Development Logs by Lucas Pope

All | Papers, Please

Cramming 'Papers, Please' Onto Phones

Aug 06, 2022

PAPERS, PLEASE
STORY

I created Papers, Please in 2013 specifically for desktop computers with mouse control. Now, here, in 2022, desktop computers no longer exist and all computing is done via handheld mobile telephone. Time to update this dinosaur.

These thousands of words and megabytes of images will cover some bits of porting the game from big desktop to little phone. As a winking throwback to days past, this is a big dump of text and inline img tags instead of exciting splashes of quick-cut video. In 9 years, I'll port this post to VR.

Beyond the minimal amount necessary to make the interface work, there are no content changes or additions for this port. No wild story twists, no new characters, no voiced dialog, no stereoscopic ray-traced graphics, and most disappointingly, no cosmetic unlockables.

Same meal, different plate.

Step 1: To Unity

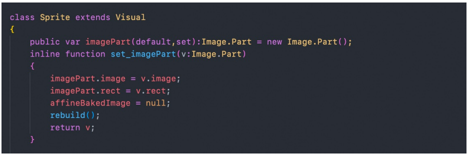
Papers, Please was originally written in [redacted]/[redacted], a combo of modern ECMA-ish language, Flash-alike API, and multi-platform build system. This was a great environment for getting the game together quickly in 2013 but over the years Haxe has moved away from its Flash-targeting roots, and keeping up with OpenFL changes to make game updates has required outsized effort.

When I finally committed to this port my first decision was to rewrite the game in C#/Unity. After Obra Dinn I'm a solid fan of [redacted] – the editor, the entity/component design, the build system, the ubiquity, just about everything.

I made it a few days into rewriting before finding that although I like C#, I like Haxe more. In making the game I'd leveraged a bunch of core Haxe features like super enums and implicit typing and it was all getting trickier and trickier to implement by hand in C#.

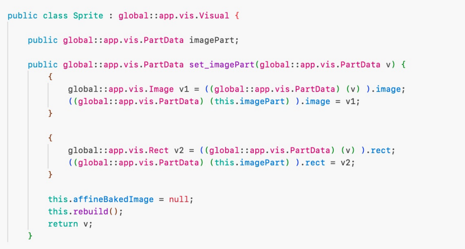
One option would've been to keep the Haxe part but switch from OpenFL to [redacted], another Haxe-based engine and build system. Dead Cells, Northgard, and a few other games are built with Haxe/Heaps so it's a perfectly capable system. Still it's hard to overstate how appropriate Unity is for someone in my position: a solo developer targeting multiple platforms and desperate for a popular, proven, and well-supported engine and build system.

Fortunately, Haxe is a transpiled language, meaning that you write in one language (Haxe) and it gets converted to another language (Javascript, PHP, Python, C++, C#, Lua, etc) before being compiled/interpreted for whatever target you've got. So it's possible to write code in Haxe and have it transpiled to C# that can be loaded and compiled in Unity. Which is how I decided to roll this.



```
class Sprite extends Visual
{
    public var imagePart(default,set):Image.Part = new Image.Part();
    inline function set_imagePart(v:Image.Part)
    {
        imagePart.image = v.image;
        imagePart.rect = v.rect;
        affineBakedImage = null;
        rebuild();
        return v;
    }
```

Haxe code



```
public class Sprite : global::app.vis.Visual {

    public global::app.vis.PartData imagePart;

    public global::app.vis.PartData set_imagePart(global::app.vis.PartData v) {
        {
            global::app.vis.Image v1 = ((global::app.vis.PartData) (v) ).image;
            ((global::app.vis.PartData) (this.imagePart) ).image = v1;
        }

        {
            global::app.vis.Rect v2 = ((global::app.vis.PartData) (v) ).rect;
            ((global::app.vis.PartData) (this.imagePart) ).rect = v2;
        }

        this.affineBakedImage = null;
        this.rebuild();
        return v;
    }
```

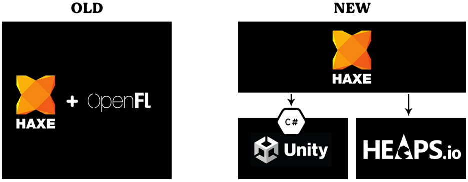
The same code transpiled to C#

First, I stripped all usage of the Flash-like API that OpenFL was providing. The display tree, input events, bitmap manipulation, resource management, sound playback - basically everything that wasn't core game logic. Then I rebuilt it all with custom code, the goal being to create a sort of PAPERS-PLEASE black box that could take user input and spit out a list of quads to draw and audio commands to execute every frame.

This was probably the most fun part of the project. The full Flash API was overkill for Papers Please anyways, and building a fresh engine specifically for what I needed was cathartic.

I was slightly worried about performance here, since much of the OpenFL API is written in C++ and I was replacing it all with higher-level Haxe code. The requirements for the game are earth-core low though, and Haxe is pretty performant anyways so I only had to be a little bit careful to get good framerates.

The end result requires a minimal host shell to send input to the black box and render the quads and audio commands that it outputs every frame. This minimalism meant it was manageable for me to create two hosts: one in Heaps and one in Unity.


OLD
NEW
HAXE
+ OpenFL
HAXE
C#
Unity
HEAPS.io

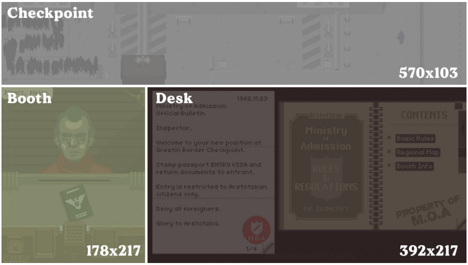
All perfectly clear

Haxe itself compiles almost instantly and is well-supported in Visual Studio Code so writing and debugging in Haxe/Heaps is quick and easy. For making releases, I transpile to C#, tab over to Unity, and build the project there.

Step 2: Phoneterface

Maintenance aside, the main thing holding me back from getting this game onto phones was its user interface. Papers, Please was always meant to be played on a large device. The actual resolution is laughably low (570x320) but these pixels need to be big.

