GameObject is not getting activated after build (Windows, Mac, Linux), but gets activated and works as expected in the editor - Unity

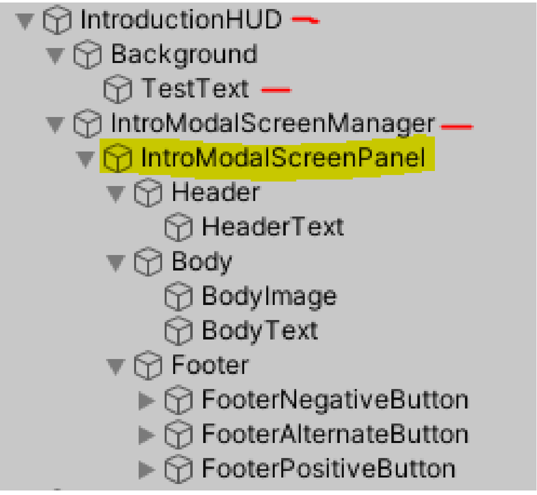
Asked 2 years, 1 month ago Modified 1 year, 9 months ago Viewed 604 times

1

I tried all the recommended steps and potential solutions suggested over the forums. However, this problem still persists. Its been 5 days and I am still trying to find why it happens only in build export and not in unity editor.

[Edit (1st Dec 2022): Recently I found that this issue doesn't happen when I "Build and Run" the project the first time. However, If I close and launch the game again using .exe file, the issue is still coming up. I don't know how it runs as expected only the first time when I do "Build and Run" instead of just "Build" or "Clean Build".]

Issue Details:

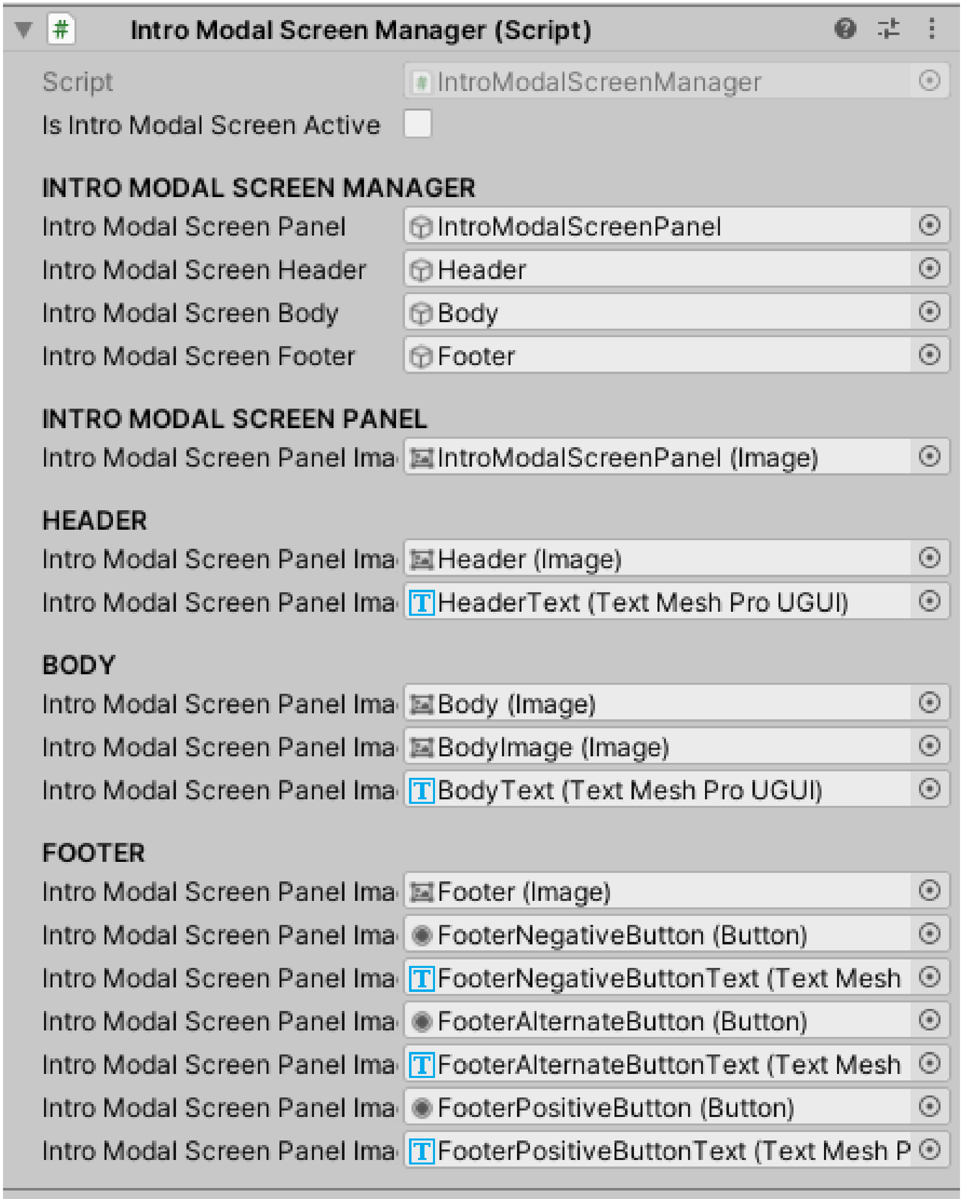


```
▼ IntroductionHUD
  ▼ Background
      TestText
  ▼ IntroModalScreenManager
    ▼ IntroModalScreenPanel
      ▼ Header
          HeaderText
      ▼ Body
          BodyImage
          BodyText
      ▼ Footer
        ▶ FooterNegativeButton
        ▶ FooterAlternateButton
        ▶ FooterPositiveButton
```

