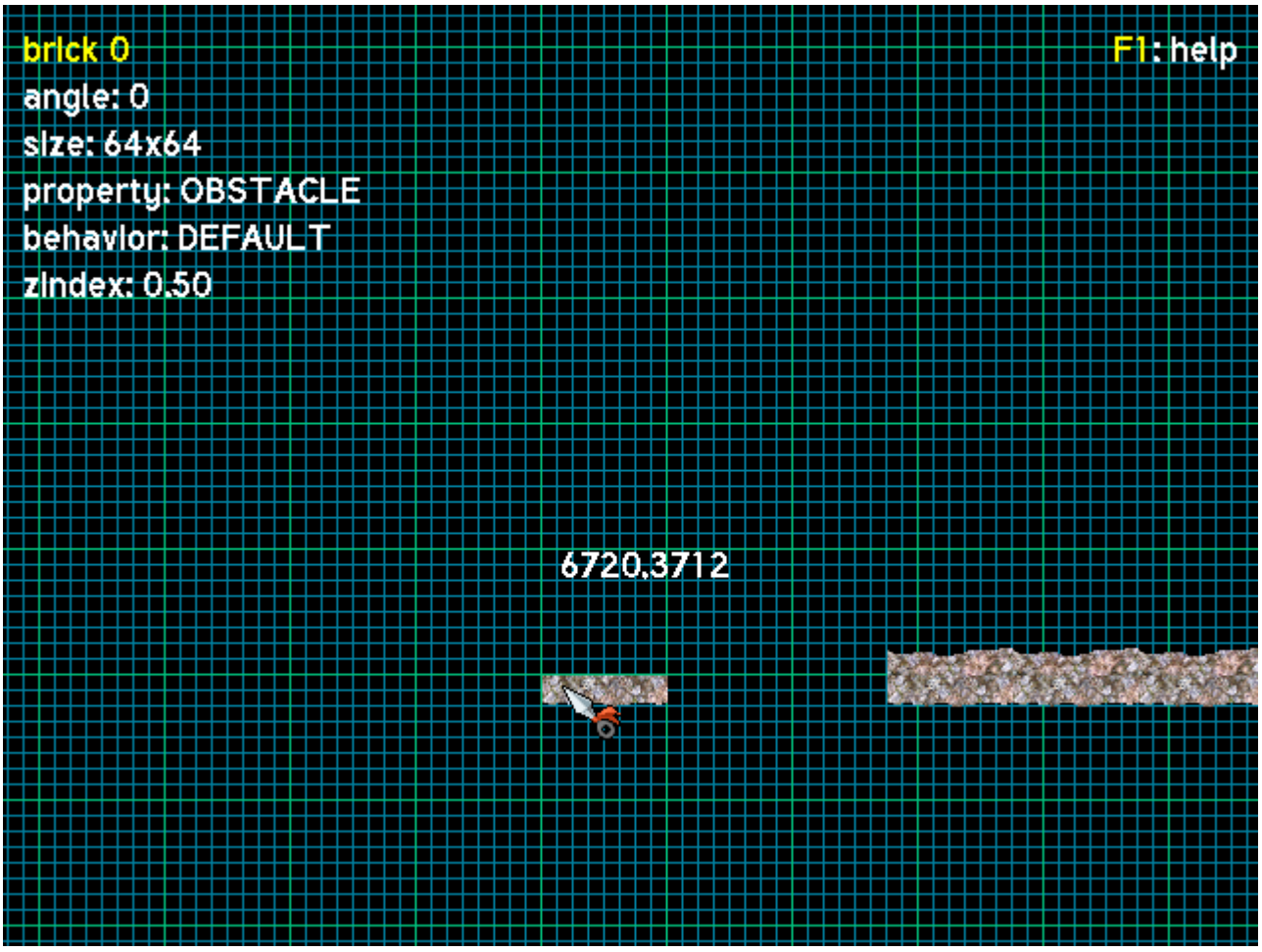
Building levels in 2.5D with Open Surge Engine

