DTT
APPS WEB GAMES

SINCE 2010

Screen Rotation Management

Documentation | 19-05-2022

Table of Contents

1. Get started quickly

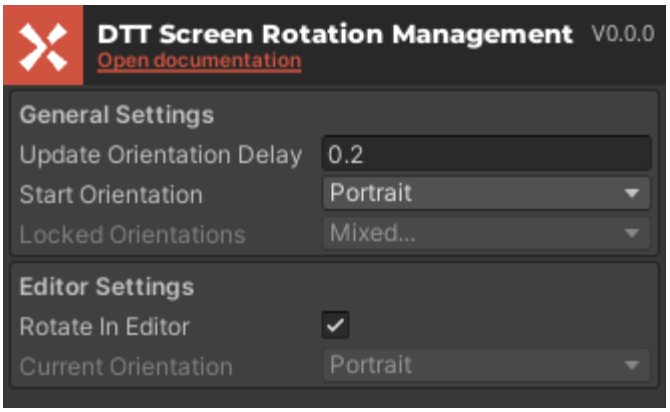
To get started, go to 'Tools/DTT/Screen Rotation Management/Window'. Here you'll find the general and editor settings of the asset.

Now to rotate the screen in code, add the 'DTT.ScreenRotationManagement' namespace to your script. Call **ScreenRotationManager.SetOrientation** to set the orientation of the screen to a certain **ScreenOrientation**. To toggle automatic rotation on, set **ScreenRotationManager.AutoRotate** to true. With this you can start implementing screen rotation into your project!

2. Introduction

DTT Screen Rotation Management allows you to easily manage and implement screen rotations into your project. Using the editor window, you can configure the settings for screen rotations in your project. The package also allows you to test screen rotations in the Unity editor by rotating the game view when the orientation changes in playmode.

3. Set-Up

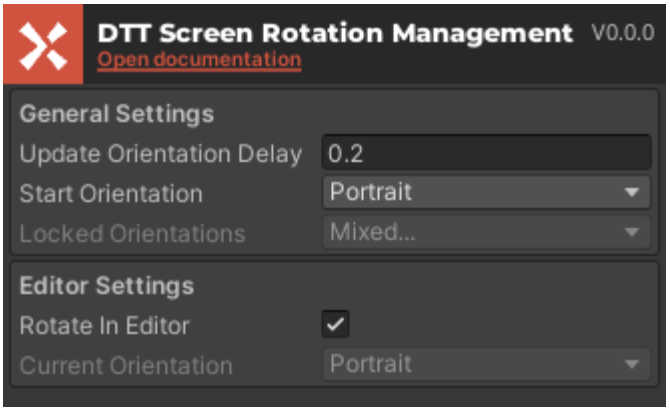


Start off by opening the editor window of the asset under 'Tools/DTT/Screen Rotation Management/Window'. Here you'll find the settings for screen rotation in your project. Under the General Settings you can set the time interval at which screen rotations are checked, the starting orientation of the application and the locked orientations for when automatic rotation is toggled on. Under Editor Settings you can toggle on whether you want the editor game view to rotate when the screen orientation changes during play mode. This can be used to test your UI. You can also set the orientation yourself from the editor window by changing the Current Orientation.

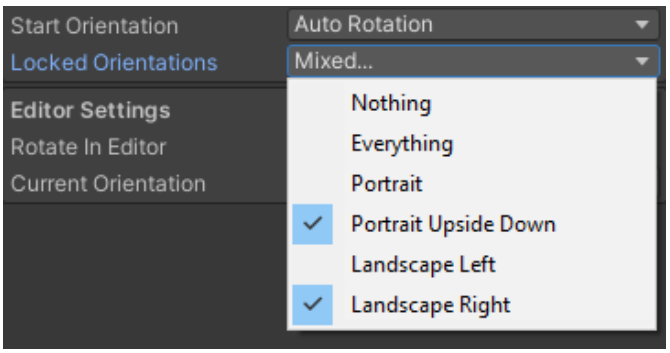
Now to rotate the screen from code, open a script and add the 'DTT.ScreenRotationManagement' namespace to the script. This allows you to call the **ScreenRotationManager**. To rotate the screen to a set orientation, use **ScreenRotationManager.SetOrientation** and pass in a **ScreenOrientation**. To set the orientation to automatic, set **ScreenRotationManager.AutoRotate** to true. If you want to lock certain orientations when automatic rotation is on, call **ScreenRotationManager.LockOrientations** and pass in the orientations you want to lock.

NOTE: Setting the locked orientations overrides the last locked orientations.

4. Editor



1. *Update Orientation Delay*
   The intervals in which the screen orientation is checked. In this case, every 0.2 seconds the **ScreenRotationWorker** will check if the orientation of the screen has changed.
2. *Start Orientation*
   Set the starting orientation of the application.



3. *Locked Orientations*
   This field is enabled when the *Start Orientation* is set to 'Auto Rotation'. This field allows you to select certain orientations you want to lock when automatic rotation is on.
4. *Rotate In Editor*
   Allows you to toggle the editor game view rotation on and off. When toggled on, the game view will rotate when the orientation changes to portrait or landscape.
5. *Current Orientation*
   This field is enabled when the editor is in play mode and *Rotate In Editor* is toggled on. It displays the current active orientation and allows you to set the screen orientation to test different orientations.

5. API

ScreenRotationManager

```
private void Start()
{
    // Allows you to set the orientation of the screen.
    // NOTE: This turns automatic rotation off,
    // unless ScreenOrientation.AutoRotation passed in.
    ScreenRotationManager.SetOrientation(ScreenOrientation.LandscapeLeft);

    // Turns automatic rotation on/off.
    ScreenRotationManager.AutoRotate = false;

    // Locks the orientations passed in the parameters
    // from being rotated to when automatic rotation is on.
    ScreenRotationManager.LockOrientations(ScreenOrientation.LandscapeRight,
        ScreenOrientation.PortraitUpsideDown);

    // Locks the opposite orientations of the passed orientation.
    // In this case the Portrait and Portrait Upside Down orientations will be locked.
    ScreenRotationManager.LockOppositeOrientations(ScreenOrientation.LandscapeLeft);

    // Unlocks all orientations.
    ScreenRotationManager.UnLockOrientations();

    // Gets the current orientation of the screen.
    ScreenOrientation currentOrientation = ScreenRotationManager.CurrentOrientation;

    // This event is invoked when the orientation of the screen has changed.
    ScreenRotationManager.ScreenRotated += orientation =>
    {
        // Do stuff on a screen rotation...
    };
}
```

6. Known Limitations

- **Opposite Orientation Change:** When automatic rotation is on and the user rotates their phone from portrait straight to portrait upside down (without first rotating to any landscape rotation), Unity doesn't detect an orientation change. This means the current orientation could be portrait, even though it's really portrait upside down (this applies for landscape left to landscape right as well).

7. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out. Always feel free to contact us at:

unity-support@d-tt.nl

(We typically respond within 1-2 business days)

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

DTT stands for Doing Things Together

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

https://www.d-tt.nl/en/