



PROBUILDER

SEAMLESS LEVEL DESIGN

(THE POST-MORTEM)

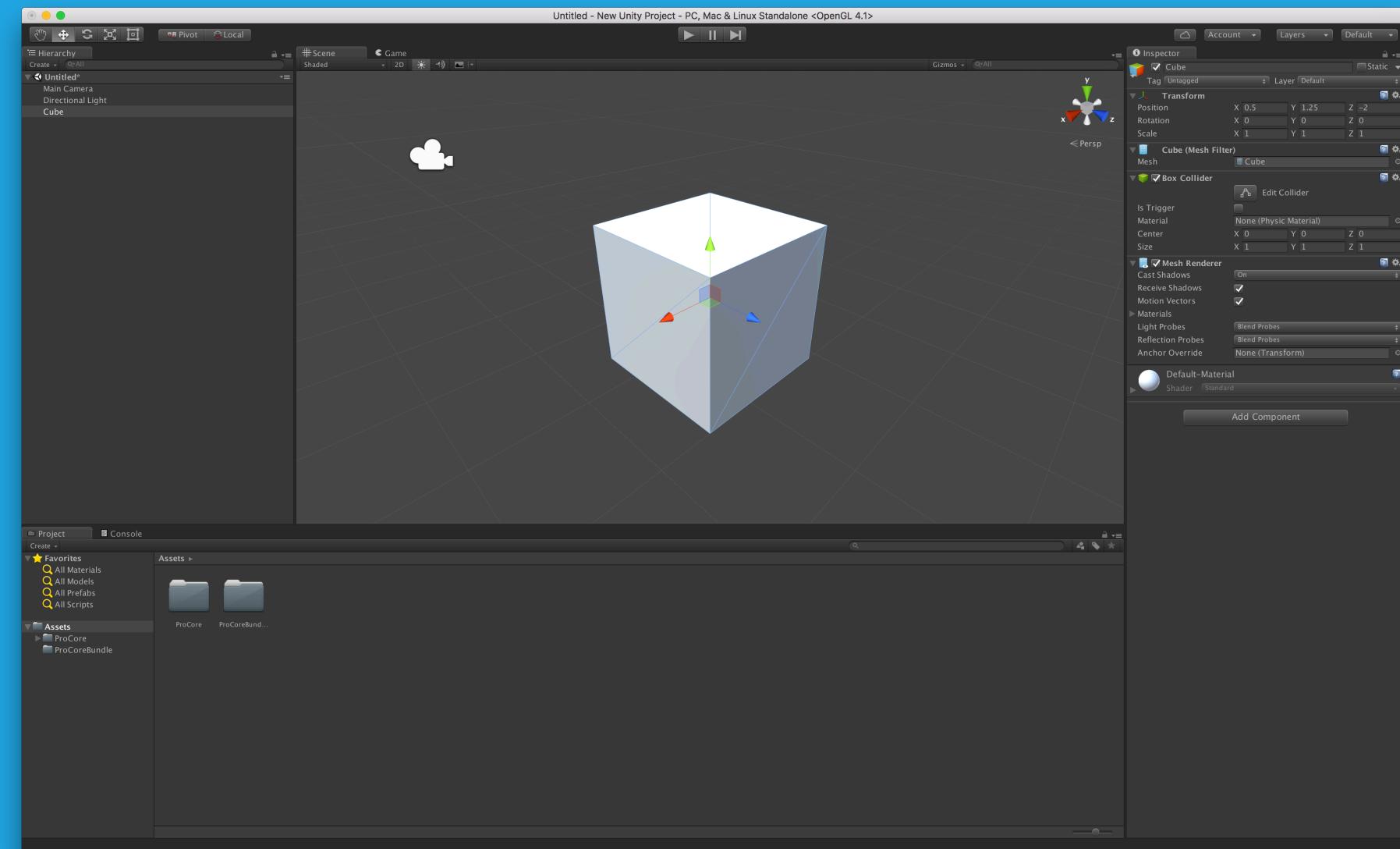
PROJECT SUMMARY

- » What are Unity Plugins / Asset Store?
- » What is ProBuilder
- » What were it's initial goals
- » What were the challenges getting there?
- » How did we overcome those challenges?

PROJECT SUMMARY (CONTINUED)

- » Where is it now
- » What are it's future goals
- » Lessons learned from pricing

WHAT ARE UNITY PLUGINS / ASSET STORE?



EXTENSIONS, TOOLS, MIDDLEWARE (ETC)

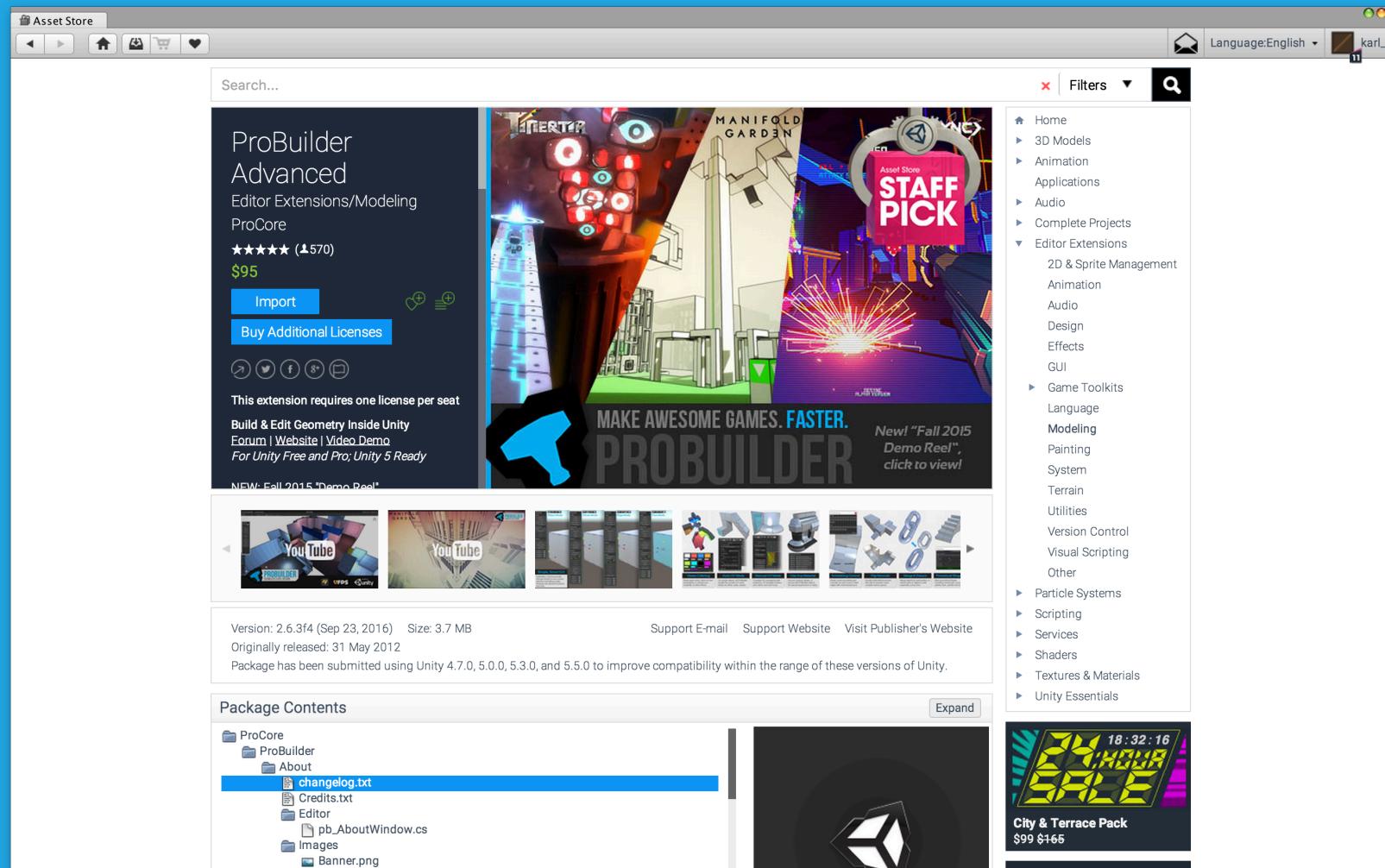
- » Plugins are tools built on top of software to add functionality
- » Exporting OBJ files in Maya
- » An image filter in Photoshop
- » A new keyframing tool in video editing software