'IntroductionHUD' (Canvas) is the canvas of this scene. 'TestText' (TextMeshPro) is used to test if the 'IntroModalScreenPanel' (Image) is active or not. 'IntroModalScreenManager' (Empty Object) has the script to handle activation and deactivation of IntroModalScreenPanel (Image) and its children.

There is an awake method in the 'IntroModalScreenManager' script which sets the 'IntroModalScreenPanel' and its children to inactive.

And there are other methods in the script which activates and deactivates the 'IntroModalScreenPanel' and its children accordingly. The objects are activated and deactivated using SetActive(bool) in the script. The objects are assigned through the inspector.



Intro Modal Screen Manager (Script)

Script	IntroModalScreenManager
Is Intro Modal Screen Active	☐

INTRO MODAL SCREEN MANAGER

Intro Modal Screen Panel	IntroModalScreenPanel
Intro Modal Screen Header	Header
Intro Modal Screen Body	Body
Intro Modal Screen Footer	Footer

INTRO MODAL SCREEN PANEL

Intro Modal Screen Panel Ima	IntroModalScreenPanel (Image)

HEADER

Intro Modal Screen Panel Ima	Header (Image)
Intro Modal Screen Panel Ima	HeaderText (Text Mesh Pro UGUI)

BODY

Intro Modal Screen Panel Ima	Body (Image)
Intro Modal Screen Panel Ima	BodyImage (Image)
Intro Modal Screen Panel Ima	BodyText (Text Mesh Pro UGUI)

FOOTER

Intro Modal Screen Panel Ima	Footer (Image)
Intro Modal Screen Panel Ima	FooterNegativeButton (Button)
Intro Modal Screen Panel Ima	FooterNegativeButtonText (Text Mesh
Intro Modal Screen Panel Ima	FooterAlternateButton (Button)
Intro Modal Screen Panel Ima	FooterAlternateButtonText (Text Mesh
Intro Modal Screen Panel Ima	FooterPositiveButton (Button)
Intro Modal Screen Panel Ima	FooterPositiveButtonText (Text Mesh P

For some reason, which I have no idea of, the 'IntroModalScreenPanel' doesn't get activated after it deactivates in the awake method. Additionally, this weird problem only exists when I do a build export. Whereas, it works perfectly fine and as expected in Unity Editor.

The 'TestText' is used to test if the 'IntroModalScreenPanel' is active or not and every time it turns to be not active in the build export.

Another weird part is (if you see the hierarchy snapshot) the 'Background' and 'TestText' works fine being the children of the same canvas object (IntroductionHUD) as of 'IntroModalScreenManager'. As per my understanding the problem is only with the 'IntroModalScreenPanel' and its children.

I would really appreciate if anyone could advise on this. Any suggestions will do.