1. Paving the way

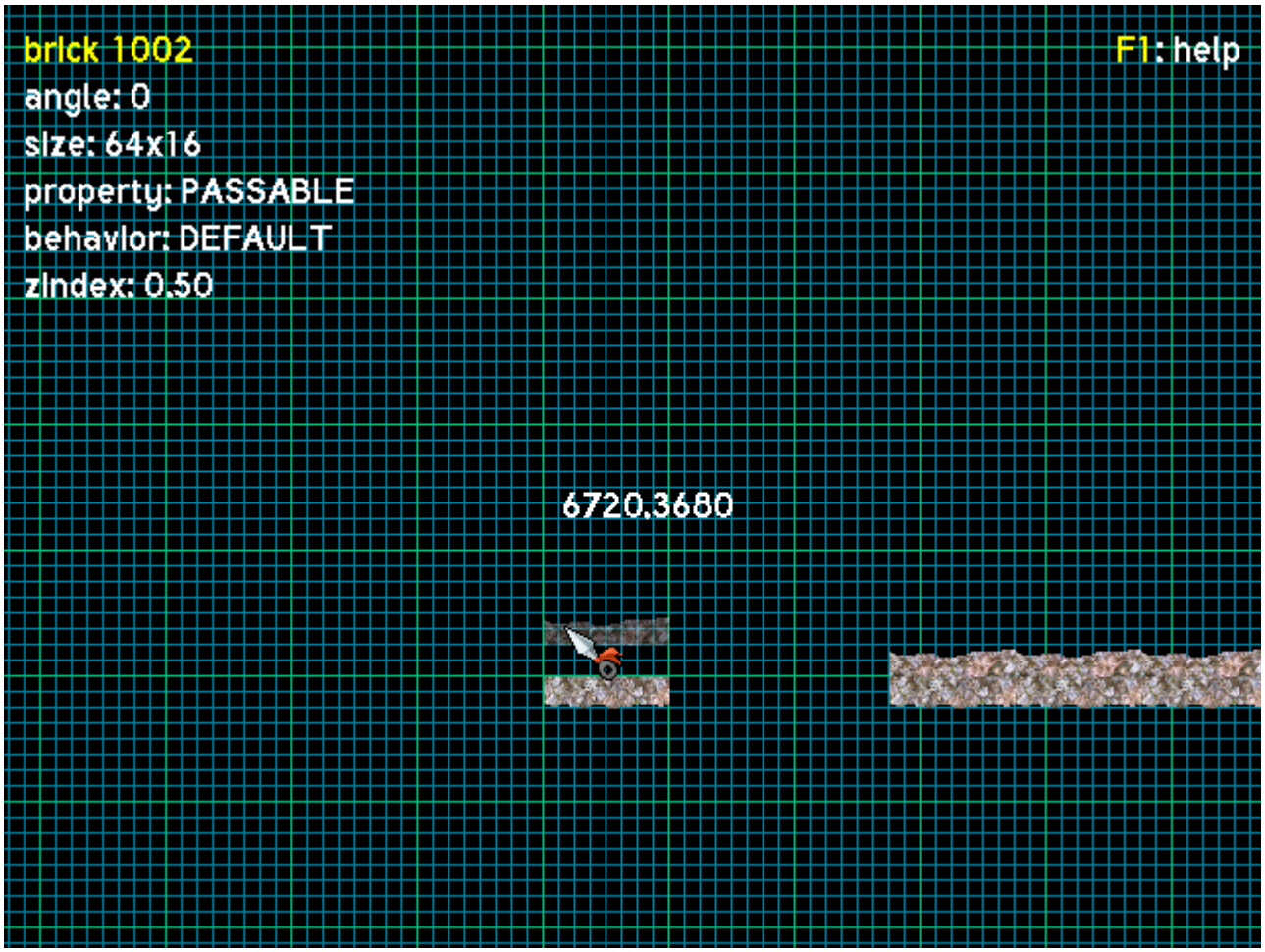
Having solid floor where actors can stand and walk on is crucial for a platformer. On a pure sidescroller, with a flat 2D view, it is easy to achieve since the edges of solids don't have a background continuity. In a 2.5D view, however, the depth illusion is based on how the solids match the background.