The screen is divided into three always-visible regions:


Checkpoint
570x103
Booth
Papers, please.
178x217
Desk
392x217
Arstotzkan
Ministry of Admission
RULES & REGULATIONS
for Inspectors
CONTENTS
Basic Rules
Regional Map
Booth Info
PROPERTY OF M.O.A
M.O.A
1/4

1982.11.23
Ministry of Admission
Official Bulletin

Inspector,

Welcome to your new position at Grestin Border Checkpoint.

Stamp passport ENTRY VISA and return documents to entrant.

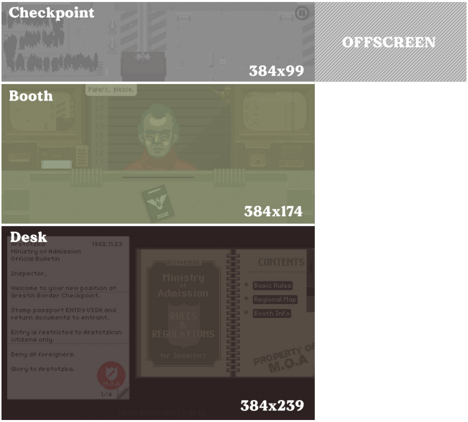
Entry is restricted to Arstotzkan citizens only.

Deny all foreigners.

Glory to Arstotzka.

Checkpoint, Booth, and Desk

Back in 2014, I separated out a chunk of interface layout code from the core logic in order to create an iPad version of the game. With a few adjustments I was able to preserve the “three regions” layout and drag-n-drop gameplay. The biggest changes are that the regions are stacked vertically with slight dynamic sizing ability, and the checkpoint area shows a horizontally scrollable window instead of the whole thing at once.


Checkpoint
384x99
OFFSCREEN
Booth
Papers, please.
384x174
Desk
384x239
Arstotzkan
Ministry of Admission
RULES & REGULATIONS
for Inspectors
CONTENTS
Basic Rules
Regional Map
Booth Info
PROPERTY OF M.O.A
M.O.A
1/4

1982.11.23
Arstotzka
Ministry of Admission
Official Bulletin

Inspector,

Welcome to your new position at Grestin Border Checkpoint.

Stamp passport ENTRY VISA and return documents to entrant.

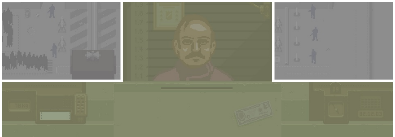
Entry is restricted to Arstotzkan citizens only.

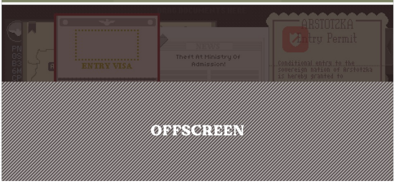
Deny all foreigners.

Glory to Arstotzka.

iPad layout. Checkpoint requires scrolling.

James Gray built on this to create the Vita version, which needed even more layout changes but kept all the same gameplay. The small screen required overlapping the checkpoint area with the booth and fullscreen vertical scrolling but the Vita’s combination of physical and touch controls offset these compromises pretty well.



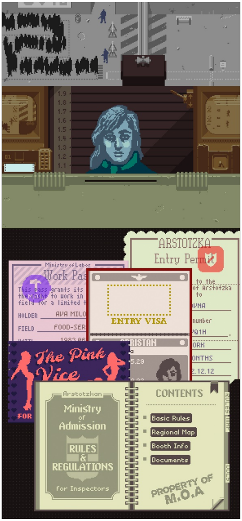

ENTRY VISA
NEWS
Theft At Ministry Of Admission!
ARSTOTZKA
Entry Permit
Conditional entry to the sovereign nation of Arstotzka is hereby granted to
OFFSCREEN

Vita layout. Overlapped booth/checkpoint and fullscreen vertical scroll.

For phones I wanted something that felt natural to the device, which meant a few subjective things:

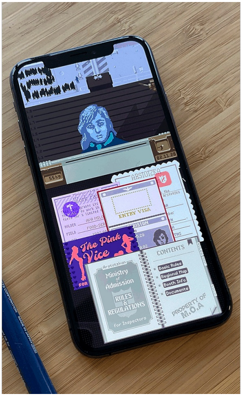
1. Played in portrait mode.
2. All three regions visible at all times.
3. No squinting, zooming, or precision required to read/manipulate documents.

Starting with the iPad layout and throwing together a quick mockup using the latest iPhone screen aspect:


ARSTOTZKA
Entry Permit
Ministry of Labor
Work Pass
ENTRY VISA
The Pink Vice
Arstotzkan
Ministry of Admission
RULES & REGULATIONS
for Inspectors
CONTENTS
Basic Rules
Regional Map
Booth Info
Documents
PROPERTY OF M.O.A

Wrenched into an extreme 10:22 aspect ratio

I tried a few permutations of this, all alike. Converting 16:9 content to 10:22 involves a special kind of desperation. One variation that I didn't hate had the face a bit bigger:

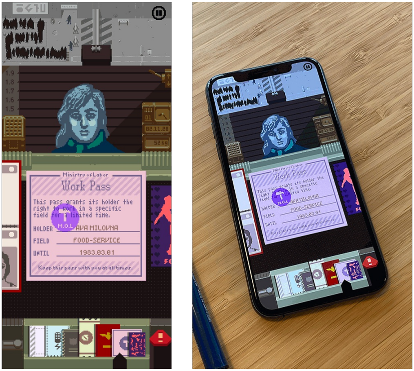

55kg
82.11.28
ARSTOTZKA
Work Pass
ENTRY VISA
The Pink Vice
Arstotzkan
Ministry of Admission
RULES & REGULATIONS
for Inspectors
CONTENTS
Basic Rules
Regional Map
Booth Info
Documents
PROPERTY OF M.O.A

Bigface

Some conclusions after churning through a bunch of mockups:

1. The border checkpoint view at the top will have to scroll left/right even more than the iPad version. Thankfully, the important stuff still fits without scrolling.
2. The documents are too small and the desk area too crowded. There's a fundamental conflict between readability and having enough space for arranging things.
3. Love that big face.

