

《我是如何思考及打磨 Eagle 这款产品的？》

作为设计师，其实我们心理都知道：「挑别人的产品毛病很容易，但要亲自弄一个更好用的，决不是三言两语的事情」，而在这些产品背后，究竟是那些产品开发者/设计师遇到了哪些困难、限制，以及他们是如何思考解决这些问题的，往往都不是我们站在外面可以清楚看到的。

在这小节，我希望能透过自身 3 年独立开发的切身经验，和大家分享在这三年中开发 Eagle 的过程中，我遇到的那些难题、困难，以及当时面对这些难题背后我是如何思考的，以及我最终使如何解决这些问题的，希望我的这些经验，可以带给大家及同行从业者一些启发。

挑战：更简单、直觉的管理素材

曾经使用文件夹电脑管理素材、灵感的朋友应该都有遇到过这样的情境，我们将素材依据不同的类型，放在电脑各个路径、文件夹，但是时间一久，每当需要使用某个素材文件时，却总是找不到、忘记它们究竟放在哪个硬盘、哪个路径，使用关键字搜索除了很慢以外，有时甚至不晓得该用哪些哪些关键字来搜索。

我的想法：

解决搜索电脑文件问题有很多思路，比如我们可以使用 Alfred 这类的文件搜索神器，高效找出电脑上任何服务搜索条件的文件，这很棒，但这个方式并没有降低脑袋的负担，

另外，类似 Adobe Bridge 这类型的产品，本质上还是和文件管理器是同样的思路的设计方式，使用这个方式管理素材，时常遇到素材和其它不相干的文件混杂显示在一起，无法有效专注查找、使用，而且和文件管理器的一样，我们得把所有素材的路径记忆在脑袋中，相当辛苦。

让素材管理工具能够直接管理本地文件夹结构，并不高效，软件如果要够高效，就不能每次载入都要依赖读取硬盘上的信息，但单靠文件本身的信息又不足够，通常这些工具会将软件内对应的相关本地文件、文件夹的信息事先进行索引，并保存在软件的架构下，但这也导致，每当我们关闭这些软件并使用其它工具（如系统内建

的 Explorer) 新增或修改相关路径的文件、文件夹时，这些软件并不晓得，为了确保软件显示的数据和本地路径是一致的，每一次打开这些工具时，它们都需要在背后进行大量的硬盘路径扫描，确保文件一致性，有时候这样的任务一等就是几十秒，软件无法响应，浪费了许多资源成本，也不高效，相信有深度用过 Adobe Bridge 人应该都曾有过类似的经验。

我希望 Eagle 可以本身运行可以更高效，并能够让收集、查找素材这样的情境可以更无脑、专注些，不需要为了管理素材，让大脑还需要记住大量额外信息。

解决方案：

我希望 Eagle 就像是「印象笔记」、「微信」、「音乐」应用设计的思路一样，专为某种场景提供服务，减少脑袋负担；比如：想要写笔记，直觉的打开印象笔记即可，不需要知道具体每一篇笔记究竟保存在电脑硬盘的哪个路径，真的太累了。

因此，Eagle 一开始就决定采用这种设计逻辑来开发，需要收集、使用设计素材？别再想太多，打开 Eagle 吧！

挑战二：不依赖任何第三方套件，让 Eagle 可以搭配同步工具在多台设备同步（Dropbox、Google Drive、OneDrive、坚果云等）

Eagle 是一款本地素材管理应用，将素材保存在本地有许多优点，比如反应快、拖拉直接可以使用、自己能掌握数据安全，但有时候我们会在台设备上工作，如果素材库只能存放某台设备中，无法在多台电脑同步使用，会是个很困扰的问题。

我的想法：

一般设计素材文件大小都非常巨大（百兆、千兆），对于目前 Eagle 的人力、物力来说，开发自己的云同步不是一个很适当的方式，不仅无法负担巨额的流量、储存成本，甚至需要考虑其它开发外的外在因素，比如：内容违权、内容法律规范管理等问题。另外，我们也清楚有越来越多的设计工具开始用订阅制模式进行收费，大家的负担也越来越重，所以，我希望能在自己开发同步与不能同步两种极端选项中找出另一条出路。

幸运的是，文件同步这类需求，市场上早已存在许多健全的服务，如 Dropbox、OneDrive、Google Drive、坚果云等工具，它们定价非常的合理，也很稳定（甚至提供 30 天无限版本还原备份功能）。