So, first, we find a solid brick (obstacle property). 0º is flat ground so it's a good start. You can see part of a completely built path in the right side.



This will be the actual base where actors will stand, so the smoother the contact side is, the better.

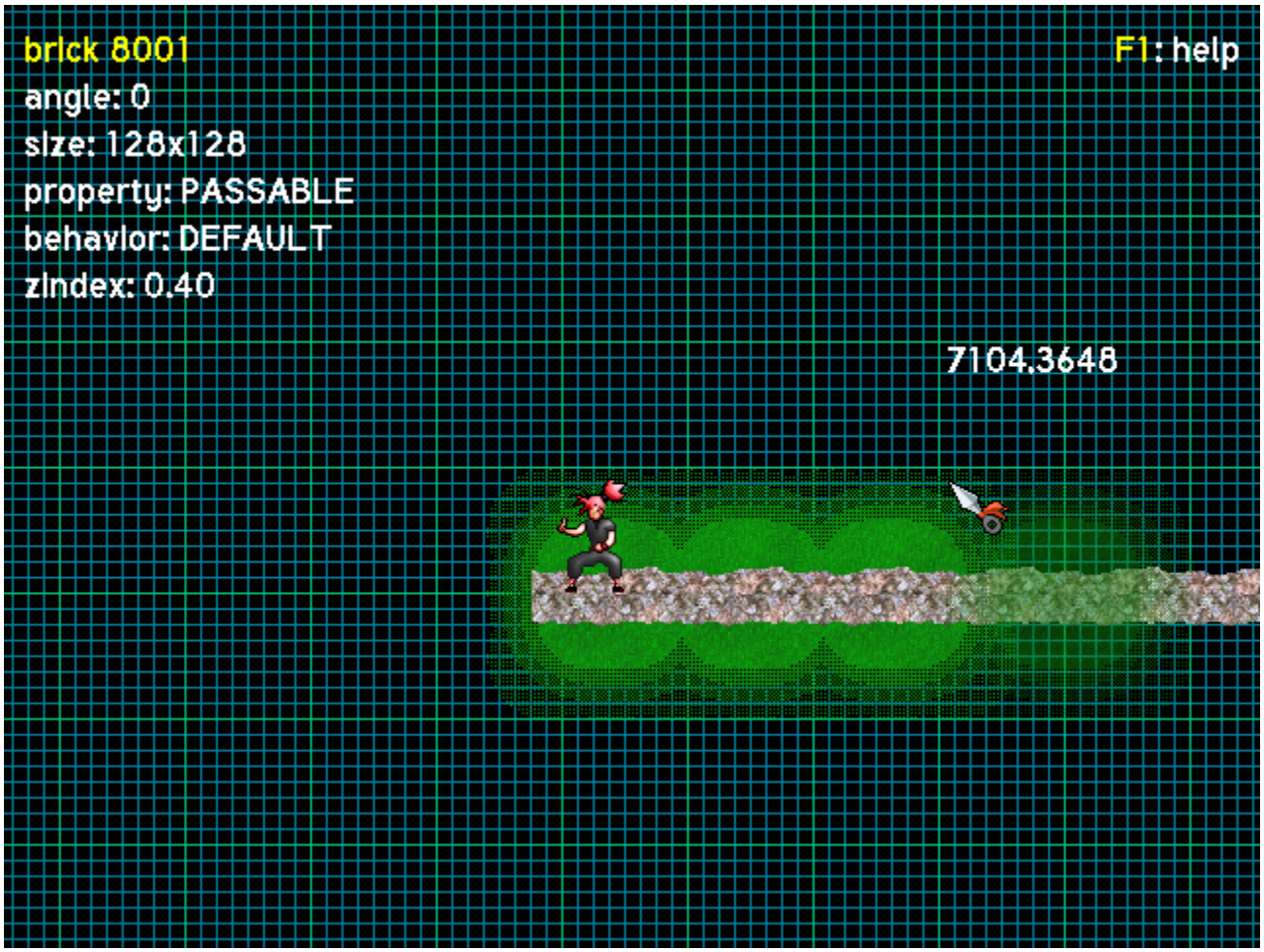
Next, we find the matching background piece.



