The amazing life & achievements of...

TASBot

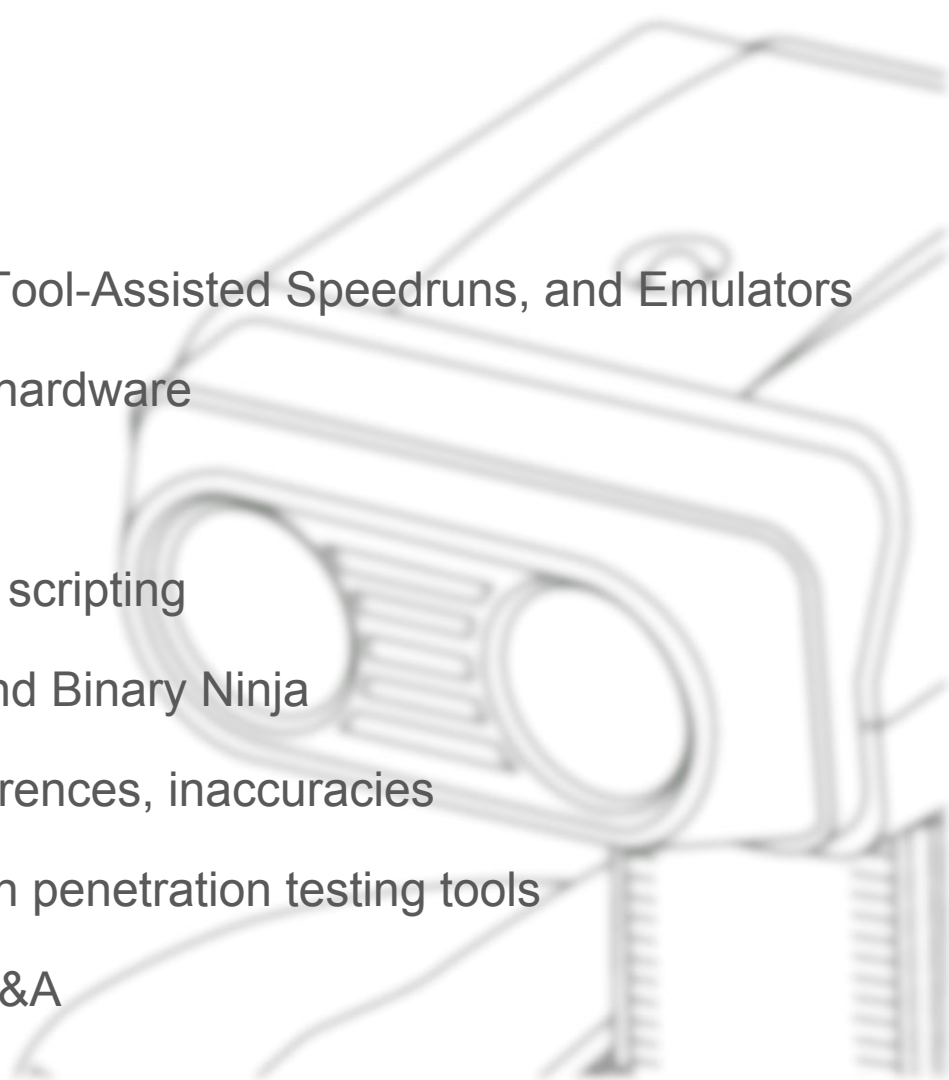
THE PERFECTIONIST

dwangoAC

DEFCON

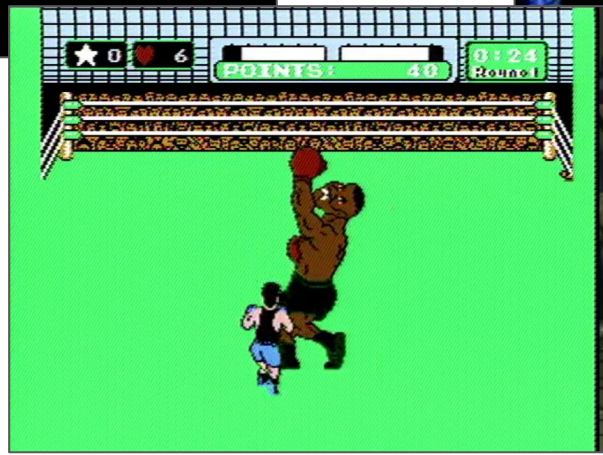
Agenda

- Intro to Speedrunning video games, Tool-Assisted Speedruns, and Emulators
- TASBot: Playing back a TAS on real hardware
- TAS techniques, history, evolution
- Emulator tools - memory search, Lua scripting
- Beyond emulators - disassemblers and Binary Ninja
- Remaining limitations - emulator differences, inaccuracies
- Key point: TAS tools are like really fun penetration testing tools
- Interactive demo of Pokemon Red, Q&A