Script:

```
using System;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class IntroModalScreenManager : MonoBehaviour
{
    public bool IsIntroModalScreenActive() { return _isIntroModalScreenActive;
}

    //Bool variables...
    [SerializeField] private bool _isIntroModalScreenActive;

    //INTRO MODAL SCREEN MANAGER
    [Header("INTRO MODAL SCREEN MANAGER")]
    [SerializeField] private GameObject _introModalScreenPanel;
    [SerializeField] private GameObject _introModalScreenHeader;
    [SerializeField] private GameObject _introModalScreenBody;
    [SerializeField] private GameObject _introModalScreenFooter;

    //INTRO MODAL SCREEN PANEL
    [Header("INTRO MODAL SCREEN PANEL")]
    [SerializeField] private Image _introModalScreenPanelImage;

    //HEADER
    [Header("HEADER")]
    [SerializeField] private Image _introModalScreenPanelImageHeaderImage;
    [SerializeField] private TextMeshProUGUI
_introModalScreenPanelImageHeaderImageHeaderText;

    //BODY
    [Header("BODY")]
    [SerializeField] private Image _introModalScreenPanelImageBodyImage;
    [SerializeField] private Image
_introModalScreenPanelImageBodyImageBodyImage;
    [SerializeField] private TextMeshProUGUI
_introModalScreenPanelImageBodyImageBodyText;

    //FOOTER
    [Header("FOOTER")]
    [SerializeField] private Image _introModalScreenPanelImageFooterImage;
    [SerializeField] private Button
_introModalScreenPanelImageFooterImageFooterNegativeButton;
    [SerializeField] private TextMeshProUGUI
_introModalScreenPanelImageFooterImageFooterNegativeButtonText;
    [SerializeField] private Button
_introModalScreenPanelImageFooterImageFooterAlternateButton;
    [SerializeField] private TextMeshProUGUI
_introModalScreenPanelImageFooterImageFooterAlternateButtonText;
    [SerializeField] private Button
_introModalScreenPanelImageFooterImageFooterPositiveButton;
    [SerializeField] private TextMeshProUGUI
_introModalScreenPanelImageFooterImageFooterPositiveButtonText;

    public static IntroModalScreenManager Instance { get; private set; }

    private void Awake()
    {
        if(Instance != null && Instance != this)
        {
            Destroy(this);
        }
        else
        {
            Instance = this;
        }

        DeactivateModalScreenDisplay();

        //DontDestroyOnLoad(this.gameObject);
    }

    //Called by Awake() in the begining and by other script to deactivate modal
screen when needed
    public void DeactivateModalScreenDisplay()
    {
        DeactivateAndClearAll();

        _introModalScreenPanel.SetActive(false);

        _isIntroModalScreenActive = false;
    }

    void DeactivateAndClearAll()
    {
        DeactivateHeader();
        DeactivateBody();
        DeactivateFooter();
    }

    //Called by another script to activate modal screen when needed
    public void ActivateModalScreenDisplay(bool isBook, bool
isBookWithAlternateOption,
        bool isSinglePage, bool isSinglePageWithNegativeOption, bool
isSinglePageWithNegativeAndAlternateOption,
        bool needBodyImage,
        string headerText = "",
        Sprite bodyImageSprite = null, string bodyText = "",
        string footerNegativeButtonText = "", string footerAlternateButtonText
= "", string footerPositiveButtonText = "",
        Action negativeActionOption = null, Action alternateActionOption =
null, Action positiveActionOption = null)
    {
        _introModalScreenPanel.SetActive(true);

        //Book - has front and back navigation options
        if (isBook)
        {
            ActivateHeader(headerText);
            ActivateBody(needBodyImage, true, bodyImageSprite, bodyText);
            ActivateFooter(true, false, true, footerNegativeButtonText,
footerAlternateButtonText, footerPositiveButtonText,
                negativeActionOption, alternateActionOption,
positiveActionOption);
        }
        //BookWithAlternateOption - Book with an addition of alternate option.
(Eg: Skip button)
        else if (isBookWithAlternateOption)
        {
            ActivateHeader(headerText);
            ActivateBody(needBodyImage, true, bodyImageSprite, bodyText);
            ActivateFooter(true, true, true, footerNegativeButtonText,
footerAlternateButtonText, footerPositiveButtonText,
                negativeActionOption, alternateActionOption,
positiveActionOption);
        }
        //SinglePage - has one button (positive option - Eg: Continue, start,
etc. buttons)
        else if (isSinglePage)
        {
            ActivateHeader(headerText);
            ActivateBody(needBodyImage, true, bodyImageSprite, bodyText);
            ActivateFooter(false, false, true, footerNegativeButtonText,
footerAlternateButtonText, footerPositiveButtonText,
