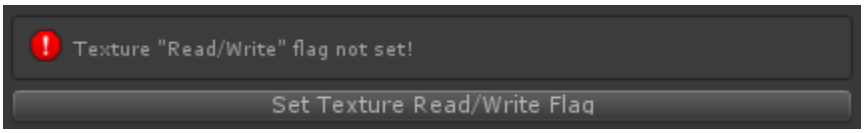
Sprite Shatter (Version 1.3.0)

Sprite Shatter provides a quick and easy way to make your 2D sprites "shatter" into numerous configurable pieces. Using the simple user interface, any game object with an associated sprite renderer can have its sprite cut into any number of pieces, which are created as separate game objects at run-time. A single method call then shatters the sprite. Sprite Shatter comes will full source code (including editor code) and, as of version 1.2.0, full prefab support.

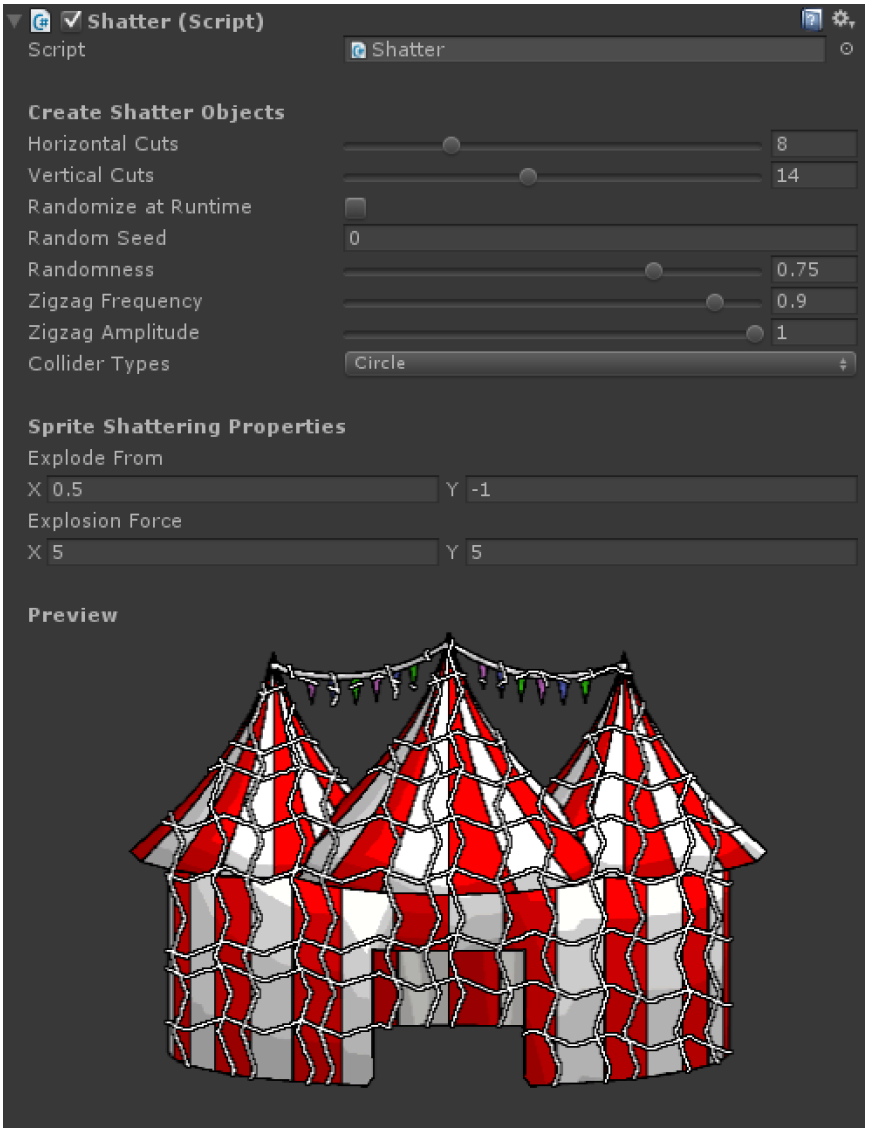
Below are instructions on how the example scene included in the package (the exploding circus!) has been set up. Feel free to modify this scene – you can create a separate sprite and follow the instructions for that, or you can modify the existing circus sprite. First, select the game object that contains the sprite renderer component for your sprite, and add a **Shatter** game component. Note that the **Shatter** game component requires a sprite renderer, so one will be added to the game object if it is not already present.

After a sprite is assigned to the sprite renderer, the various Sprite Shatter properties will then be displayed in the inspector window under the new **Shatter** component. Note that if you see the following…



…you will need to set the Read/Write flag on the sprite's texture. This is because Sprite Shatter requires access to the sprite's pixel data, which is only available if this flag is set. You can click the **Set Texture Read/Write Flag** button to achieve this – doing so will automatically change the flag on the asset and reimport it, causing the error message to disappear.

The Inspector window should now look something like this:



The various settings allow you to configure how your sprite should be shattered – for example you can specify how many pieces, how random the pieces should look, whether they should have a zigzag shape, and what type of colliders, if any, should be applied to them. You can hover over any of the fields to get a detailed explanation of what that field does. To summarize the fields:

- **Horizontal Cuts** – the number of cuts to make horizontally across the sprite. Note that for horizontal and vertical cuts, more cuts mean more (smaller) pieces but too many pieces will adversely affect performance, particularly if colliders are included.
- **Vertical Cuts** – the number of cuts to make vertically down the sprite.
- **Randomize at Runtime/Random Seed** – you can specify a random seed to use when generating random numbers. If you have randomness applied, you can change this value to get an arrangement of cuts that you are happy with. The same random seed will always produce the same cuts on the same sprite. Alternatively, you can check **Randomize at Runtime** to generate a different random seed, and hence a different arrangement of cuts, every time the game runs.

- **Randomness** – how much the horizontal and vertical lines are randomly moved. A larger value produces a greater range of sizes of shattered pieces, whereas a value of zero produces uniform pieces.
- **Zigzag Frequency** – combined with zigzag amplitude, generates a zigzag effect on the cuts to give more jagged-edged pieces.
- **Zigzag Amplitude** – see zigzag frequency. You can play around with these two settings to get a decent-looking zigzag effect.
- **Collider Type** – how each shattered piece should collide with the terrain, if at all. The options are:
    - **None** – no collisions.
    - **Circle** – most efficient but not always a very accurate shape.
    - **Box**
    - **Polygon** – least efficient but the most accurate shape.

  If you have a lot of pieces, circle colliders are usually enough to create realistic shattered pieces. Sprite Shatter will optimize the size of circles and boxes to give the best collisions with the least overlap.
- **Explode From** – the position to explode the shattered pieces from (where in the X direction, 0 is the left of the sprite and 1 is the right, and in the Y direction 0 is the bottom of the sprite and 1 is the top). For example to explode the pieces out from the centre, set this to (0.5, 0.5).
- **Explode Force** – the force with which to explode the shattered pieces. A greater force will make them fly further away, and a force of zero will make them just fall where they are.

Below these fields is a preview of how the sprite will be shattered. Altering the parameters above will update this preview in real-time.

When you are happy with the Sprite Shatter settings, it is as simple as calling **shatter()** or **reset()** at run-time on the **Shatter** component that is automatically attached to the sprite. To illustrate this, the circus example calls **shatter()** if you left-click anywhere in the scene and **reset()** if you right-click.