UNITY IS BUILT FOR PLUGINS

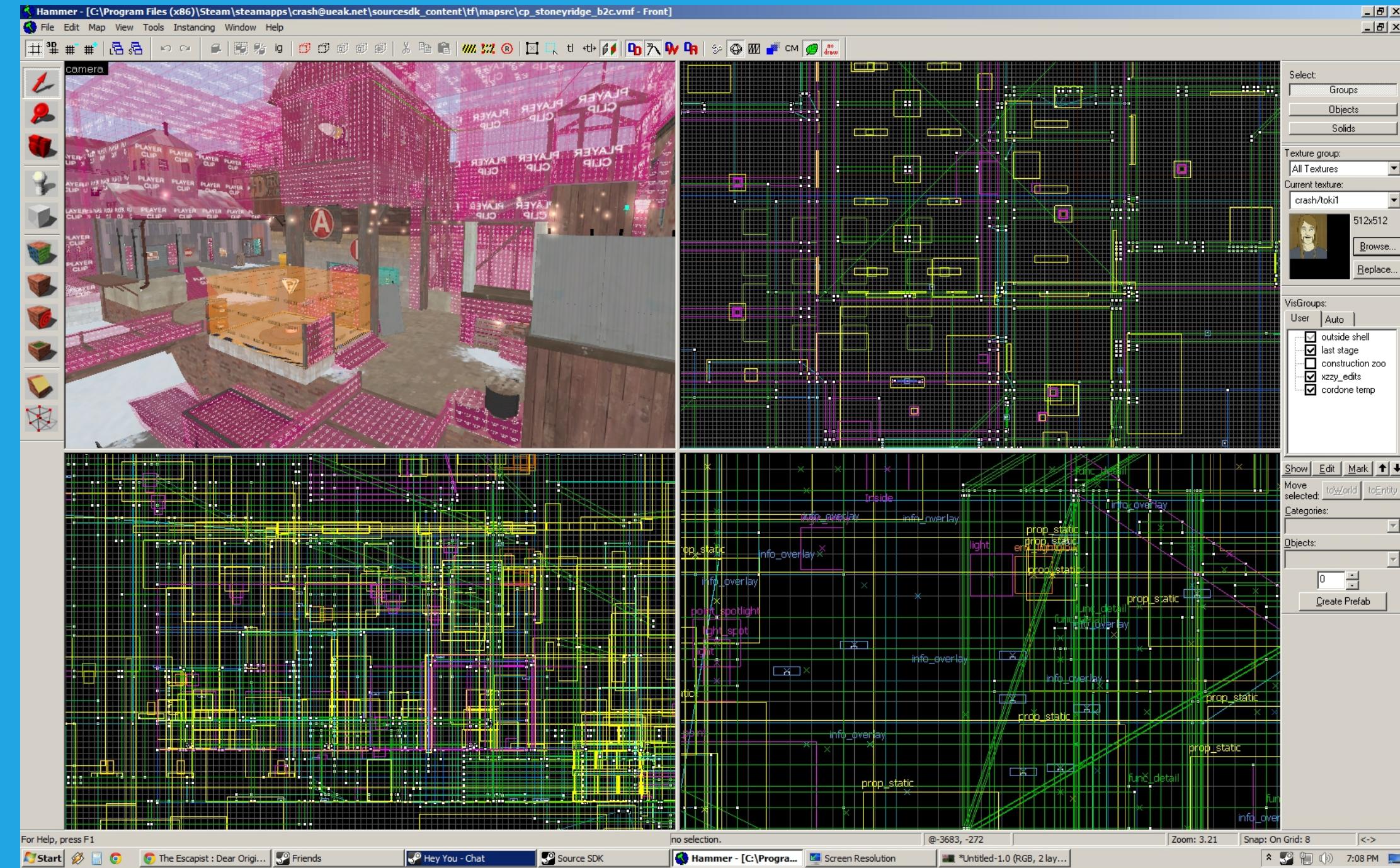
- » Much of the editor is built with the same API made available to developers
- » C# makes for very fast iteration (ie, not re-compiling the editor for every change)
- » Excellent documentation

ASSET STORE

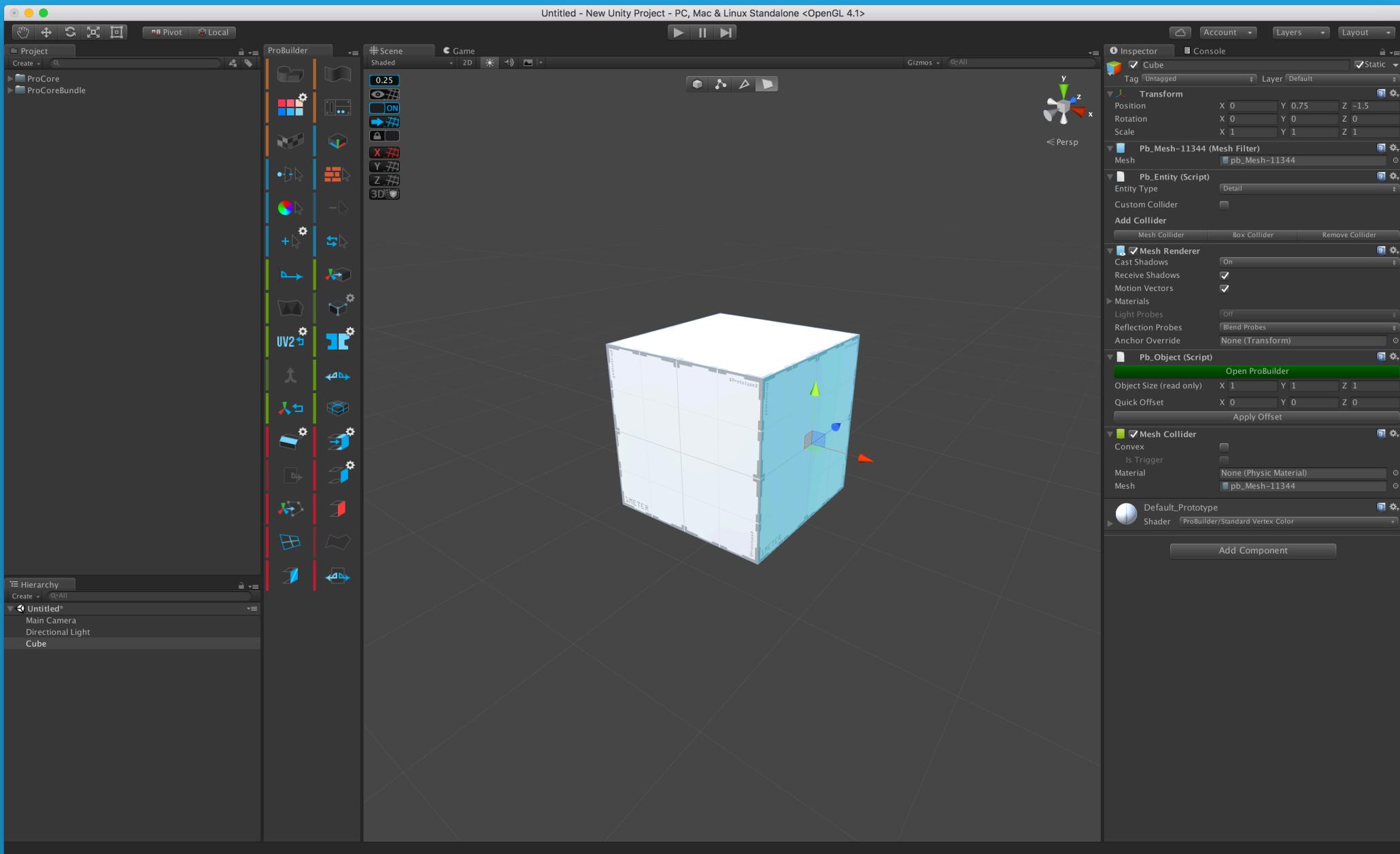
» The asset store is a place to buy and sell plugins (and art)



WHAT IS PROBUILDER?



WHAT IS PROBUILDER?

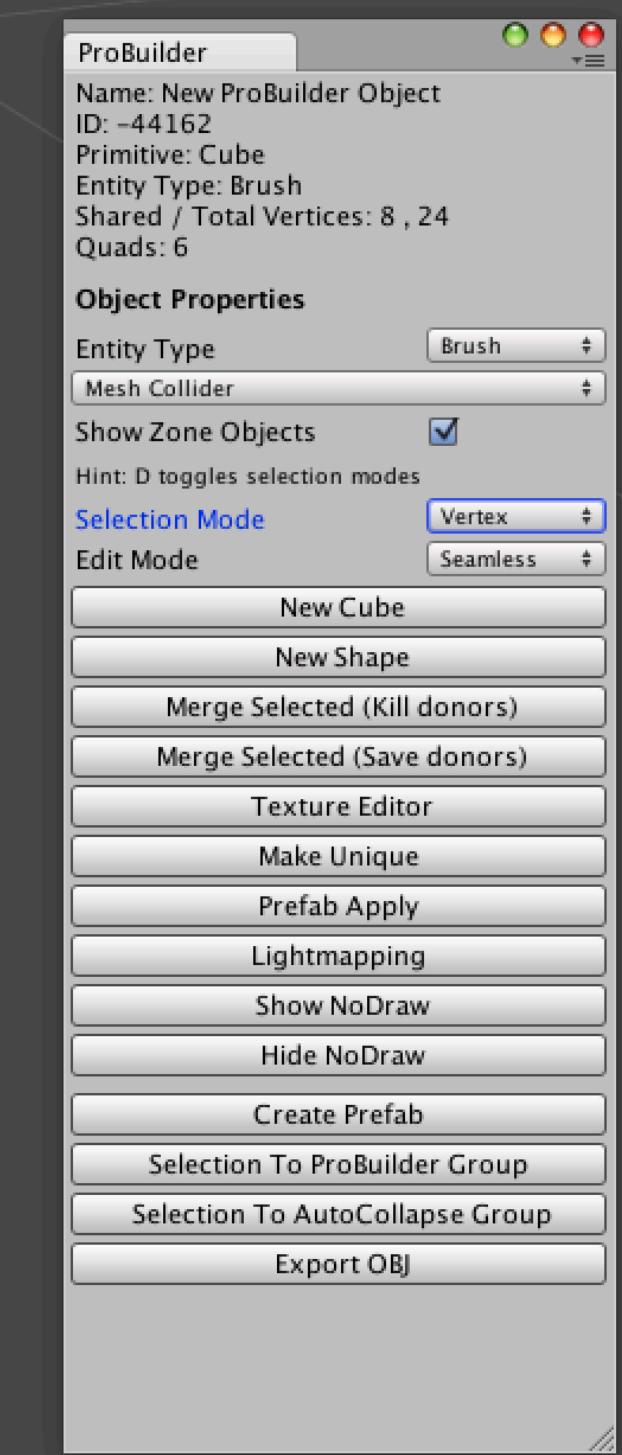


WHAT IS PROBUILDER?