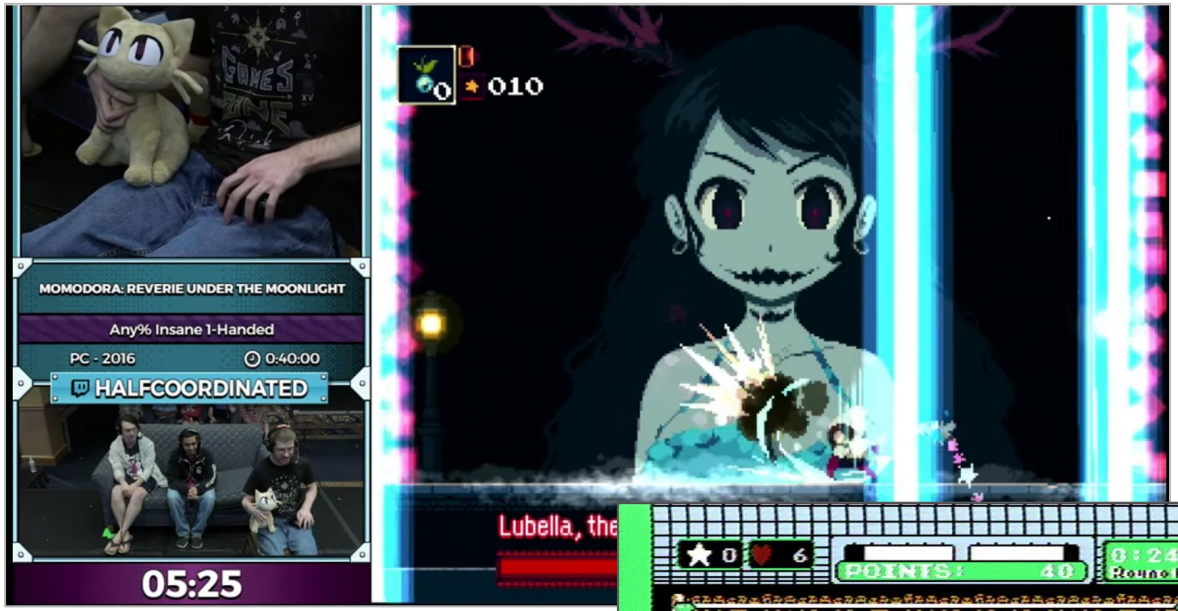


Speedrunning - playing games fast

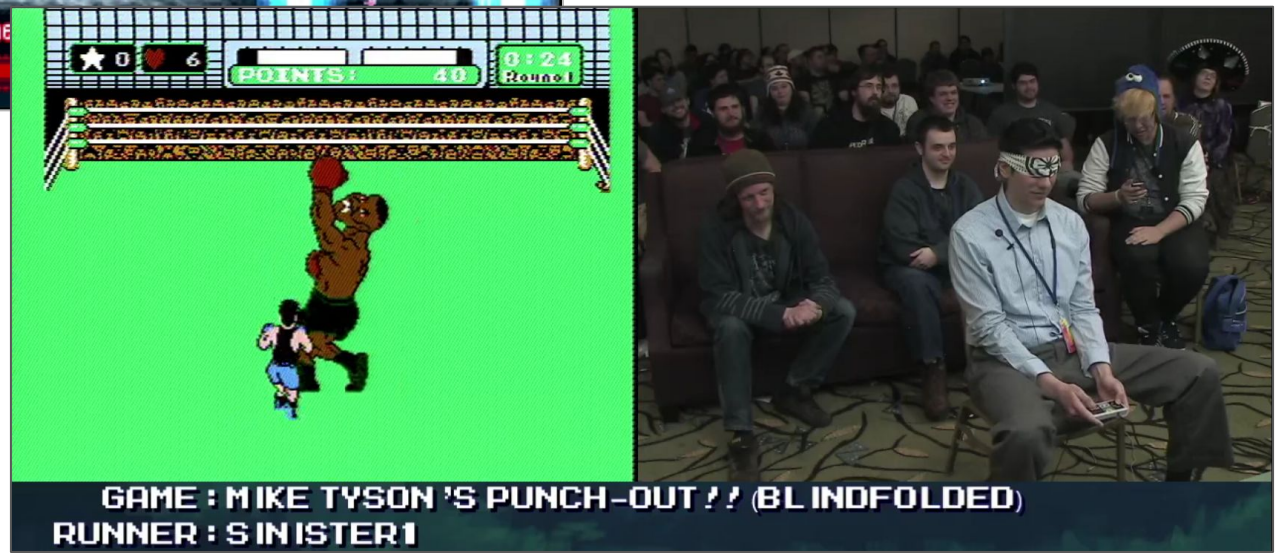
- Inspired in part by in-game completion timers (Metroid)
- Many categories, ranging from "any%" to "low% no major glitches"
- SpeedDemosArchive.com and others track fastest completion times
- Strict rules and peer review ensure no cheats or macros are employed
- Highly entertaining, especially for a game you've played normally



Speedrunning records verified from video captures or live at GDQ events



Even beyond standard limits: blindfolded, 1-handed...



