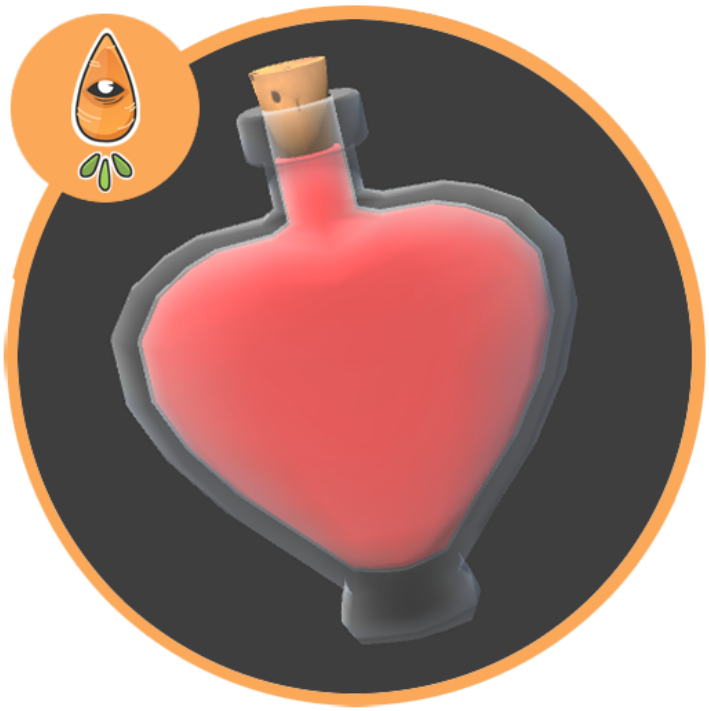


Magic Potions with Liquid Simulation

V2.0 - November 2017

Index

New Features in v2.0

- Added legacy (mobile friendly) liquid shader for potion liquid objects
- Added legacy potion demo scene to demonstrate how legacy shader works/ looks
- Added ‘workshop’ demo scene where users can visualise the potions working in an environment. Pick up and drop potions to see how they shatter, slosh around.
- Cleaned up project materials
- Cleaned up prefab bottles folder
- Made demo folder structure cleaner for users to delete when looking to clear out the demo related content

Magic Potions - Overview

This asset pack has been developed to allow you to easily integrate breakable potions with liquid simulation into your next Unity game project. Taking the prefabs we have developed you will be able to quickly drag and drop these potions into your project as environmental props, or interactable items for players to collect.

Secondly, this pack is designed to allow you to easily create your own potions making use of the shaders and scripts we have developed.

Included in v2.0 of this pack is:

- 8 potion bottles, set up for your use and to demonstrate functionality of pack
- Easy to use liquid shader - allows users to define liquid fill level and draws a ‘fill face’ where the liquid mesh is culled - possesses a texture channel to add detail to liquid surfaces as well as emission, smoothness and metallic sliders and colour pickers
- A mobile friendly ‘legacy’ shader for liquid meshes
- A glass shader with metallic, gloss, texture and normal channels to allow for unique bottle designs
- A ‘liquid volume animator’ script which allows users to simulate approximate liquid physics and define the behavior of the liquid within the predefined parameters
- A cork mesh and texture
- Bottle smash particle effects and broken glass meshes for each bottle
- A bottle smash script that allows users to manage a bottle smash event, determining which particle effects play at the time of breakage and how the liquid puddle renders on the ground plane below the smash location
- A workshop demo scene filled with assets

Demo Scene Overview

Workshop Demo
An interactive scene that demonstrates how the bottles can work within a game environment. Pick up potions, drop them around the scene and watch them smash against the furniture. Sloshing potions around will also demonstrate the liquid ‘physics’.

Potion Demo
The potion demo allows users to demonstrate the liquid ‘physics’ and smashing interactions close up. This is the best place to see the pack in its entirety.

Transparency Demo
The transparency demo displays how users can set liquid transparency and distortion/ what that looks like when the liquid volume is looked through.

Pouring Demo
The pouring demo shows how one liquid volume can be tipped and poured into another. It also demonstrated how liquids can blend together in one volume, mixing colour.

