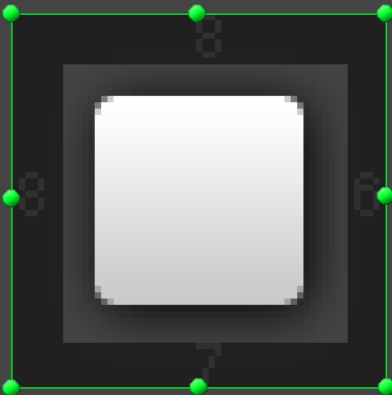


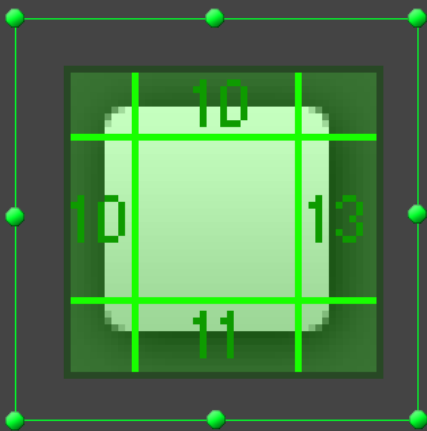
Original Texture
Width = 59
Height = 59

Padding = L:8, B:7, R:6, T:8



Drawn Sprite
Width = $(59 - 8 - 6) = 45$
Height = $(59 - 8 - 7) = 44$

Border: L:10, B:11, R:13, T:10



Known Data:

- Original texture's **Rectangle**

Given the original texture of 59 by 59, the rectangle would be (X:0, Y:0, W:59, H:59) if the texture is not a part of an atlas.

If the texture is a part of an atlas, then the rectangle is expected to contain the sprite's rectangle within the atlas texture (specified in pixels).

This rectangle may be smaller than the original texture if the edges got cut away because they contained fully transparent rows or columns.

Known Data:

- **Padding** (left: 8, bottom: 7, right: 6, top: 8)

The actual drawn area can differ from the selection box. It's quite common for it to be notably smaller than the original texture because parts of the texture may be transparent and don't need to be drawn.

When the sprite's pixels get cut away, they get replaced with "padding" instead. Padding information needs to be specified for all four sides in order to be able to position and draw the sprite correctly when the pivot changes.

When the sprite is scaled, the padding scales with it by the ratio of current size divided by original size. This essentially means that the sprite acts as if the cut pixels are still there.

Known Data:

- **Border** (left:10, bottom: 11, right: 13, top: 10)

Sliced sprites use border information to calculate the 9 quads used to draw the sprite. The border values essentially "inset" the sprite from the drawn edges.

When the sliced sprite is scaled, it cannot become smaller than the sum of padding + border in size, as neither of these values scale with sliced sprites. This behavior differs from that of a regular sprite (where the padding scales with it).