Tool-Assisted Speedruns: playing games even faster

- Also called Tool-Assisted Superplays, used as a noun or verb as TAS, TASing, TAS'ed, etc.
- Early TAS's were usually made with tools built in to specific PC games (Doom, Quake)
- By the late 90's, Doom Done Quick was well known, beating the game in 19:41

Tool-Assisted Speedruns (TAS)

- Tool-Assisted Speedruns push the limits of the hardware and game rather than the human
- Emulator tools include saving / loading game states, frame advance, and scripting
- Movie files deterministically record every button press for later playback
- Let's be honest, it's basically the Doped Olympics, with no rules
    - Bad idea when it comes to humans, but a lot of fun when beating games
    - A 2003 run of SMB3 by Morimoto was unlabeled, causing much controversy
- TASVideos.org formed by Bisqwit, now hosts runs for many platforms

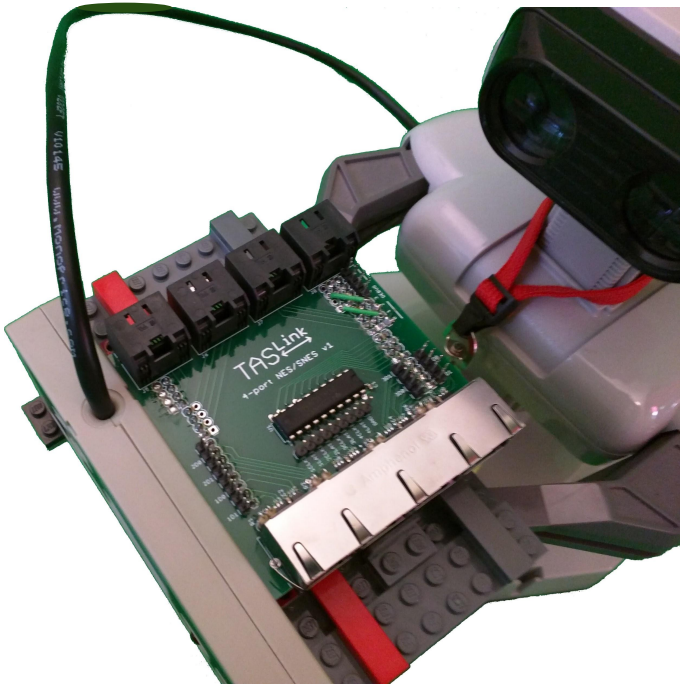
2003 SUPER MARIO BROS. 3

by Morimoto

TAS VIDEOS

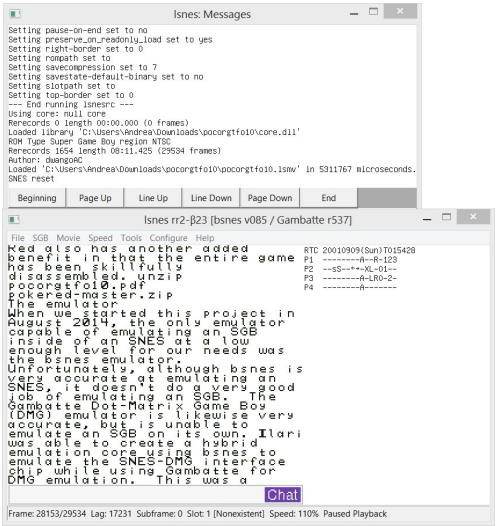
What, a live demo already?

- Live demo: TASBot playing back a movie of SMB3 on real hardware
- Explanations forthcoming while TASBot happily mimics a real controller



Rerecording enabled video game emulators and frameworks

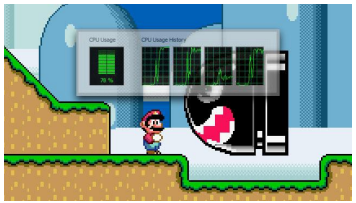
- TAS techniques are enabled by emulators of video game consoles
  - FCEUX (NES), lsnes (SNES), VBA (Game Boy), BizHawk (multiple platforms)
- Some platforms, such as Windows, have rerecording frameworks
  - Hourglass, specialized projects like nethack-tas-tools
- Some emulators are very accurate, with scripting and memory search tools



Emulation accuracy has improved over time

- Early emulators were highly inaccurate, often unusably so
- Emulation accuracy was improved through clean room reverse engineering
    - OK, sometimes using not-so-clean techniques and stolen manuals
- Some took emulation accuracy to extreme levels (Byuu) at the cost of usability
- This obsession allows movie files to match actual hardware, frame for frame

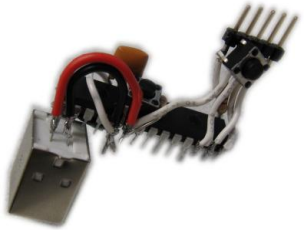
bsnes EMULATOR

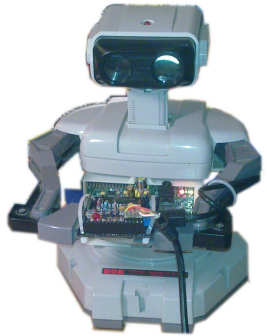


"Console verification", all without voiding any (long expired) warranties

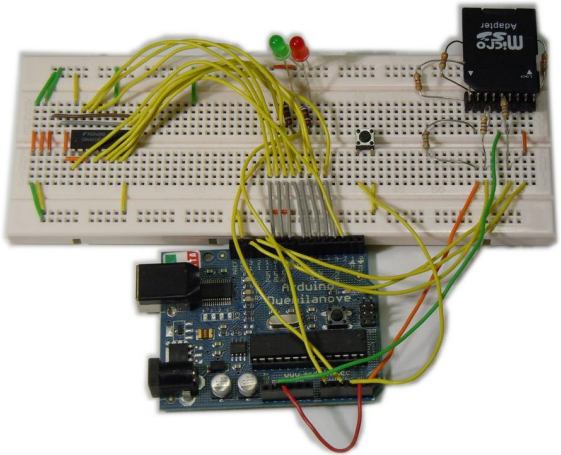
- In 2009, true of TASVideos.org used a PIC microcontroller to press NES buttons
- By 2011, micro500 built his NESBot and demonstrated the first replay of SMB1
    - DarkKobold used an NESBot at SGDQ 2011 showing SMB2 and Wizards and Warriors 3