- » 3D Modeling in Unity
- » Powerful tools with a newbie friendly approach
- » Makes iterating on level designs lightning fast

SCOPE OF WORK

- » ProBuilder 1.0:
 - » Move faces and vertices
 - » Basic texturing (what I was originally brought on to do)
- » Feature creep
- » ...but in a good way



PROBUILDER 2.0

- » Complete re-write
- » Met initial goals, then some
- » Overally successful launch
- » That sounds great! No problems then?

AN ASSORTMENT OF CHALLENGES (THE INTERESTING ONES)

LONG BETA PERIOD

- » It took much longer than expected to exit beta stages
- » Feature creep was a part of it
- » Largely due to desire for backwards compatibility

BACKWARDS COMPATIBILITY

- » ...is the worst
- » Needed to handle
 - » ProBuilder 1 scenes
 - » Beta 2.x scenes
- » Core parts of ProBuilder still constantly changing
- » I didn't plan for it well

PROBLEMS

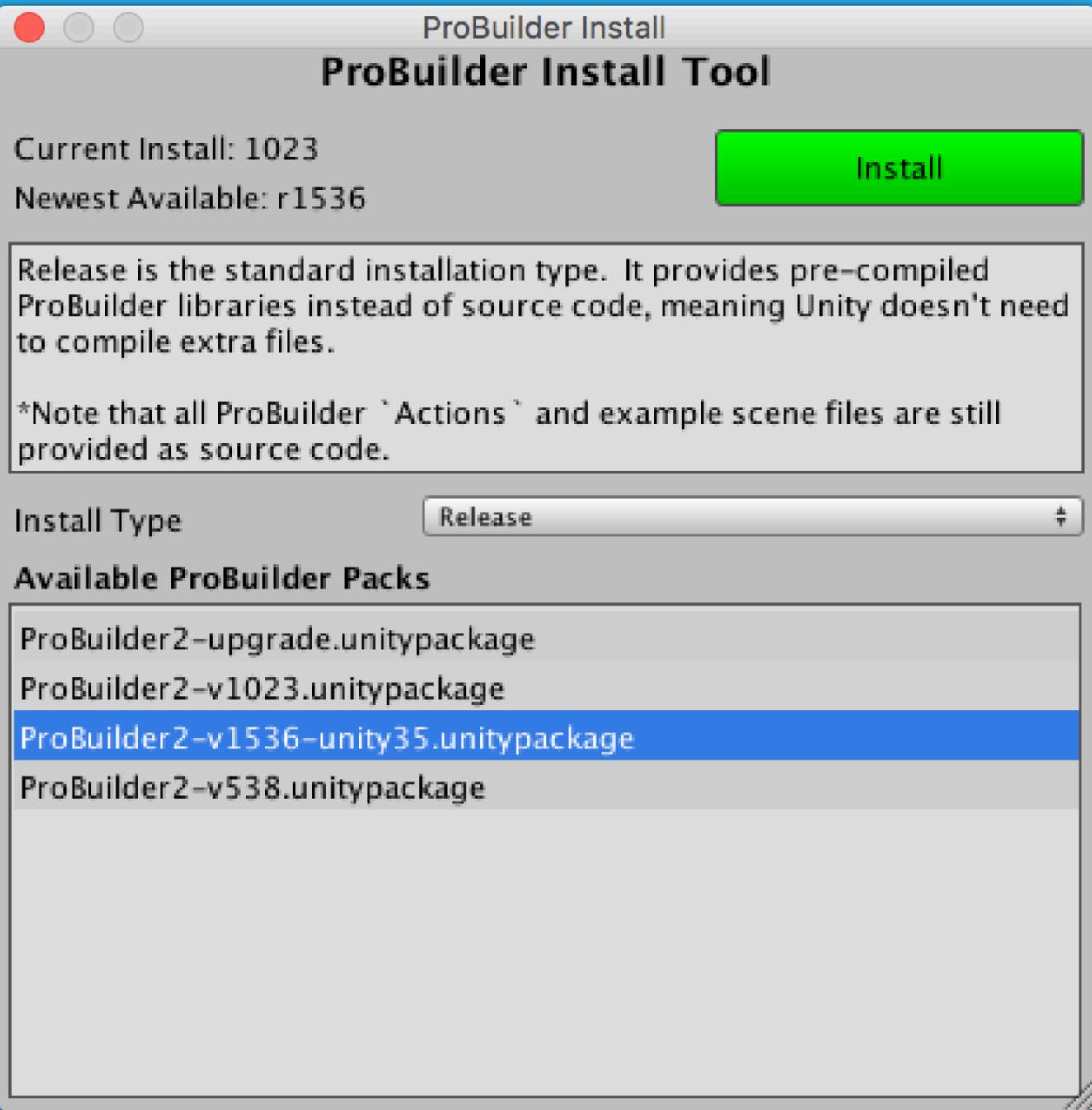
- » Lots of different versions of ProBuilder (each more incompatible than the last!)
- » No standard ProBuilder file format
- » Every upgrade was a moving target
- » Spent a lot of time writing bridge code

A MOVING TARGET

- » Didn't define a file spec early
- » Results in incredibly complicated upgrade path
- » Instead of cutting losses just tacked on more

WHAT WE DID

- » Wrote a lot of one-off conversion scripts
- » Spent a lot of time on support, often manually upgrading projects for users
- » Time consuming and draining
- » At some point settled on an installer



Nice when it worked

THE INSTALLER DEBACLE

- » Not in line with Unity workflow
- » Super fragile
- » Incredibly fragile
- » Not user friendly (still required user interaction)
- » Complicated packaging & distribution

INSTALLER WASN'T ENTIRELY BAD

- » Allowed greater flexibility in changes (could move and delete stuff)*
- » Arguably looked more professional than readme with upgrade instructions

*Moving and deleting exposed files wasn't great either

THE SOLUTION

- » Stop changing the file spec
- » Be aware of compatibility breaking changes
- » When removing things, mark deprecated instead of deleting

THE INTERFACE

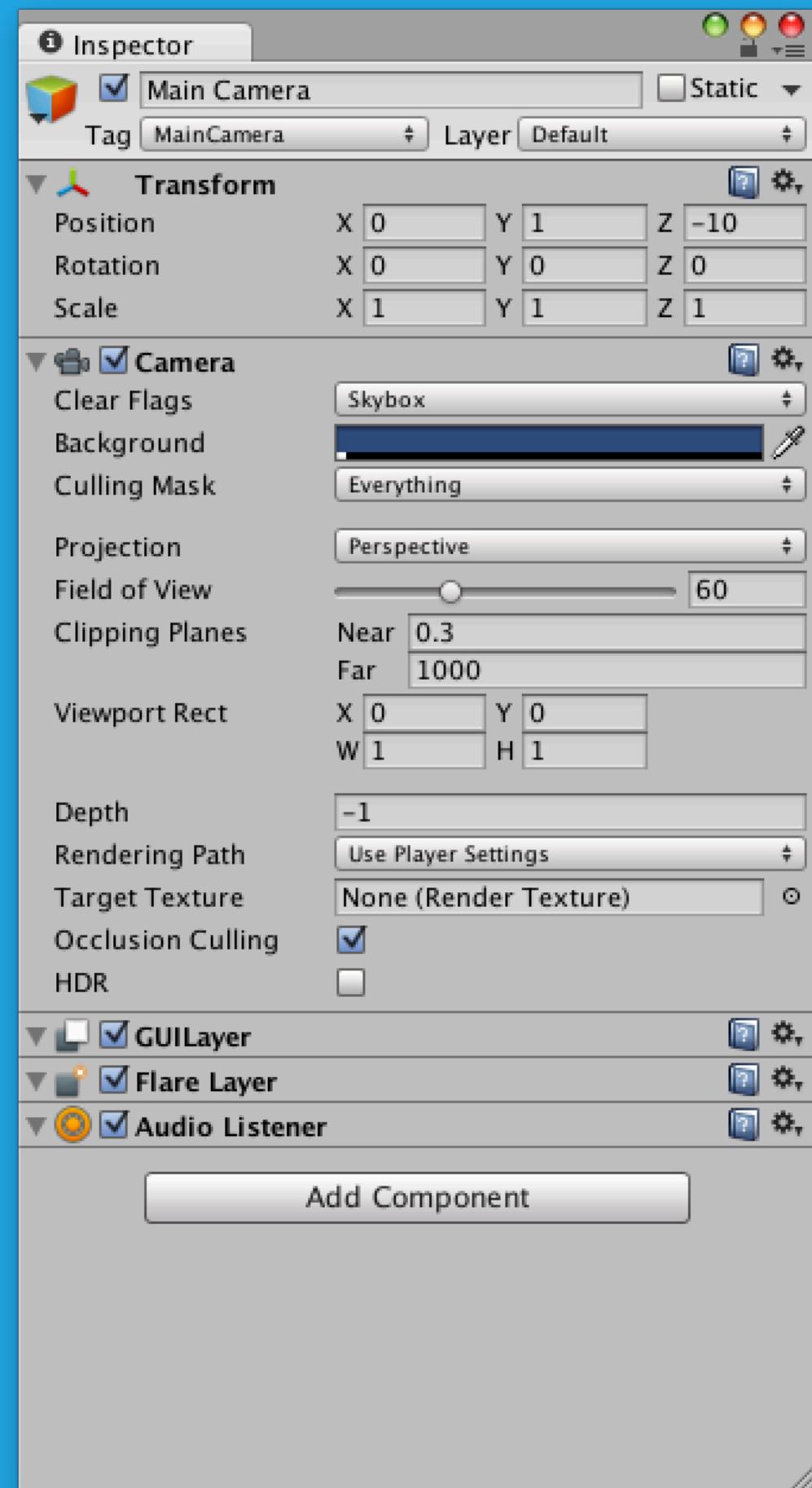
- » Has seen 3 major iterations
- » Many many smaller tweaks

INTERFACE GOALS

- » Feel native to Unity
- » Be functional
- » Look pretty

INTERFACE CHALLENGES

- » Unity's default interface isn't prettiest¹
- » Needs to work with both dark and light skins



¹ In our opinion

INITIAL OFFERING

Icons!