As you can see, the property is *passable* which means it will never block you or any object.

2. Textures

Textures are only passable bricks, but the way they are made and used is what makes them so special.

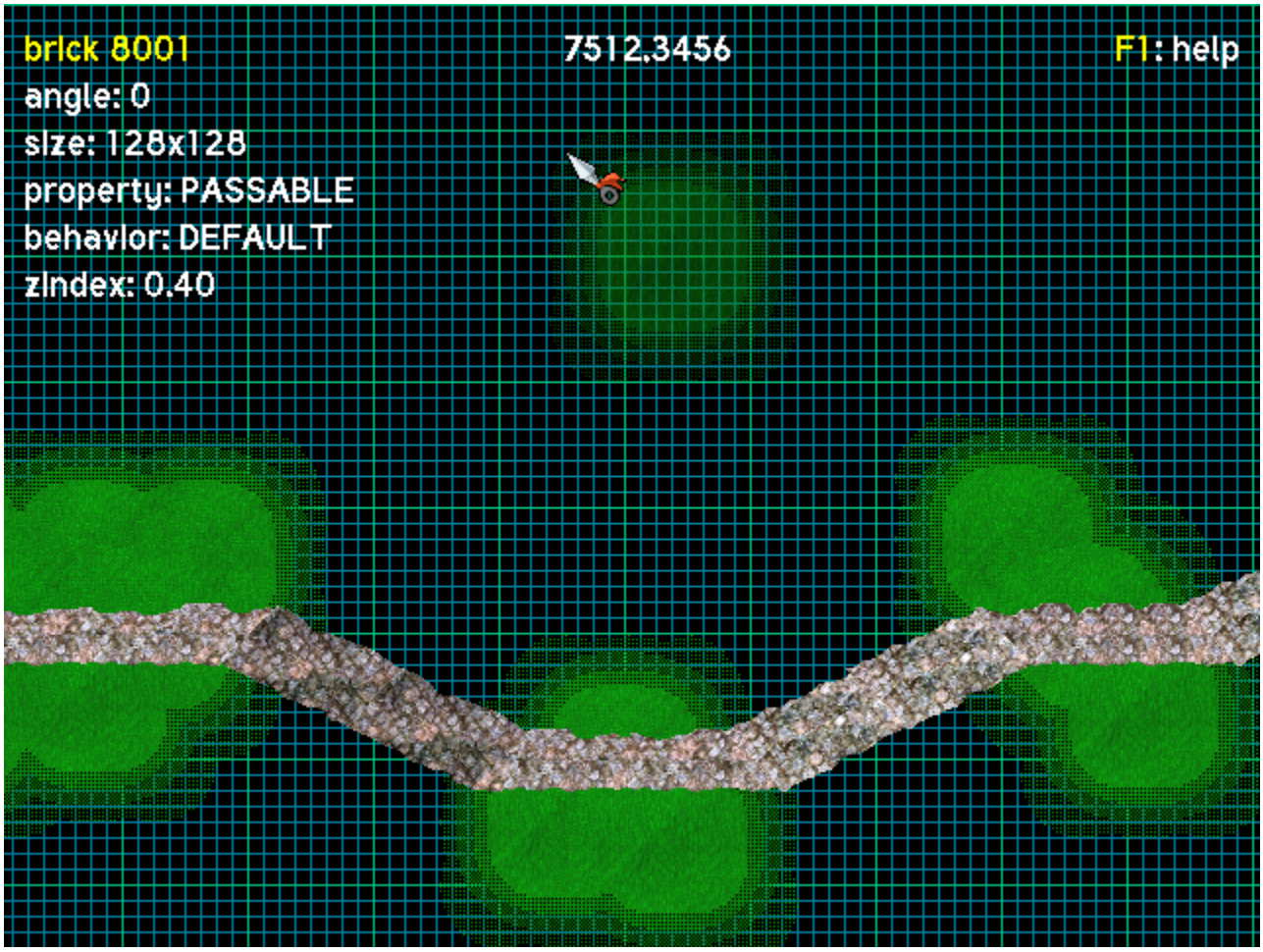


Basically, it is a passable brick, which has a z-index lower than 0.5, so it always appears as