

Playdate Adventures

von Toot bis Catalog

Rico Becker

Vorwort

- Viel Ökosystem
- Wenig Code – aber gefährliche Xcode-Demo
- Keine Tests
- das Gerät ist sehr klein: hoffentlich sieht man was

Toot

Alles begann in der Timeline



Finn Voorhees
@finnvoorhees@mastodon.social

Someone had to do it 😊

[github.com/finnvoor/PlaydateKi...](https://github.com/finnvoor/PlaydateKit)



PlaydateKit

GitHub

GitHub - finnvoor/PlaydateKit: Create games for Playdate using Swift.

Create games for Playdate using Swift. Contribute to finnvoor/PlaydateKit developm...

Mar 14, 2024 at 06:38 PM · 🌎 · Ivory for Mac

29 boosts · 80 favorites



...

Outline

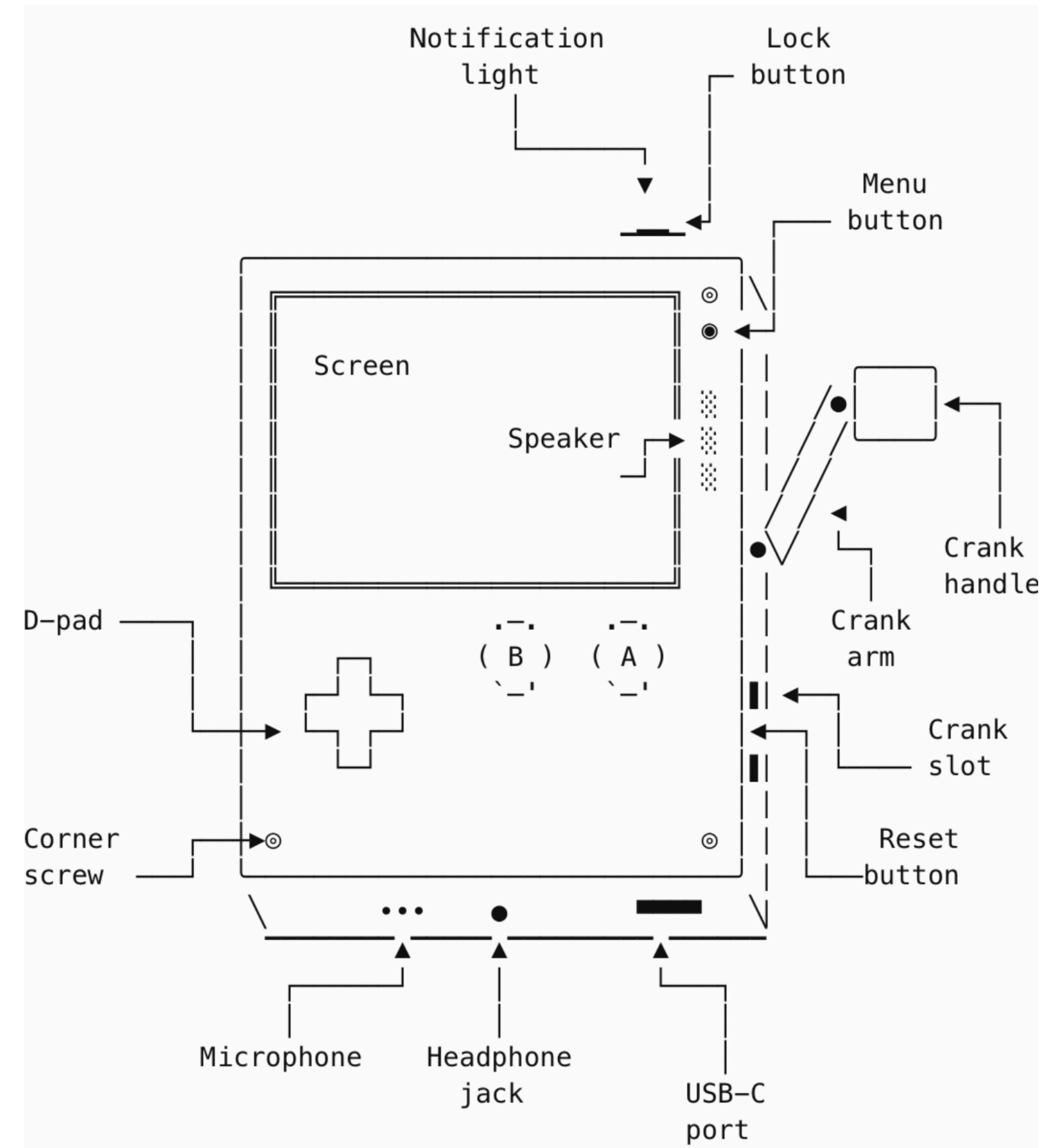
von Toot bis Catalog

- Hardware (Display, Crank, D-Pad, ...)
- SDK (Lua/C, Simulator, Graphics, Fonts, ...)
- Embedded Swift / PlaydateKit
- Xcode-Demo: Runloop, Deployment
Simulator, Sideload Device
- Testing
- Catalog Listing