The jumbo face felt so nice that I concluded the documents should also be big, and that uncorked the brew that defines this port: no more desk, no more drag-n-drop. I swung all the way to readability and eliminated the arrangeability requirements completely:


82.11.28
52kg
M.O.L

Ministry of Labor

Work Pass

This pass grants its holder the right to work in a specific field for a limited time.

HOLDER AVA MILOVNA

FIELD FOOD-SERVICE

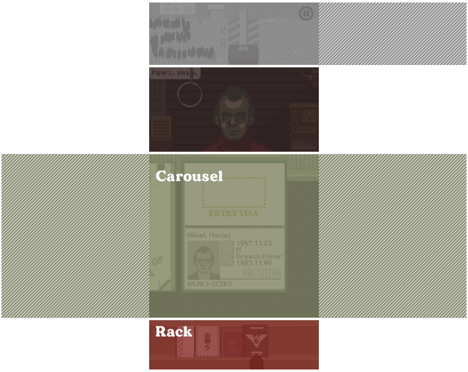
UNTIL 1983.03.01

Keep this pass with you at all times.

Goodbye drag, so long drop

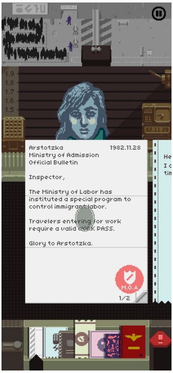
The desk where documents were manipulated has been replaced by two separate elements: a carousel for closeup work and a rack for quick navigation.

The Carousel and the Rack


Carousel
Rack

The carousel extends offscreen on both sides

With the carousel, documents are displayed full-size in a long, horizontal, snap-scrolling list. The rack at the bottom shows a representation of all documents in smaller form. Navigation is done either by swiping the documents directly, or tapping/dragging anywhere in the rack.


Arstotzka 1982.11.28
Ministry of Admission
Official Bulletin

Inspector,

The Ministry of Labor has instituted a special program to control immigrant labor.

Travelers entering for work require a valid WORK PASS.

Glory to Arstotzka.

Navigatin'

My first worry when considering this interface was that it changed the gameplay too much. Instead of organizing documents on a 2D surface, you're just swiping around and viewing each document in near-isolation. The banner task from the desktop game is totally missing now.

That's a good clue to turn back but I knew this port would require some blood. After testing my first basic carousel implementation, the arterial spray I expected from losing document management turned out to be more of a minor abrasion.

In the end, I'm happy with the trade-offs. Swiping feels natural on the phone, and dealing with document inspection has its own rhythm – different from the desktop version but still fun. Instead of arranging the documents and darting your eyes around to correlate information, you manipulate the documents as a group and eye-dart a little less.

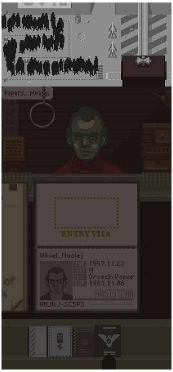
This carousel + rack represents the big sea-change adaptation for the phone port, and it had ripple effects throughout every single other element of the game's main interface. I'll go through the more interesting of these changes in exhausting animated-gif-supplemented detail below.

But first a word about the pixel grid.

The Pixel Grid

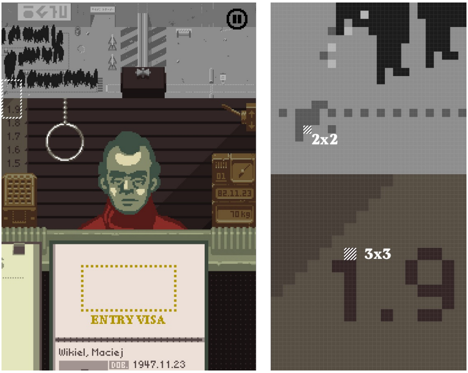
In working on a pixel-art game, one decision every developer has to make is how honest they'll be about their pixel grid. There's usually no actual restriction that all your pixels need to be the same size and even Nintendo's Mario Maker series aerial kicks pixel size consistency directly in the face.

Resolution-wise, the phone interface settled into 208x405 based on how the game's existing documents would fit legibly onto a modern non-Max iPhone. Not a lot of pixels quite frankly, and man that checkpoint region at the top is bulky.


Papers, please
ENTRY VISA
Wikiel, Maciej
1947.11.23
M
Orvech Vonor
1983.11.08
ARSTOTZKA
AHJWJ-SC5P3

Chunky checkpoint

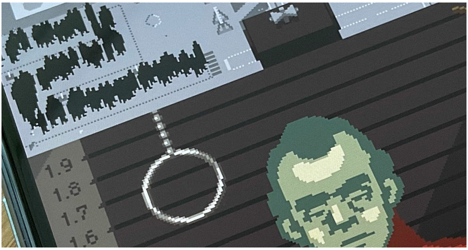
I definitely prefer maintaining a single consistent pixel grid but we've all got our price. In this case, cheating to two pixel grids solved a lot of problems. With a base scale of 3x, the main booth and document regions can run at 3x while the checkpoint is a more manageable 2x.


1.9
1.8
1.7
1.6
1.5
01
82.11.23
70 kg
ENTRY VISA
Wikiel, Maciej
DOB.
1947.11.23
2x2
3x3

Checkpoint at 2x, everything else at 3x

In code, this is handled with fractional 2/3 pixel scaling on the 2x stuff, which multiplies out to integer coordinates in the final 3x buffer.

With the pixel grid sorted there's still the problem of scaling the whole image onto a phone screen. Thankfully the effective resolution of the game is basically only 208x450 and phone screens – especially retina ones – are delirious with pixels. A combination of integer scaling and bilinear filtering doesn't hurt so bad here. The phone version handles aspect adjustments within a certain range and then scales the result to fill the device's screen.

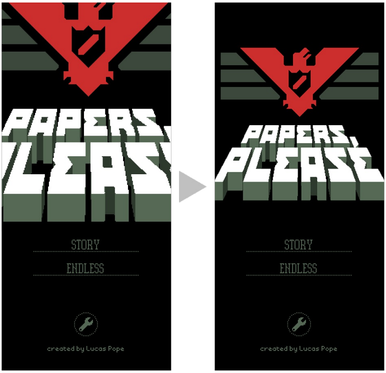


Final image integer + bilinear scaled up to fill the screen

On to some specifics of how the core interface changes played out.