background. However they have fading edges, as seen in the image, which simulates a smooth transition between overlapped bricks.

Textures come in 4 shades, for easing the simulation of depth, only through lighting.

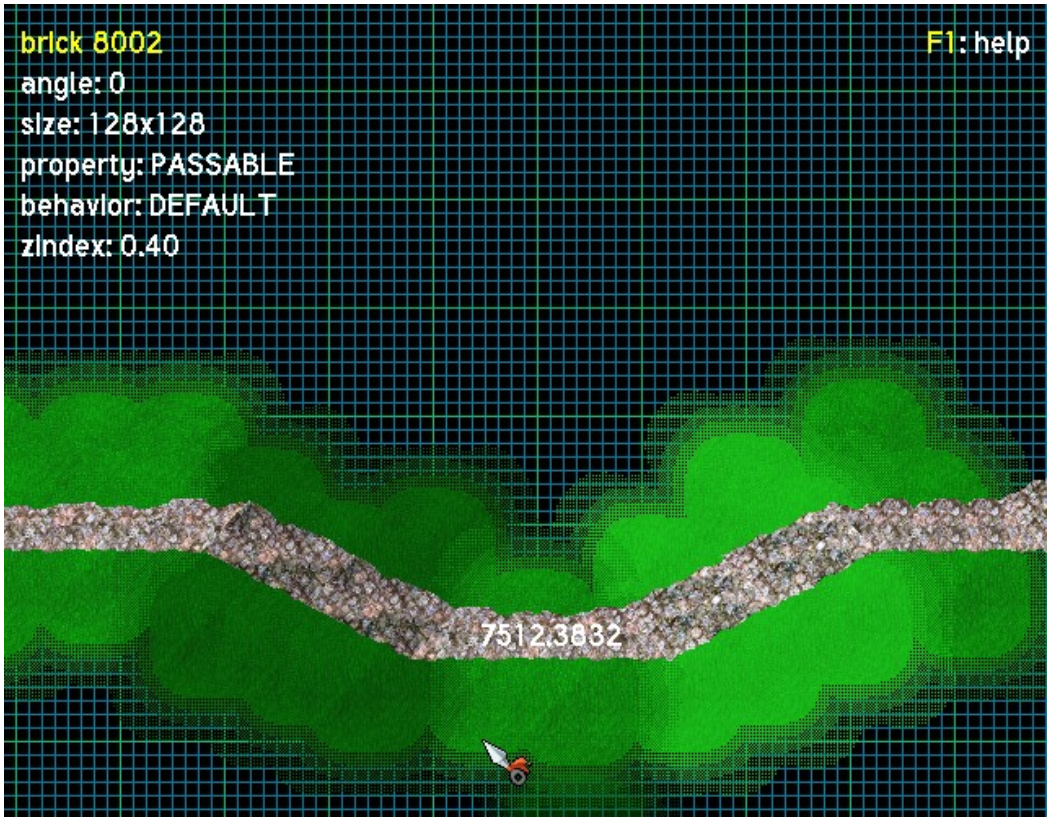
You may use the yellow layer to place a texture behind everything else, including other textures.