Hardware

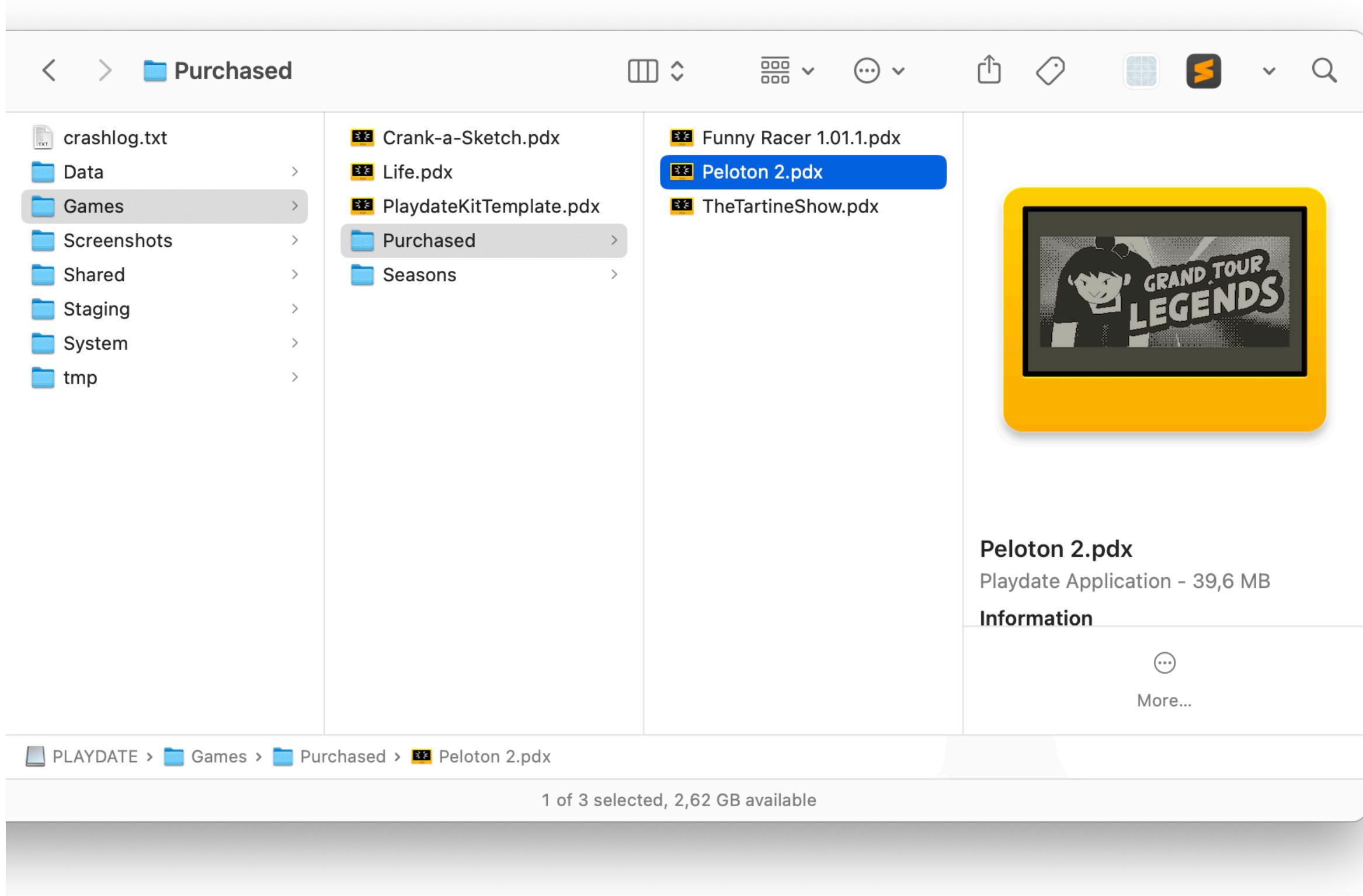
- Cortex-M7 Processor
- 1bit LCD 400 x 240 Pixel
- 16 MB RAM
- Accelerometer
- Crank
- Bluetooth
- WiFi



Software

Stepping back a bit, it's really wild to think that **Panic has** made not only this handheld gaming device that's 100% custom, but also **developed its operating system from scratch**, developed a full SDK that third-party developers have been using for a few years now, plus this entire web-based development environment where anyone can make their own Playdate games that will then run on any Playdate device.

Software



- Device als Data Disk mountbar
 - Sideload via Web/USB
 - DRM nur für Executable File
- Container bei Catalog Games

Software

<https://github.com/cranksters/playdate-reverse-engineering>

- Dateiformate
- USB-Interface
- Server-API

Documentation

- File Formats
 - Playdate game formats
 - [.luac](#) - Lua bytecode
 - [.pdz](#) - File container
 - [.pda](#) - Audio file
 - [.pdi](#) - Image file
 - [.pdt](#) - Imagetable file
 - [.pdv](#) - Video file
 - [.pds](#) - Strings file
 - [.pft](#) - Font file
 - Other formats
 - [.fnt](#) - Font source file
 - [.strings](#) - Strings source file (TODO)
- Server
 - [Playdate API](#) - Main Playdate server API
- Misc
 - [USB](#) - USB serial interface
 - [Streaming](#) - Video/audio streaming protocol (via USB serial), used by Playdate Mirror

Tools

- [pdz.py](#) - Unpacks all files from a `.pdz` file container
- [pdex2elf.py](#) - Converts a `pdex.bin` to an ELF file that can be analysed in a decompilation tool such as Ghidra
- [usbeval.py](#) - Uses the Playdate's USB `eval` command to evaluate a Lua script over USB. Has access to the Lua runtime of the currently loaded game, except for system apps.

⚠ This documentation is unofficial and is not affiliated with Panic. All of the content herein was gleaned from reverse-engineering Playdate tools and game files, and as such there may be mistakes or missing information.

Playdate SDK

- Download <https://play.date/dev/>
- auch für Linux 
- (und Windows)
- Playdate Compiler (Mac: ARM embedded compiler)
- baut die benötigten Dateiformate (bspw. wav > pda)

SDK

PULP

LINKS

DEV HELP

CATALOG

Playdate SDK

All you need: Lua and C APIs, docs, as well as a Simulator for local development, with profiling and more.

