More Than an Emulator

- Living documentation
  - Device-based emulation
  - Running original software
  - Interactive debugger
- Preservation of digital heritage

Software is Culture

- Eligible for copyright protection
- Shared experiences
- Entertainment and storytelling medium

All Culture Needs to be Preserved

- Hopes and fears of a generation
- Window into a way of life
- Passing down memories
- Software is fragile

Dragging MAME into the 21st Century

Practicalities of a large open source project with two decades of history
(this is not a technical talk)

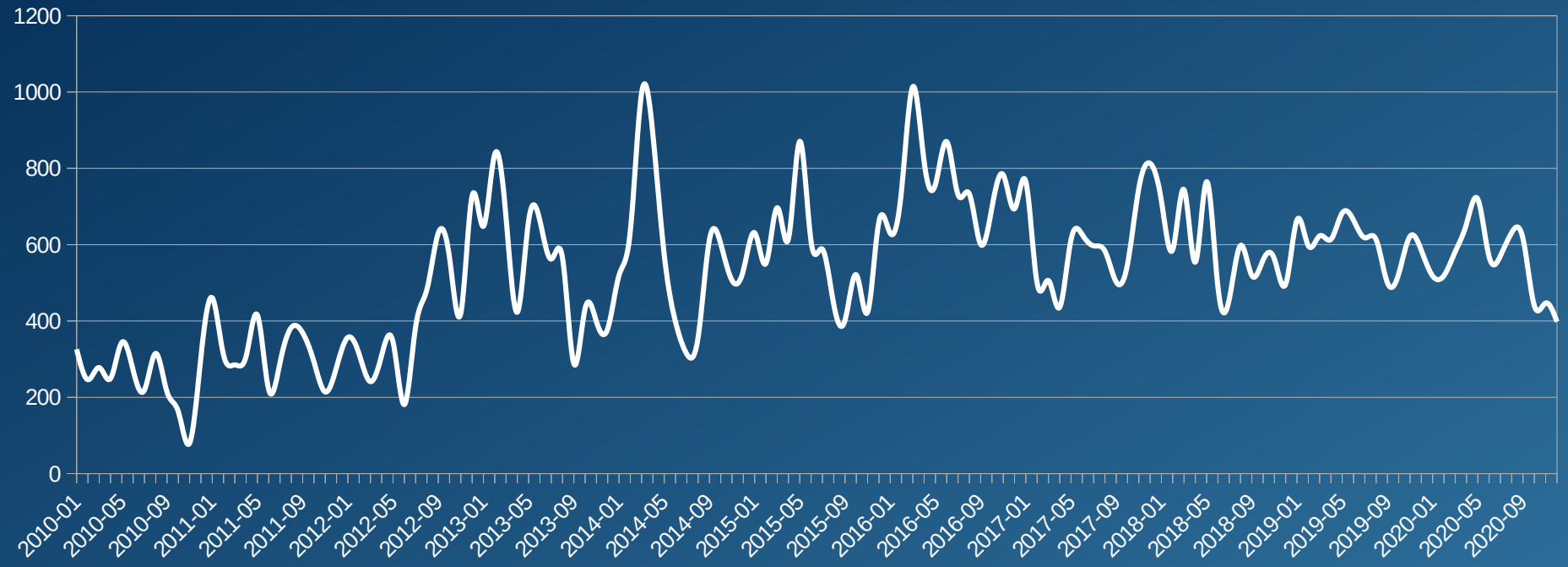
Milestones

- Feb ’97: Multi-Pac becomes MAME
- Jun ’98: First MESS release
- Oct ’14: Source on GitHub
- May ’15: MAME absorbs MESS
- Mar ’16: GPL re-licensing completed

