



Steam Assets TeX Kit User Manual

Neruthes

2024-04-05

Chapter 1

Introduction

Producing a series of store assets with mandated specific dimensions can introduce a significant mental overhead. If you are familiar with concepts like dependency trees and functional programming, you might envision a more efficient process where a minor change in the logo automatically updates all store assets that depend on it. Imagine a scenario where a simple modification to the logo seamlessly propagates across all associated assets, streamlining the design process and ensuring consistency throughout. This automated synchronization could not only enhance productivity but also contribute to a more harmonized and efficient workflow.

We at Nekostein invented a LaTeX-based internal workflow to generate store assets for Steam right before we published our [first store page](#)¹ on Steam. This document is the manual of the open-source toolkit which is abstracted from the internal workflow, which resides in a large repository where all non-code stuff² of the project is managed.

We assume that you are working on a modern GNU/Linux desktop, or using an OS with similar capabilities. For mainstream desktop user who prefer Windows PC, the workflow described in this manual could unnecessary. But if you insist, try using this toolkit inside a Debian WSL2 environment.

This manual is designed for users working on a modern GNU/Linux desktop or a system with similar capabilities. For Windows PC users who prefer a mainstream desktop environment, the workflow outlined in this guide may not be directly applicable. However,

¹Game title: KumaWelt 1: Honey Frontier.

²Non-code stuff: Branding assets, legal documents, marketing materials, Steam keys, videos splashes, etc.

if you wish to proceed on a Windows system, you can explore using this toolkit within a Debian WSL2 environment. If you are unfamiliar with WSL2, consult Microsoft official documentation.

1.1 Dependencies

Before you start, make sure that you have installed the following dependencies on your machine.

- **TeX Live**

We will use `xelatex` command to compile LaTeX files into PDFs.

<https://tug.org/texlive/quickinstall.html>

- **Poppler**

This toolkit is used to convert PDFs to images.

<https://poppler.freedesktop.org/>

1.2 Download Files

This repository is available on GitHub at the following URL.

<https://github.com/nekostein/steam-assets-texkit>

You may simply download the ZIP archive. You will hardly want to fetch from the upstream later.

Chapter 2

Prepare Components

2.1 Download Example Components

You can run the following command to have a quick overview, before using your own components.

```
./make.sh steam
```

Put your files in the “components” directory, reusing the filenames.

2.2 Background (bg.png)



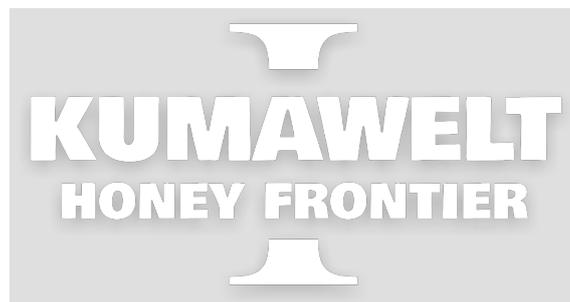
Example component: bg.png
© Alessio Soggetti (via [Unsplash](#))

You should prepare some generic empty background image and save to “/components/bg.png”.

1920x1080 should be a good size. Use the 16:9 ratio unless you feel good with manually adjusting a few parameters for absolute positioning.

This background image can be concept art or screenshot. Make sure that its elements and colors are simple, otherwise your logos will not be legible.

2.3 Large Logo (logo1.png)



Example component: logo1.png

This file should be a transparent PNG. For best compatibility, a horizontal logo is prefer-

able. The aspect ratio should be around 1.25:1 to 2.5:1.

2.4 Small Logo (logo2.png)



Example component: logo2.png

You might want to use a different logo image in smaller assets for better legibility. Typically, a solid drop shadow (like outline) behind the main shape will be good. In CSS terms: “text-shadow: rgba(0,0,0,0.5) 1px 1px 0px”.