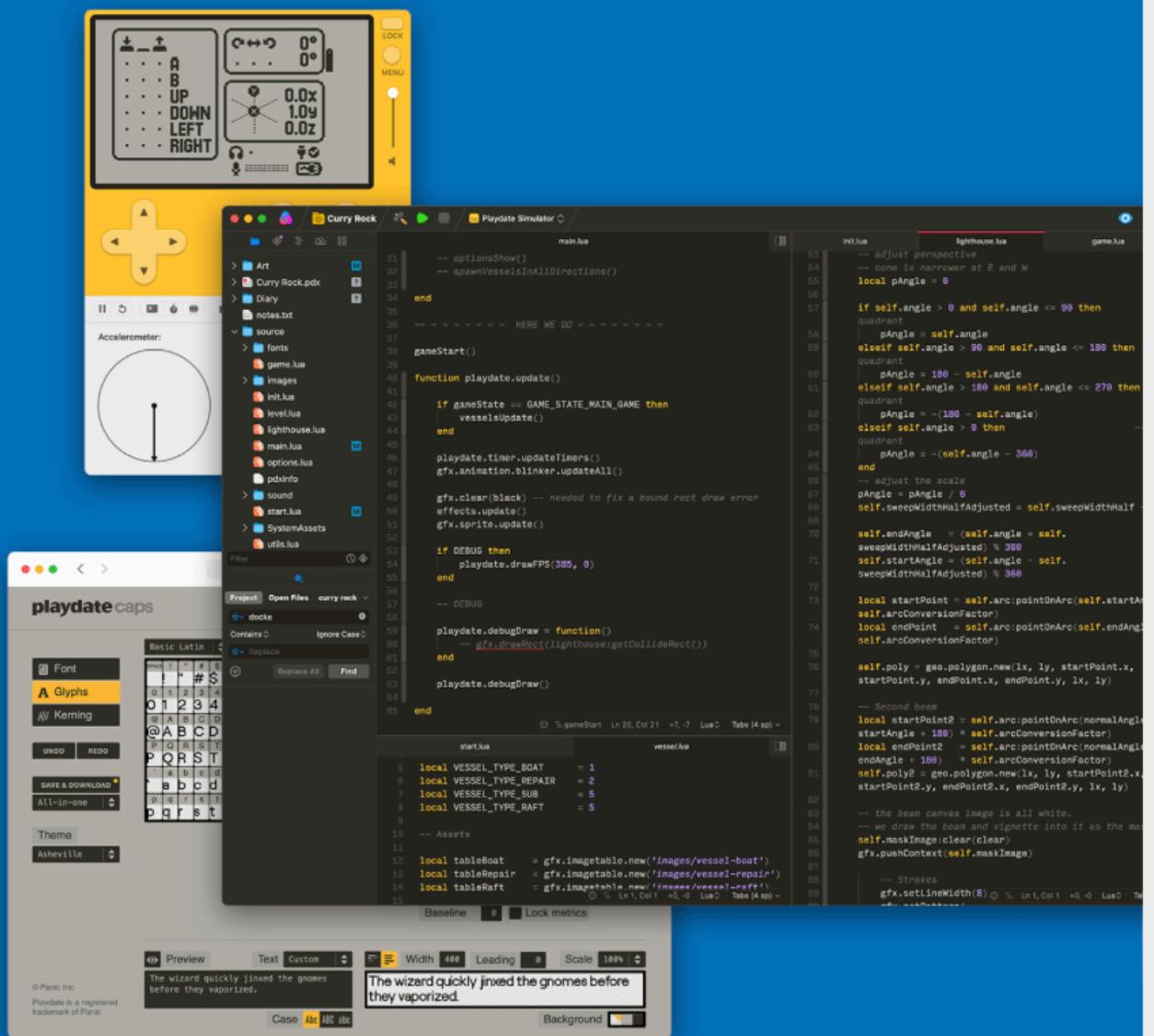
I agree to the [Playdate SDK License](#).

Download Playdate SDK 2.4.2 for macOS

[Download for Windows](#)

[Download for Linux](#)

 See what's new



Inside Playdate

The official reference documentation for programming Playdate games.

[Inside Playdate →](#)

[Inside Playdate with C →](#)



Libraries & Tools

Tutorials, editors, importable libraries, engines, templates,



Swift for Playdate

The Swift team's new embedded language support means you can

Playdate SDK

- Verfügbare Sprachen
- Lua / Pulp populär
- C für maximale Performance
- C + Embedded Swift: “**Byte-sized Swift: Building Tiny Games for the Playdate**”
- <https://github.com/apple/swift-playdate-examples>

SDK

PULP

LINKS

DEV HELP

CATALOG

Playdate SDK

All you need: Lua and C APIs, docs, as well as a Simulator for local development, with profiling and more.