我和许多用户本身其实就是 Dropbox、Google Drive 的付费使用者，于是我就开始思考，能不能让 Eagle 来支持所有这类型的同步工具，让文件同步任务用更低的负担，更

高的安全来达成。（使用这些服务，你甚至可以同步 Eagle 以外任何其它场景需要的文件，这是相对超值的）

让 Eagle 支持「所有」类型的同步工具这个任务并不容易，「所有」表示我不能依赖任何第三方工具提供的 SDK 套件开发，同时我也不能使用任何本地数据库（例如：SQLite）来开发这套软件，主要原因是，使用本地数据库（.db /.db3 /.sqlite3）开发方式，数据文件随著数据越来越多，文件也会逐渐变得越来越大，这会带来同步成本无限上升，也就是如果用户只修改一个文件，对于同步工具来说，还是需要把整个 .db 文件重新上传/下载到所有电脑，文件越多，同步负担越大。

解决方案：

我决定自己来开发自己的数据库套件，不采用已知的那些开源数据库工具（如：SQLite），来保存每一个图片素材的数据，满足大部分场景数据查询（Query）的性能同时，也能支持所有同步工具差异化同步的工作（在 Eagle 修改一张素材信息，对同步工具来说，仅需要同步一个大小约 0.5-1kb 的小文件，比起数据库需要数十、数百兆差异是巨大的）。

挑战三：提高 Eagle 打开速度，无论资源库保存多少数量、占用多少大小，并将读取硬盘次数降到一次

最早版本的 Eagle（约在 2017/04 - 2017/08 其间的版本），在性能上确实有很多进步的空间，比如随著保存的数据越来越多，打开应用的速度也开始受到影响，特别是在性能比较不理想的设备上，最早的 Eagle 确实不快。


我的想法：

这个问题在我的预期中，毕竟决定自己开发数据库模型是一个相当艰难的挑战，在保证能够支持同步工具差异化同步功能的同时，又能确保软件可以以无比的速度快速打开。

解决方案：

我针对原来的架构进行了大量优化，引入最重要的「数据检索缓存」功能，这项功能让 Eagle 应用打开性能指数级别的提升，即使你在 Eagle 保存了数万张图片，软件打开的过程也仅需载入这「一个数据检索」文件就能完成，无须逐一载入，不受到硬盘读写能力的影响，我在多台设备测试，无论使用什么硬盘，采用这项技术后，Eagle 载入数据的性能可以达到 40,000 - 60,000 个/秒。

以前有部分用户曾使用 Finder / Explorer 打开 Eagle 资源库，发现文件越多越慢的状况，这个原因是因为 Finder / Explorer 它们需要展现的信息栏比较丰富（比如同时需要知道创建日期、修改日期、标签、缩略图等），所以采用了相当复杂、费时

的查询方法得到数据列表；但 Eagle 内部并不是这样运做的，Eagle 仅查询必要的信息，仅需一次硬盘读取，这也是 Eagle 打开极快的原因。

挑战四：打开任何文件夹，就能立即预览列表上的内容

Eagle 很早期的版本就开始支持各种常见的文件格式，比如 psd、ai、sketch、mp4 等，在早期的版本中，每当我们打开文件夹想要查找素材时，和所有的文件管理工具一样（Finder、Explorer），时常因为文件类型、文件大小太大等原因，无法立即在列表上查看到文件预览图，都需要等待软件在背后进行载入、计算才能看到结果，这样的问题，让查找、管理素材的体验变得非常糟糕。

我的想法：

在开发圈的这句话：「你要用时间换空间，还是要用空间换时间」可以直接反应这个问题背后的原因。为了让体验更极致，我认为预先产生缩略图这样的技术应该可以解决这个问题，我考虑了这两项开发方式：

1. 采用 Thumbs.db 技术开发（Windows 文件管理器采用此方式）
2. 独立保存每一个缩略图文件（主流 SaaS 几乎都采用此方式）

解决方案：

为了同时保证能满足缩略图性能以及能兼容所有同步工具以及操作系统，传统的 Thumbs.db 技术并不适合 Eagle，最后我采用「独立保存缩略图」这个方式来开发。

事实上，macOS 及 Windows 操作系统的文件管理器默认的做法并没有错，只是事先将整台电脑的所有文件缩略图都制作保存起来，这种做法对于整个系统这种量级来说，并不符合成本、效益的。但是像是 Eagle 这样使用场景明确且文件数量是在相对有限、可估量的前提下，预先制作缩略图，是可以带来许多好处的，比如：