- Through 2012, devices for other consoles were added, such as Genesis and N64
- I (as dwangoAC) pitched TAS's for AGDQ 2014 resulting in true making a new device
    - I combined a board with a R.O.B. using Legos and others named him TASBot

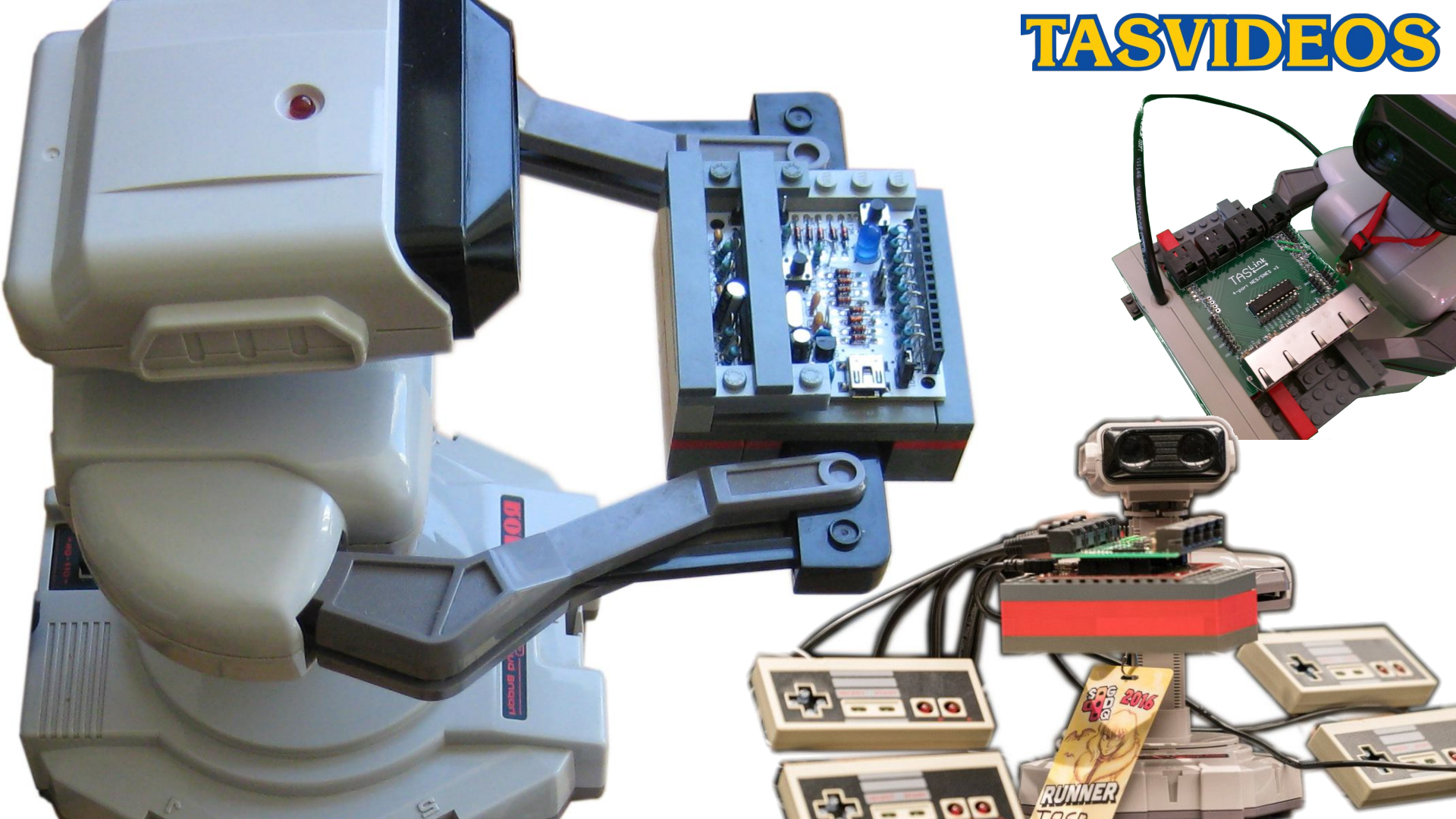




Console verification devices over time

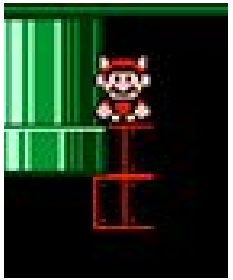


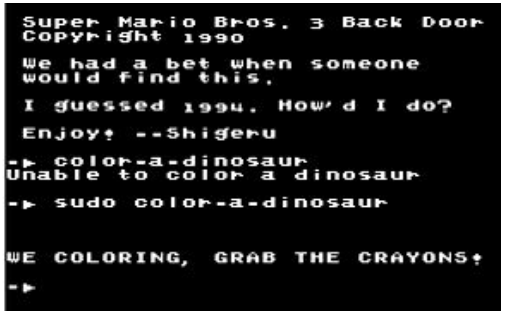
- 2011
  - NESBot from micro500 - original Instructable, breadboard design
  - Droid64bot from SoulCal - first 3D console verification
- 2012
  - N64 bot from micro500
- 2013
  - SNES and Genesis Arduino bot from GhostSonic
  - NES/SNES replay device from true - streaming capable and inexpensive but slow
  - Multireplay device from true - self-contained device, faster datatarates
- 2015
  - Game Boy Player Player from endrift - used for GBA games on a GameCube
- 2016
  - TASLink - FPGA based, expensive, but very flexible and fast

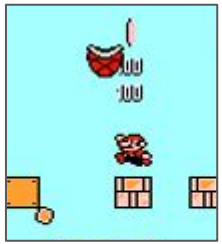

TASVIDEOS
RUNNER
2016