- » Looked nice
- » Not many actions; learning curve was small
- » New menu items now required two people



ICONS CONTINUED

ProBuilder starts getting more features added.

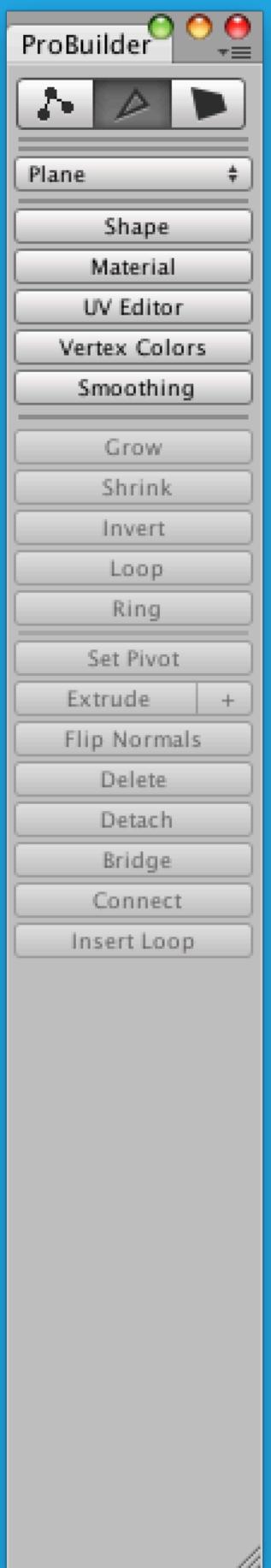
- » Pain to keep up with new menu icons
 - » Complex mesh operations are hard to represent in 32px
 - » Users had to memorize large array of icons



SWITCH TO TEXT

Too many items overwhelms icon approach, swap to dynamic text-based menu.

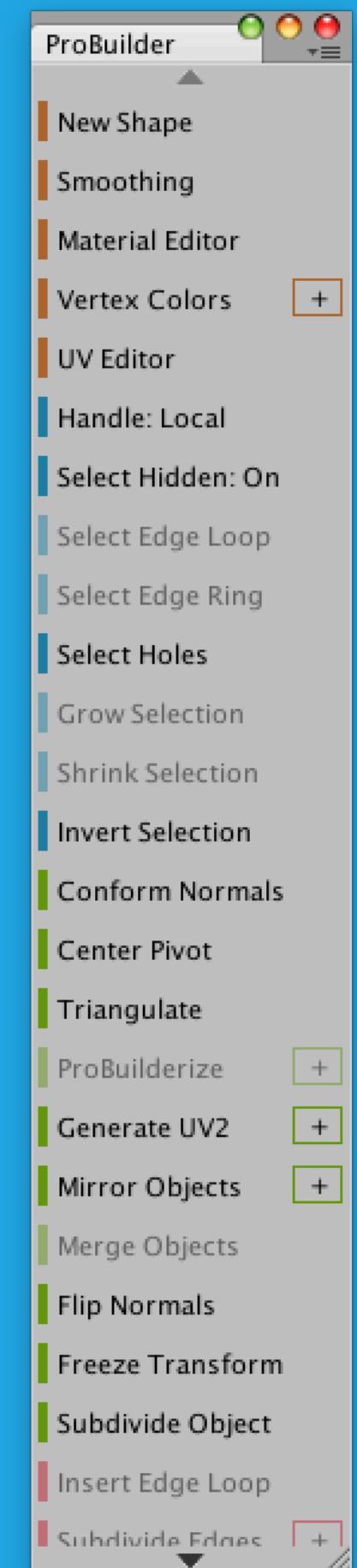
- » Was easier to handle in code
- » Adding new items was fast
- » Could hide and show items based on context
- » Fit Unity UI (but looked meh)



THE GREAT COMPROMISE

Or rather, the lack there-of.

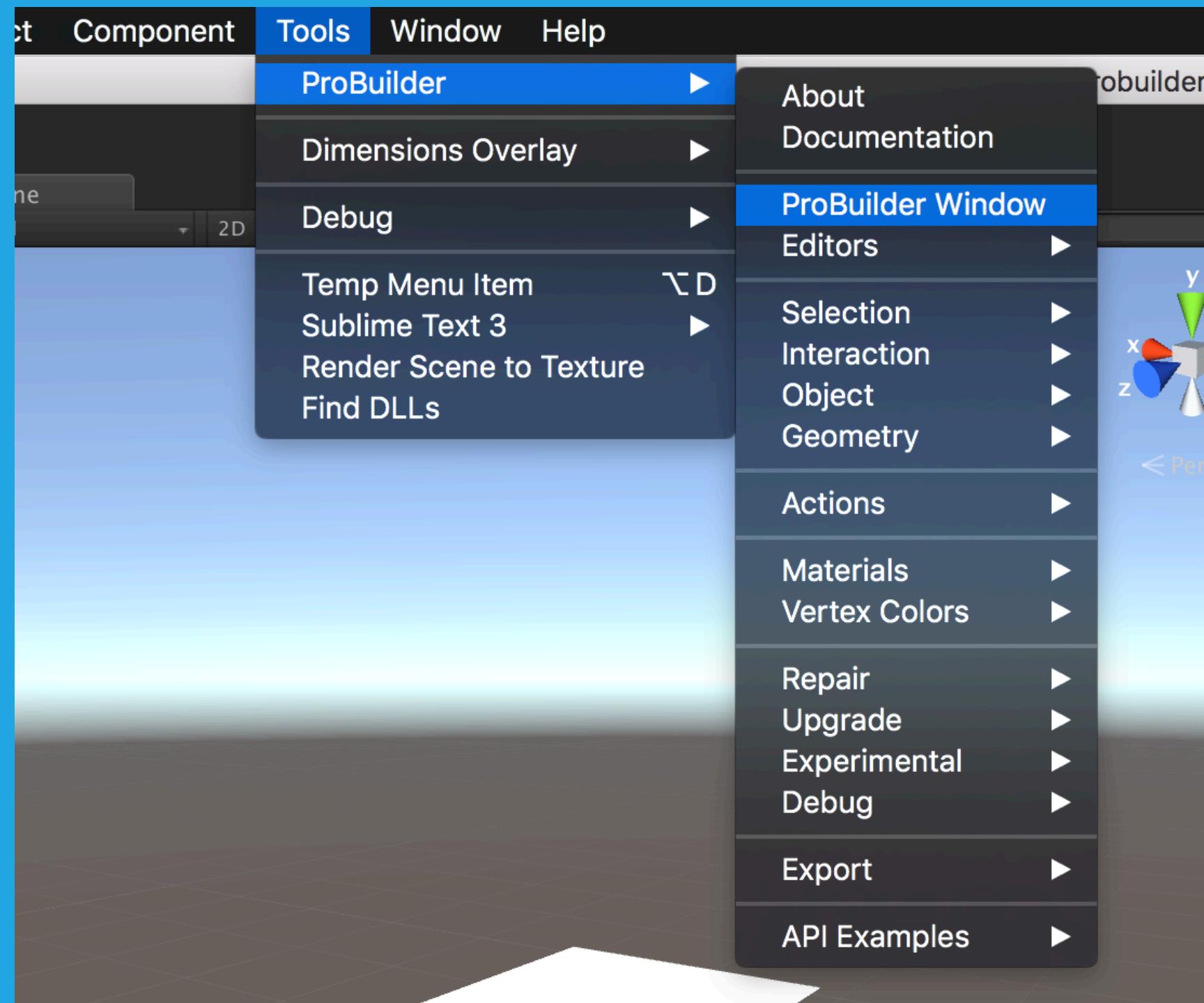
- » Option for icon or text based
- » Still context based
- » Re-written with extensibility in mind



MENU ITEMS NOW GENERATED

Menu bar used to be the wild west - no consistent strategy.