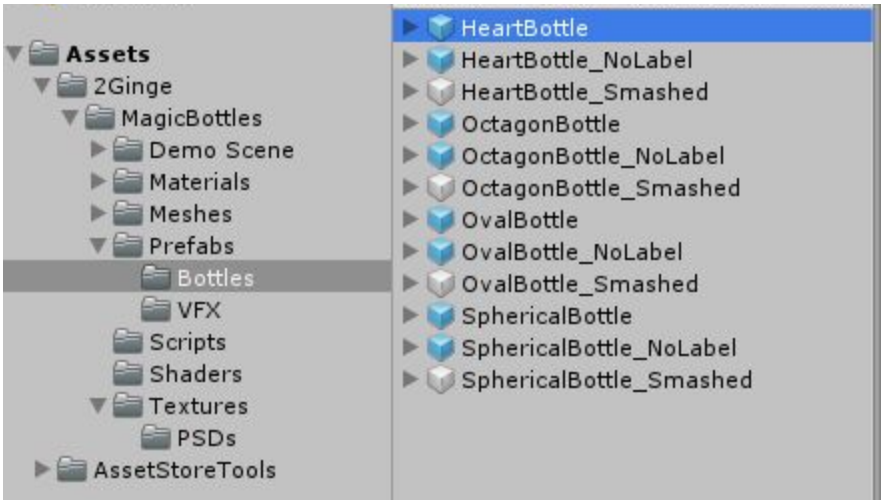
Legacy Demo
The legacy demo is a static scene that demonstrates how the legacy shader looks and contains a pre-made legacy bottle for users to look at the setup of.

Animated Texture Demo
The animated texture demo shows users how they can set up an animated texture on the surface of liquids using coffee foam as an example.

How to use Magic Potions - first time users

Importing pre made bottles

First time users can expect to find prefab bottles within the 2Ginge folder following import from the Unity Asset Store. These bottles are already set up to break when the change in velocity of the bottle object is greater than the predefined break velocity and simulate stylised liquid physics when in an unbroken state. Simply drag these prefabs into your scene and you’re ready to go!



If you are creating your own bottles you will require five components to mimic the effects and functionality of our pre made assets. You will require a bottle mesh, a liquid mesh, a cork mesh a label mesh and finally a series shattered glass objects. You can create these shattered segments manually, or perhaps look at implementing a mesh fracturing tool to do this for you.

Once you have the objects described above you should import them into your project and set them up in the scene using the hierarchy shown to the left. Ensure the smashed particles are turned off by default as the bottle smash script will enable these when the break conditions are met. Use the ‘glass_potion’ shader for the bottle mesh and the ‘Liquid_Potion’ shader for the liquid. Both of these can be found in the shader drop down under 2Ginge > Potion.

Most of you will be aware, but it bears mentioning that it is best practice to center your parent objects in world space while organising the initial hierarchy to ensure all objects are aligned correctly with one another.

At this point ensure that the necessary objects have rigidbodies and colliders set up. The bottle parent object should have a rigidbody attached, the bottle object should have a mesh collider and the smashed bottle parts and cork should have their own rigidbodies and mesh colliders in order to resolve their own physics after they are turned on. The liquid object and label do not require any colliders or rigidbodies as they will despawn on the smash event.

If you are generating your own smash particle effects, also include them in the setup described above underneath the bottle parent object, positioned correctly. The particle system will be activated by the bottle smash script and should remain turned on by default.

Once you have set up your objects in a hierarchy similar to that outlined above you are ready to attach the provided scripts and shaders to ensure correct functionality. The ‘BottleSmash.cs’ script should be attached to the bottle parent object and the ‘LiquidVolumeAnimator.cs’ script should be attached to the liquid object within your bottle.