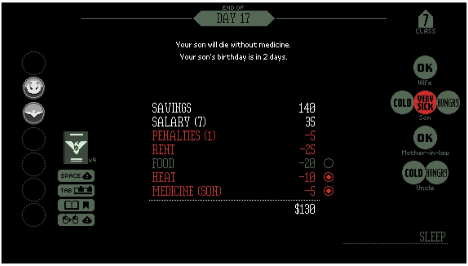
Too Narrow, Doesn't Fit

Several of the game's single visual elements don't fit in the base 208 pixel width. Instead of redrawing them, the extra 2/3 scale option made some fixes easy, as with the title screen:


PAPERS, PLEASE
STORY
ENDLESS
created by Lucas Pope
PAPERS, PLEASE
STORY
ENDLESS
created by Lucas Pope

Using selective 2/3 scale on that big eagle there

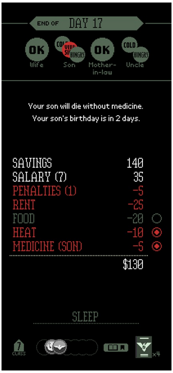
The night screen was a little more complex. It's got way too much business going on, coasting easily in the desktop's ocean of 570x320 but requiring a completely new layout and conditional 2/3 scaling to fit in 208x450.


END OF
DAY 17
7
CLASS
OK
Wife
COLD
VERY SICK
HUNGRY
Son
OK
Mother-in-law
COLD
HUNGRY
Uncle
x4
SPACE
TAB
SLEEP

Your son will die without medicine.
Your son's birthday is in 2 days.

SAVINGS	140	
SALARY (7)	35	
PENALTIES (1)	-5	
RENT	-25	
FOOD	-20	○
HEAT	-10	●
MEDICINE (SON)	-5	●
	$130	

Night screen's roomy desktop layout


END OF DAY 17
OK
Wife
COLD
VERY SICK
HUNGRY
Son
OK
Mother-in-law
COLD
HUNGRY
Uncle

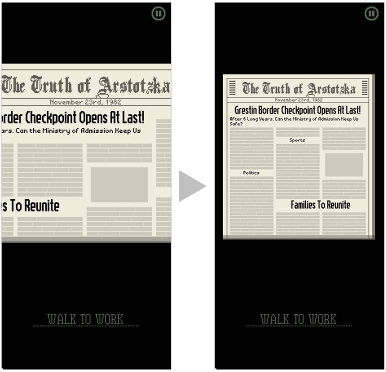
Your son will die without medicine.
Your son's birthday is in 2 days.

SAVINGS	140	
SALARY (7)	35	
PENALTIES (1)	-5	
RENT	-25	
FOOD	-20	○
HEAT	-10	●
MEDICINE (SON)	-5	●
	$130	

SLEEP
7
CLASS
x4

Adjusted phone layout with 2/3 scaling to fit status icons, tokens, etc

The start-of-day newspaper was another big boy and needed both the 2/3 scale plus a narrower 3-column layout to replace the orignal 4-column layout:


The Truth of Arstotzka
November 23rd, 1982
rder Checkpoint Opens At Last!
ars. Can the Ministry of Admission Keep Us
s To Reunite
WALK TO WORK
The Truth of Arstotzka
November 23rd, 1982
Grestin Border Checkpoint Opens At Last!
After 6 Long Years. Can the Ministry of Admission Keep Us Safe?
Sports
Politics
Families To Reunite
WALK TO WORK

Changed layout and scale to fit newspaper

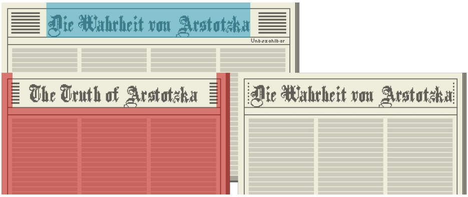
This “needs new background image” requirement for the newspaper segues to a core engine feature implemented for the port: load-time image transmogrification.

Image Transmogrification

Papers, Please has a boatload of documents and images that contain text. A good share of these are generated outside the game and have to be localized for each supported language. There’s tools to help with that but one of my goals for this port was to do as little localization work as possible.

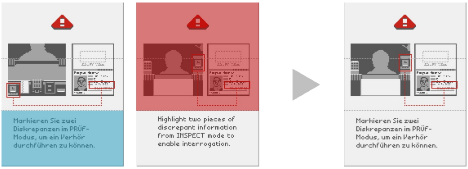
Instead of generating new images just for the phone layout, the game has a “mogrification” step when loading image assets that makes edits procedurally based on which platform it’s running. This is handled at a pretty low level so the higher level game logic doesn’t need to worry about it.

One example is the newspaper background, which has the localized name baked into the image. The mogrification process loads both the desktop-formatted localized image and the phone-formatted unlocalized image, then copies a section from one to the other to get a phone-formatted, localized image.


Die Wahrheit von Arstotzka
Unbezahlbar
The Truth of Arstotzka
Die Wahrheit von Arstotzka

Localizing the newspaper banner at load-time

Another case is the bulletin page with inspection correlation instructions. Here, the mogrifier combines the localized lower half of this desktop-specific bulletin page with the unlocalized phone-specific upper half to get a localized phone-specific bulletin.



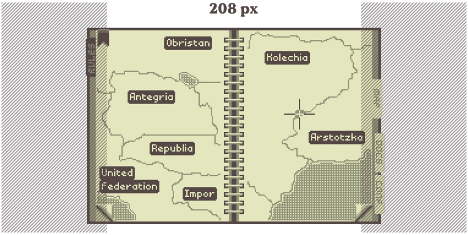
Markieren Sie zwei Diskrepanzen im PRÜF-Modus, um ein Verhör durchführen zu können.

Highlight two pieces of discrepant information from INSPECT mode to enable interrogation.

Markieren Sie zwei Diskrepanzen im PRÜF-Modus, um ein Verhör durchführen zu können.