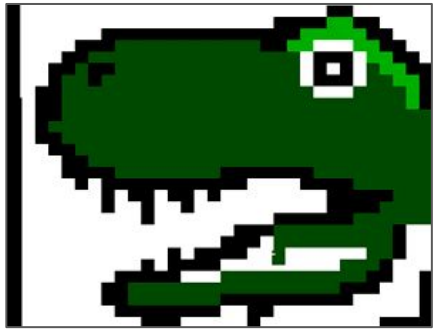
Arbitrary Code Execution / Exploit

- Effectively like having total control; the game becomes a playground









OK, what the heck did I just see?

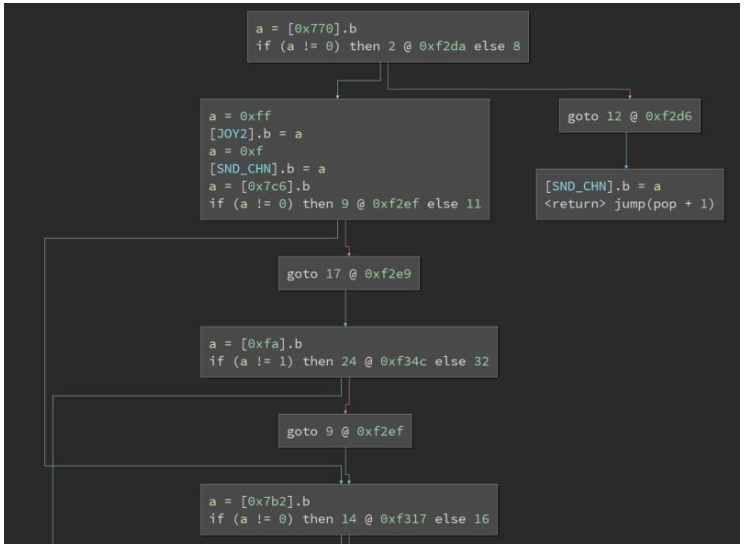
- The tools that allow us to beat games quickly also allow us to glitch them
- Sometimes we can make games execute opcodes of our choosing
- Doing so requires delving deeper into TAS tools

Advanced emulator tools: Memory searching, Lua scripting, disassembly

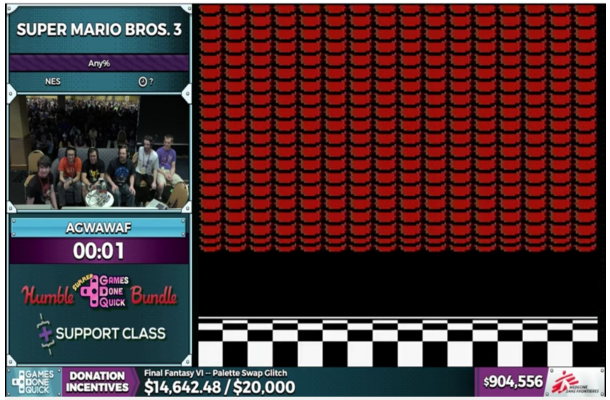
- Search tools combined with frame advance and savestates can be very powerful
- Find Mario's speed: Save a state, reset memory search, run forward
  - By eliminating values that don't increment you can find the correct address
- Disassembly of RAM or ROM can tell you what will happen if it is triggered
  - Dissassemblers range in ability and level of integration but are very helpful

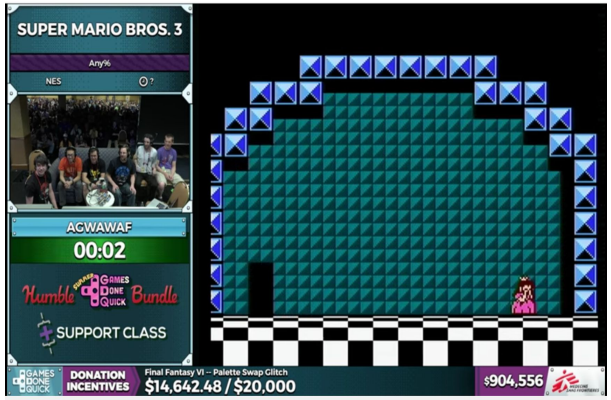
Binary Ninja: Adding Reverse Engineering to the TASing toolbelt