Scripts Overview

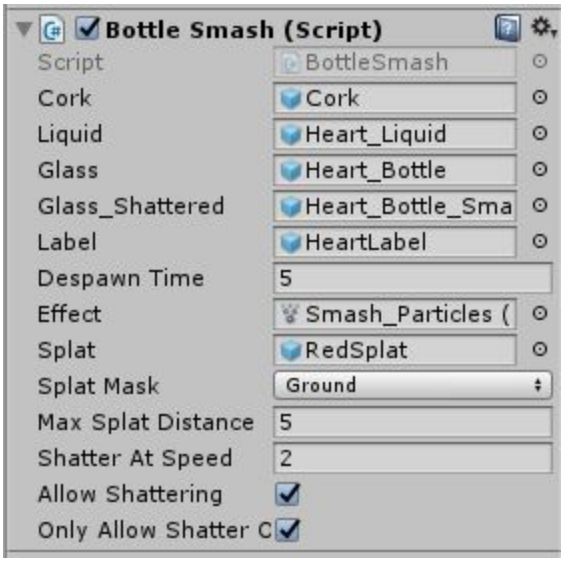
Described below are the functions of the two core scripts within the asset pack. If you are unclear on the functionality of either of these scripts and their many components please do not hesitate to contact us for assistance. You can find our support contact details at the bottom of this document.

BottleSmash.cs outline:

‘Cork’, ‘Liquid’, ‘Glass’ and ‘Label’ are all gameobjects to be destroyed. Cork is unique in that it will despawn after the despawn timer has completed, so be sure to have a disabled mesh collider with a rigidbody (with kinematic ticked) on it to take full effect.

Glass_Shattered is a disabled container full of the glass shards (each with their own rigidbody and mesh collider).

Despawn Time is how long parts of the mesh will stay around for (including the shattered glass).



Effect is the particle effect at the center of the bottle (splash effect essentially).
Splat is the mesh that gets instantiated on the ground.

Mask is the layer that the 'ground' exists on.

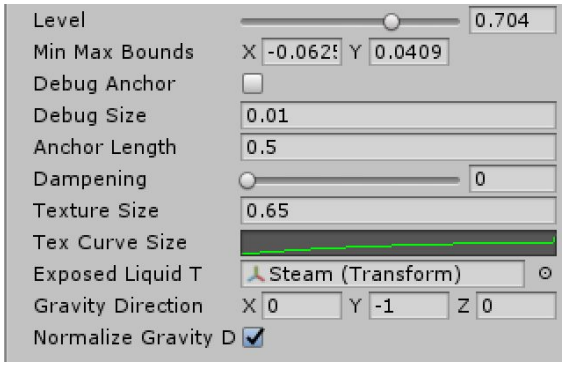
Splat Distance means that any ground that is greater distance than this, the splat will not occur .

ShatterAtSpeed is the speed at which the potion needs to be travelling in (subtracting its previous velocity) in order to 'break', think stopping very quickly or starting very quickly.

AllowShattering will disable any normal shattering logic (however it can still be shattered through code.

OnlyAllowShatterOnCollision means that there is a small window after colliding with another collider that the potion can shatter (0.2 seconds).

LiquidVolumeAnimator.cs Outline:



Level is the slider which fills or empties the mesh (0, being empty, 1 being full).

MinMaxBounds is used to debug if the volume is being approximated correctly.

Debug Anchor allows you to see the physicality of the liquid as a pendulum object.

Anchor Length will increase (closer to 0) and decrease (closer to positive infinity) the effect of rotation and movement on the bottle.

Dampening controls the intensity of liquid movement that will take place during the bottles rotation or transformation (1 will have no effect while 0 will have full effect).

Texture Size the size of the cutoff surface texture

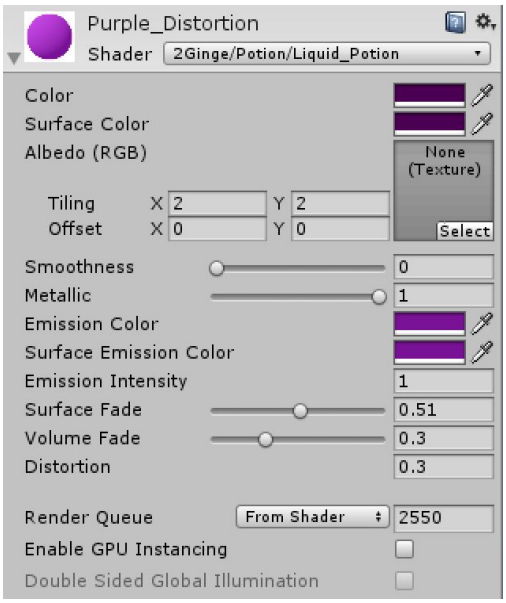
Texture Curve Size the texture curve graph allows users to set the texture size “keys” so that when the liquid volume empties and fills, the cutoff texture scales with the cutoff surface size at any level of fill. This can take some playing around with, but a good approach is set a start, middle and end key and fill in where scaling is not matching surface size where necessary.