This here plus that there

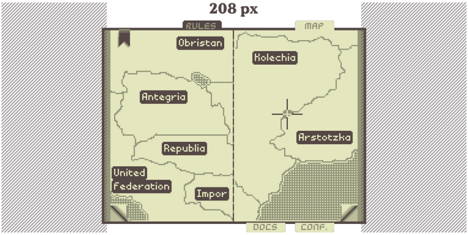
These load-time adjustments go beyond images and into some document layouts too. For example, the rulebook is the widest document in the game and it, horrifyingly, doesn’t fit in 208 pixels. It’s close but not quite, and there are important tabs and page-turn links on the edges that need to be comfortably onscreen.


208 px
RULES
Obristan
Kolechia
Antegria
Arstotzka
Republia
United Federation
Impor
MAP
DOCS
CONF.

Original rulebook, designed for landscape layout on desktops

I wanted to keep the entire booth/documents area in one pixel grid, so 2/3 scaling wasn’t an option here.

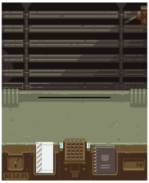
Load-time mogrification works a number on this to crop each page, move all the active links, and rotate/re-position the tabs to fit within the 208 pixel width.


208 px
RULES
MAP
Obristan
Kolechia
Antegria
Arstotzka
Republia
United Federation
Impor
DOCS
CONF.

Mogrified rulebook for the phone portrait layout. RIP spiral.

Hand Over Your Documents

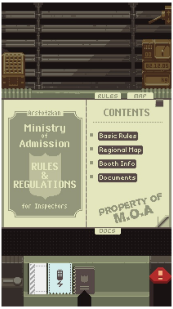
In the desktop build, each traveler drops their documents on the countertop after they enter the booth. Plain and simple.



Countertop docs

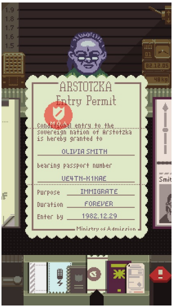
Due to the lack of space, the phone layout doesn't have a functional countertop. An obvious alternative would be to have the docs just drop all the way down to the rack at the bottom and appear in the carousel automatically.

That seemed a bit passive to me though, losing the initial "ok fine I'll take these" activity from the original desktop version when you drag them from the counter to the desk for reading. The solution I came up with is to float the documents first, requiring a tap to get them onto the rack and into the carousel. The order they're tapped determines their order in the list.


82.12.05
kg
RULES
MAP
Arstotzkan
Ministry
of
Admission
RULES
&
REGULATIONS
for Inspectors
CONTENTS
Basic Rules
Regional Map
Booth Info
Documents
PROPERTY OF
M.O.A
DOCS

Floating docs

This design highlights something used throughout the phone version: overlapping elements. The main carousel is intended to feel like you're holding documents up to scrutinize them, enhanced a bit by overlapping the traveler and booth console. To stay practical, it's possible to pull the carousel down for a better look.


1.9
1.8
1.7
1.6
1.5
01
82.12.05
48 kg
M.O.A
Smit

ARSTOTZKA
Entry Permit

Conditional entry to the sovereign nation of Arstotzka is hereby granted to

OLIVIA SMITH

bearing passport number

VE4TN-K1KAE

Purpose IMMIGRATE
Duration FOREVER
Enter by 1982.12.29

Ministry of Admission

The original design of the booth was always pretty spacey but there are some distinct activity modes. For this port it was generally possible to optimize the use of space by masking/overlapping things that weren't needed in one activity with something else needed right now.

Custom UI API OK

A notable benefit of writing my own UI system for this was that I could do some very specific things with touch and pointer input. Papers, Please looks like a pretty simple 2D game at a glance but representing all the document and interface interactions through standard input event callbacks in the original Haxe/OpenFL required a lot of uncomfortable hacks. The new input system doesn't use callbacks and is cleaner now, which let me implement all the overlapping and passthrough input logic in a more manageable way.

```
private function this_onMouseDown(e:PointerEvent):Void {
    ...
}

private function this_onMouseLeave(e:Event):Void {
    ...
}

private function this_onMouseOver(e:PointerEvent):Void {
    ...
}

private function this_onMouseUp(e:PointerEvent):Void {
    ...
}

private function this_onMouseClick(e:MouseEvent):Void {
    ...
}
```

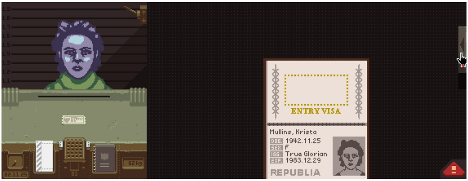
Old event system, where input handling logic is spread around in various callback functions

```
public override function react(input:Input) {
    var pointerOver = input.getPointerOver(this, worldRect());
    if (pointerOver != null)    {
        // all pointer logic
        ...
    }
    if (input.getKeyDown(shortcutKey)) {
        // keyboard logic
        ...
    }
}
```

New react() event system, more centralized and flexible

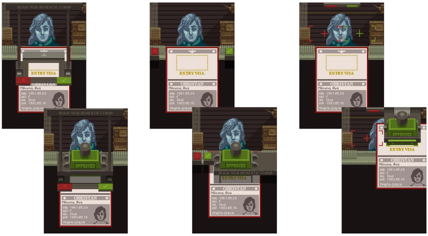
Stamping

The desktop version has a stamp bar that slides out from the right side and hangs over the desk. To stamp a passport you just align it under the stamp and click.


ENTRY VISA
Mullins, Krista
DOB 1942.11.25
SEX F
ISS True Glorian
EXP 1983.12.29
REPUBLIA
62kg
82.11.25

Original desktop stamp bar

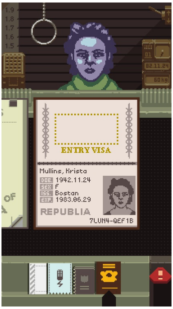
I had some trouble redesigning this for the phone layout because (A) you can't just drag the passport around to align it under the stamp, and (B) I didn't want to give up the satisfaction of stamping on press/touch. Without this second requirement it's simple to just make the stamp itself draggable, to be dropped on the passport where it automatically stamps. But then you're stamping on release/un-touch, which never felt right to me. Some concepts I worked through:



Ceiling attachment, Split bar, Laser sensor

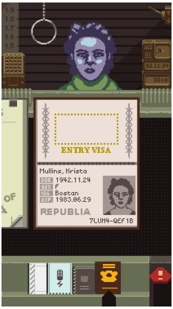
These all felt weird, confusing or just flat out didn’t work. There was something implicitly sensible with the desktop stamp bar that I wasn’t getting.