So, now that we found the correct shade for flat ground, we lay it out on flat spots.

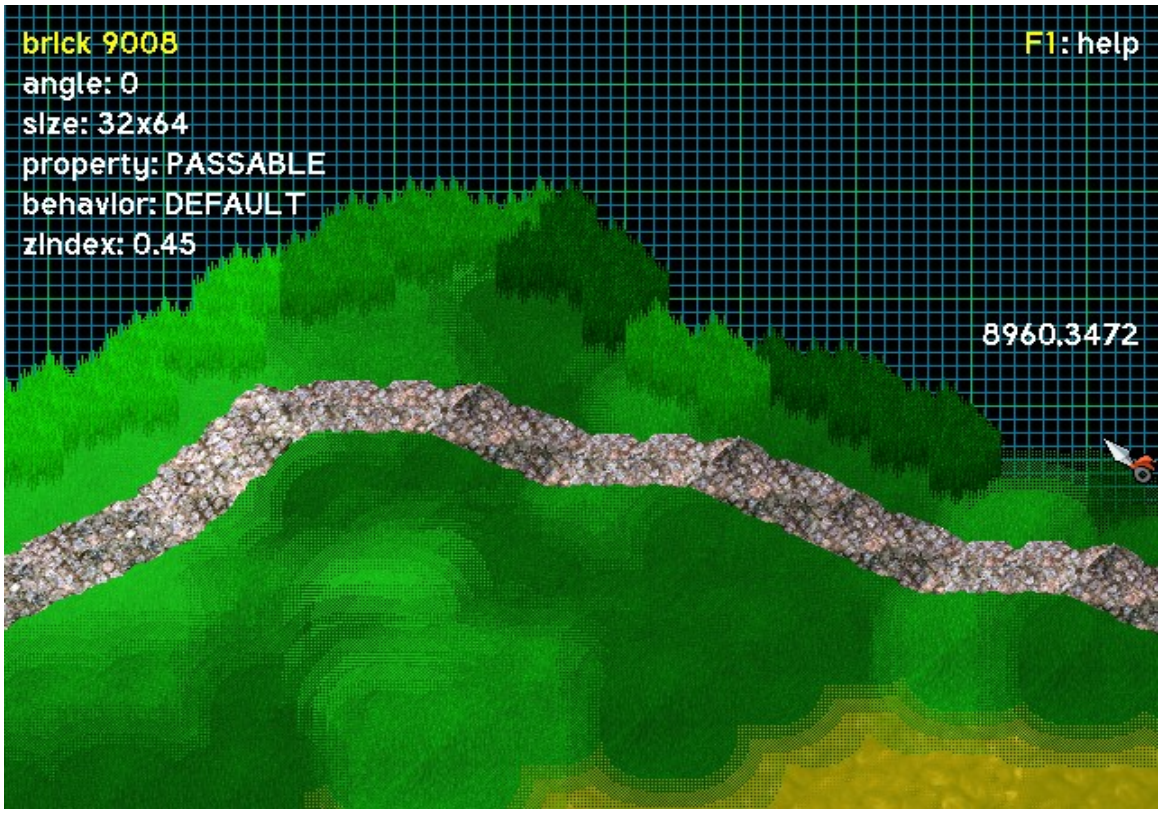


When creating large patches of the same texture, be careful with overlapping them only where they are fully opaque.

Next, is adding shades for slopes. Light on the right, dark on the left.



Add a bit more detail...



Notice that at the top edges, shaded grass is used to cover them. Shading some other non-texture passables is recommended to achieve better immersion. The image can't show it, but the grass is also animated, giving both the feeling of depth and a sense that there is some wind blowing.