                negativeActionOption, alternateActionOption,
positiveActionOption);
        }
        //SinglePageWithNegativeOption - Single page with an addition of
negative option (Eg: Confirmations)
        else if (isSinglePageWithNegativeOption)
        {
            ActivateHeader(headerText);
            ActivateBody(needBodyImage, true, bodyImageSprite, bodyText);
            ActivateFooter(true, false, true, footerNegativeButtonText,
footerAlternateButtonText, footerPositiveButtonText,
                negativeActionOption, alternateActionOption,
positiveActionOption);
        }
        //SinglePageWithNegativeAndAlternateOption - Single page with negative
option and an aditional alternate button (Eg: 3rd option in any choice making
scenes)
        else if (isSinglePageWithNegativeAndAlternateOption)
        {
            ActivateHeader(headerText);
            ActivateBody(needBodyImage, true, bodyImageSprite, bodyText);
            ActivateFooter(true, true, true, footerNegativeButtonText,
footerAlternateButtonText, footerPositiveButtonText,
                negativeActionOption, alternateActionOption,
positiveActionOption);
        }

        _isIntroModalScreenActive = true;
    }


    void ActivateHeader(string headerText)
    {
        //Only if the header text has some value, this component will be
active... Implemented this only for header & footer, as header & footer is used
for all types of modal screen layouts.
        if(headerText != "" && headerText != null)
        {
            //HEADER
            _introModalScreenHeader.SetActive(true);//Header Object


_introModalScreenPanelImageHeaderImageHeaderText.gameObject.SetActive(true);//Head
Text Object
            _introModalScreenPanelImageHeaderImageHeaderText.text =
headerText;//Header Text
        }
    }

    void DeactivateHeader()
    {
        _introModalScreenPanelImageHeaderImageHeaderText.text = "";//Header
Text
        if
```

```
(_introModalScreenPanelImageHeaderImageHeaderText.gameObject.activeSelf)
        {

_introModalScreenPanelImageHeaderImageHeaderText.gameObject.SetActive(false);//Hea
Text Object
        }

        if (_introModalScreenHeader.activeSelf)
        {
            _introModalScreenHeader.SetActive(false);//Header Object
        }
    }

    void ActivateBody(bool needBodyImage, bool needBodyText, Sprite
bodyImageSprite, string bodyText)
    {
        //BODY
        _introModalScreenBody.SetActive(true);//Body Object


_introModalScreenPanelImageBodyImageBodyImage.gameObject.SetActive(needBodyImage);
Image Object
        if
(_introModalScreenPanelImageBodyImageBodyImage.gameObject.activeSelf)//Body
Image Object Image Sprite
        { _introModalScreenPanelImageBodyImageBodyImage.sprite =
bodyImageSprite; }


_introModalScreenPanelImageBodyImageBodyText.gameObject.SetActive(needBodyText);//
Text Object
        if
(_introModalScreenPanelImageBodyImageBodyText.gameObject.activeSelf)//Body Text
Object Text
        { _introModalScreenPanelImageBodyImageBodyText.text = bodyText; }
    }

    void DeactivateBody()
    {
        _introModalScreenPanelImageBodyImageBodyText.text = "";
        if (_introModalScreenPanelImageBodyImageBodyText.gameObject.activeSelf)
        {

_introModalScreenPanelImageBodyImageBodyText.gameObject.SetActive(false);//Body
Text Object
        }

        _introModalScreenPanelImageBodyImageBodyImage.sprite = null;
        if
(_introModalScreenPanelImageBodyImageBodyImage.gameObject.activeSelf)
        {

_introModalScreenPanelImageBodyImageBodyImage.gameObject.SetActive(false);//Body
Image Object
        }

        if (_introModalScreenBody.activeSelf)
        {
            _introModalScreenBody.SetActive(false);//Body Object
        }
    }
    void ActivateFooter(bool needFooterNegativeButton, bool
needFooterAlternateButton, bool needFooterPositiveButton,
        string footerNegativeButtonText, string footerAlternateButtonText,
string footerPositiveButtonText,
        Action negativeActionOption, Action alternateActionOption, Action
positiveActionOption)
    {
        //FOOTER
        _introModalScreenFooter.SetActive(true);//Footer Object

        //When this footer button is needed, only if the footer text and action
has some value, this component will be active... Implemented this only for
header & footer, as header & footer is used for all types of modal screen
layouts.
        if (needFooterNegativeButton && footerNegativeButtonText != null &&
footerNegativeButtonText != "" && negativeActionOption != null)
        {

_introModalScreenPanelImageFooterImageFooterNegativeButton.gameObject.SetActive(ne
Negative Button

            if
(_introModalScreenPanelImageFooterImageFooterNegativeButton.gameObject.activeSelf)
            {