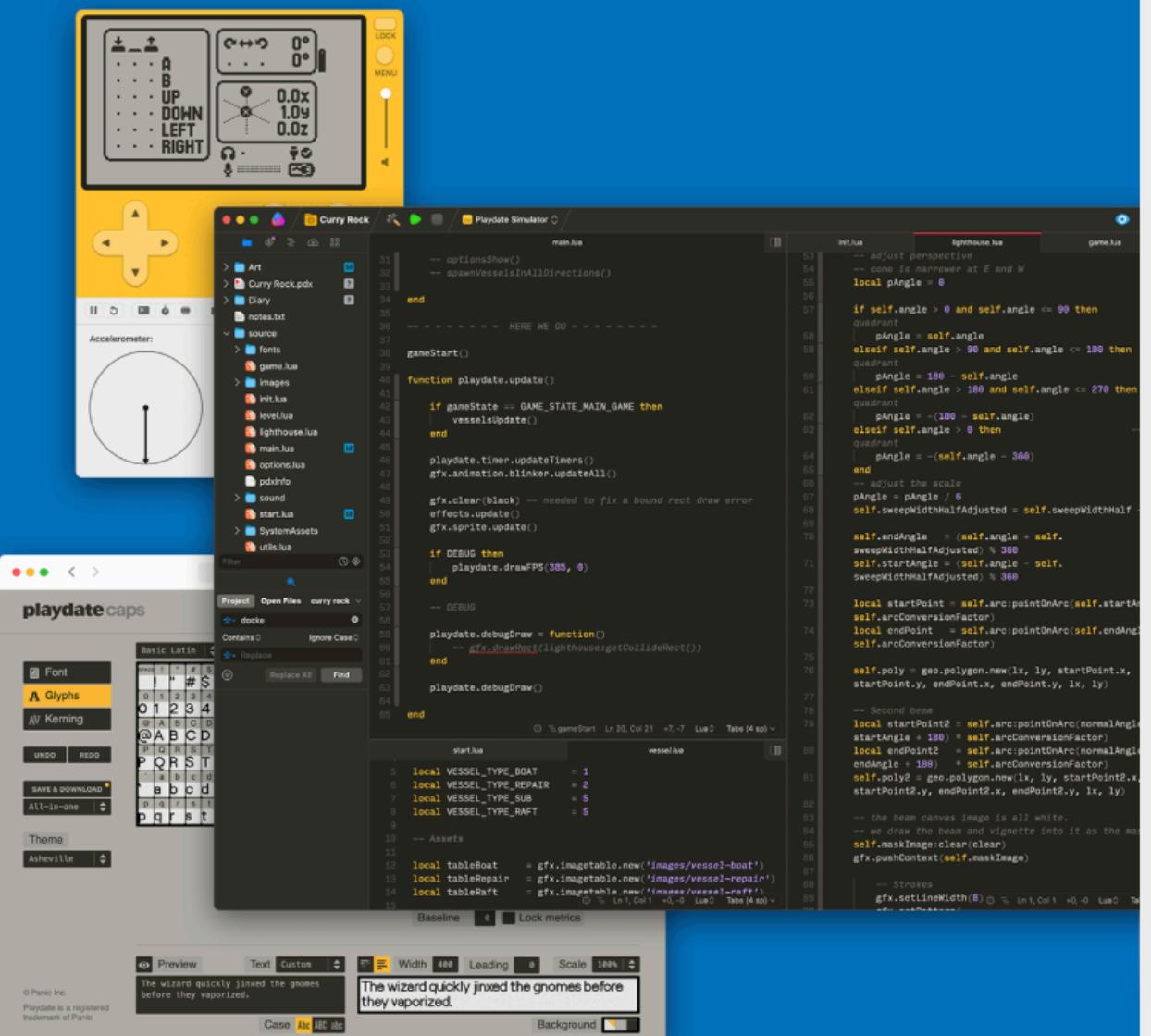
I agree to the [Playdate SDK License](#).

Download Playdate SDK 2.4.2 for macOS

[Download for Windows](#)

[Download for Linux](#)

☀ See what's new



Inside Playdate

The official reference documentation for programming Playdate games.

[Inside Playdate →](#)

[Inside Playdate with C →](#)