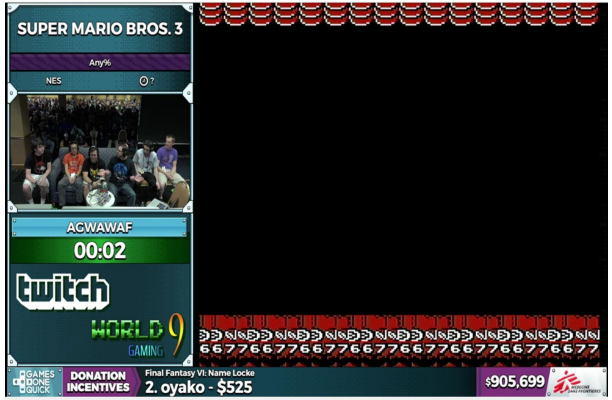
- Recent tool focused on Reverse Engineering like IDA but more flexible
- Graph view visual representation with low level IL and annotation support
- Python scripting comes with NES support and ability to add new mappers
- Still in beta, future versions will add advanced searching and multi-module UI



BINARYNINJA

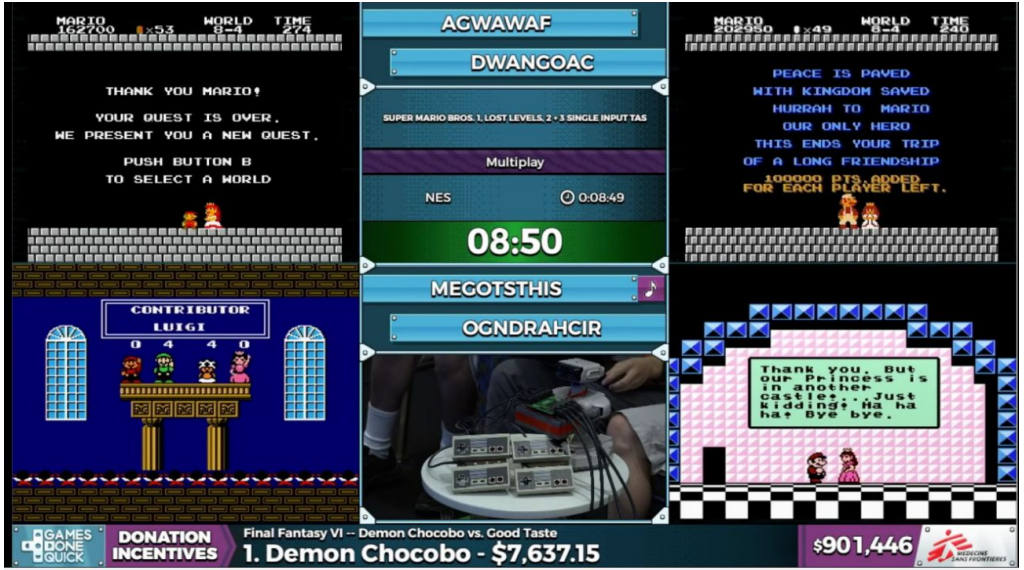




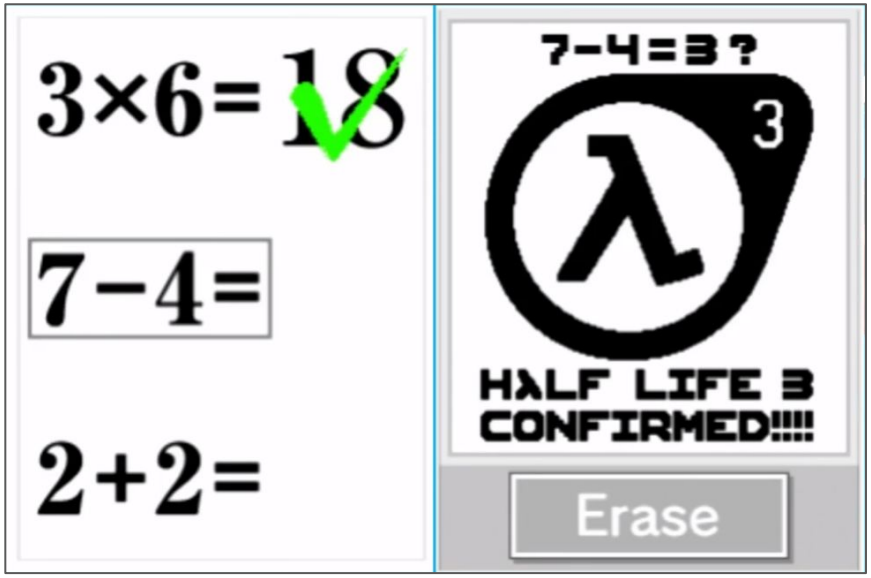


From boot to ending in 16 frames by changing input every other poll, eventually executing the controller addresses as opcodes and jumping to the end credits

