CHEVY RAY'S PIXEL FONTS

Table of Contents

1. Fonts & Files

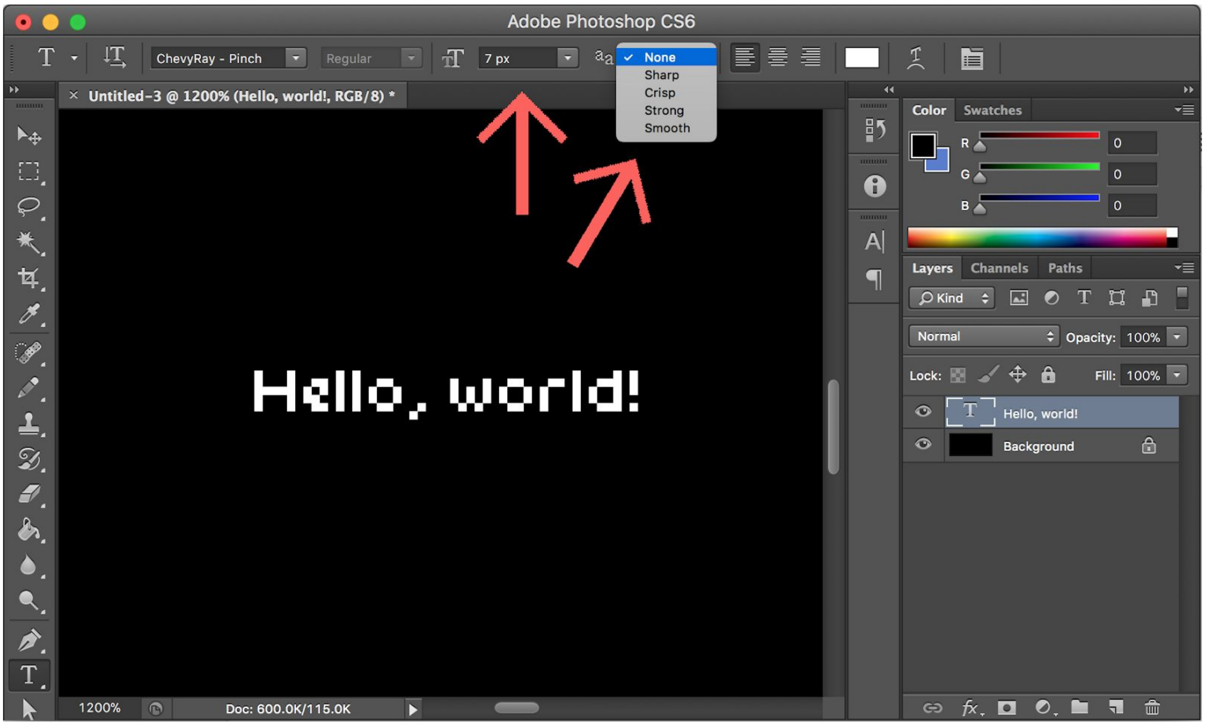
Thanks for purchasing Chevy Ray's Pixel Fonts! Each of these fonts has a specific size it is designed to be displayed at. You can also use multiples of that size to scale up the text.

Font Name	Size
Babyblocks	8 (16,24,32...)
Babyblocks Mono	8 (16,24,32...)
Bird Seed	10 (20,30,40...)
Bubble Time	21 (42,63,84...)
Chocolate	11 (22,33,44...)
Chocolate Mono	11 (22,33,44...)
Classified	16 (32,48,64...)
Column	11 (22,33,44...)
Crates	20 (40,60,80...)

Curse	**14**	(28,42,56...)
Express	**9**	(18,27,36...)
Express Mono	**9**	(18,27,36...)
Headline	**26**	(52,78,104...)
Lantern	**17**	(34,51,68...)
Lazytown	**9**	(18,27,36...)
Lazytown Mono	**9**	(18,27,36...)
Little League	**5**	(10,15,20...)
Little League Mono	**5**	(10,15,20...)
Love Bug	**9**	(18,27,36...)
Magic Book	**12**	(24,36,48...)
Oeuf	**13**	(26,39,52...)
Picory	**16**	(32,48,64...)
Pinch	**7**	(14,21,28...)
Pinch Mono	**7**	(14,21,28...)
Rise	**32**	(64,96,128...)
Roundabout	**32**	(64,96,128...)
Skullboy	**16**	(32,48,64...)
Skullboy Mono	**16**	(32,48,64...)
Slapface	**13**	(26,39,52...)
Softsquare	**9**	(18,27,36...)
Softsquare Mono	**9**	(18,27,36...)
Softsquare Wide	**9**	(18,27,36...)
Space Loot	**20**	(40,60,80...)
Stapler	**15**	(30,45,60...)
Tentown	**10**	(20,30,40...)
Thicket	**12**	(24,36,48...)
Thicket Mono	**12**	(24,36,48...)
Thinstix	**15**	(30,45,60...)
Vector	**9**	(18,27,36...)
Vector Mono	**9**	(18,27,36...)

