assets in a New Scene
- Demo Scenes
- Interactable Objects
- Ability System
- Ability Script Structure/Creating Custom Abilities
- Usable Item System (FPS)
- Usable FPS Item Structure/Creating Custom FPS Items
- Sharing Your Interactables, Abilities, or Items
- Standard Menus
- Credits

- Support This Project!

Package Dependencies:

The Etra's Starter Assets package requires the Cinemachine and TextMeshPro packages to work.

It is *highly recommended that* you download the Input System package. Partial support for Unity's default input system is included in the StarterAssetsInputs.cs script.

The packages will automatically install when you import a Starter Assets package into your project. This is handled by the *PackageChecker.cs* script.

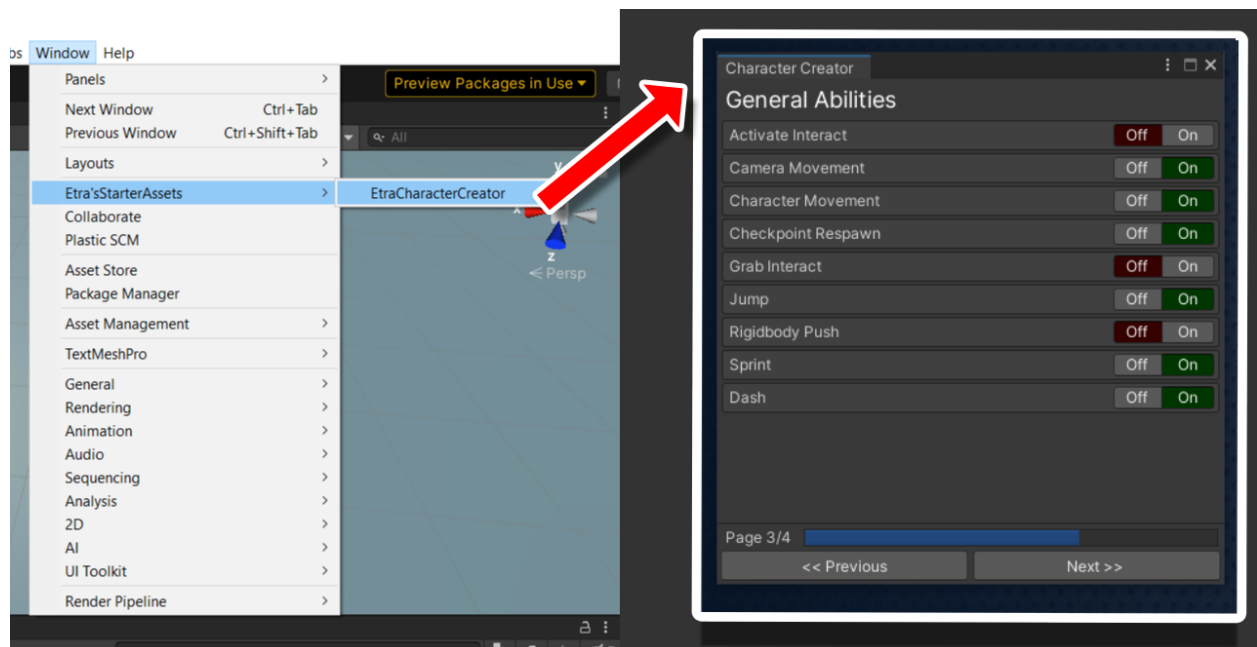
Please accept the Input System pop-up and Editor restart to successfully install the New Input System. If you accidentally decline, you can install the New Input System through the Package Manager.

Set up The Starter Assets in a new Scene:

There are two ways to set up the Etra Starter Assets:

Character Creator:

If you want to add a character controller to a new scene, simply go to Window->Etra'sStarterAssets->Etra'sCharacterCreator



From there select what general abilities and character type you wish to create. Then press next and eventually the “CREATE” button for the character controller to be added to your scene.