_introModalScreenPanelImageFooterImageFooterNegativeButtonText.text =
footerNegativeButtonText;//Footer Negative Button Text

_introModalScreenPanelImageFooterImageFooterNegativeButton.onClick.AddListener(new
UnityEngine.Events.UnityAction(negativeActionOption));//Add listener to the
button's OnClick
            }
        }
        else
        {

_introModalScreenPanelImageFooterImageFooterNegativeButton.gameObject.SetActive(ne
Negative Button
        }

        //When this footer button is needed, only if the footer text and action
has some value, this component will be active... Implemented this only for
header & footer, as header & footer is used for all types of modal screen
layouts.
        if (needFooterAlternateButton && footerAlternateButtonText != null &&
footerAlternateButtonText != "" && alternateActionOption != null)
        {

_introModalScreenPanelImageFooterImageFooterAlternateButton.gameObject.SetActive(n
Alternate Button

            if
(_introModalScreenPanelImageFooterImageFooterAlternateButton.gameObject.activeSelf
            {

_introModalScreenPanelImageFooterImageFooterAlternateButtonText.text =
footerAlternateButtonText;//Footer Alternate Button Text

_introModalScreenPanelImageFooterImageFooterAlternateButton.onClick.AddListener(ne
UnityEngine.Events.UnityAction(alternateActionOption));//Add listener to the
button's OnClick
            }
        }
        else
        {

_introModalScreenPanelImageFooterImageFooterAlternateButton.gameObject.SetActive(n
Alternate Button
        }

        //When this footer button is needed, only if the footer text and action
has some value, this component will be active... Implemented this only for
header & footer, as header & footer is used for all types of modal screen
layouts.
        if (needFooterPositiveButton && footerPositiveButtonText != null &&
footerPositiveButtonText != "" && positiveActionOption != null)
        {

_introModalScreenPanelImageFooterImageFooterPositiveButton.gameObject.SetActive(ne
Positive Button

            if
(_introModalScreenPanelImageFooterImageFooterPositiveButton.gameObject.activeSelf)
            {

_introModalScreenPanelImageFooterImageFooterPositiveButtonText.text =
footerPositiveButtonText;//Footer Positive Button Text

_introModalScreenPanelImageFooterImageFooterPositiveButton.onClick.AddListener(new
UnityEngine.Events.UnityAction(positiveActionOption));//Add listener to the
button's OnClick
            }
        }
        else
        {

_introModalScreenPanelImageFooterImageFooterPositiveButton.gameObject.SetActive(ne
Positive Button
        }
    }

    void DeactivateFooter()
    {
        #region COMMENTED PART - NEDD TO RESOLVE LATER - KEPT FOR FUTURE
UPGRADE...
        //To make the button not be in selected state after clicking once: is
to set it's Navigation drop down option from inspector to none.
        //Or can do it by code as done for each button below.
        //Unfortunately, I tried, but it doesn't work that way. Maybe I did it
wrong. For now lets do the inspector method.
        //https://forum.unity.com/threads/clicking-a-button-leaves-it-in-
mouseover-state.285167/
        #endregion


_introModalScreenPanelImageFooterImageFooterNegativeButton.onClick.RemoveAllListen
all listeners to the button's OnClick
        //var nav =
_introModalScreenPanelImageFooterImageFooterNegativeButton.navigation;
        //nav.mode = Navigation.Mode.None;
        if
(_introModalScreenPanelImageFooterImageFooterNegativeButton.gameObject.activeSelf)
        {
```

```
_introModalScreenPanelImageFooterImageFooterNegativeButton.gameObject.SetActive(fa
Negative Button
        }


_introModalScreenPanelImageFooterImageFooterAlternateButton.onClick.RemoveAllListe
all listeners to the button's OnClick
        //nav =
_introModalScreenPanelImageFooterImageFooterAlternateButton.navigation;
        //nav.mode = Navigation.Mode.None;
        if
(_introModalScreenPanelImageFooterImageFooterAlternateButton.gameObject.activeSelf
        {

_introModalScreenPanelImageFooterImageFooterAlternateButton.gameObject.SetActive(f
Alternate Button
        }


_introModalScreenPanelImageFooterImageFooterPositiveButton.onClick.RemoveAllListen
all listeners to the button's OnClick
        //nav =
_introModalScreenPanelImageFooterImageFooterPositiveButton.navigation;
        //nav.mode = Navigation.Mode.None;
        if
(_introModalScreenPanelImageFooterImageFooterPositiveButton.gameObject.activeSelf)
        {

_introModalScreenPanelImageFooterImageFooterPositiveButton.gameObject.SetActive(fa
Positive Button
        }

        if (_introModalScreenFooter.activeSelf)
        {
            _introModalScreenFooter.SetActive(false);//Footer Object
        }
    }
}
```

