Complete Top-Down Controller

Quickstart Guide

Thank you for purchasing the Complete Top-Down Controller. This quickstart guide will you get up and running in minutes so that you can begin creating your very own top-down game. If any further explanation is needed for any of these steps, please view the full ReadMe file included in the package.

Initial Set-Up

1.) Navigate to: ***Edit -> Project Settings -> Player -> Other Settings -> Scripting Define Symbols*** and add "CROSS_PLATFORM_INPUT" and "MOBILE_INPUT" to your project

2.) Ensure that tags are set up properly:

- "Player" tag should be used for the player character's parent object only.
- "Walls" tag should be used by a single layer on your tilemap.
- "Item" tag should be used by any objects that use the "TDC_InteractableObject" script.

WARNING: Step 3 will overwrite any input data that is currently set up in your project. If you want to avoid this, view Step 2, Method 1 in the full ReadMe file.

3.) Right-click in the project window and select "Show In Explorer" to open your project folder outside of Unity. Navigate up until you find a folder named "Project Settings". Download the "InputManager.asset" file from here and place it in the "Project Settings" folder in your project. Overwrite any existing files and replace them with this new downloaded file.

4.) Drag and drop the "Player" and "WorldMap" prefab objects into your game scene. You can also add the "Walls" tag to an existing layer in a tilemap instead of using the "WorldMap" prefab. The "WorldMap" game object has its transform set to (x: -0.5, y: -0.5, z: 0), while the rest of the child objects are set to (x: 0, y: 0, z: 0). If you are using an existing tilemap that was created, make sure to adjust its transform to match this.

5.) Right-click in your project window and select: ***Create -> Animator Override Controller.*** Choose the controller type (GridAnimationController or RBAnimationController) and create animation clips that correspond with the variables shown. Put your created clips in the variable slots on the Animator Override Controller and click on the "Player" in your scene. Drop your new Animator Override Controller into the "Anim Override Controller" slot on the "TDC_Movement" script. Also change the "Sprite" variable on the "Sprite Renderer" component to an idle sprite to match your new animations.

Interactable Objects

There are many prefabs that you can use to quickly add items to your game scene. Keep in mind that, by default, these items do not have any functionality beyond disappearing when the player interacts with them. This can be changed as needed by editing the scenario script attached to the item. All items from this asset have the same 2 scripts attached, "TDC_InteractableObject" and "TDC_ItemScenarios".

"TDC_InteractableObject" should **NOT** be edited in any way at all, as it is the core behavior script for in-game items. "TDC_ItemScenarios" is used to determine what should happen when the player interacts with an item. Open "TDC_ItemScenarios" in your script editor and you will see additional notes on how to add functionality to your script. It is recommended to view the full ReadMe text for a better understanding before making any edits to this file.

Teleporting

The Teleport Tile object is a trigger box that will fade the screen, change the music, and teleport the player to a specified location, either in the same scene or to a new scene. The player character only needs to be added once to the starting scene of the level. You may notice a warning in the console regarding 2 audio listeners when returning to your starting scene. This is normal due to the player initially being active there, but is corrected automatically by the script and can be safely ignored.

Anytime that multiple teleport tiles would be used next to each other, they should be replaced with a single teleport tile that is stretched to be wider. If the player walks between 2 teleport tiles at the same time it will crash your game. Instead, use a single Teleport Tile and make it wider to fit your needs.

Adding Buttons

1.) Locate the game object named "btnA" that is a child of the "Touch Controls" game object under the player. Duplicate it, rename it to "btnX", and move it above the A button in your game scene.

2.) Select the new button that was created and change its sprite to the image that should be shown when the button is not pressed. Next, locate the "Button" component and change both the "Highlighted Sprite" and the "Pressed Sprite" values to the image that should be used when the button is being pushed down. Last, find the "TDC_ButtonHandler" component and change the "Button Name" to "X".

3.) Navigate to: ***Edit -> Project Setting -> Input Manager*** and add an additional input for the new X button. Copy the same settings that are used for the A button, but change the "Positive Button" to a new value and rename the input to "X".

4.) Locate the player in your scene hierarchy, select it and open the "TDC_Actions" script for editing. Once opened, you will notice 2 templates at the top of this script that can be copied and pasted under the Custom Button Scripts area. Copy the entire Get Button Down Template function, paste it below and un-comment it by selecting the text and pressing Ctrl+K+U. Now locate the phrase "CustomButton" four times in the pasted function and replace it with "X".

Conclusion

I hope that this documentation has been helpful in setting up your Complete Top-Down Controller. If this guide did not adequately explain a step, then please view the full ReadMe file for a deeper explanation. If you have any issues with this package, find a bug, or need help with set up then please feel free to reach out to me via email at FrodoUndead@gmail.com I will do my very best to help you with any problems you may encounter. Thank you again for your purchase. Happy development! 💻