Libraries & Tools

Tutorials, editors, importable libraries, engines, templates,



Swift for Playdate

The Swift team's new embedded language support means you can

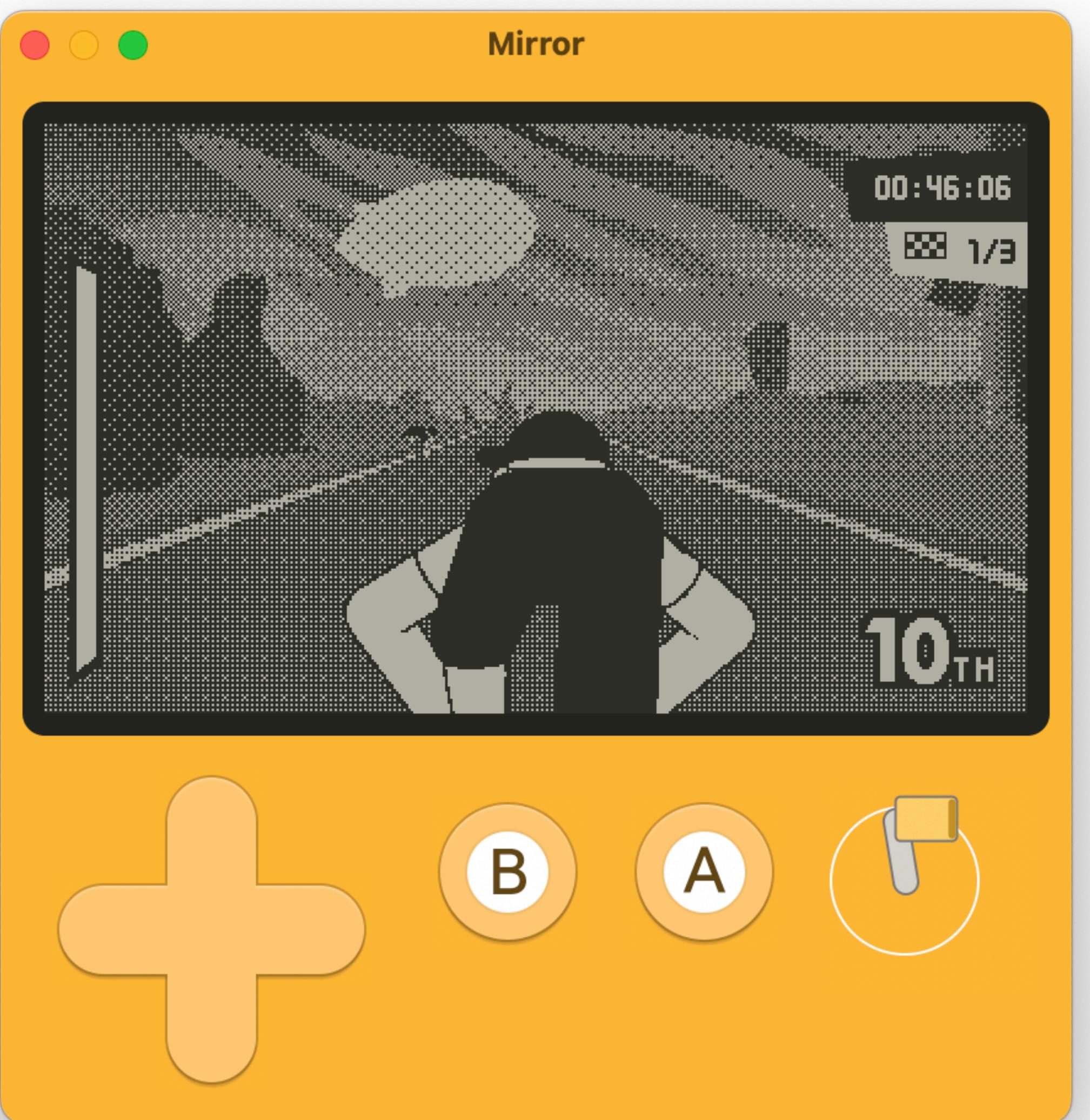
Playdate SDK

- Tools – Simulator
- Standalone App
- Lädt App-Bundle (**PlayDate eXecutable**) aus lokalem Verzeichnis
- Sideload: Deploy aufs Gerät
- Kann Hardware-Input (Crank, Accelerometer, D-Pad) benutzen



Playdate SDK

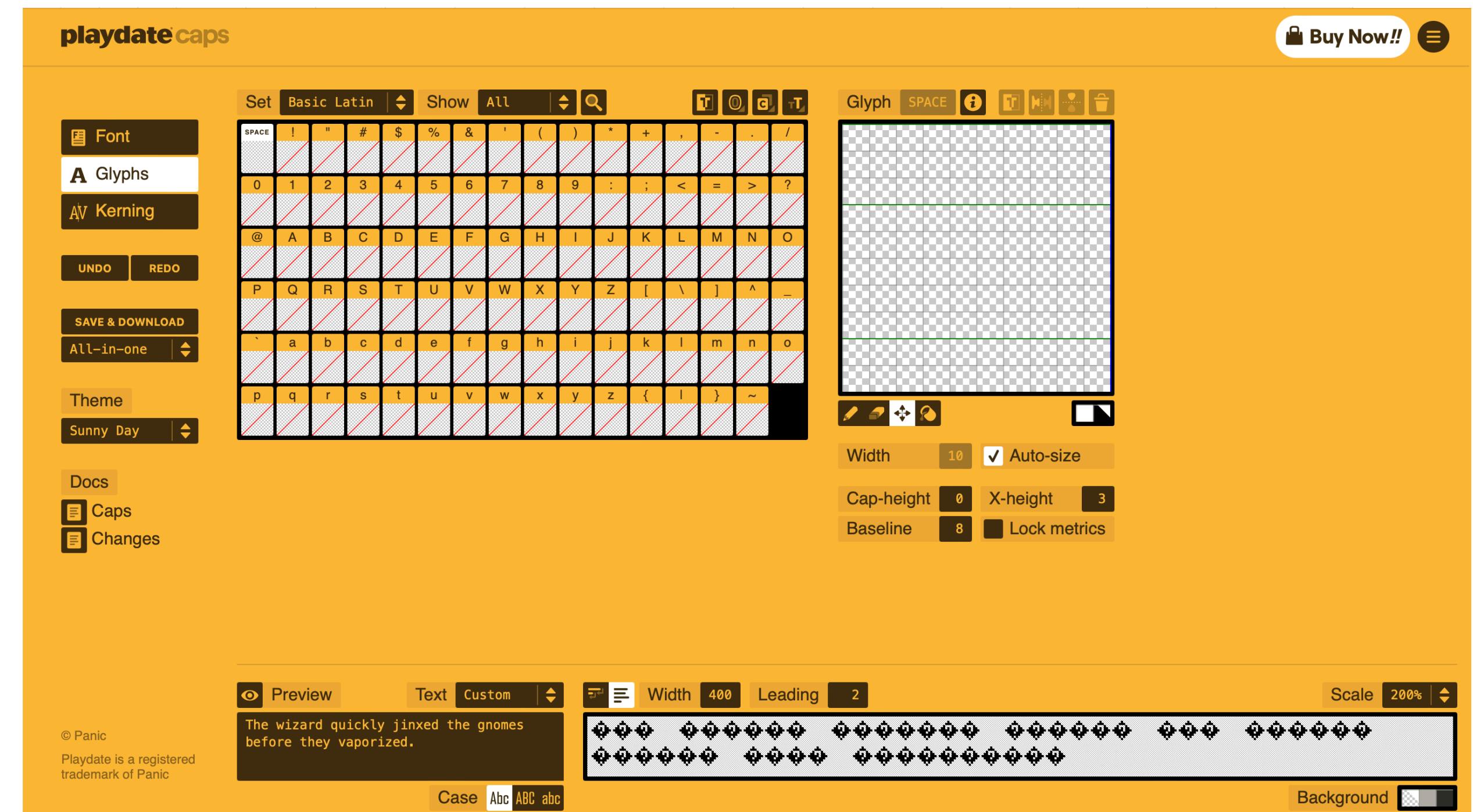
- Tools – Mirror
- Streamt Display auf Mac
- Separater Download:
<https://play.date/mirror/>



