Magic Splitscreen System

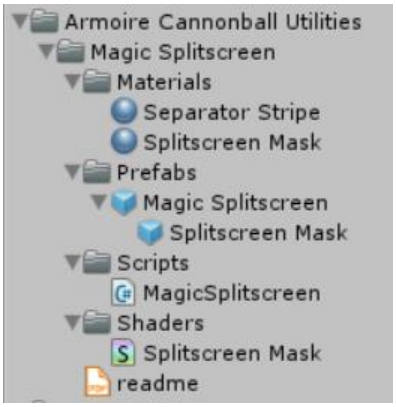
Thanks for downloading the Magic Splitscreen system from Armoire Cannonball. This package allows you to add a dynamically controlled splitscreen for showing two targets to your Unity project, and it works for both the Free and Pro licenses.

When two targets are close enough to each other to show on a single screen, they are shown on the same screen with no split. Once they move far enough apart, the screen dynamically splits to track both targets separately, keeping them both visible with the separation showing their relative directions. The effect is similar to the two-player splitscreen behavior in the LEGO adventure video games.

Now, let's cover what you've got here and how to use it.

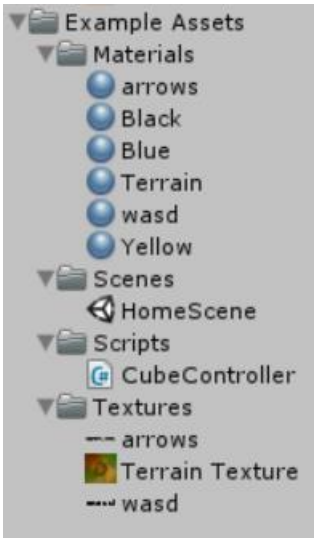
Tour of the Package

The Magic Splitscreen System



- Magic Splitscreen prefab: the primary control for the whole Magic Splitscreen system. It consists of the MagicSplitscreen.cs script, which does the Magic, the Splitscreen Mask mesh, and the Separator Stripe.
- Splitscreen Mask mesh: a small plane which is positioned in front of the secondary camera when the screen is split that uses the Splitscreen Mask shader and material to prevent the secondary camera from rendering on the whole screen. By default, the Mask is set to Layer 31, which has been named "SplitscreenMask." You are free to set this to a different, otherwise unused Layer, and the Magic Splitscreen will still work properly.
- Separator Stripe: a small stripe that appears along the division between the screens when split. It uses the Separator Stripe material, allowing you to change the color of the stripe.
- Readme.pdf: this document (which can also be found on our website at http://armoirecannonball.com/unitydemos/MagicSplitscreen.pdf).

The Example Project



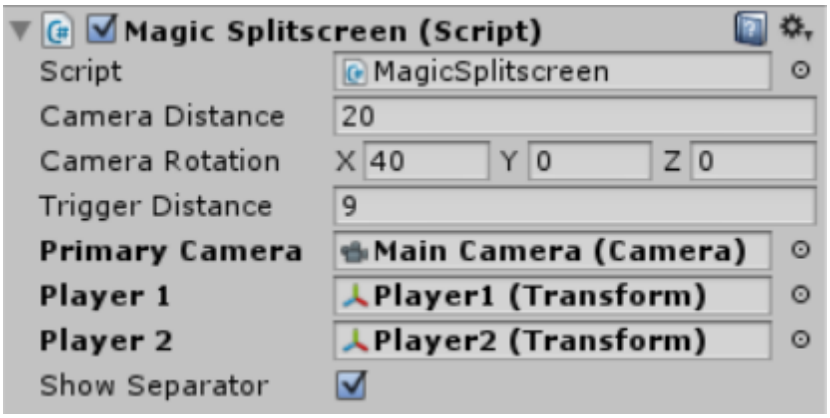
- HomeScene: a scene with the example implementation of the Magic Splitscreen system. It consists of two player cubes, an arena for the players to walk around in, a camera, and the Magic Splitscreen prefab.
- Some materials: simple materials to give the moving players and environment some distinctive colors.
- Some textures: colors for the terrain and textures for quads above the players to show their controls ("WASD" or "Arrows").
- CubeController.cs script: a very basic movement script for the two player objects that allow them to move around the arena.

You may notice that the Input axes have been set to allow two player controls in this project. They are only set to make it easier to show the product. You can configure Input however

you'd like for your project without having to change anything in the Magic Splitscreen scripts or settings.

Setting Up Magic Splitscreen For Your Project

Using the Magic Splitscreen is very simple. Once you have two objects moving around your world that you want to track, add the Magic Splitscreen prefab to your scene, set a few values, and you're ready to go.



- Camera Distance: set how far away from the target(s) the camera(s) will be.
- Camera Rotation: set the pitch, yaw, and roll of the camera (in degrees).
- Trigger Distance: how far apart the targets need to be to split the screen. If the targets are closer than this distance, the screen won't split. If they are farther apart, it will split to track both targets.
- Primary Camera: set this to your main camera. Its configuration will be used to create the secondary camera while the screen is split. Note that if there is an AudioListener attached to this camera (which there is by default), it will be removed and a separate one will be created at runtime.
- Player 1: set this to the Transform on one of the players.
- Player 2: set this to the Transform on the other player.
- Show Separator: check this to show a stripe along the separation between screens when it is split.

Updating Camera Movement

If you'd like to update the precise behavior of the cameras, you can change the code in the MoveCamera function in MagicSplitscreen.cs. The cameras are always moved via this script whether the screen is split or not. The provided logic is currently very simple, but it does the job.

```
/// <summary>
/// Move a camera to look at a specified position
/// </summary>
/// <param name="camera">The camera to move</param>
/// <param name="targetPos">The position for that camera to look at</param>
/// <remarks>This is the place to add specialized camera movement behavior if you want it.</remarks>
private void MoveCamera(Camera camera, Vector3 targetPos)
{
    camera.transform.localRotation = this.cameraQuaternion;
    camera.transform.position = targetPos - (camera.transform.forward * this.cameraDistance);
}
```

Known Limitations

The secondary camera uses the Depth Only clear flags, not the Skybox clear flags, to properly display views of both targets. As a result, your game won't look right if you use the Unity skybox and it becomes visible while the screen is split. If you intend for the sky to be visible, you will need to create your own different skybox or skydome solution instead.

The AudioListener is placed equally between both players when they separate. If you're using 3D sounds and the players get far apart, the sounds may be very hard to hear. We suggest the use of 2D sounds if players can travel far apart from each other.

Currently, changing the orientation of the camera during play isn't immediately supported with the code as it is.

Contact Us

If you have any questions about using the Magic Splitscreen, want to report an issue, make a request for something we should update this tool with, or feel the need to tell us how awesome we are, email us at support@armoirecannonball.com.

If you want to see what else we're up to, check us out online on ArmoireCannonball.com, Facebook, Twitter, or YouTube.

Version History

Version 1.0 (May 2014)

- Initial release

Version 1.1 (August 2014)

- Added tooltips to the Inspector fields
- Added the separator stripe

Version 1.2 (October 2014)

- Updated Splitscreen Mask to automatically resize during runtime to fit any arbitrary camera (including proper support for Orthographic cameras)
- Added support to make the asset work in Unity 4.3.0 (down from Unity 4.5.1)