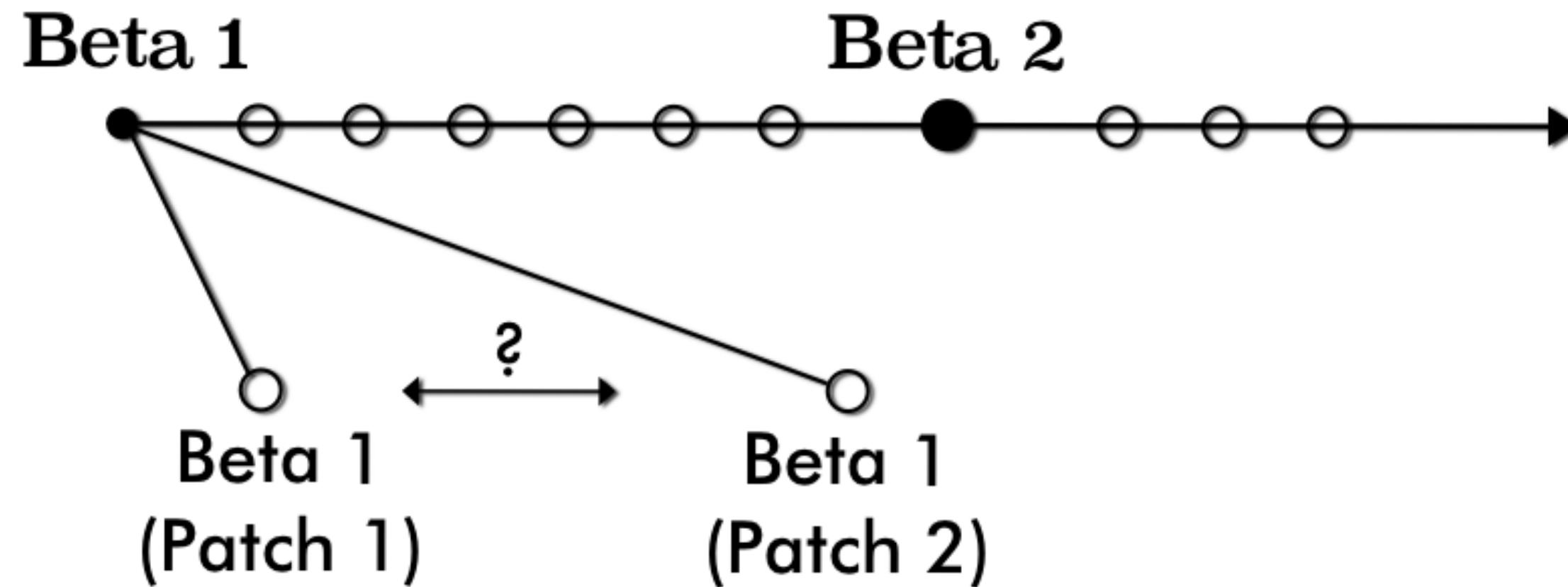
- » Menu code generated directly from toolbar source
- » Far more flexible, and less chance for inconsistencies



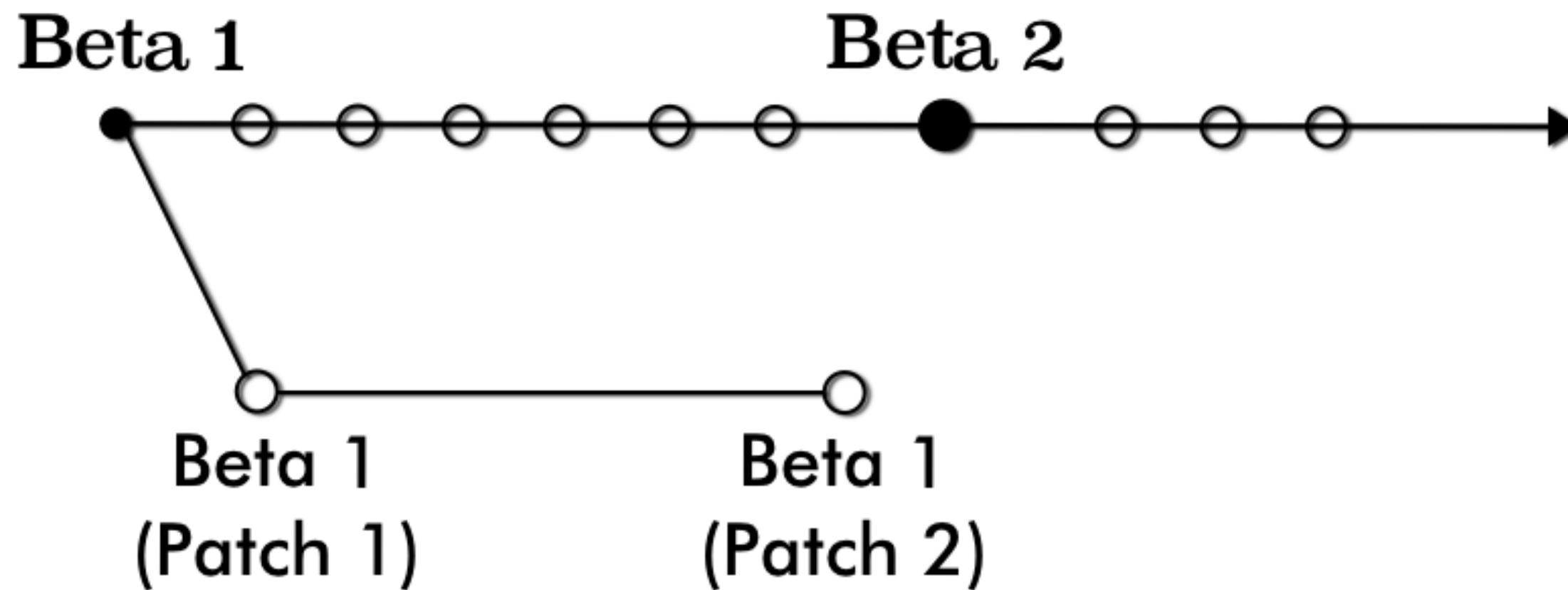
MANAGING RELEASES & PATCHES

- » Was using SVN as glorified Dropbox
- » No tags/branches for releases
- » Patches were few and varying in quality

MANAGING RELEASES & PATCHES



MANAGING RELEASES & PATCHES



RELEASES & PATCHES: THE GOOD

- » Kept changelog & and version info up to date
- » Maintained bin/Release vigilantly (albeit by hand)

RELEASES & PATCHES CURRENTLY

- » Now using git
- » Releases are tagged
- » New features developed on branches
- » Master branch always compiles

DISTRIBUTING PROBUILDER

- » We have to support a wide range of Unity versions
- » ProBuilder is distributed in DLLs
- » Currently we build for 5 separate targets

BUILDING AND PACKAGING INITIALLY DONE BY HAND

- » Tedious
- » Error prone
- » Did ensure I tested each target

ENTER BUILD SCRIPT

The build process is ripe for automation:

- » Less room for error
- » Significantly reduced build times
- » Adding a new build target is trivial

DOWNSIDES OF BUILD SCRIPT

Not too many to speak of

- » High startup cost
- » Have to be diligent about human testing

THE MESH CLASS IT DOES WAY TOO MUCH

- » Currently handles:
 - » Storing vertex & triangle data
 - » Submesh and material structure layout
 - » Normals calculation
 - » Element selection
 - » Collision volume sizing & placement

MESH CLASS

- » Makes changes to these methods scary (don't want to break existing models)
- » Couples data (vertex positions, triangles, etc) with operations (bad form)
- » File size is much bigger than it needs to be (hard to navigate)

MESH CLASS STILL DOES TOO MUCH

- » Cost of "just get it working then optimize"
- » Still working on paring down the monolith
- » Automated tests help with this process

DOCUMENTATION

A TOOL WITH POOR (OR NO) DOCUMENTATION ISN'T ANY GOOD

- » ProBuilder didn't ship with a manual until sometime last year
- » Always had web docs, but they were infrequently updated
- » Support forum has been active and monitored regularly

HOW IT USED TO BE

- » One gigantic word document
- » Then one gigantic markdown document
- » Updating was a pain
- » Navigation was also a pain

LEVERAGING STATIC SITE GENERATORS

- » Enter mkdocs
- » Lets us break monolithic markdown into separate files
- » Abstracts web stuff from technical writing
- » Easy to update
- » Spits out pretty & functional website

ProBuilder Documentation

Docs » Home

Home

ProBuilder Advanced vs. ProBuilder Basic

Optional: Register for access to Source Files and Online Downloads

Helpful Links and Info

Fundamentals

Toolbar

Overview

Tool Panels

Selection Actions

Object Actions

Vertex Actions

Edge Actions

Face Actions

Element Actions

Texture Mapping

Overview

UV Editor Toolbar

Auto UVs Actions

Manual UVs Actions

Menu Actions

Preferences

Troubleshooting

Faq

Upgrading

What Upgrade Procedure Should I Follow?

Standard

DLL Rename

Upgrade Kit

Prototype

Credits

Changelog

Next »

What is ProBuilder?

ProBuilder is a tool for building and editing 3D objects, directly in Unity. You can use it for visual things like buildings or custom vehicles and weapons, or to make custom collision geometry, trigger zones, nav meshes, etc. The choice is yours!



ProBuilder Advanced vs. ProBuilder Basic

ProBuilder Basic is entirely free, and includes the core ProBuilder features for building and editing basic geometry.

[Download ProBuilder Basic](#)

ProBuilder Advanced is a paid extension that adds advanced features such as Edge Loop Modeling, UV Unwrapping, and specialized optimization.

[Purchase ProBuilder Advanced](#)

Important!

This documentation encompasses *all* features, actions, and tools available in both ProBuilder Basic and Advanced. If you are using ProBuilder Basic, please view the full [Advanced vs Basic Comparison](#).

Optional: Register for access to Source Files and Online Downloads

#fancy

WHAT ABOUT A MANUAL?