Chapter 3

Generating Output Images

3.1 The Shell Command

You can run the following command to build all images.

```
./make.sh steam
```

3.2 Magic Explained

If we set document paper width to 300bp and convert the produced PDF to PNG using 72 DPI, the PNG will be exactly 300px wide.

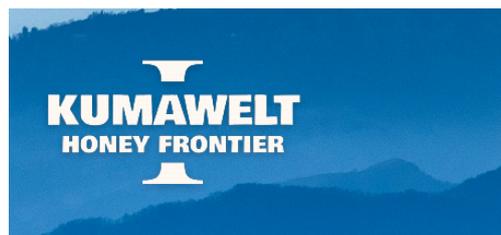
BP is a length unit defined as $\frac{1}{72}$ inch. TeX originally defined 1 PT as $\frac{1}{72.27}$ inch, but later Adobe appropriated the word with a different definition: 1 PT = $\frac{1}{72}$ inch. TeX later added the new unit $\frac{1}{72}$ inch for compatibility, but a different name “bp” is assigned.

Chapter 4

Example Outputs



*Main Capsule (616*353)*



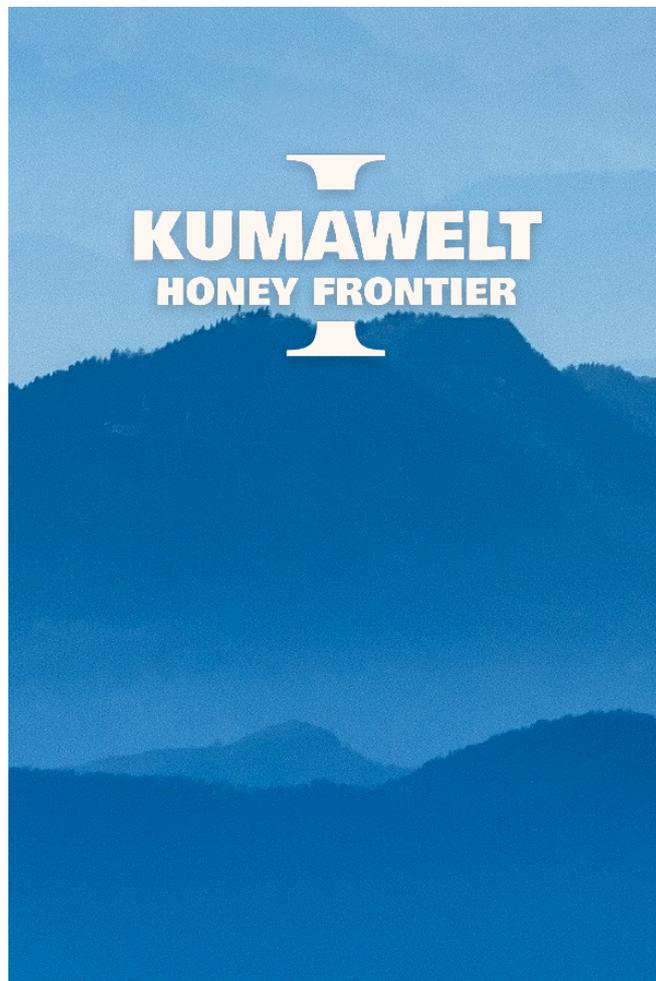
*Header Capsule (460*215)*



*Small Capsule (321*87)*



*Vertical Capsule (374*448)*



*Library Capsule (600*900)*

Chapter 5

Tweaking Positions

If you want to manually tweak the positioning things, you may dive into the “steam/*.tex” files to adjust length numbers.

Chapter 6

Afterwords

6.1 Obtaining This Manual

You should be able to get this manual from either of the following URLs:

- [Link 1](#)
- [Link 2](#)

6.2 Ways To Support Us

- Wishlist [KumaWelt 1: Honey Frontier](#) on Steam.
- Star [this repository](#) on GitHub.
- Make a [donation](#).

Copyright © 2023-2024 Nekostein. All rights reserved.

The word mark "KumaWelt" is an unregistered mark used by Nekostein for trademark purposes in commercial activities.