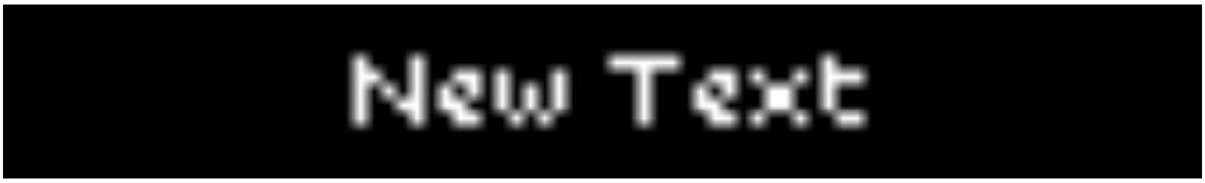
2. Using with Photoshop

In order for pixel fonts to display properly, you have to disable hinting and font smoothing in whichever program you're using. Also, each of these fonts is designed to look For example, in Adobe Photoshop CS6, you have to set the font to the correct size (see the above table) and then disable hinting with this option on the toolbar:

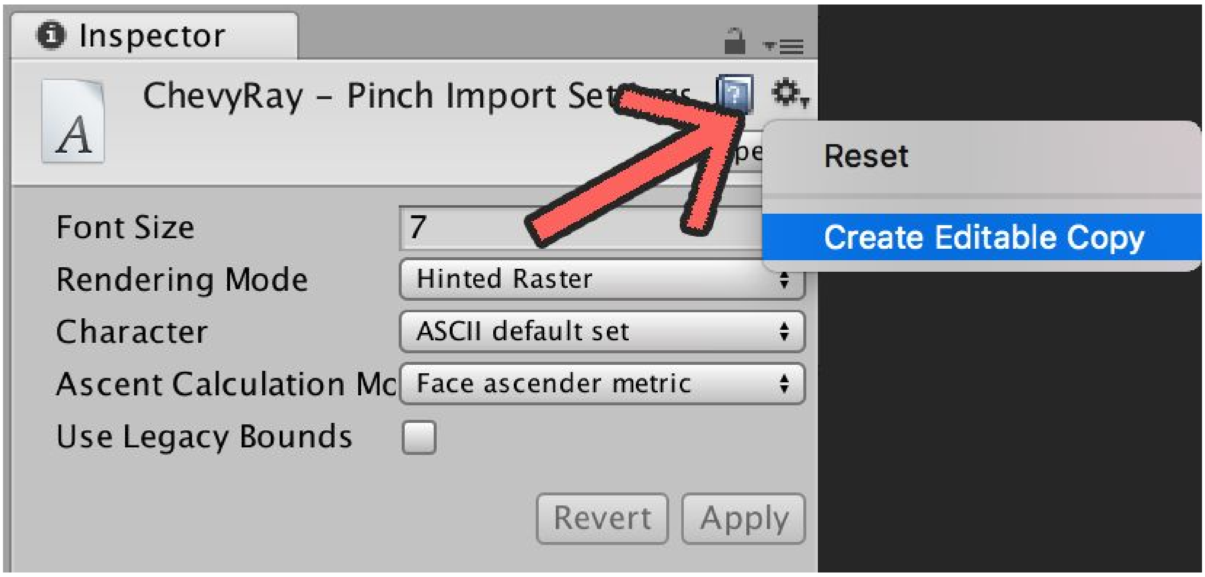


3. Using with Unity

In Unity, by default the fonts will display very ugly and blurry, like this:

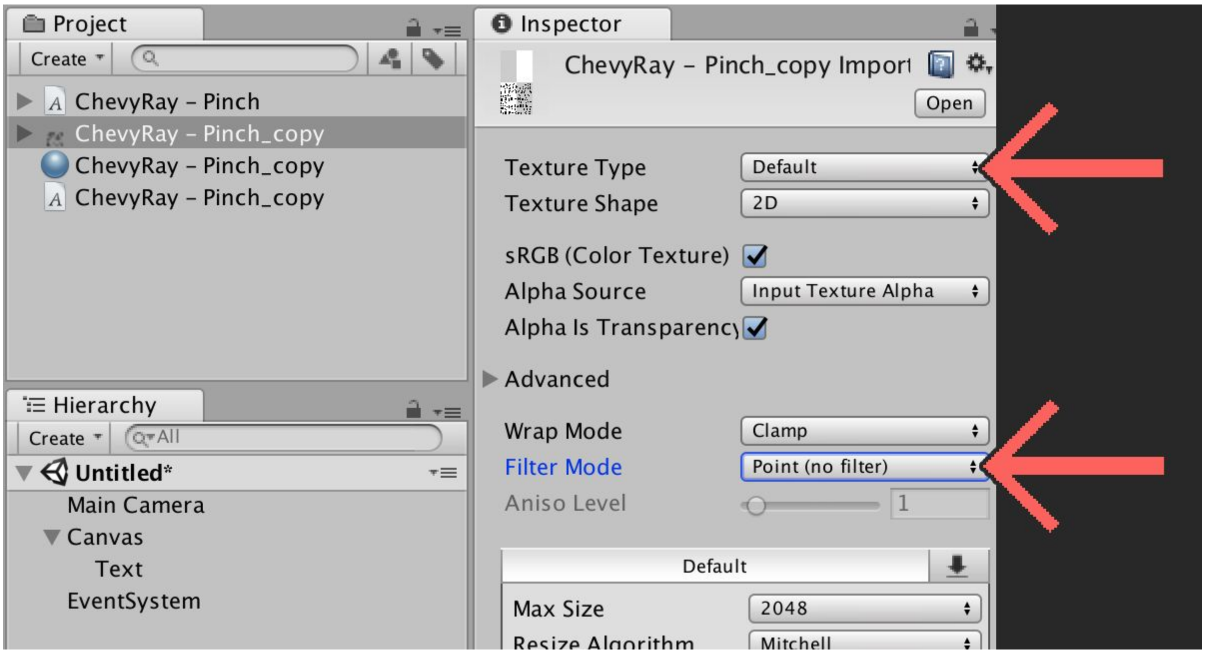


We have to jump through a few hoops to fix this. First, we have to make an **editable copy** of the pixel font. The gear icon in the corner of the font's inspector will give you this option:



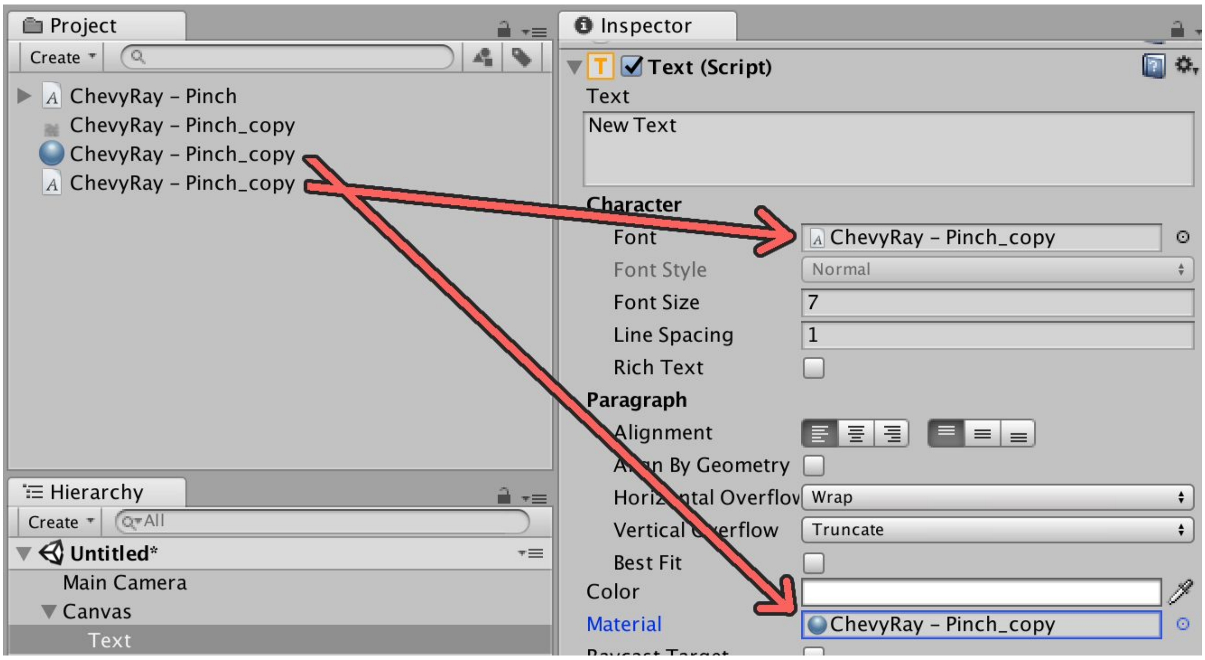
Font Size 7
Rendering Mode Hinted Raster
Character ASCII default set
Ascent Calculation Mc Face ascender metric
Use Legacy Bounds ☐