Exposed Liquid T transform of particle effect where liquid is exposed (in the case of steam in coffee cup). Not required on normal bottle set ups.

Gravity Direction -1 = world down, 1 = world up, 0 = crazy rotation time. Set the direction of “gravity” that the liquid references or more simply the direction the pendulum debug object ‘hangs’.

Normalize Gravity On/ off

Shader Overview



Liquid_Potion Shader

Colour overall colour, can be overridden with code (displays current colour)

Surface Colour set the colour of the cut off surface independently from the remainder of the liquid volume

Albedo Liquid volume texture

Smoothness / Metallic set values as needed

Emission Colour if using emission, set colour for liquid volume here

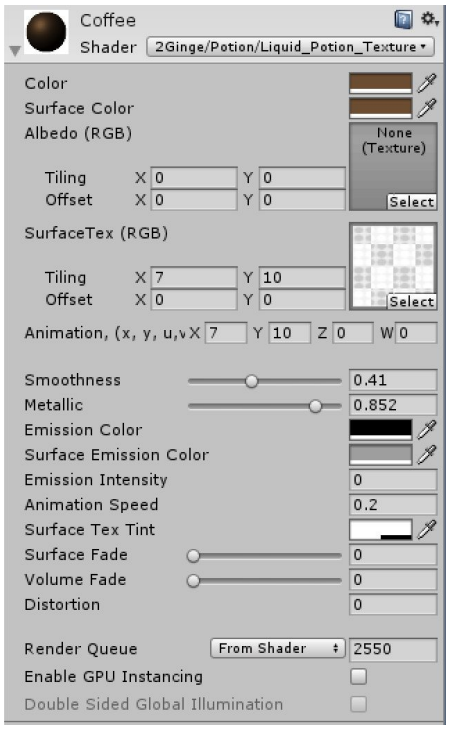
Surface Emission Colour if using cutoff surface emission, set colour value here

Emission Intensity set emission intensity for liquid volume total

Surface Fade set “opacity” of cutoff surface

Volume Fade set “opacity” of liquid volume (semi-transparent has subtle distortion on objects behind bottles when looking through liquid volume)

Distortion set the distortion value that determines how distorted objects appear through liquid



Liquid_Potion shader with cutoff texture

This shader is effectively the same as the above, except that is possesses a channel for a cutoff surface texture, as well as animation values if the texture is animated (bubbles popping, foam, etc…). Below the unique elements of this shader are listed, see above for any standard liquid shader features.

SurfaceTex the texture to display on the cutoff surface (size set on liquid animator script)

Tiling the tiling value for the cutoff surface texture

Animation the number of vertical and horizontal rows/ columns in the attached sprite sheet if texture is animated.

Animation Speed speed of cycling through animation frames

Legacy_Liquid_Potion

This shader is for use on lower end devices/ mobile and does not possess the surface texture elements, and additional maps/ variables that the more complicated ‘Liquid_Potion’ shader does.

Surface Emission Color simply the color of the cut-off surface emission

Volume Emission Color the color of the liquid volume emission

Emission Intensity the intensity of emission across the liquid volume and cut-off surface

Additional Help/ Contact

If you are having any issues in setting up your own bottles with the provided scripts and shaders, please watch the video demonstration which can be found on our Youtube channel. There you will also be able to find a video detailing the modelling practices put into place when creating the meshes required to create your own bottle assets.

Feel free to contact us with any issues you may be having via any channel. We are always happy to support our customers and will address bug fixes as soon as possible. Please do not hesitate to contact us with feature requests either! We'd love to continue to make our tools and assets better wherever possible.

Email: contact@2ginge.com
Website: www.2ginge.com
Twitter: @TwoGinge | @PezzSp | @JairMcBain

If you'd like to hear about our other projects and tools, please find our newsletter signup form at www.2ginge.com or check out our Unity Asset Store developer profile.