1. 大幅提高软件性能

在双击打开大图以外的场景，软件不需要载入较大的原文件，不管是打开软件或者列表载入性能，都会得到巨大的性能提升，同时也提高了用户体验。

2. 特殊格式文件（psd, ai, pdf）可以更快速的在列表呈现

大部分情况，我们并不需要在列表上完整载入数十 MB 的文件进行预览，这样的行为对电脑硬件的负担也相当巨大，缩略图可以有效帮助软件在列表高速预览这些特殊格式的文件。

注1：其实任何系统都有缩略图技术，之所以我们没发现，只是因为系统默认将这类文件设置为隐藏文件，故意让我们看不到而已。（Eagle 也能隐藏，但实际上没必要）

注2：有些图片本身预览性能极高，并不需要缩略图的协助，所以 Eagle 不会对所有的图片产生缩略图，对于那些本身文件就支持预览的格式，Eagle 不会浪费额外空间来制作缩略图。

挑战五：支持将同一个素材文件分类到多个文件夹，并减少原文件进行搬移、修改等成本较高的操作，将低应用读写硬盘的成本

设计素材文件与其它电脑文件在分类管理上，有这一个很不同地方，很多时候我们设计素材是同时属于多个分类的（例如有个素材同时属于插画素材、手绘风格两种分类），传统上，如果们透过电脑文件夹来管理素材，会有个相当不便的（除非使用 In 指令），一般我们很难将文件同时放在多个文件夹进行分类。另外，假如我们想要进行文件分类整理，使用传统文件管理器，我们只能透过物理搬移文件的方式来进行，相当费时。

我的想法：

假如，我们每天都需要对数万、数千个文件疯狂不断的分类、修改，而且也不写入 EXIF 元数据的需求，那么使用「数据库」这种开发技术肯定是最高效；但是，一般日常整理的设计素材，我们并不这样使用，素材的累积往往是日积月累的。

解决方案：

根据这样的需求，Eagle 采用非物理分类的方式来设计分类功能，Eagle 将素材的各种数据保存在一个大约 1kb 左右的 json 文件上，因此在进行分类的时候，它不需要特别复杂的物理搬移操作，也不会因为分类到多个文件夹，占用电脑两倍甚至多倍的占用空间。

选择较于传统更优良，又没有数据库带来种种限制的「设计方案」，对 Eagle 才是更弹性、有效的方式。

注：单纯不考虑任何因素：传统方法 < Eagle 采用的方法 < 数据库开发方法

注2：我曾遇到一些不客观的评论者，时常会把各种极端的文件操作方式作为批评产品的理由，比如他们举例：每天都需要修改几万、几十万个文件这种操作，或许真的有少部分的人会这样使用电脑，但是这种操作无论透过什么文件管理工具，这操作本身其实就在加速硬件寿命的损耗。

假如市场上真实存在这种需求，或许需要一套针对这种情境设计的解决方案，但这并不是 Eagle 或其它常规文件管理器的设计初衷。

挑战七：将不同目的素材，分开进行管理，让完全不相关的标签、文件夹不会混合在一起，影响管理体验

挑战：针对中文深度优化

大部分国外开发者开发的工具，很少会针对中文语言上的细节进行深度的优化，使用中文长期体验下，总会感觉到不足，比如搜索不精确、搜索无法自动提示、无法使用拼音搜索、界面字体不好看等问题，这不能怪这些产品，毕竟很难要求非母语开发者做到这些事情、理想程度。

我的想法：

中文就是我自己的母语，对「断词、词性、拼音、简繁体」这些观念算是熟悉，所以我希望能把中文的这些细节都添加到 Eagle 的设计流程中，让 Eagle 可以更符合中文使用者的使用习惯。

解决方案：

目前版本的 Eagle，已经能够支持这些特性：

1. 文字断词、词频分析

Eagle 有能力针对任何属性、文本进行中文字词分析，例如使用浏览器扩展收藏图片时，默认会在当下分析该网页可能出现的相关词，透过词频也能抓出该网页的关键字，并在用户输入标签时进行推荐。

2. 拼音搜索

在 Eagle 你可以使用拼音的方式快速输入、查找你需要的内容，比如想要搜索出「小米手机」标签，你只需要输入「xiam」就能迅速找出。

3. 中文、拼音模糊搜索 (Fuzzy Search)