Playdate SDK

- Tools – Caps

- Online Bitmap-Font-Editor
- <https://play.date/caps/>



Playdate SDK

Dokumentation

- Sehr ausführlich
- Hilfreich!
- Offline verfügbar

playdate

2.1. Set PLAYDATE_SDK_PATH Environment Variable
3. IDEs
3.1. Xcode/CMake
3.2. Xcode/Make
3.3. CLion/CMake
4. Command line
4.1. CMake
4.2. Make
5. Building on Windows
5.1. Visual Studio/CMake
5.2. Install Development Tools
5.3. Set PLAYDATE_SDK_PATH Environment Variable
Building
5.4. Building for the Simulator using Visual Studio
5.5. Building for the Simulator using NMake
5.6. Building for the Playdate using NMake
Building for Release
6. Game Initialization
7. API reference
7.1. Utility functions
Memory allocation
Logging
Interacting with the System Menu
Time and Date
Miscellaneous
7.2. Audio
Channels
SoundSource
AudioSample
FilePlayer
SamplePlayer
PDSynth
Synth parameters
PDSynthInstrument



You may also want to add <path to SDK>/
pdc, pdutil and the Simulator from any lo

3. IDEs

3.1. Xcode/CMake

To use Xcode to build the example C projects, you will first need to open the CMake file. To do this, you'll need CMake installed, we recommend Homebrew. Open a terminal window, run the command to install CMake, and then run the command to clone the Playdate repository.

```
mkdir build
cd build
cmake .. -G "Xcode"
```

This will create an Xcode project in the `build` directory. You can then open the project and run and debug in the Simulator. When you want to build for the Playdate, navigate to the `build` folder (or create a second folder), then run these commands:

```
cd build
cmake -DCMAKE_TOOLCHAIN_FILE=<path to SDK>/C_API/toolchain.cmake
make
```

3.2. Xcode/Make

Using Xcode for Playdate development requires a few additional steps to get your environment working.

1. Begin by making sure you have a working Makefile installed. The `makefile` example project can serve as a starting template.
2. Create a new Xcode Project with a macos Library target. In the target settings, under the `Build` tab, make sure you add the word 'Simulator' to the target name.
3. Add your source files to the target's "Compile Sources" list.

Playdate SDK

- Tools – Unterstützte IDEs
- VSCode
- Nova
- xCode (👋)

SDK

PULP

LINKS

DEV HELP

CATALOG

Playdate SDK

All you need: Lua and C APIs, docs, as well as a Simulator for local development, with profiling and more.

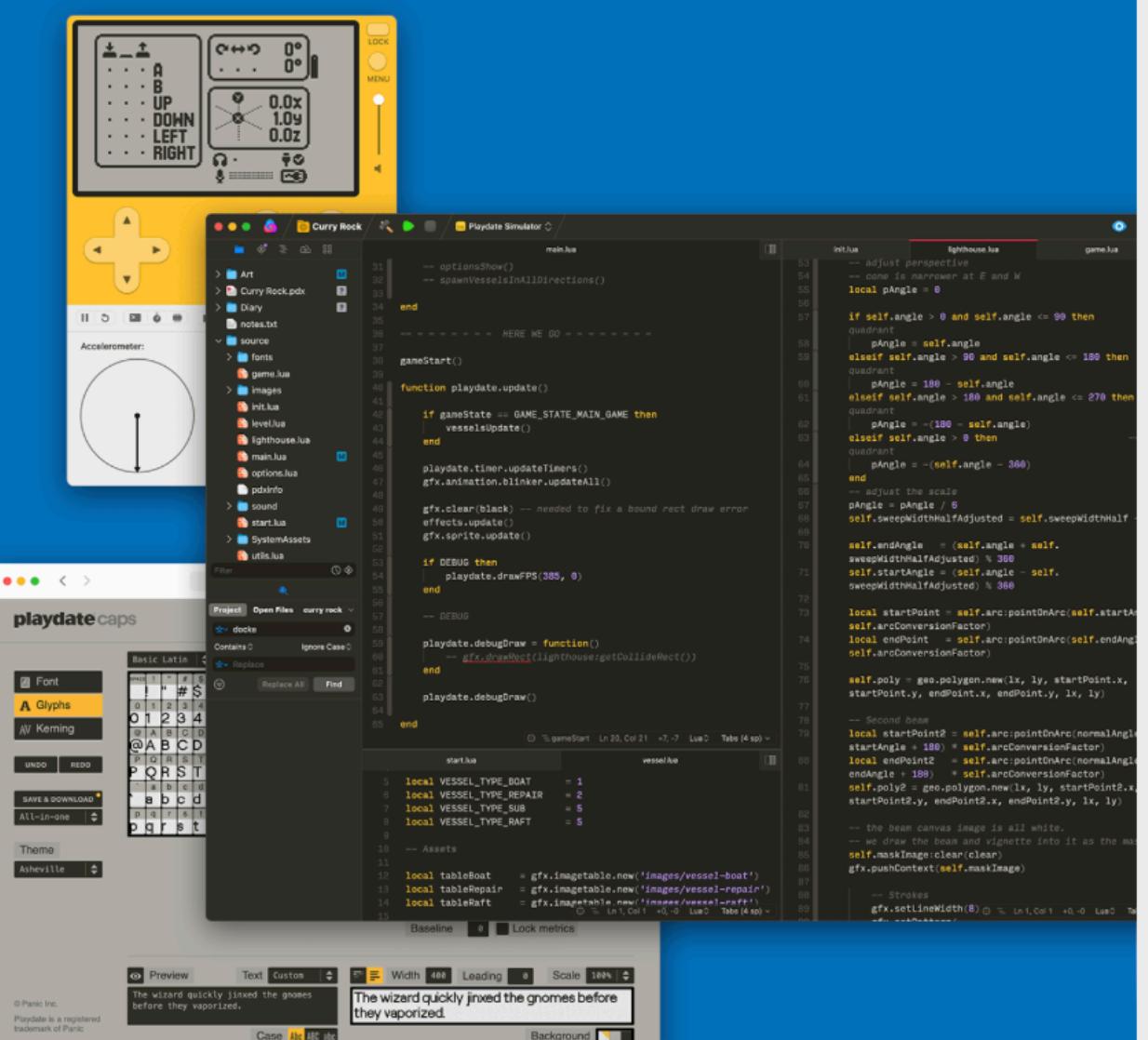
I agree to the [Playdate SDK License](#).