This will generate a new texture, material, and font that you are allowed to modify. First, modify the texture to disable the blurring, like so:



Texture Type Default
Texture Shape 2D
sRGB (Color Texture) ☑
Alpha Source Input Texture Alpha
Alpha Is Transparency ☑
Advanced
Wrap Mode Clamp
Filter Mode Point (no filter)
Aniso Level 1
Max Size 2048
Resize Algorithm Mitchell

Next, we need to assign both the **font** and the **material** to our text object:



Project:
- ChevyRay – Pinch
- ChevyRay – Pinch_copy
- ChevyRay – Pinch_copy
- ChevyRay – Pinch_copy

Hierarchy:
- Untitled*
  - Main Camera
  - Canvas
    - Text

Text (Script)

Text: New Text

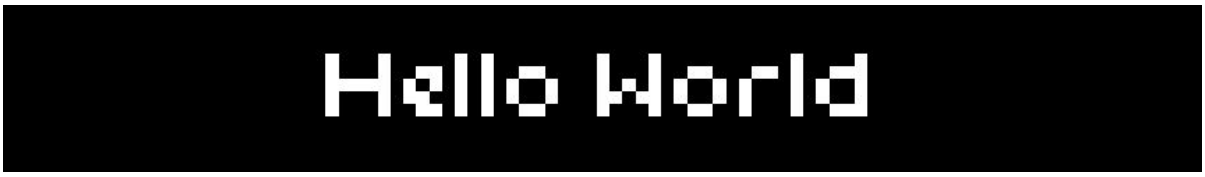
Character

Font	ChevyRay – Pinch_copy
Font Style	Normal
Font Size	7
Line Spacing	1
Rich Text	□

Paragraph

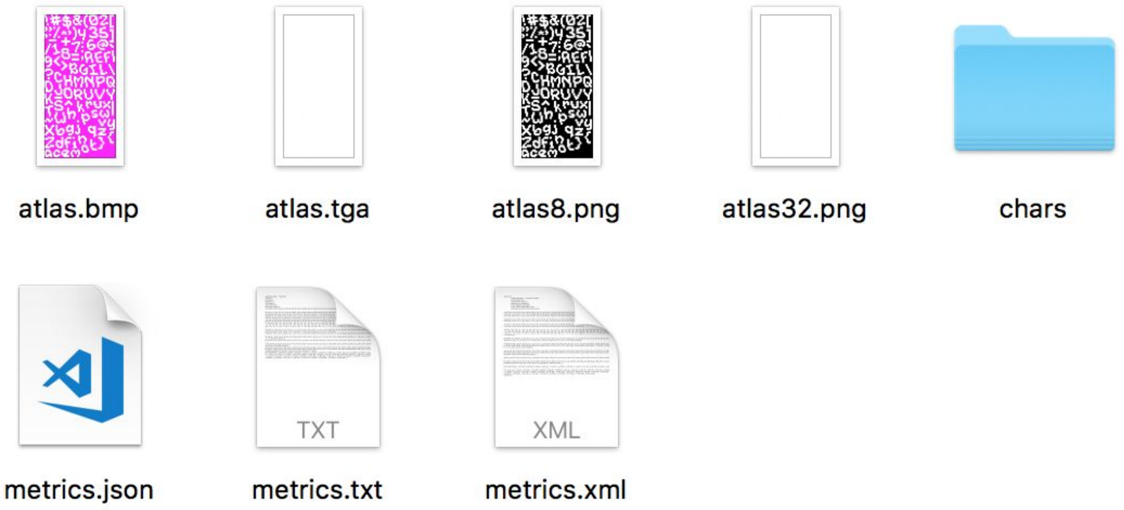
Alignment	
Align By Geometry	□
Horizontal Overflow	Wrap
Vertical Overflow	Truncate
Best Fit	□
Color	
Material	ChevyRay – Pinch_copy
Raycast Target	□

Once you've done this, the text will now display nice and pixelated, with no blurring!