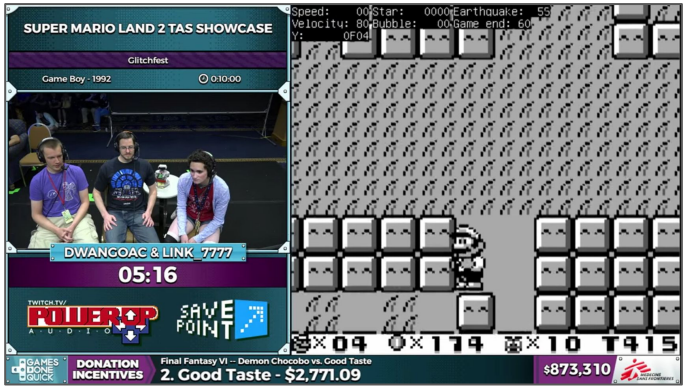
SGDQ 2016
SMB1+2+3+Lost Levels simultaneously



AGDQ 2016: Brain Age



Visual memory editor

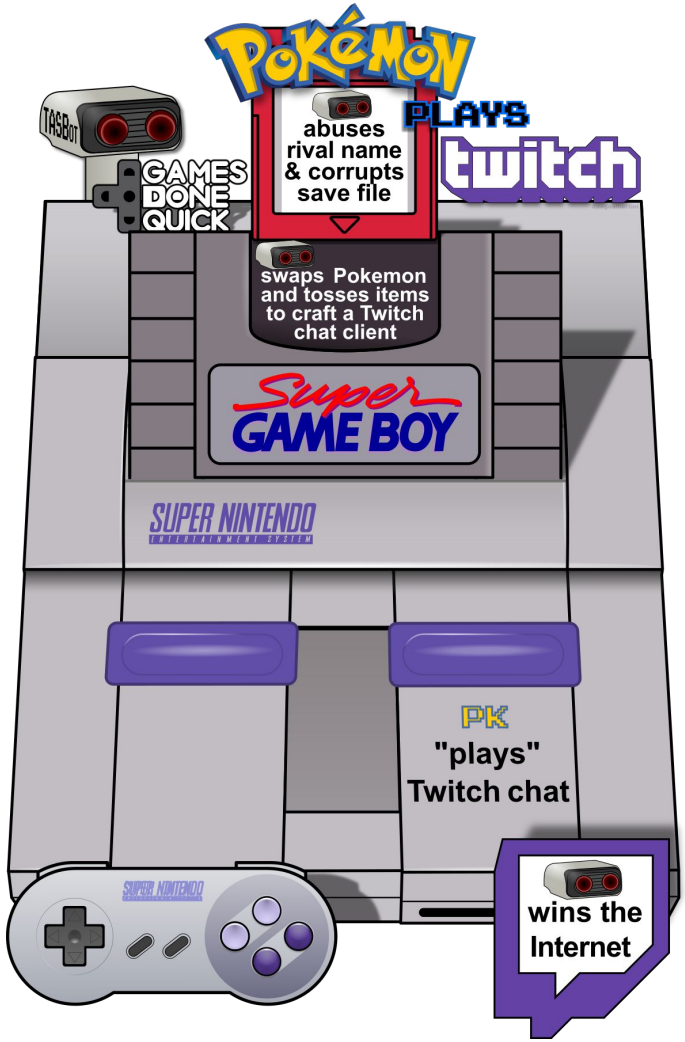


The tools and terminology in making a TAS translate to security research

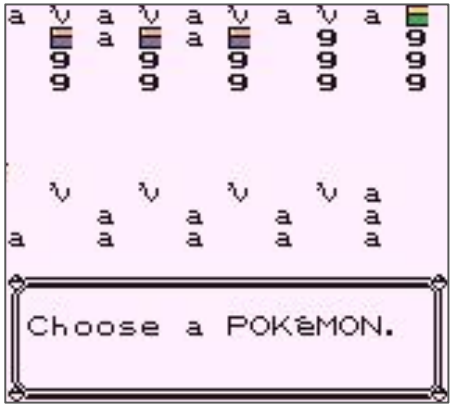
- Vernacular differences abound, but the principles are the same
  - Savestate = VM snapshot
  - Frame advance = VM CPU tick
  - Glitch = Vulnerability
  - Total Control = Pwned / Arbitrary Control Exploit / root exploit (if consoles had such a concept)
- Learning how to make a TAS can be a fun and educational experience

Anatomy of a complex Arbitrary Code Execution

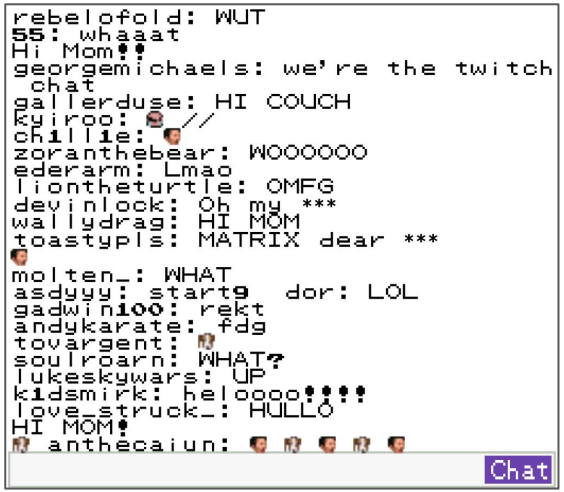
- Pokemon Red can be compromised in a very unique way
- Values in the controller register treated as opcodes allow taking over SGB
- Once full access to SNES is gained, anything is possible


Pokémon
PLAYS
twitch
TASBot
GAMES DONE QUICK
abuses rival name & corrupts save file
swaps Pokemon and tosses items to craft a Twitch chat client
Super GAME BOY
SUPER NINTENDO
PK
"plays" Twitch chat
wins the Internet


RIVAL's NAME?
lower case


Choose a POKéMON.


