

# IRDC 2012

Porting POWDER

Jeff Lait

# Overview

- How to Port Everywhere
  - Create a Bedrock
  - Opening to Outsiders
- Porting Hazards
  - Event Model
  - Controls
  - Screen Dimensions
  - Compilers

# Bedrock

- Bedrock Programming
  - To write everything at the lowest level
  - Seen as:
    - Direct hardware access
    - Assembly
    - Raw UDP packets
    - Foolish way to waste time

# Bedrock

- *Persistence implemented correctly is an architectural simplification, rather than yet another proprietary knob dumped on top of a tall, stinking heap of the same.*
  - Stanislav on Loper-Os

# Bedrock

- Bedrock Programming
  - Why do should we accept too low of a bedrock?
  - Note that:
    - Screens are arrays of pixels
    - Assembly isn't machine code
    - Raw UDP packets are abstracts of a network driver
  - Even if you **think** you are on the true-bottom layer
    - You aren't. Welcome to DOSBOX.
    - Or Pentium Pro

# Engine Programming

- POWDER was my first non-Engine game
- Game logic and engine are intertwined
- But...
  - They aren't

# GBA Bedrock

- Memory mapped graphics
  - 4 independently scrollable tiled arrays
    - With 1 bit transparency, shared tilelist
  - Some number of sprites
    - 1 bit transparency, independently placed
- Memory mapped NVRAM
- No disk
- Memory mapped button states

# Windows Port

- Write the GBA bedrock in SDL
  - SDL is itself a nice bedrock
- Create memory mapped locations on startup
- Bedrock is
  - Easy to write
  - Highly fault resistant – garbage in doesn't crash

# Every platform under the sun

- Write a cycle-for-cycle GBA bedrock
- Already done for almost every machine out there!
- GBA downloads outlive the GBA due to people playing POWDER on devices which have GBA emulators!

# How to write bedrock

- I am no expert, Nintendo showed me how!
- Think of it as a hardware problem
  - Very simple interface
  - Memory model rather than API
    - Allow partial updates!
  - Should **not** be affected by new game features!
    - Should not occur to you to dive lower!
- SDL, GL
- libtcod
  - Interesting as it is **not** from hardware

# Outsiders

- Source Code Available
- Does not have to be open source
- Requires:
  - Enough players
  - Simple dependencies!

# Open Source Rant

- Engine Free but Art Closed
  - So why do we say the game logic has no art?
  - What about scripts?
  - What about art that becomes scripts?
- Bedrock Free, Art Closed
  - There is no shame in coding the game in the same language you write the bedrock in.

# POWDER's Dependencies

- C++ Compiler
- LibSDL
  - Very well ported
- Everything bootstraps from there!
- Not even image readers are needed (bmp2c)

# Porting Hazards

- Event Model
- Controls
- Screen Dimensions
- Compilers

# Event Model

- POWDER does not have an event loop!
- `msg_asksno()`
  - Busy waits for an answer!
- Easier to program
- GBA hardware handled rendering, etc

# SDL Events

- Separate event thread
  - Updates memory mapped global registers
- Explicit `gfx_isnewframe()` invoked in all busy waits
  - Blocks to new frame
  - Triggers screen update
  - Useful on GBA for power saving

# iPhone Events

- Display must be done in event loop!
- Objective C++ interface for events
- Ideally use co-routines
  - Execute game logic from last block until next block
- C++ lacks co-routines
  - Separate game thread, block on condition variables

# iPhone Display

- Keep SDL bedrock, but replace SDL\_Image with a char \*
- gfx\_isnewframe still updates this bitmap
- External Objective C++ thread retains copies
  - Maps to GL texturemaps
  - Displays
- All of POWDER's game logic is blissfully ignorant of the substitution, since it occurs below the bedrock.

# Controls

- Initially 10 buttons only
  - Windows port directly mapped buttons to key states
- `#define SDL_HASKEYBOARD` and expose keyboard stream in the bedrock
- DS adds touch
  - Touch x/y/state global variables
  - Game made entirely stylus controlled
  - Windows Mouse then was trivial

# iPhone Touch

- A stylus is literally orders of magnitudes more accurate than a finger.
- DS sized buttons unhittable on iPhone
- Die a little inside and add onscreen controls
- Add giant buttons – 3x normal size
  - Live only in Objective C layer
  - POWDER's gameplay logic does not know about them!
  - Instead, Action Queue acts as a new input type

# Android Port

- TBD!
- Use NDK to run the POWDER thread
- Add Java wrappers for the input/output queues used by iPhone
- Add Java widgets for controls.

# Screen Dimensions

- GBA → Windows
  - Pixel doubling
- GBA → DS
  - Expand screen area
- DS → iPhone
  - Change base tile dimensions
  - Instead of 8x8, use 10x10 or 12x12
  - Pixel perfect is necessary!
- iPhone → Android
  - Cry.

# Compilers

- If you treat warnings as errors, you are not on enough platforms yet.
- `enum BAR { A, B, C };`
- `BAR foo = 4;`
- `if (foo == 4) { /* does this run? */ }`
  - Yes, I am still bitter
  - And the optimization was revoked in a later build!

# Summary

- Bedrock Programming is Good
- Release your Source