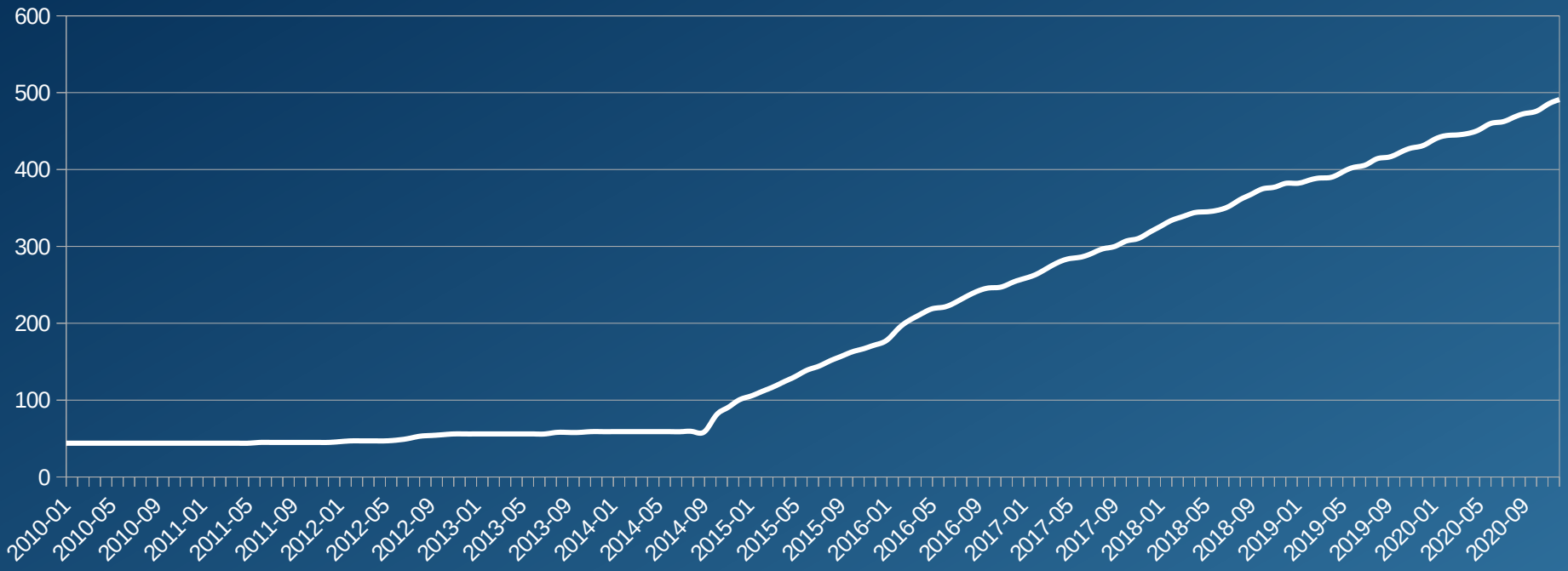
Rumours of Demise

- Systems no-one cares about
- MAME always gets slower
- Grumpy old men

Commit Activity



Unique Authors



Developers!

- MAME lives on active development
- Users follow development
- Inactive projects wither

Challenges

- MAME is mature
  - Basic functionality is complete
  - The easy stuff is done
  - Decades of legacy
  - Core changes are difficult

Challenges

- Constraints
  - Working within MAME’s architecture
  - Keeping up with core changes
  - Single-system emulators are simpler
  - MAME philosophy

Attractions

- Interesting challenges
- Nostalgia
- Not your day job
- Big device library

Scope

- MAME was exclusive
  - Arcade video games only
  - No gambling systems
  - No low-effort bootlegs

Scope

- Proliferation of forks
  - Duplicated effort
  - Development silos

Scope

- Benefits of absorbing the forks
  - More test cases
  - Improvements benefit everyone
  - Talent under one roof
  - No need to choose a fork

Approachability

- Something for everyone
  - Sourcing and dumping media
  - Reporting emulation issues
  - Layouts for non-video systems
  - Documentation

Approachability

- Regular releases
  - Users see progress faster
  - Checkpoints for tracking regressions
  - Infrequent releases are unwieldy

Approachability

- Public version control
  - Frequent updates are easier to follow
  - See changes as they happen
  - Quicker community feedback

Approachability

- Transparent review process
  - MAME had this wrong for years
  - Everyone benefits from public feedback
  - Tools can really help