模糊搜寻是个很伟大的发明，他可以让我们不需要输入完整字词就能够找到符合关键字的内容，这种算法在开发工具上（比如 VSCode、Sublime Text）行之有年，Eagle 不仅支持英文的模糊搜索，他还能同时支持中文、拼音作为关键字，以上面的

案例为例，在 Eagle 搜索「小米手机 (xiao mi shou ji)」标签，可以直接输入「xmsj」来搜索。

4. 机器学习用户自定词

断词系统本身受限于词库的限制，并无法辨识出一些新名词、用户偏好用语，因此我在断词系统的基础上，而外设计了一套自动学习用户词语的功能，让断词、推荐关键字都能够更精准。比如：断词系统对「海报设计」断词的结果「海报、设计」两个词，但如果用户曾经在 Eagle 中明确的添加过「海报设计」这样的标签时，这个时候系统会自动学习起来，并再下次进行断词时，将分析结果增强为「海报设计、海报、设计」。

5. 推荐标签

添加标签是一件苦力活，Eagle 能够根据素材本身名称、分类、标签等属性，在后台计算出可能用户会输入的相关字词，并依据内部的算法排序推荐给用户，透过这种方式，大幅提升标签输入的效率及体验。

6. 中文简繁体搜索

过去，如果收藏了许多简体及繁体文字作品，由于文字编码不同，我们很难透一次将相同意思的内容都找出来。Eagle 支持中文简繁体搜索功能，无论关键字是繁体还是简体，Eagle 都能够帮你把那些相同文字的结果给找出来。

挑战：让收集网页图片的任务可以更轻松、更容易

设计师时常需要在网路上浏览设计灵感、大神作品，看到喜欢的图片时，我们我想要将它们下载保存下来，传统上我们可以透过右键保存图片的方式，将网页上的图片下在到电脑保存，但长期下来，使用这个方法遇到几个比较麻烦的问题：

1. 操作过于繁琐

如果要下载大量图片，一直右键保存，步骤繁杂且浪费时间。

2. 无法批量处理

如果我们需要下载一个页面上的所有图片，一张一张右键下载明显不是很聪明的作法。

3. 无法保留图片来源

下载下来的文件，没有办法保留图片的来源信息，无法在未来寻找来源。

4. 网站锁右键、或禁止右键保存、背景图等防下载技术

许多网站都有防下载技术（如 Instagram、Flickr），造成保存上的麻烦。

5. 分类很费时

如果下载到电脑的同时，还要同时选择不同文件夹进行分类，光是打开文件夹、返回上一层、打开文件夹，这样的操作非常累人。

我的想法：

我希望简化收集图片的流程及交互逻辑，讓收集圖片所需要的步驟降到最低，针对那些做了防护的网站，Eagle 应有能力进行一定程度的破解，直接可以获取真实图片的数据，如果可能，透过各种分析技术，帮用户取得图片的额外信息（如链接、标签、名称等），尽可能不需要自己输入补齐。

解决方案：

Eagle 浏览器扩展能够很有效的解决上述问题，在开发时，我考虑的许多不同的技术、交互方式，满足各种不同场景下的需求，以下是目前版本（v2.0.13）已经拥有的功能：

#1 多种更高效收藏方式

Eagle 提供四种收藏相当高效的方式，收藏眼前的任何图片：

[图-拖拽收藏]

拖拽收藏

[图-Alt右键收藏]

Alt+右键收藏

[图-右键菜单]

右键菜单

[图-批量收藏]

批量收藏

#2 添加破解锁右键、防下载机制

我为 Eagle 扩展特别设计了一套串透技术，可以从根本上解决锁右键、防下载等网站的限制，如 Instagram、500px、Flickr 等，轻松保存图片。

[动画-收集 instagram 图片]

#3 支持 background-image / srcset 等属性

Eagle 扩展可以保存任何可能包图片的网页元件，不像其它扩展仅限于 图片标签，这项技术可以精准获取鼠标下方任何 HTML 标签，并自动识别标签的图片链接，除

除了可以保存基本的 src 属性外，就连那些使用 background-image、srcset 属性的其它元件也都能够轻松保存。

#4 智能获取文件名、链接

收藏网页图片，Eagle 会自动将 alt 属性作为文件名默认的考量，假如图片没有 alt 属性，则采用图片原始名，若原始名是无意义的乱码（比如常见图床工具），最后 Eagle 会以当前网页名来命名图片，这样的设计方式可以有效帮助用户使用关键字找到相关的图片；另外，任何采用 Eagle 扩展保存的图片，都会将图片来源链接保存到软件中，让未来可以回头找到图片来源。