Ater some more experiments I decided to re-introduce the original setup, behind a modal switch. From the carousel, passports can isolate into a temporary stamp desk to be dragged around and stamped willy nilly.


1.9
1.8
1.7
1.6
1.5
01
02.11.24
60 kg
OF
ENTRY VISA
Mullins, Krista
DOB. 1942.11.24
SEX F
ISS. Bostan
EXP. 1983.06.29
REPUBLIA
7LUN4-QEF1B

Opening a temporary desk just for stamping

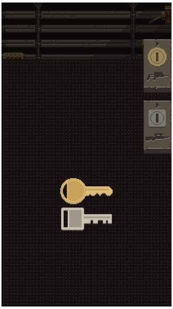
This temporary stamp desk is opened with a pull chain, which allows me to save some screen space and helpfully indicate its availability only when a stampable document is focused. Also it’s marginally fun to play with the chain.


1.9
1.8
1.7
1.6
1.5
01
02.11.24
60 kg
OF
ENTRY VISA
Mullins, Krista
DOB. 1942.11.24
SEX F
ISS. Bostan
EXP. 1983.06.29
REPUBLIA
7LUN4-QEF1B

Swinging chain

Key Desk Too, Sure Why Not

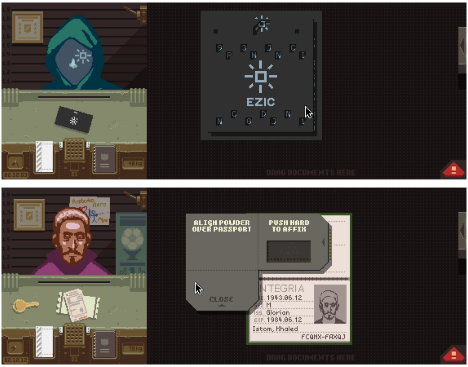
This same temporary-desk mechanic is used for the rifle and tranquilizer gun sequences. When a border attack is in progress, it quickly flips to the keys on an isolated desk. Dragging the keys and unlocking a gun works the same here as the desktop version.



Temporary key desk for unlocking guns

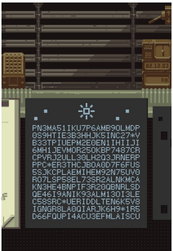
Pincher

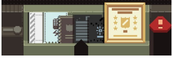
The need for a modal stamp/key desk was a gradual realization born from failed experimentation. In contrast, I knew from the start of this port that I'd need a way to let the player align certain documents on top of each other.


EZIC
DRAG DOCUMENTS HERE
ALIGN POWDER OVER PASSPORT
PUSH HARD TO AFFIX
CLOSE
NTEGRIA
1943.06.12
M
ISS. Glorian
EXP. 1984.06.12
Istom, Khaled
FCQNX-FAXQJ
DRAG DOCUMENTS HERE

'Puzzles'

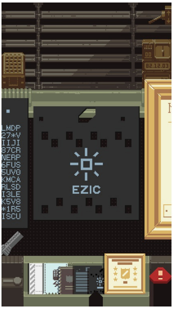
To replicate this functionality in the new carousel mobile design, I added an alligator-clip style pincher. Like with the stamp desk pullchain, this pincher appears only when certain documents are focused in the carousel.


PN3MA51IKU7P6AMB90LMDP
0S9HTIE3B3HHJK5INC27*V
B33TPIUEPM2E0EN1IHIIJI
6MH1JEVMOR250KBP7487CR
CPVRJ2ULL30LH2Q3JRNERP
PPC*ER3THCJB0A0D7F6FUS
5SJKCPLAEMIHEM92N75UV0
R07LSP58EL73SR2ALNKMCA
KN3HE4BNPIF3R20QBNRLSD
QE46I9ANIK93ALM130I3LE
C58SRC*UERIDDLTEN6K5V8
IGNGR8LA0QIARJK6H9*1R5
D66FQUPI4ACU3EFMLAISCU



Alligator-clip pincher appearing near pinchable document, for pinching

Tapping the pincher will grab the document and hold it above and out of the carousel, where it can be dragged around separately. The carousel remains scrollable underneath so things can be lined up.

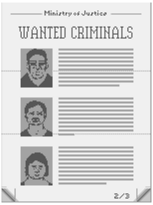

02.12.03
LMDP
27*V
IIJI
87CR
NERP
6FUS
5UV0
KMCA
RLSD
I3LE
K5V8
*1R5
ISCU
EZIC

Pincher action

None of this is explained in-game so let's assume the concept and mechanics are self-evident.

Wanted: Criminal Poster

One emergent play pattern in the desktop version is how to deal with the daily criminal bulletin.



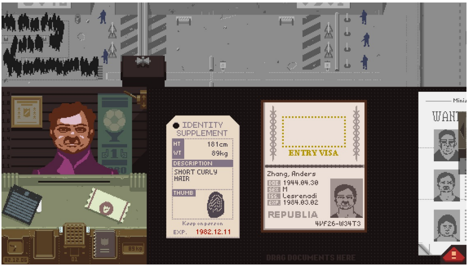
Ministry of Justice

WANTED CRIMINALS

2/3

Detain these people

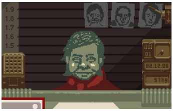
Any traveler matching a photo here needs to be interrogated, and the best way to keep this in mind while working is to tuck it off to the side of the desk and leave it there throughout the day.


IDENTITY SUPPLEMENT
HT 181cm
WT 89kg
DESCRIPTION
SHORT CURLY HAIR
THUMB
Keep on person
EXP. 1982.12.11
ENTRY VISA
Zhang, Anders
DOB 1944.04.30
SEX M
ISS Lesrenadi
EXP 1984.03.02
REPUBLIA
4VF26-W34T3
DRAG DOCUMENTS HERE
02.12.05
89 kg
WANT

You look familiar

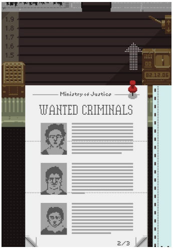
Since the freeform desk is gone in this new phone layout, it’s easy to swipe away from the bulletin and completely forget about the mugshots. This is a problem I didn’t catch until playing through the mostly-finished port, where I missed having a way to see those three faces at all times.