Download Playdate SDK 2.4.2 for macOS

[Download for Windows](#)

[Download for Linux](#)

☀ See what's new



Inside Playdate

The official reference documentation for programming Playdate games.

[Inside Playdate →](#)

[Inside Playdate with C →](#)



Libraries & Tools

Tutorials, editors, importable libraries, engines, templates,



Swift for Playdate

The Swift team's new embedded language support means you can

Vorbereitung

Install all the things!

- Recently Nightly Toolchain von Swift installieren:

[https://www.swift.org/download/
#snapshots](https://www.swift.org/download/#snapshots)

- Playdate-SDK installieren:

<https://play.date/dev/>

- PlaydateKit-Template-Repo clonen:

[https://github.com/finnvoor/
PlaydateKitTemplate](https://github.com/finnvoor/PlaydateKitTemplate)

- Xcode Toolchain konfigurieren

Swift 6.0 Development

Swift 6.0 snapshots are prebuilt binaries that are automatically created from `release/6.0` branch. These snapshots are not official releases. They have gone through automated unit testing, but they have not gone through the full testing that is performed for official releases.

Download	Date	Architecture	Docker Tag
Xcode	May 14, 2024	Universal Debugging Symbols	Unavailable

**Download Playdate SDK 2.4.2 for
macOS**



Toolchains			
Toolchain	Created	Origin	Size
Xcode 15.4		Xcode	
Swift 6.0 Development Snapshot	26.03.24	Swift Open Source	4,92 GB
Swift 6.0 Development Snapshot	06.04.24	Swift Open Source	4,94 GB
Swift 6.0 Development Snapshot	27.04.24	Swift Open Source	5,13 GB

Machen wir's uns bequem

Wrapper um C-Library

Pointer unsicher auspacken & mehr

```
404     /// Provide all sprites with collision rects containing the point at x, y to
405     /// the given `body` closure and return its result.
406     public func querySpritesAtPoint<T>(x: Float, y: Float, body: (UnsafeBufferPointer<OpaquePointer?>) -> T) -> T {
407         var count: Int32 = 0
408         let rawSprites = spriteAPI.querySpritesAtPoint.unsafelyUnwrapped(x, y, &count)
409         let sprites = UnsafeBufferPointer(start: rawSprites, count: Int(count))
410         defer { rawSprites?.deallocate() }
411         return body(sprites)
412     }
```

- Swift-Code übersichtlich halten
- Deallocationen sicherstellen
- Callbacks für Lifecycle
- Callbacks für Events

Gibt's da eventuell schon was?

PlaydateKit

“aims to be as Swift-y as possible”

- Vervollständigung des Wrappings aus ursprünglichen Template
- Lifecycle Management
- Run Loop

Currently, the following sections of the API are implemented:

- Display
- File
- Graphics
- JSON
- Lua
- Scoreboards
- Sound
 - FilePlayer
 - SamplePlayer
- Sprite
- System

<https://github.com/finnvoor/PlaydateKit>



Swift 6.0 Playdate SDK 2.4.2 Build passing

Now it can start

Fehlt eigentlich nur noch eine Idee.

Etch... ähm: Crank-A-Sketch

Was mit Kurbel ist bestimmt gut

- beides 1-Bit-Displays
- zum Löschen muss man schütteln (Accelerometer)
- Steuerung durch Drehung (Crank)
- Richtungswechsel mit D-Pad



Xcode Demo: Crank-A-Sketch

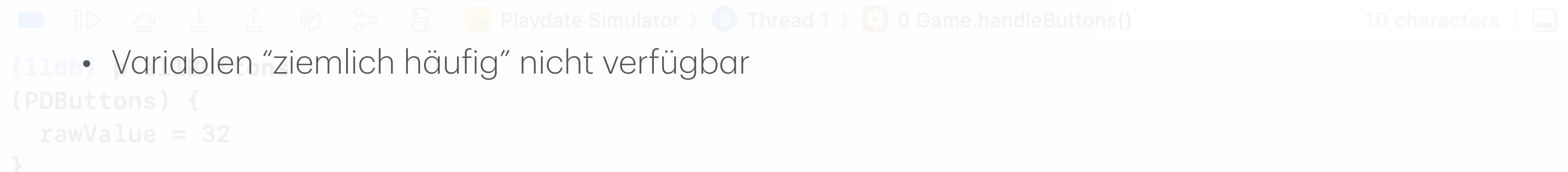
Wird schon schiefgehen ...

- Auswahl Toolchain
- PDCPlugin
- Package.swift
- Boilerplate entry code
- “Game” Code
- Shaker
- Sound
- Run in Simulator
- Deploy

Development

Nicht immer einfach

- Nur StaticString verfügbar, aber es gibt Hoffnung:
<https://forums.swift.org/t/embedded-swift/67057/183>
- Debugger muss manuell attached werden
- Breakpoints halten “ziemlich nah an der Stelle”
- Variablen “ziemlich häufig” nicht verfügbar



The screenshot shows the Xcode debugger interface. At the top, there's a toolbar with various icons. Below it, a stack trace is displayed, starting with "Playdate Simulator > Thread 1 > 0 Game.handleButtons()". To the right of the stack trace, there's a text input field with placeholder text "10 characters" and a small text icon. The main area of the debugger shows some code snippets and error messages related to button handling.

```
(lldb) p oldButtonState  
(PDBButtons) {  
    rawValue = 32  
}  
  
118.035 HALC_ProxyIOContext.cpp:1.328 HALC_ProxyIOContext::IOWorkLoop: skipping cycle due to overload
```

```
(lldb) p oldButtonState  
error: Expression evaluation failed. Retrying without binding generic parameters  
error: Could not evaluate the expression without binding generic types.
```

Testing

Computer says no

- Kein XCTest für Embedded Swift
- “Logik auslagern” ist uns nicht gelungen, da Plattformen nicht mischbar

Testing

"Akademischer Bullshit"

Markus M.