#5 标签推荐

Eagle 针对许多设计师网站做了深度的优化，比如 Dribbble、Behance、Pinterest、站酷、花瓣网，收集这些网站的图片时，Eagle 会自动分析这些网站的原始码，将原作者的标签、网站的关键字抽取出来，然后建议到标签输入元件中，加速用户输入标签的效率。

[图-推荐标签]

由于 Eagle 标签推荐，用户不需要输入完整单词就能迅速将标签输入完成。

#6 减少重复下载

浏览器扩展会智能判断图片是否已经在浏览器端下载完成，假如收藏已经下载的图片，Eagle 扩展会直接将图片转为二进制数据，直接发送给 Eagle 软件，不会以链接方式发给软件，让重新下载一次，浪费时间也消耗无谓流量。

#7 智能转大图

灵感网站的图片列表，通常分辨率都不大，收藏这种缩略图意义不大，但想收藏大图，要一张张的打开大图查看，再进行收藏，这样的非常浪费时间，体验也极度不爽！

Eagle 支持主流设计师网站小图智能转大图功能，看到喜欢的图片直接收集就能下载最大尺寸，不需花时间打开再预览。

目前智能转大图功能支持这些网站：Pinterest、Behance、Dribbble、Artstation、Medium、Designspiration、花瓣、站酷、500px、Instagram、Flickr、新浪微博、QQ 相册、豆瓣相册、POCO、淘宝、天猫、蘑菇街、1688、AD、Archiproducts、Houzz、HouseBeautiful、Officesnapshots、Archilovers、Archdaily 中文、ArchDaily 国际、Dezeen、interiordesign、所有 Wordpress 开发的网站、所有 Wix 开发的网站、所有 Squarespace 开发的网站。

挑战：让分类素材变成一种愉悦的体验

[痛点][低效][进进出出][重复操作过多][文件夹很多的时候，是地狱][无法让素材分类到多个分类][不容易批量处理]

我的想法：

我希望「分类素材变得更高效，即使用数百种分类也能秒速完成分类」

解决方案：

#1 多种更高效的分类工具（颠覆传统搬移逻辑）

[拖拽分类][高效分类工具（F）][Video或GIF，演示什么是高效分类]

#2 历史记录

[加速选取的过程，历史记录是非常有效的]

#3 强大搜索能力

[让想到就能找到][模糊搜索、拼音搜索]

#4 自动找出未分类素材

[可以在想整理的时候一起整理，Eagle 提供一个未分类导航功能]

#5 自动分类（智能文件夹）

挑战：解决素材散落各处，找不到素材的窘境

[目前痛点][关键字的不足][传统工具太慢][缺少多种维度、颜色、时间、形状][传统工具不是为了这个目标设计的]

我的想法：

我希望「无论资源库中有多少文件，都可以在 1 秒内找到需要的素材」

我希望「可以更快速的进行多种属性筛选，我重新设计了传统筛选器设计模式」

我希望「收藏网路图片可以更聪明、简单，不必一张一张慢慢下载」[让这件事情的操作可以减到最少][可以更高效，比如批量处理][默认取得更多的信息][协助用户输入修改内容][破解右键]

解决方案：

1. 更高效的收藏方式

辟谣：Eagle 搜索并不会扫描硬盘，无稽之谈...

挑战：优化颜色搜索功能，让筛选结果更符合人眼感知

挑战：在数百、数千个文件分类中切换自如，想到就能看到

[痛点][随著累积、分类越来越多、找资料的速度指数级下滑、肉眼查找非常浪费时间]
[demo、讨论临时需要查看相关参考，往往要花上一段时间，甚至重新 google 都比较快]

1.