Drag and Drop (Prefabs):

Alternatively, you can simply drag and drop a pre-built character controller from *Etra'sStarterAssets/1-UserAssets/CharacterAsset/Prefabs*

Demo Scenes:

In *Etra'sStarterAssets/1-UserAssets/Scenes* there are three demo scenes for testing out the Etra Character Controllers.

- *FPSCharacterDemo.unity* showcases a FPS character with all general abilities, FPS abilities, and FPS Items.
- *TPSCharacterDemo.unity* showcases a TPS character with all general and TPS abilities. It also showcases available models for the TPS character.
- *NoCharacterDemo.unity* showcases an empty playground where you can build a custom character.

Interactable Objects:

Several interactable objects are included in the Etra's Starter Assets package. You can find them *Etra'sStarterAssets/1-UserAssets/InteractableObjects*.

Ability System:

The abilities on a character are managed by the *EtraAbilityManager.cs* script on the EtraAbilityManager object. Code for character controller actions are separated into many small ABILITY scripts. These ABILITY scripts all inherit from the *EtraAbilityBaseClass.cs* parent script. Most ability scripts work for both the FPS and TPS gameplay types. There are ability scripts exclusive to both gameplay types.

Ability Script Structure/Creating Custom Abilities:

To create your own new abilities please follow the digital tutorial in the following video.

To tag an ability as general, fps, or tps exclusive use the AbilityUsageAttribute class.

Link: <https://youtu.be/okOk8vS8D2w>

FPS Usable Item System:

The Usable FPS Items on a character are managed by the *EtraFPSUsableItemManager.cs* script on the EtraFPSUsableItemManagerPrefab object.

The code for each item is contained in its own FPSUsableItem script. These FPSUsableItem scripts all inherit from the *EtraFPSUsableItemBaseClass.cs* parent script. The manager only enables the active item's script and disables the rest.

FPS Usable Item Structure/Creating Custom FPS Items:

To create your own FPS items please follow the digital tutorial in the following video.

Link: <https://youtu.be/okOk8vS8D2w>

Sharing Your Interactables, Abilities, or Items:

Once your additions are created, feel free to join my Discord, Etra's Lab to share your abilities with other users.

Link: <https://discord.gg/zZdGJQvNvq>

Standard Menus:

For more information on the standard menus go to the following path and read the readme.txt.

Assets/Etra Games/Etra's Starter Assets/1-User Assets/Etra Standard Menus

Credits:

Several creators have helped assemble this starter assets package. Each original asset has an appropriate license near it. All other third party content is listed in

ThirdPartyNotices.txt. The original asset credits are summarized here for convenience.

Thanks to all who've helped out so far :)

Version 1.0.0

Buffalou - <https://www.youtube.com/@buffalou>

-FPSUsableItemSwayAndBobAnimations.cs

CodyCantEatThis - <https://www.youtube.com/@CodyCantEatThis>

-Voxel Character Models

-VoxelAnimator.cs

Krissy - Kris#0001

-Treadmill Model

-Treadmill_Interactable.cs

-Weighted Button Model

-Interactable_WeightedButton.cs

amysour - <https://www.youtube.com/@aMySour>

- Standing Button Model
- ObjectInteraction.cs
- SciFi Door Model
- SciFiDoor.cs
- Physics Grabber Model
- USABLEITEM_FPS_Physics_Grabber.cs
- Blaster muzzle flash particle system

Version 1.0.1

Dock Frankenstein - https://www.youtube.com/channel/UCq_7pbSyOvrurXLAMi_Ss1w

- Revised Character Creator

Support This Project! :

If you want consistent updates on Etra's Starter Assets or other free projects I am working on. Feel free to support my Patreon or subscribe to my Youtube channel. Thanks!

Patreon: <https://www.patreon.com/Games4NonGamers>

Youtube: <https://www.youtube.com/@EtrasGames4NonGamers>