CANCEL
SUPER REPEL
Threw away
CANCEL.


rebelofold: WUT
55: whaaat
Hi Mom!!
georgemichaels: we're the twitch chat
gallerduse: HI COUCH
kyiroo:
chillie:
zoranthebear: WOOOOOO
ederarm: Lmao
liontheturtle: OMFG
devinlock: Oh my ***
wallydrag: HI MOM
toastypls: MATRIX dear ***
molten_: WHAT
asdyyy: start9 dor: LOL
gadwin100: rekt
andykarate: fdg
tovargent:
soulroarn: WHAT?
lukeskywars: UP
kidsmirk: helooooo!!!!
love_struck_: HULLO
HI MOM!
anthecaiun:
Chat

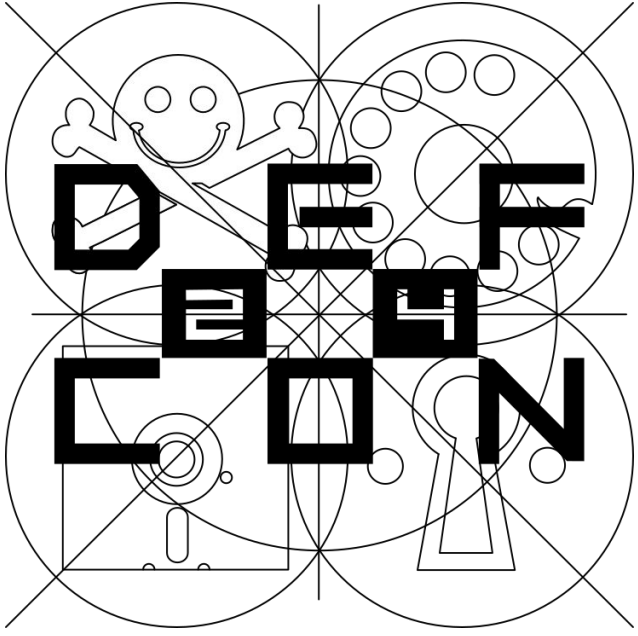

Pokemon Plays Twitch Credits:
dwangoAC: Main project organizer and presenter, primary tester, Stage 0/1 movie files, PR
Ilari: Emulator coder (lsnes), Stages 3-5 developer, payload tester, game mechanic researcher
p4plus2: Primary payload author, encoding scheme creator, SNES expert, general stage 3-x help
Masterjun: Stage 0/1 original idea and research, SNES advice
micro500: Wiring harness build, Python IRC to bot streaming, poll speed firmware modification
true: Creator of NES/SNES Replay device, reset handling updates
TheAxeMan: Python script support
tais523: Data encoding assistance
Nach: SGB docs, site defacement
Tompa: On-site hardware support
padz: Pokemon Red research
Vulaj in: Camera serial interface
twm: Initial IRC Python script
Full article in next PoC||GTFO!
Chat
2015
RUNNER
TAS Bot
GAME BOY

References, thanks, and bibliography

- http://tasvideos.org/TASBot.html - history of TASBot / historical information on how I organized teams of people to participate at Games Done Quick events, ultimately helping raise over $234k for charity and presenting in front of nearly 200,000 live viewers
- http://arstechnica.com/gaming/2014/01/how-an-emulator-fueled-robot-reprogrammed-super-mario-world-on-the-fly/ - This is the first presentation I gave at a Games Done Quick event; this initial exploit caused TASBot to become a known name in the community
- http://arstechnica.com/gaming/2015/01/pokemon-plays-twitch-how-a-robot-got-irc-running-on-an-unmodified-snes/ - This was quite possibly the most difficult (and arguably impressive) feat to date
- https://www.alchemistowl.org/pocorgtfo/pocorgtfo10.pdf page 6 - Pokemon Plays Twitch: This journal article written by myself as well as the author of the lsnes emulator and the author of the chat payload
- http://arstechnica.com/gaming/2016/07/how-to-beat-super-mario-bros-3-in-less-than-a-second/ - Details on the 16-frame SMB3 completion

The antics described in this talk would not have been possible without the help of a very, very long list of talented TAS'ers and hackers both from TASVideos.org and elsewhere, including: micro500 - co-presenter, Ilari - Emulator coder, p4plus2 - payload author, Masterjun - TAS glitchfinder, true - hardware dev, TheAxeMan - Python script support, ais523 - Mathematician, pretty much everyone in #tasvideos, and a long list of others I'm forgetting as I always seem to do whenever I'm thanking people. Thanks also goes to the staff of Games Done Quick for organizing an awesome event and giving us a reason to do all the crazy things we do.

Questions?



Presented and written by Allan Cecil (AKA dwangoAC)

SLIDES BY ANGE ALBERTINI