- » Online documentation is great, but we still need to ship with a PDF manual
- » Could manage 2 separate documents, but that's error-prone and a pain
- » Solution: python script to build PDF from website

```
#!/usr/bin/env python

import os
import pdfkit
import yaml

directory = yaml.load( open("mkdocs-manual.yml") )
pages = directory["pages"]
site_dir = "site"
pdfs = []

# gather all the docs pages in a list of tuples (category, title, path)
print("finding doc pages in mkdocs-manual.yml")

for header in pages:
    for key in header:
        if type(header[key]) is str:
            pdfs.append( ("", key, header[key]) )
        elif type(header[key]) is list:
            for item in header[key]:
                if type(item) == str:
                    pdfs.append( ("", key, item) )
                else:
                    for kvp in item:
                        pdfs.append( (key, kvp, item[kvp]) )

gen_pdfs = []

for page in pdfs:
    if page[1] == "Home" and page[2] == "index.md":
        path = page[2].replace(".md", ".html")
    else:
        split = page[2].split("/")
        path = page[2].replace(".md", "") + "/index.html"

    if len(page[0]) > 0:
        title = page[0] + " - " + page[1] + ".pdf"
    else:
        title = page[1] + ".pdf";

    gen_pdfs.append( site_dir + "/" + path )
    print("processed: " + site_dir + "/" + path)

pdfkit.from_file(gen_pdfs, "manual.pdf", options={
    'disable-javascript': '',
    'quiet': '',
    'load-error-handling': 'ignore',
    'disable-plugins': '' });

print("output pdf to: " + "manual.pdf")
```

Took about 2 weeks to solidify documentation build process

Verdict: Absolutely worth the startup cost

SALES & MARKETING

CURRENT STATE

- » 6 Different products
- » 2 Bundle packages
- » 1 free version (ProBuilder Basic)

PRODUCT**COST**

ProBuilder Advanced \$95.00

ProBuilder Basic Free

QuickEdit: Mesh Editor \$10.0

ProGrids 2 \$20.0

QuickDecals 2 \$10.0

ProGroups \$10.00

ProCore Bundle \$150.00

ProCore Utility Bundle \$60.00

Polybrush \$65.00

QUESTIONS

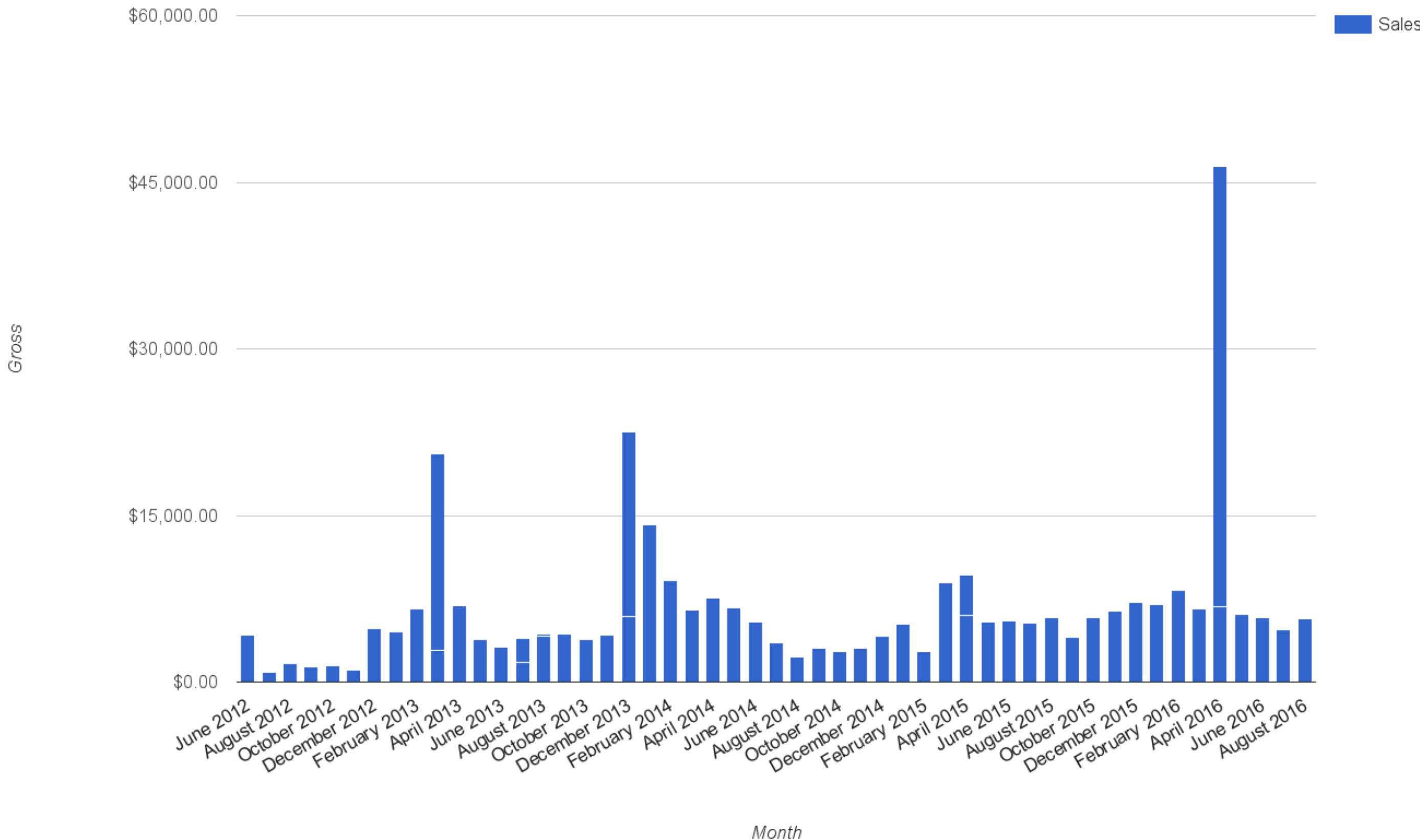
- » What's the general trend?
- » How do discounts affect sales in following months?
- » What did ProBuilder Basic do for us?
- » What effect did the Bundle have?

HYPOTHESES

Alt title: Assumptions we made prior to actually analyzing the data

- » Sales will negatively affect following months
- » A cheap ProBuilder version will sell like hotcakes
- » A free ProBuilder version will positively impact Advanced sales
- » A Bundle including ProBuilder might negatively affect sales