Space is so tight in the phone layout that there’s no perfect place to put them but I did find a little spot on the back wall where they can be hung up.


1.9
1.8
1.7
1.6
1.5
82.12.06
57kg

Mugshots helpfully posted on the wall

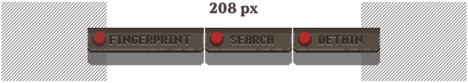
To preserve some of the desktop version’s intent, this isn’t automatic. The player needs to tap a pin in the bulletin to post them on the wall at the start of each day.


1.9
1.8
1.7
1.6
1.5
82.12.06
Ministry of Justice
WANTED CRIMINALS
2/3

Pinning from the bulletin to the wall

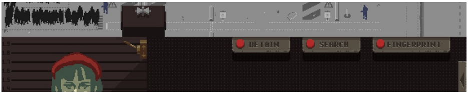
Action Buttons Don’t Fit

This could’ve gone under Doesn’t Fit. The desktop version’s FINGERPRINT, SEARCH, and DETAIN buttons don’t fit.


208 px
FINGERPRINT
SEARCH
DETAIN

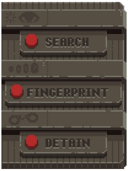
Too wide to arrange these horizontally in 208px

This was painful. These buttons drop down after an interrogation, and their appearance is a noisy eye-catching signal that the player has some new actions available. It’s rare, but possible, that all three are available at the same time.


DETAIN
SEARCH
FINGERPRINT

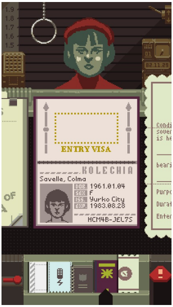
Plenty of space over here

With basically no place to put them and not wanting to scale them down I decided they couldn’t go in the main booth interface. There’s now an extra slide-out panel to hold them.


SEARCH
FINGERPRINT
DETAIN

Panel of buttons. This is an older version -- I removed the icons and right-aligned everything at the last second.

The panel pops out automatically when a new action becomes available, closes when clicking anywhere outside, and shows a flashing lever when hidden.


ENTRY VISA

KOLECHIA

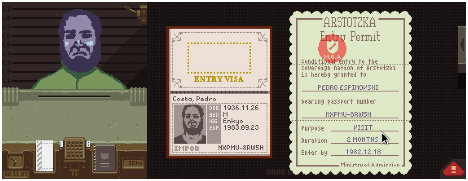
Savelle, Colma
DOB. 1961.01.04
SEX F
ISS. Yurko City
EXP. 1983.08.28
HCM4B-JEL7S

Opening the panel with a flashing lever

These buttons are fully localized as images, so reusing the same buttons and just putting them in a pop-out saves a bunch of trouble with the localization process.

Enhanced Inspect Mode

The original game treats its discrepancy-highlighting inspect mode as a quick aside. First the player arranges documents to get a clear view, then finds a discrepancy, enables inspect mode, highlights the problem, and finally interrogates. While inspect mode is active, documents can’t be moved and pages can’t be turned – things have to be in order before inspecting.

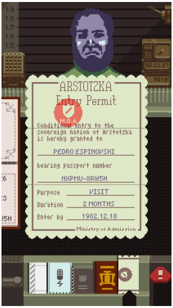

ENTRY VISA

Costa, Pedro
DOB. 1936.11.26
SEX M
ISS. Enkyo
EXP. 1983.09.23
IMPOR
NXPMU-8RW5H

ARSTOTZKA
Entry Permit
M.O.A
Conditional entry to the sovereign nation of Arstotzka is hereby granted to
PEDRO ESPINOVSKI
bearing passport number
NXPMU-8RW5H
Purpose VISIT
Duration 2 MONTHS
Enter by 1982.12.18
Ministry of Admission

Interrogating a discrepancy in the desktop version

This process is impossible if you can’t see all your documents at once, as in the phone port. The fix was conceptually pretty simple but a fairly big technical pain in the ass: the carousel and page turn buttons are still active during inspect mode, and highlighted elements remain partially onscreen when their documents scroll away.



ARSTOTZKA
Entry Permit

M.O.A

Conditional entry to the sovereign nation of Arstotzka is hereby granted to

PEDRO ESPINOVSKI

bearing passport number

NXPMU-8RW5H

Purpose VISIT
Duration 2 MONTHS
Enter by 1982.12.18

Ministry of Admission

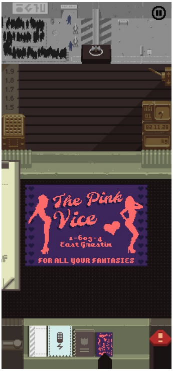
Inspect mode in the phone version

Polishing Tweaks

With all of the game’s interactions there was plenty of room for small frustrations. I did my best to fix these whenever I found them. None are critical but half the battle is just recognizing there’s some friction that could use smoothing, so I enjoy these kinds of tweaks.

Auto Sorting

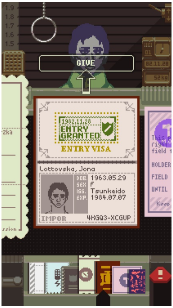
The three most important documents are the all-day ones: the bulletin, audio transcript, and rulebook. Because everything’s in a flat list now, it’s important to keep these three close to where each entrant’s new docs appear on the right. To do this, the game performs an instant re-sort when calling an entrant, without affecting the focused view.


The Pink Vice
1-603-4 East Grestin
FOR ALL YOUR FANTASIES

Instant auto-sorting to keep important docs on right side

Post-Give Smart Shift

Once a passport is stamped, the only thing left to do is give the entrant all their documents back. Giving a document back removes it from the carousel list, shifting the next document into view. My first implementation was the naive one: if document n disappears, shift to n-1. For a sequence of giving, this means it may shift to a document that’s not meant to be given back, requiring the player to manually swipe around. Not a huge deal but it’s a lot less annoying when the carousel intelligently shifts to the next givable one.