挑战：不需要双击打开素材，就能快速预览素材内容

我希望「在列表就能预览，不需要双击打开文件浪费时间（视频、音频、字体、图片）」

[痛点][寻找合适素材，过去往往需要比较麻烦的步骤打开查看，相当费时麻烦]

我的想法：

解决方案：

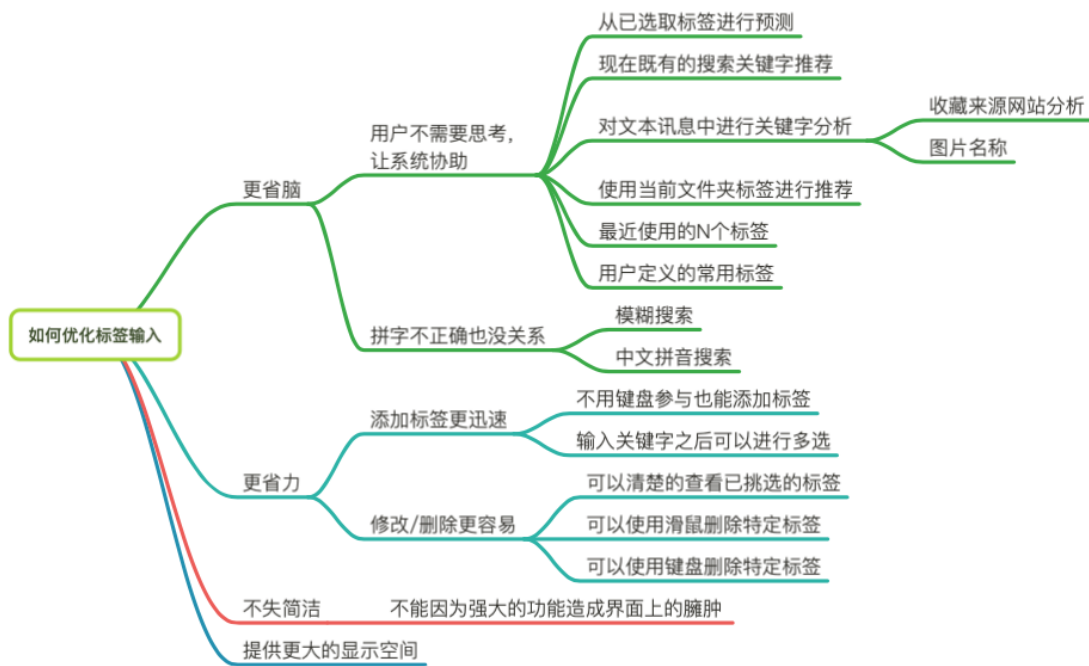
1. 空格键预览
 2. 鼠标悬停预览
-

挑战：让输入标签变得更有效率

我希望「输入标签这件事情可以变得更智能」

[痛点][传统标签输入效率不高][更轻松][更简单][更不动脑]

我的想法：



解决方案：

挑战：解决传统布局难以查看素材的问题

我希望「可依据不同场景提高浏览素材效率（布局）」

[痛点][细长图片被缩放到看不清楚][传统布局是针对文件设计不是针对素材情境设计][无法显示完整、时常出现内容被裁切][看不出素材本身的比例][影响查找素材速度]

我的想法：

[描述一下我想到素材查看的两种情境][不同情境目标不同，布局应不同][需要一种根据素材比例，智能呈现的布局方式]

解决方案：

1. 自适应布局
2. 瀑布流布局
3. 随时都能切换布局

挑战：让产品界面保持简单，不会随著功能不断迭代，越来越复杂

[痛点][每个人使用工具的目标不同][有人希望很强大][有人希望支持很多格式][有的人希望够简单、简洁][不希望 Eagle 变成 PS 那么臃肿][找不到需要的功能][界面太多面板]
传统布局方式并非为素材管理来开发，

我的想法：

为了不变成「胖子」，

我希望「Eagle 不会因为不断迭代，越来越复杂」（顺便讲一些垃圾化，比如多种需求指向一个功能，多种功能指向相同需求）

解决方案：

1. 渐进式界面设计

依据用户实际使用状况，启用对应的功能，无此需求的用户，默认隐藏

2. 允许根据偏好进行设置

用户可以在偏好设置对界面进行一定程度的调整，隐藏自己不需要的功能。

3. 主动询问用户偏好方式

有些常用但是每种族群偏好不同的设置，很难透过「良好的默认值」设计达成，把这些功能设定放在偏好设置页面，用户也很难发现

挑战：传统 Web 技术资源载入逻辑不够智能，需要自己重新开发「页面资源载入优先顺序」算法，做到滚动列表时，能够优先呈现眼前的内容，并减少不必要的载入操作

使用传统 App/Web 技术在载入页面内容，往往采用「谁先请求就帮谁先载入」的逻辑进行资源载入，这种模式在资料量较少的浏览情境下，不会遇到什么问题，但在一个拥有大量文件的列表，进行迅速的滚动时，会依序由上到下的方式载入内容，即使用户的视线已经来到列表底端，依然会继续将上方滚出界面外的内容载入后，才会继续往下载入，大幅延后用户看到眼前内容，浏览体验相当糟糕。