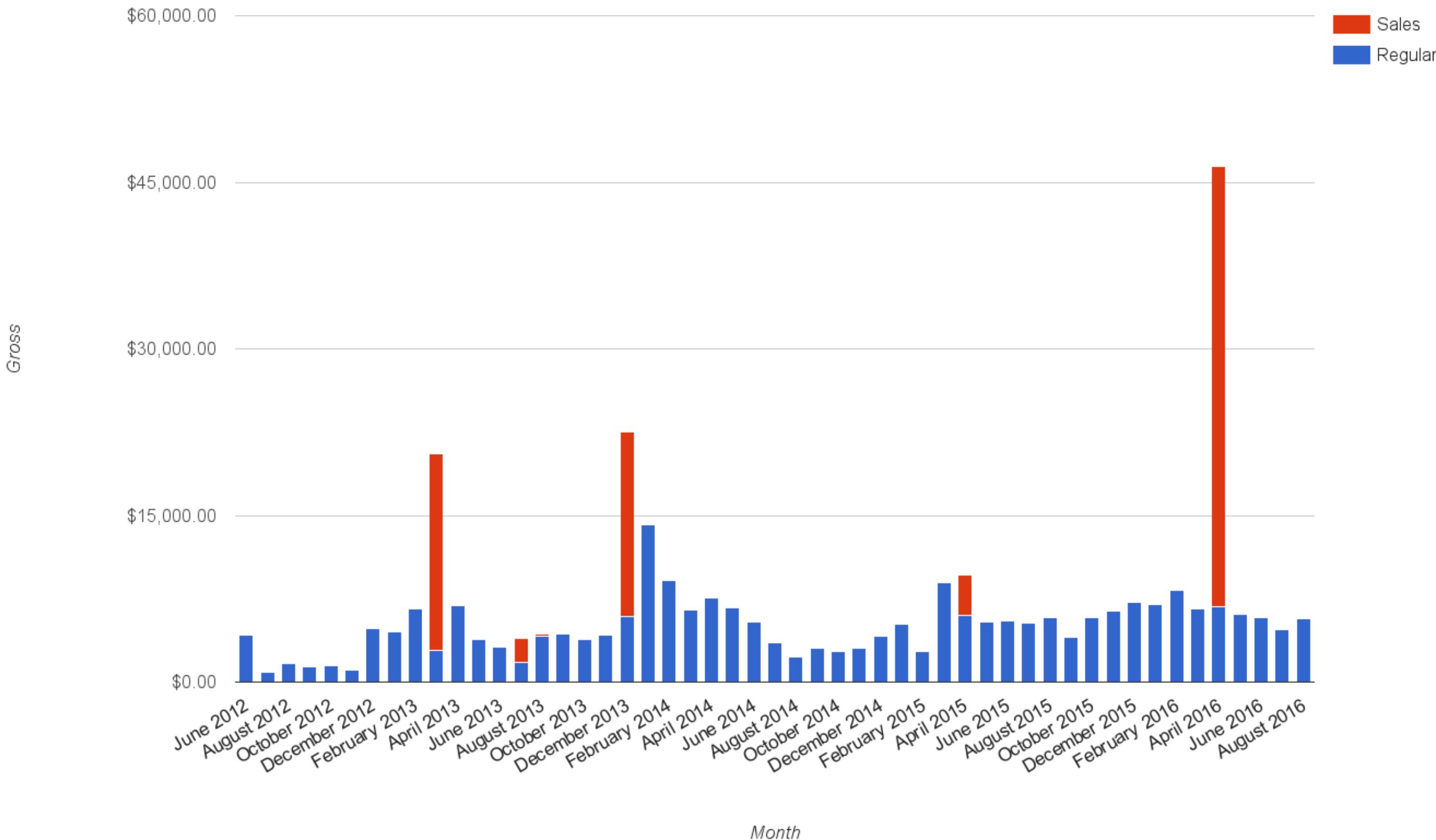
ProBuilder Advanced Sales



HYPOTHESIS

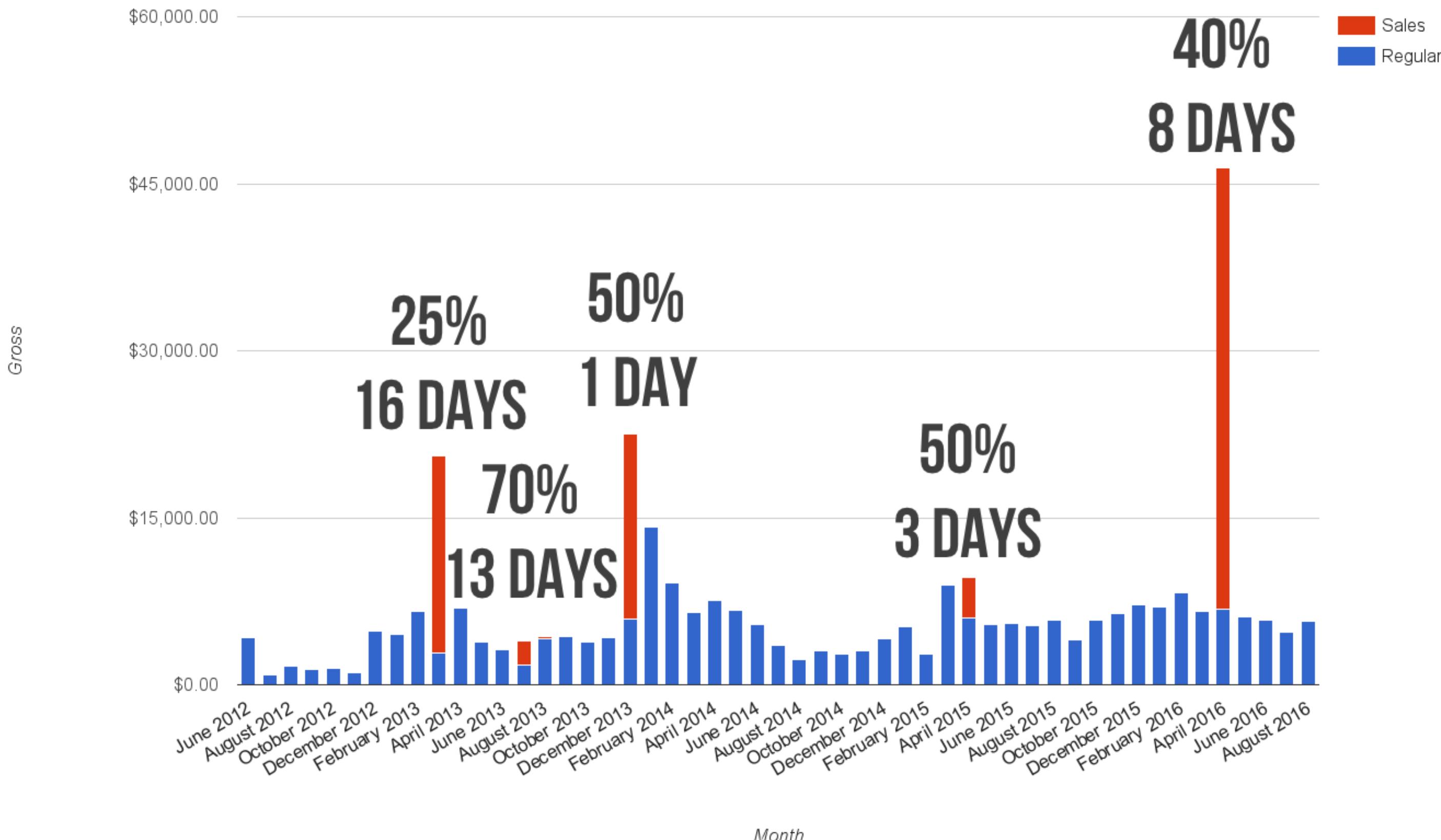
SALES WILL NEGATIVELY AFFECT FOLLOWING MONTHS

ProBuilder Advanced Sales



WHAT ABOUT SALE DURATIONS?

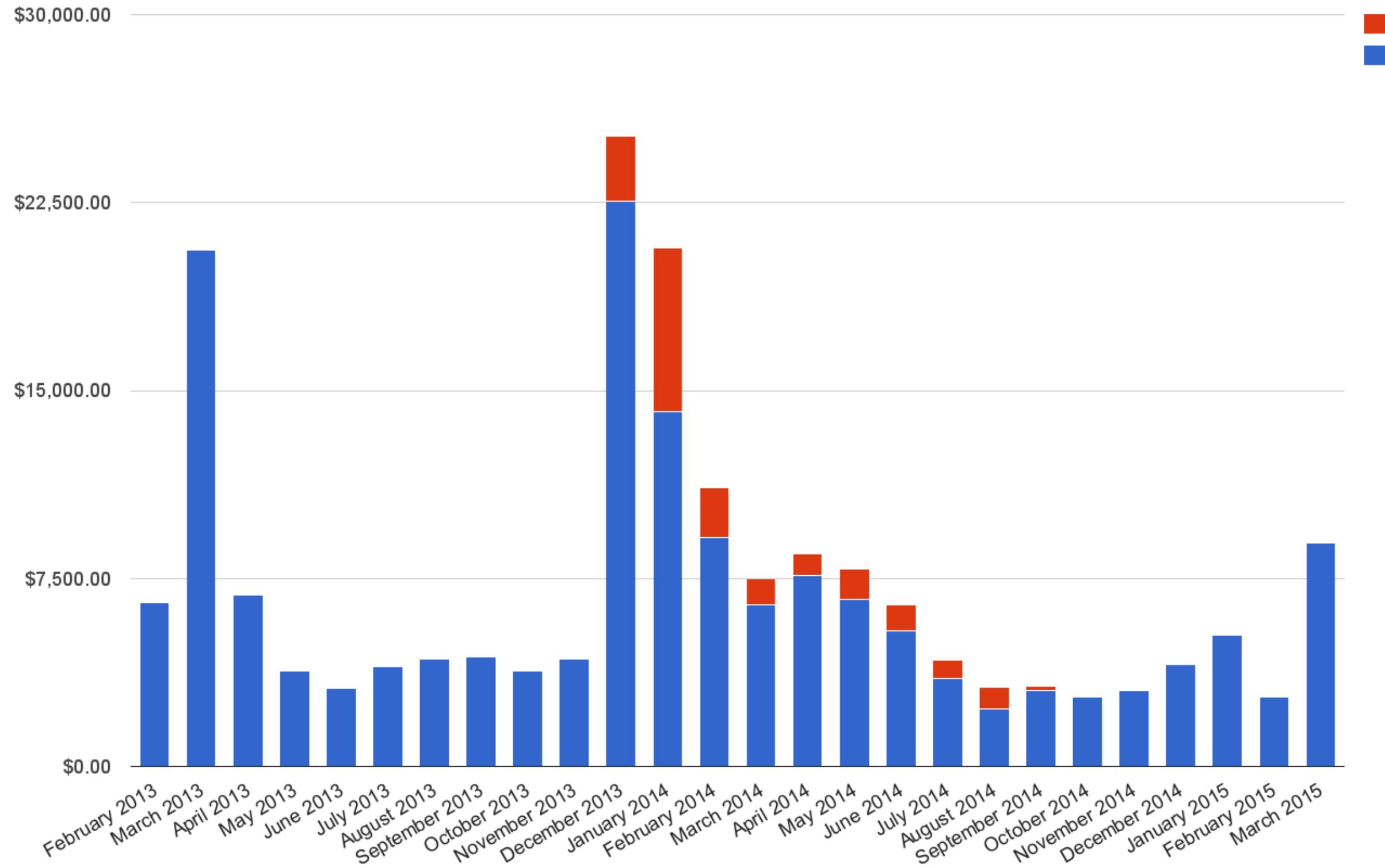
ProBuilder Advanced Sales



HYPOTHESIS

CHEAPER VERSION OF EXPENSIVE SOFTWARE
WILL BE BIG SELLER

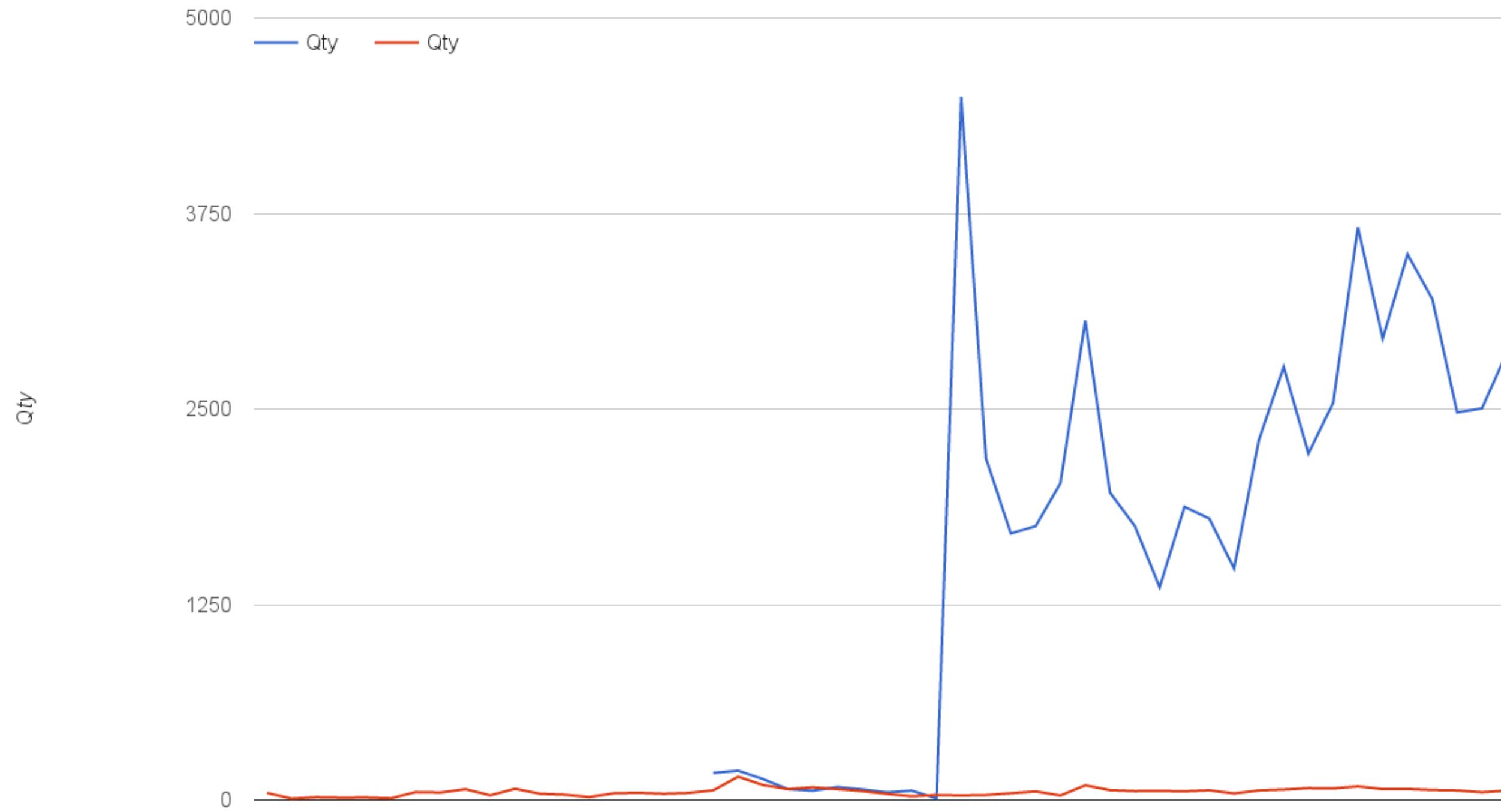
Advanced vs. Basic (Paid)