Hello World

4. Assets & Metrics



Each font has a corresponding **Assets** folder which has images and data you can use for using the font in various game engines. There is an **atlas** image, in several different formats for convenience, with all of the characters packed into it. There is also a **chars** directory with individually rendered bitmaps if you prefer to use your own texture-packing system.

If you open up a **metrics** file, you will see more information about that font to help you render it correctly:

name: after drawing this character, the next should appear this x-distance ahead
size: the em size that the font was rendered at (note: this is NOT the height)
ascent: how high the font sits above the baseline
descent: how low the font drops below the baseline

The height of the font can be calculated as (**ascent - descent**).

It also has an array of all characters, and corresponding arrays for each character's metrics. These metrics are:

advance: after drawing this character, the next should appear this x-distance ahead
offset_x: when drawing a character at (x,y), offset its x-position by this much
offset_y: when drawing a character at (x,y), offset its y-position by this much
width: the width of the character in the atlas
height: the height of the character in the atlas

pack_x: the x-position of the character in the atlas
pack_y: the y-position of the character in the atlas

There is also a **kerning** array, which is a large array of kerning triplets:

[**left_char, right_char, kerning**, **left_char**, **right_char**, **kerning**, **...**]

The size of this array will always be (**kerning_count × 3**).

To use kerning, when printing **right_char**, if it is preceded by **left_char**, this is how much you should offset the character on the x-axis by **kerning**. This will make your text fit together tightly, making it more readable, so I highly recommended implementing this.

Here is some sample C++ of how your text rendering code might look:

```cpp
string text = "Hello, world!";
char prev = 0;
int pos_x = mouse_x;
int pos_y = mouse_y - font_ascent;
for (int i = 0; i < text.size(); ++i)
{
    if (text[i] == '\n')
    {
        pos_x = 0;
        pos_y += (font_ascent - font_descent) + 1;
        continue;
    }
    pos_x += get_kerning(prev, text[i]);
    int draw_x = pos_x + char_offset_x(text[i]);
    int draw_y = pos_y + char_offset_y(text[i]);
    draw_char(text[i], draw_x, draw_y);
    pos_x += get_advance(text[i]);
    prev = text[i];
}
```

5. License

1. Grant of license

This Agreement is a license, not an agreement of sale. Licensee shall not acquire any copyright ownership or equivalent rights to any of the Licensed Content. Seller and the Licensed Content sources retain all right, title and interest in and to all of the copyrights, trademarks, and all other proprietary rights in the Licensed Content. All rights in and to

Licensed Content not expressly granted in this agreement are retained by Seller or its suppliers.

Licensee is permitted to use the Licensed Content in unlimited commercial projects. A commercial project is one defined as a Work for Distribution launched with the capability to generate revenue, or intention to generate revenue through the sale of, licensing of, or otherwise intend to generate revenue directly from the Work for Distribution.

2. Restriction on use

Licensed Content may not be used contrary to any restriction on use indicated herein.

Licensed Content may not be resold, sublicensed, assigned, transferred or otherwise made available to third parties except as incorporated into Works for Distribution. Licensed Content may not be distributed to third parties as standalone files or in a way that unreasonably permits the recipient to extract the Licensed Content for use separately and apart from the Work for Distribution.

Licensee may not distribute the Licensed Content in any library or reusable template, including but not limited to game templates, website templates intended to allow reproduction by third parties on electronic or printed products. Licensee may not distribute Licensed Content in a manner meant to enable third parties to create derivative works incorporating Licensed Content.

Licensee may not superficially modify the Licensed Content and sell it to others for consumption, reproduction or re-sale.

Licensee shall not use the Licensed Content in a manner that violates the law of any applicable jurisdiction.

Licensee shall not claim copyright or attribution of Licensed Content.

3. Term and termination

The license contained in this Agreement terminates automatically without notice from Seller if Licensee fails to comply with any provision of this Agreement. Upon termination, Licensee must with immediate effect stop using the Licensed Content, destroy, delete and remove the Licensed Content from Licensee's premises, computer systems and storage. Licensee must also make all reasonable efforts to ensure that copies of the licensed content are removed from any locations it has been distributed to.