Let's get rich

<https://panic-inc.typeform.com/catalog>



Veröffentlichungsprozess

Manuell, aber mit Liebe

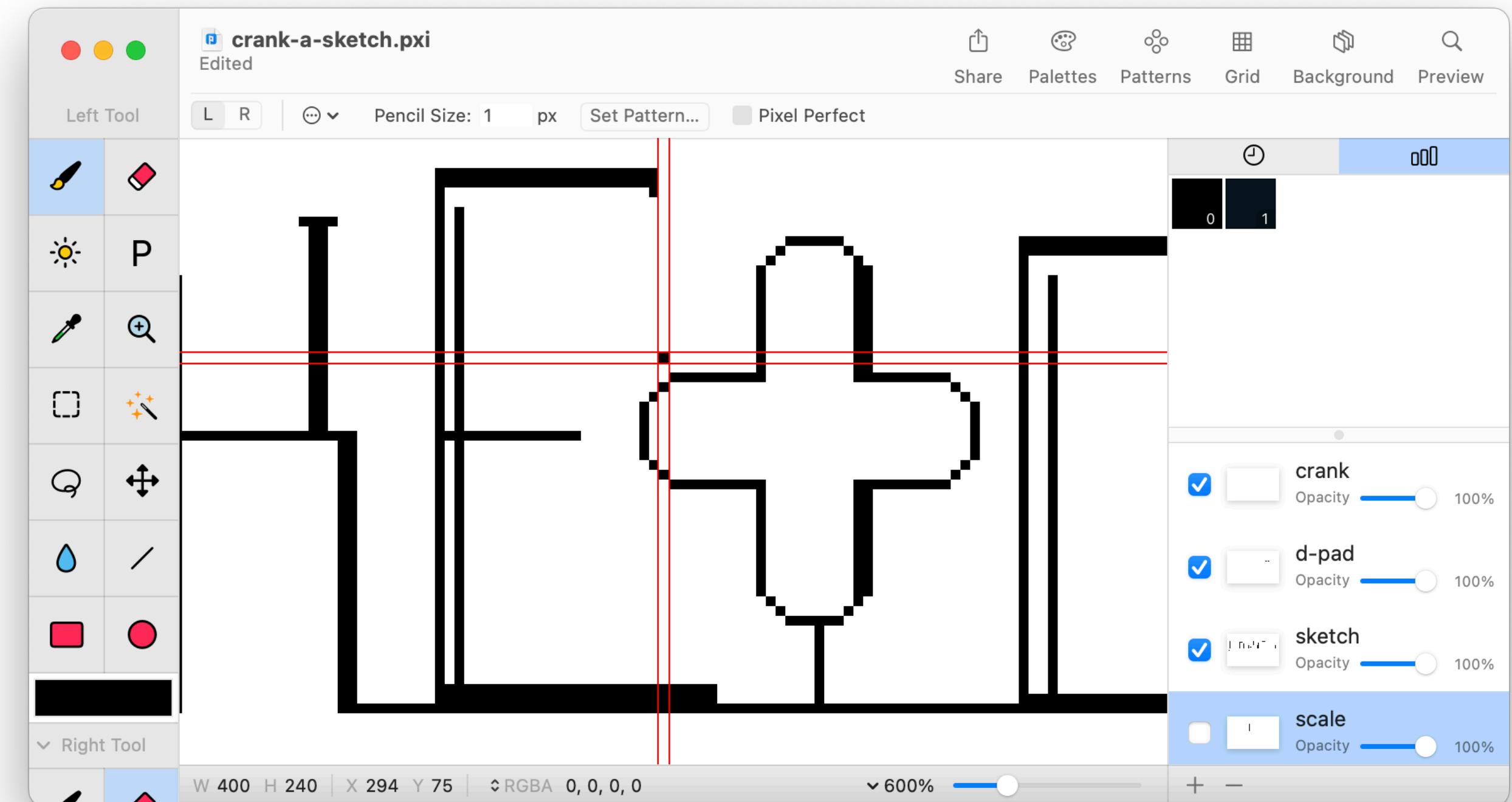
- Unterschrift Agreements
- Onboarding bei Stripe
- Tax Interview (7-Schritte für W-8BEN-E, nice!)
- Einreichung aller Assets (Grafiken, Texte)
- Upload Binary
- Alles extrem gut dokumentiert, persönlicher Kontakt via Email

Gestaltungsprozess

Manuell, aber mit Liebe

- aktuelle Tools haben wenig Augenmerk auf 1-Bit-Grafiken
- Photoshop völlig ungeeignet
- Sketch ebenfalls schwierig
- Alte Schule: Pixeln

ok, also mit photoshop kommt man nich weit wenn man 1bit png erzeugen will. ich muss da jetzt einen pixel exkurs machen, das wird tricky. fahr schön



Material

- <https://gdcvault.com/play/1034707/The-Playdate-Story-What-Was>
- <https://github.com/cranksters/playdate-reverse-engineering>
- <https://github.com/apple/swift-playdate-examples>
- <https://www.swift.org/download/#snapshots>
- <https://play.date/dev/>
- <https://github.com/finnvoor/PlaydateKitTemplate>

Danke!