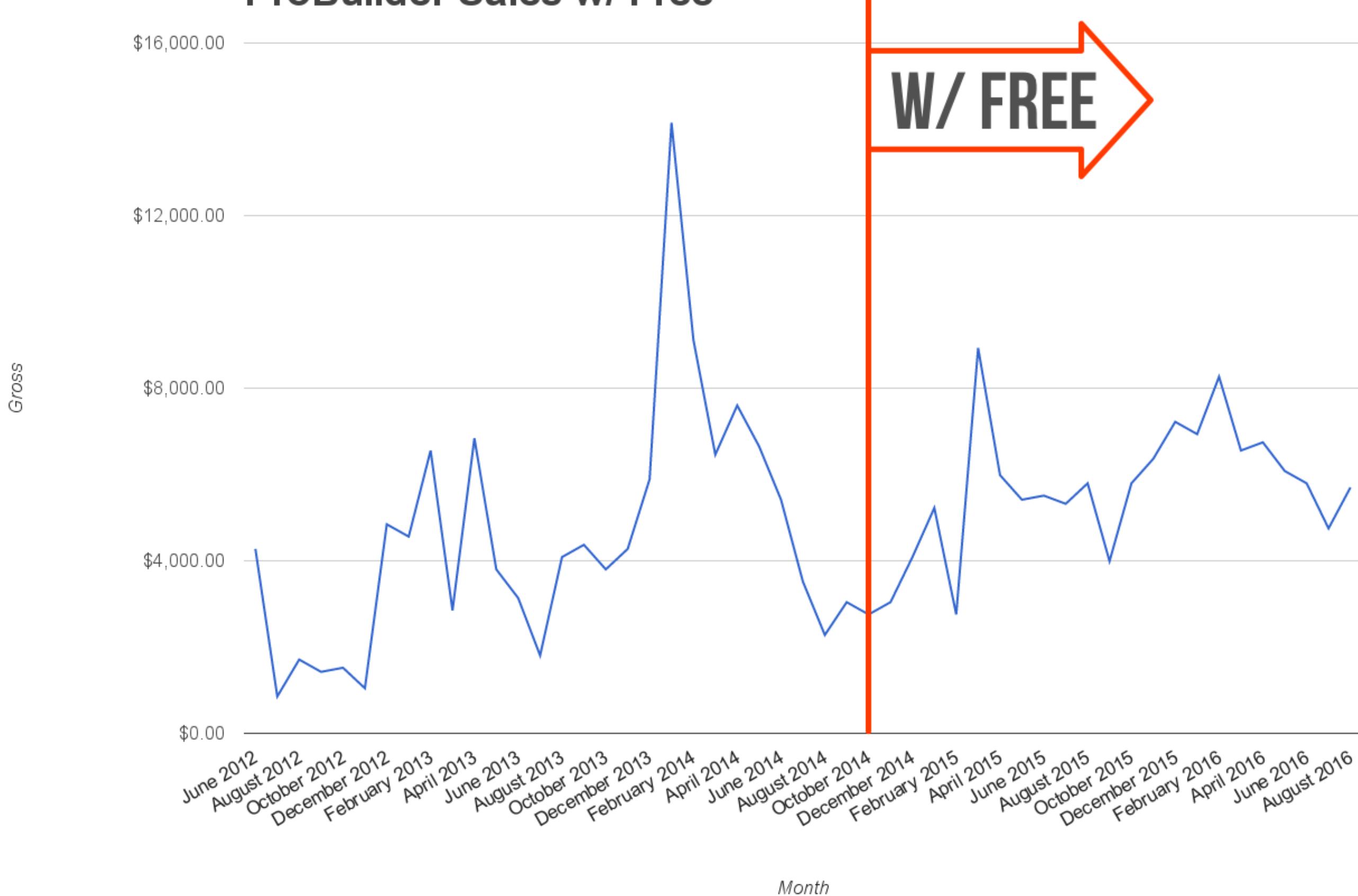
HYPOTHESIS

FREE VERSION OF EXPENSIVE SOFTWARE WILL INCREASE SALES

ProBuilder Advanced vs. Basic



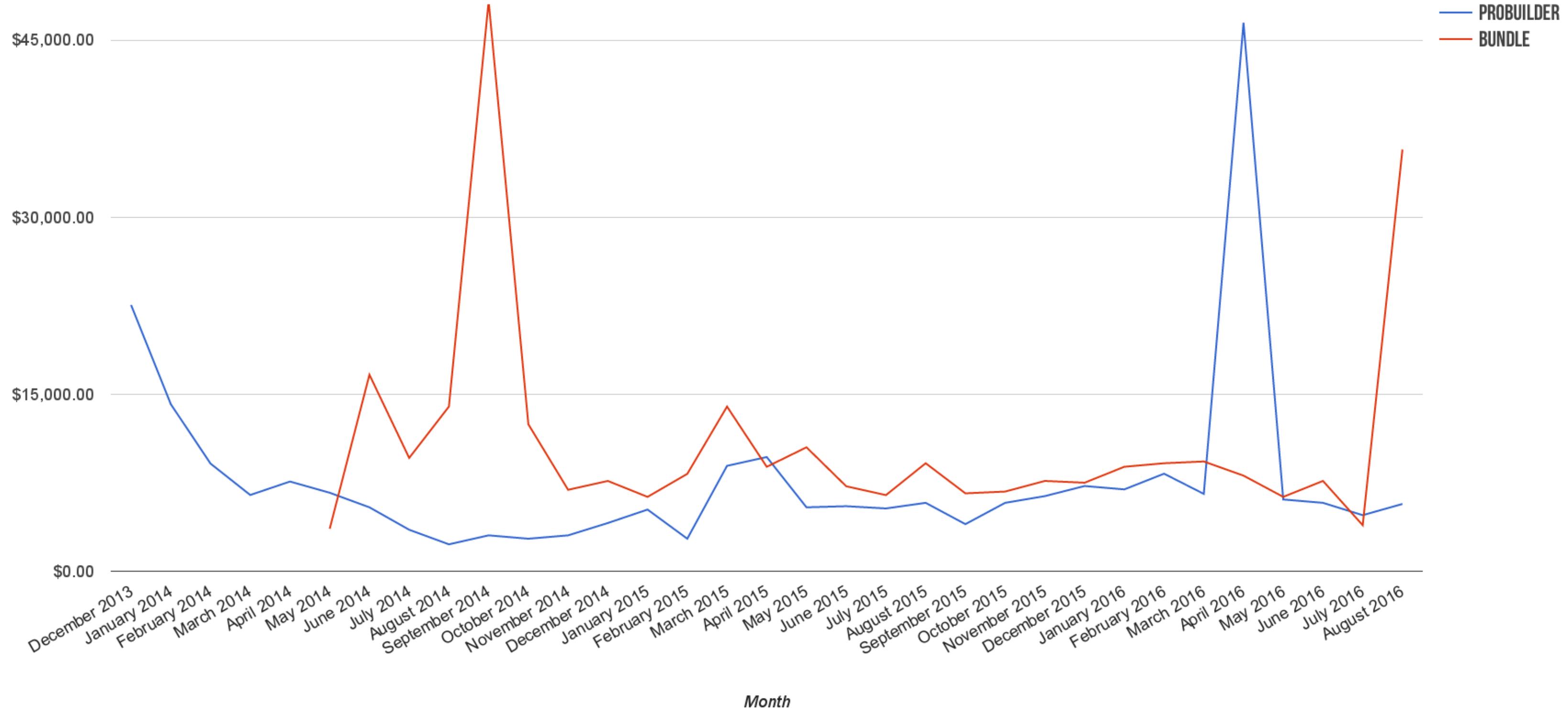
ProBuilder Sales w/ Free



HYPOTHESIS

BUNDLE MAY NEGATIVELY AFFECT SALES

ProBuilder vs. Bundle Sales



THE FUTURE

- » Plan build and distribution pipeline from start
- » Plan for a growing code-base
- » Keeping things modular (code & otherwise)
- » Focus on large scope expensive (ish) tools

QUESTIONS GO HERE