这样改可能会更容易让人理解

假如有一个超过 1000 个文件的文件夹，我们重最上方迅速滚动到最底端，

我的想法：

如果能让眼前的内容可以优先呈现，那么就必须有能力控制资源载入的顺序，我猜想这需要设计一套新的资源排架构，取代原来系统默认行为，让 Eagle 有能力控制资源载入

的优先顺序。

解决方案：

这套系统除了基本的任务排程功能外，还必需要包含几项重要的能力：

- 中断
随时可以取消已经发起的任务，让之后的新任务先被处理。
- 暂停
随时能暂停正在载入的任务，让其它任务先完成。
- 插队
直接把新任务安排在第一位执行，不需要等待其它任务完成。

由于这套系统的加入，为 Eagle 带来了更好的使用体验，比如：

1. 更快呈现当前视图内容
2. 减少不必要的载入行为

注：这个新架构是最新 Eagle 1.11 版本的重要更新项目之一。

挑战：传统无限滚动的瀑布流列表无法快速滚到底端或任意位置，需要自行开发一套方法，支持随时滚动定位功能

我希望「列表无限滚轴可以快速滚到任意位置（解释我如何解决花瓣、Pinterest 这类网站的通病）」

[痛点][绝大部分的瀑布流/无限滚动的网站的通病][无法直接滚到底][无法指定滚动到特定位置][每次刷新就得重新滚动]

很多图片收集网站（如：花瓣、Pinterest）都采用瀑布流布局来呈现列表的内容，而这些网站通常都存在一个功能上的严重缺陷，那就是我们无法滚动列表到自己想要的位置。假如你在花瓣创建一个画版，保存了 3 万张图片，我们得慢慢滚动页面到底端触发载入，一页一页载入到最下方才能看到最底端的内容，这个过程不管网速再怎么快，通常都要花上三到五分钟左右的时间才能到底，假如你不小心切换了分页或者重新刷新页面，那么刚刚的努力得再来一次...

我的想法：

一般在处理无限滚动问题时，我们可以采用总数量的方式来推算列表的总高度，比如 1 千个内容，每个内容占用 100 像素高度，当前屏幕宽度可容纳一排 5 个，那么列表总高度就是 $1000 / 5 \times 100$ ，20,000 高度。但是，像是瀑布流这种高度不固定的布局方式，我们无法采用这种简单方式来推算，而要在一开始就将所有数据下载下来再进行计算，也太过暴力，不是考虑的方式。

解决方案：

1. 自己开发一套滚轴系统
2. 记住用户上一次查看的位置

挑战：滑杆控件默认为线性操作，在缩放功能操作体验极差，需自行重新开发 Eagle 专用滑杆控件

[痛点][1-100% 这段区间是最常被使用的][采线性设计的滑杆1%800%会造成110%%可以使用的空间很少][非常难操控]

我的想法：

我希望「缩放图片滑杆更好使用，我重新开发滑杆控件」

解决方案：

- 1.

[图 1.9.x 版本的样子]

挑战：减少不必要的缩放操作

不是所有素材都适合默认采用 100% 大小浏览，软件应有能力智能判断默认比例，我希望「查看手机截图、UI 设计稿不需要额外缩放，所以我设计智能缩放功能」

[痛点][错误的默认值][传统看图工具会把所有图片一视同仁][默认方式很笨][每次打开都要自己缩放到合适尺寸][手机截图][UI设计稿][网页截图][App 长截图]

我的想法：

让 Eagle 能依据智能判断图片的类型，计算默认应该采用什么方式来呈现

解决方案：

1. 智能判断手机截图、App UI 设计稿特征（特征表）[比较图]
2. 智能判断网页长截图特征[比较图]

挑战：更全面的 GIF Viewer，支持暂停、逐帧查看、片段重复等功能

许多行业都会用到 GIF 文件，比如交互设计、动效设计，平常我们需要详细的查看 GIF 素材，比如将自动循环播放的动画暂停，或者逐帧查看每一帧细节、调整播放的速度。由于目前市场上免费、付费的 GIF 播放器体验差强人意，想要仔细查看 GIF 文件这事情仍是许多设计师的恶梦。