Approachability

- Idiomatic code – before

```
static MACHINE_CONFIG_FRAGMENT( sound_2151 )
    MCFG_SPEAKER_STANDARD_MONO("mono")

    MCFG_YM2151_ADD("ymsnd", XTAL_3_579545MHz )
    MCFG_YM2151_IRQ_HANDLER(INPUTLINE("audiocpu", 0))
    MCFG_SOUND_ROUTE(0, "mono", 0.50)
    MCFG_SOUND_ROUTE(1, "mono", 0.50)

    MCFG_OKIM6295_ADD("oki", XTAL_1MHz, OKIM6295_PIN7_HIGH)
    MCFG_SOUND_ROUTE(ALL_OUTPUTS, "mono", 0.60)
MACHINE_CONFIG_END
```

Approachability

- Idiomatic code – before

```
#define MCFG_YM2151_ADD(_tag, _clock) \
        MCFG_DEVICE_ADD(_tag, YM2151, _clock)

#define MCFG_YM2151_IRQ_HANDLER(_devcb) \
        devcb = &ym2151_device::set_irq_handler( \
            *device, DEVCB_##_devcb);

#define MCFG_YM2151_PORT_WRITE_HANDLER(_devcb) \
        devcb = &ym2151_device::set_port_write_handler( \
            *device, DEVCB_##_devcb);
```

Approachability

- Idiomatic code – before

```
template <class Object>
static devcb_base &set_irq_handler(device_t &dev, Object obj) {
    return downcast<ym2151_device &>(dev)
            .m_irqhandler.set_callback(obj);
}

template <class Object>
static devcb_base &set_port_write_handler(device_t &dev, Object obj) {
    return downcast<ym2151_device &>(dev)
            .m_portwritehandler.set_callback(obj);
}
```

Approachability

- Idiomatic code – after

```
void dooyong_z80_state::sound_2151(machine_config &config) {
    SPEAKER(config, "mono").front_center();

    ym2151_device &ymsnd(YM2151(config, "ymsnd", 3.579'545_Mhz_XTAL));
    ymsnd.irq_handler().set_inputline(m_audiocpu, 0);
    ymsnd.add_route(0, "mono", 0.50);
    ymsnd.add_route(1, "mono", 0.50);

    OKIM6295(config, "oki", 1_Mhz_XTAL, okim6295_device::PIN7_HIGH)
            .add_route(ALL_OUTPUTS, "mono", 0.60);
}
```

Approachability

- Idiomatic code – after

```
auto irq_handler() { return m_irqhandler.bind(); }
auto port_write_handler() { return m_portwritehandler.bind(); }
```

Approachability

- Make the most of the language
  - Features make languages more expressive
  - Use features where they make sense
  - Don’t use features just for the sake of using them

Refactoring

- It’s difficult
  - Language and compiler limitations
  - Catering to all use cases
  - Future-proofing
  - Time-consuming in a large project

Refactoring

- Things get worse before they get better
  - Supporting old and new syntax
  - Clashing styles
  - Not adding legacy code
  - Few examples of new syntax

Refactoring

- It pays off
  - Higher productivity
  - Lower barriers to entry
  - More contributors

Project Management

- High level and low level, nothing in between
  - Setting overall direction
  - Best practices
  - No task assignments or priorities

Project Management

- Be prepared to make decisions
  - Decisions won’t make everyone happy
  - Indecision makes everyone unhappy
  - Decisions need to be well reasoned
  - Explain your decisions

Project Management

- Set quality standards
  - Bad code is more effort to fix later
  - Explain what’s wrong with submissions
  - Document standards if possible

Project Management

- Your job is to make sure they can do theirs

Choosing a License

- Use an OSI- or FSF-approved license
  - Written by real IP lawyers
  - Widely understood
  - Perks like access to tools and services

Choosing a License

- The MAME license
  - “Redistributions may not be sold, nor may they be used in a commercial product or activity.”
- Pitfalls of custom licenses
  - Incompatible with other software licenses
  - Unintended side-effects

Choosing a License

- Switching licenses wastes time
  - Tracking down contributors
  - Rewriting code that can’t be re-licensed
  - Time that could be better spent productively

Promotion

- Keep people interested
  - Release notes
  - Progress reports
  - Social media presence
  - Low-effort requests

Random Advice

- Stay true to your goals
- Bigger than any one person
- You can’t finish something you don’t start
- If it stops being fun, take a step back
- Don’t lose sight of MAME’s purpose