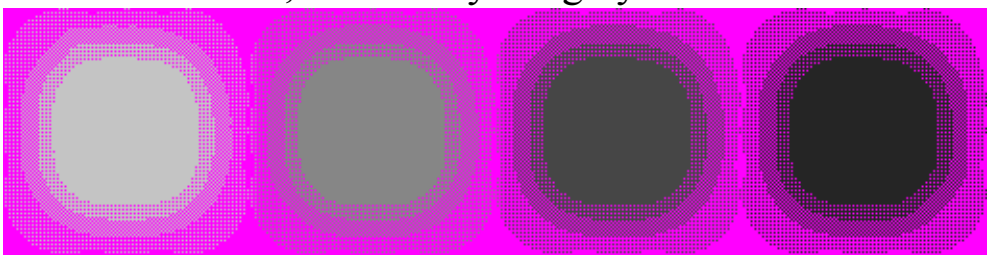
The inclusion of natural elements in the design of a level, makes it seem more... natural. Even if you are designing highly unnatural structures, accounting for lighting, wind, ambiance... it all plays out to draw the player in and make him/her feel some kind of emotional connection with your level.

So, here's what happens when you use multiple textures, and the overlapping methods previously described, with a few detail bricks:

3. Setting up the graphics

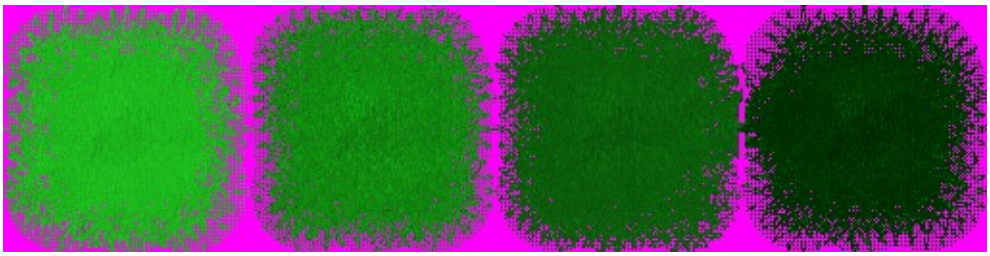
A. Textures

First, open your favorite graphics tool, and load this image. If that tool is able to do automatic selections, select only the gray areas.



Once you have isolated them from the magenta (transparency), you can start painting. If your graphics tool supports blending modes, you should use “hard light” or “multiply”. This will use the existing shade of gray and turn it to light or dark, while the texture you're painting is essentially the same in every shade.

When finished, it should look like this:



In this case you can see some grass blades “bleed” out of the opaque area. This is an intended effect that makes the grass textures look more natural when overlapped.

In the brickset script, you need to define each shade as an individual brick.

Refer to the brickset index to know how to number them for better sorting and less time searching in the editor.

B. Details

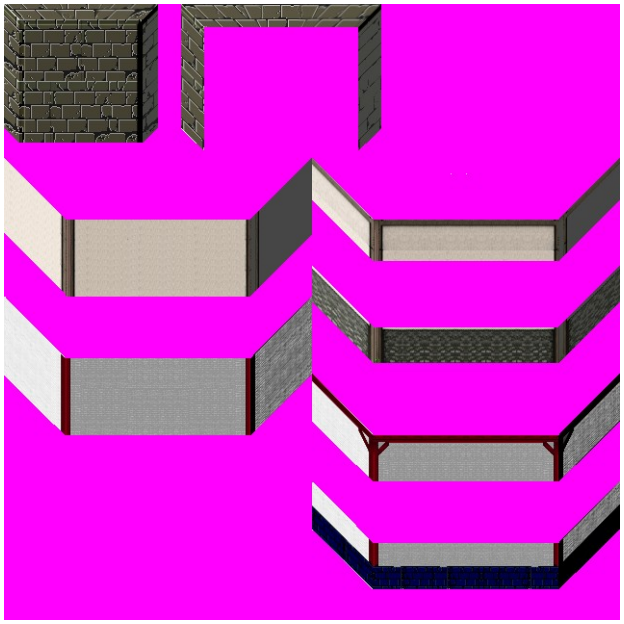
Floors and walls are meaningless without a background to support them. Besides the regular background that scrolls behind the level, in-level background is as important if not even more. It sets the mood for every screen, making every scene memorable.