我的想法：

目前能找到开源的 GIF Viewer（比如 [gif.js](#)）并不强大，而期望能解决的任务这些套件也都无法提供，所以我只能想办法自己开发一套，好在过去念计算机时学过图像编码相关的知识，我想自己开发一个能够解析 GIF 编码的播放器，应该是可以尝试的方向。

解决方案：

最后成功的把 GIF 播放器开发完成，播放器功能也相当完整，例如支持：

1. 暂停查看

在浏览 GIF 图片时，可以透过点击图片或者使用空格键，暂停查看某一帧内容。

2. 逐帧查看

如果你想要快速切换或查看某个区域，你可以拖拽下方进度条逐帧查看。

3. 框选播放范围

有时候我们会希望重复播放某一区段的动画内容，现在你可以透过左右两侧的裁剪工具，设置你想要重复查看的区段，播放器会自动在此范围内重复播放。

4. 调整播放速度

如果 GIF 默认的速度太快了让我们无法仔细查看某个细节，你可以在右下角调整动画播放的速度，软件支持播放速度调整（0.25x - 0.4x）。

5. 自定帧数作为封面

将影片暂停在您想设置为封面的画面，点击右下角“...”按钮，选择“将当前帧设为 GIF 封面”。

挑战：符合各种用户不同的操作习惯

[痛点][Eagle 是针对素材场景设计，但不免会遇到来自不同领域的用户，它们拥有不同的操作习惯][默认值无法满足所有人]支持软件大部分功能都要能够支持多种操作方式、自定义设置

我的想法：

想做出符合用户习惯的软件，最重要的就是知道产品潜在用户群平常是怎么使用同类型软件的，针对已经习惯的功能，Eagle 不应该重复发明新的操作逻辑、方式，比如常用的快速键设置。有些功能在不同领域也许没有固定的操作逻辑，针对这样的状况，软件

应提供足够程度的自定义能力，让用户可以将软件的功能设置为他们已经习惯的操作方式。

解决方案：

1. 大部分功能都提供多种操作模式
2. 绝大部分功能都支持偏好调整

用户可以在偏好设置对界面进行一定程度的调整，隐藏自己不需要的功能。

3. 主动询问用户偏好方式

有些常用但是每种族群偏好不同的设置，很难透过「良好的默认值」设计达成，把这些功能设定放在偏好设置页面，用户也很难发现

本节最后：

我非常重视产品的交互体验，平常在使用其它公司开发的产品时，也时常将操作体验上的不便、易用性问题记录下来，主动反馈给他们。工作这几年一直是这样，因此当我在开发自己的作品 Eagle 时，也是用相同甚至更严格的方式，尽可能让体验做到更好。

实际上，大家也能从上面的分享中发现 Eagle 在开发架构上考虑了非常多细节，对性能也是极其追求。这其实和一些称懂些技术的评论者，单纯以外部结构推测，所得到的结论是截然不同的，“仅凭外表就评价一个人，直接把假设当结论，这样既不专业，也不负责”。

一人独立开发及设计产品，会遇到的挑战还有非常多，每一次遇到挑战背后的细节，都是很棒的学习经验，很高兴能透过这篇文章，总结了这些年的实战经验，限于篇幅，这个部分就分享到这里，假如大家对这些细节感兴趣，可以持续关注 Eagle 的改版日志，我都会在改版时尽可能向大家说明。

随著技术的进步，一定会有更棒的方法是我还不知道、还没来得及考虑的，上面提到的各种挑战，在未来我还是会继续保持初心，持续优化，也欢迎大家和我分享你的想法。

备注：截至目前为止，其实还有一些大功能是我还没有成功攻克，希望未来能顺利解决：

- 如何基于浏览器技术，支持所有类型视频文件播放（限制：不能事先转档、不能产生额外储存成本）
- 如何时时监测局域网硬盘文件修改事件，让 Eagle 可以在 NAS 上多人同时编辑修改（限制：不能使用轮询方式开发）

- 如果使用多种颜色作为条件，筛选相关素材（非技术问题，主要还在考虑 UI 要如何设计）
- 如何让 PS、Sketch 创建的新文件，可以直接保存添加到 Eagle