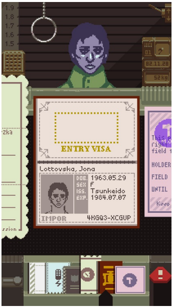

GIVE
1982.11.28
ENTRY GRANTED
ENTRY VISA

Lottovska, Jona
DOB. 1963.05.29
SEX F
ISS. Tsunkeido
EXP. 1984.07.07
IMPOR 4HGQ3-XCGUP

Shifting to the next givable document

Peeking

While looking for correlations and discrepancies there's a lot of flipping back and forth. With swiping this is straightforward but if you keep your finger held on the screen and just slide it left and right, the reach isn't enough to actually see whole documents on either side. Not immediately obvious but measurably frustrating while playing. My fix was to implement a "peek" feature that snaps the adjacent document fully into view when sliding just a little bit.



Lottovska, Jona
DOB. 1963.05.29
SEX F
ISS. Tsunkeido
EXP. 1984.07.07
IMPOR 4HGQ3-XCGUP

Peeking at adjacent documents

Auto Player Testing

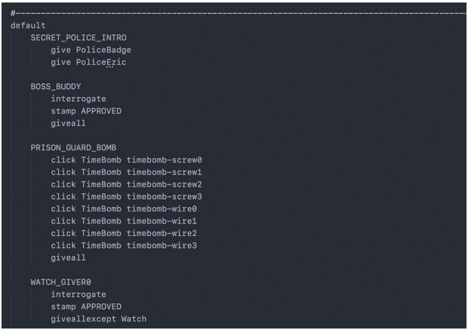
The original desktop game has a debug "soak" test mode for stressing memory and gameplay logic. Basically, a process plays through the game by calling internal debug functions to load a day, stamp documents, shoot attackers, etc. This was useful for making sure there were no errant memory leaks or serious logic errors but because it called directly into high-level functions it wasn't much use for testing the whole game and engine environment.

Having full control over the entire software stack for this port enabled me to enhance this system considerably. Instead of a memory soak test, it's now a full automatic player that controls the game using the same input messages that a real player would.

Auto player in action, showing the route, day, traveler, time, memory, etc.

The auto player is scriptable and follows “routes” to:

1. Play through to all 20 endings, starting from the title screen.
2. Earn all tokens.
3. Play a few minutes of each Endless style and course.
4. Seed each playthrough with a different random seed.



```
#------------------------------------------------------------------------------------------
default
    SECRET_POLICE_INTRO
        give PoliceBadge
        give PoliceEzic

    BOSS_BUDDY
        interrogate
        stamp APPROVED
        giveall

    PRISON_GUARD_BOMB
        click TimeBomb timebomb-screw0
        click TimeBomb timebomb-screw1
        click TimeBomb timebomb-screw2
        click TimeBomb timebomb-screw3
        click TimeBomb timebomb-wire0
        click TimeBomb timebomb-wire1
        click TimeBomb timebomb-wire2
        click TimeBomb timebomb-wire3
        giveall

    WATCH_GIVER0
        interrogate
        stamp APPROVED
        giveallexcept Watch
```

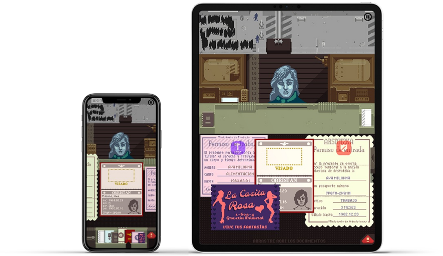
Defining a route plan

And because the entire rewritten game+engine is modular, it’s possible to run this auto player as fast as possible, with or without visuals. On a modern iPhone rendering every 1,000th frame, a full sweep through all ~30 routes takes about 15 minutes.

Creating this system was at least a solid month of initial work and a good amount of maintenance throughout. Quite a long time honestly but developing and using it has helped me find and fix countless bugs in the game. Since I develop on a Mac using Haxe/Heaps and deploy in Haxe/Unity, it’s also an invaluable way to verify that Heaps and Unity give the same result, and that Mac/iOS/Android all behave correctly. Smaller unit testing can validate some of this but nothing beats full playthroughs.

Platforms, Plural

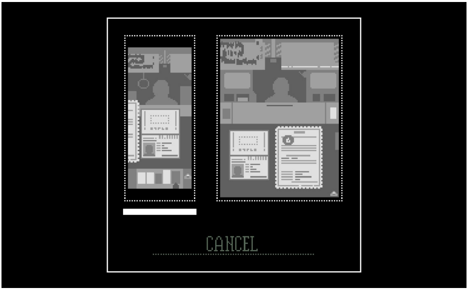
There are now three core interface modes for the game: desktop, tablet, and phone. This port deals with just the mobile interfaces and supports phone and tablet layouts in the same app/binary.


ARSTOTZKA
Permiso de entrada
VISADO
OBRISTAN
La Casita Rosa
VIVE TUS FANTASÍAS

Both mobile layouts

Since everything is in one binary, the game needs to know where it's running in order to pick the right interface on startup. On iOS, this is easy as there's an explicit check you can make for iPhone or iPad. On Android, surprisingly, there's no clear delineation between phones and tablets and the game has to wing it based on resolution and DPI.

In the cases where it guesses wrong, there's an added setting to explicitly choose which mode you want.



Some devices get to choose

The final breakdown looks something like this:

- iPhone: phone layout only.
- iPad: default tablet layout, optional phone layout.
- Android Device: Optional phone and tablet layouts, default best guess.

Finally

A proper mobile version of Papers, Please has been a long time coming. Since the desktop release 9 years ago I'd dismissed the idea that the game could even work on phones but I guess a switch must've flipped in my head last year.

From start to finish this port took around 8 months of work. That sounds like an eternity but I enjoyed most of it so won't complain too loudly. The sharpest cut is that it's kept me away from working on anything else.

After putting out whatever fires light up after release I'll take a short break and then get back to making new stuff. See you then.

Copyright © 2022 Lucas Pope

All | Papers, Please