Above the blue line, the script divides the image in regions of 64x64 pixels. This one is a bit messed up, as it is incomplete and has some smaller texture bricks that don't belong in there, but the point is to put all details in a separate image, and use 64x64 blocks to divide them. Some are modular, others not so much. It all depends on what type of level you are designing.

Note: The line says "Manually added bricks" because we previously used an automatic division tool called Rapid Brickset Editor.

C. Modular Perspective Walls

If you want certain structures to stand out and “un-flat” them, it's a good idea to create some modular walls. Taking in account the light source used in all other bricks, make some slanted walls. 45º is the easiest to make.

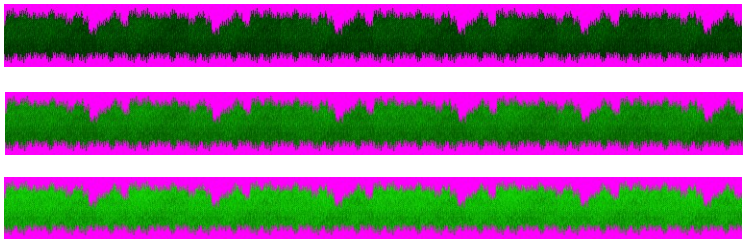


These are a bit harder to divide correctly, so refer to the brickset in the resource pack to see how they were divided. Basically the walls in perspective are two bricks each, the ones facing front are three (edges and center loop).

D. Animated elements

Since the beginning of this tutorial, we have been looking for elements that bring out the feeling of natural. “WAS there” rather than “WAS PUT there”, is the look you should always try to achieve.
The animated elements must not be completely oblivious to the atmosphere and setting of the level, they should play along and complete the experience of being in the level.

For a forest setting, wind blowing in the foliage is a nice touch.

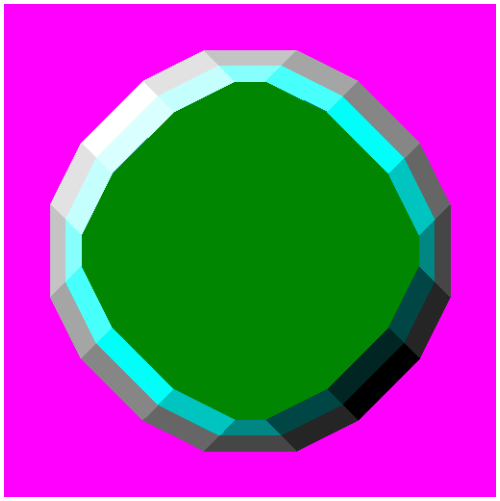


The division and animation is driven by the brickset script, so it requires much less resources and effort than creating an object for the same effect.

For the grass, once again, shading was used. Since grass it the element closest to the ground, and is itself, made into a texture, this small touch makes it blend better and immerse the player more with the illusion of lighting.

E. Path Bricks

This is probably the hardest part of all, so for your convenience, here's a template:



As you can see, they were made to fit together. You don't need to paint textures on both rings, the gray ring (outer one) is enough to start.

Also the blue ring may give you shades that do not match the ones from the gray one. The color is meant only to distinguish them.

After all the painting and separating, you should have two images: One with the solid blocks, and another one with the passable blocks.

If you prefer, you can edit the ones on the resource pack, or see how I’ve laid them out.

Appendix – Indexing Bricks

Here's a sorting standard that will make your bricksets easier to navigate in the editor. Don't be scared by the big numbers, you can skip numbers on brickset scripts without problems.

0-999 : Path - Solid
1000-1999 : Path - Passable

2000-4999 : Scenery - Misc
5000-6999 : Modular Walls (passable)
7000-7999 : Prefabs
8000-8999 : Textures (4 shades each) (up to 128*128) (zindex < 0.5)
9000-9999 : Animated