c# unity-game-engine gameobject

Share

Improve this question

Follow

edited Dec 1, 2022 at 21:53

asked Oct 26, 2022 at 19:14

Aravind Shanmugam

31 ● 5

My first guess is as theres no code. That awake is running before the item its looking for is finished being made – BugFinder Oct 26, 2022 at 19:32

@BugFinder My bad. Added script now. However, I quite didn't understand your first guess. Could you please explain a bit in detail? – Aravind Shanmugam Oct 26, 2022 at 20:38 ✎

Objects dont all appear instantly. They are made. In order so as awake is almost the constructor. Perhaps the item ots referring to isnt made yet. – BugFinder Oct 26, 2022 at 23:04

3 Answers

Sorted by: Highest score (default)

▲
0
▼

The order in which scripts are run is not guaranteed, and are very likely to be different between editor and builds.

The only guarantee you have is that all Awakes() will run before first Start(), use it to your advantage. As a rule of thumb, initialise the object in Awake, but only interact with other objects in Start() and never in Awake() as other objects might not have been initialised yet, especially if you use Singletons.

It is likely that, because your script start order is different in build,

Share Improve this answer Follow

answered Oct 27, 2022 at 10:14
zambari
5,025 ● 1 ● 13 ● 22

Thank you for your insight @zambari. However, I still have this issue haunting me. Recently I noticed that when I do a 'Build and Run' instead of just 'Build' or 'Clean Build' the game builds and runs as expected with no issues the first time. But, If I quit the game and again launch it using the .exe from build folder, the same issue continues. – Aravind Shanmugam Dec 1, 2022 at 21:18

▲
0
▼

Open up Edit/Project Settings and go to Script Execution Order

Add all of these scripts to the list, and put them in the order that you need to them run in. Then, test it in the editor. It should now behave the same way in editor as in the build, so if it works in the editor it should work in the build. If it doesn't work, change the orders about first.

Share Improve this answer Follow

answered Oct 27, 2022 at 16:41
deek0146
984 ● 6 ● 21

Thank you for your insight @deek0146. I tried to work with the Script Execution Order to put them in the order that is expected to run. However, I still have this issue haunting me. Recently I noticed that when I do a 'Build and Run' instead of just 'Build' or 'Clean Build' the game builds and runs as expected with no issues the first time. But, If I quit the game and again launch it using the .exe from build folder, the same issue continues.
– Aravind Shanmugam Dec 1, 2022 at 21:21

0

Currently getting the same issue, the only way so far I've found for the .exe to work is to place the build of the game directly into my project folder.

Share Improve this answer Follow

answered Mar 10, 2023 at 11:05
Cmarsa
1

Your answer could be improved with additional supporting information. Please edit to add further details, such as citations or documentation, so that others can confirm that your answer is correct. You can find more information on how to write good answers in the help center. – Community Bot Mar 16, 2023 at 13:39