

# Analisi del progetto

MAME™ - Multiple Arcade Machine Emulator

Andrea “Pawz” Pavan



*Una visione di MAME rispetto le tecnologie di sviluppo Open Source*

---

---

Ultimo aggiornamento: 24 ottobre 2010  
Allineata con la versione 0.140

---

## Indice

<b>Prefazione</b>	<b>3</b>
<b>1 Introduction</b>	<b>3</b>
<b>2 Vision</b>	<b>3</b>
<b>3 Story</b>	<b>6</b>
<b>4 Market</b>	<b>9</b>
4.1 Giocatori . . . . .	10
4.2 Gestori di sale giochi . . . . .	10
4.3 Collezionisti . . . . .	11
4.4 Programmatori . . . . .	11
<b>5 License</b>	<b>12</b>
5.1 Lavori derivati . . . . .	13
5.2 Trademark . . . . .	14
<b>6 Business Model</b>	<b>15</b>
<b>7 Development Process</b>	<b>17</b>
<b>8 Community</b>	<b>20</b>
8.1 Le “build” di MAME . . . . .	23
<b>9 Development and Information Management Tools</b>	<b>24</b>
9.1 Sito web . . . . .	24
9.2 Mailing List . . . . .	25
9.3 Versioning . . . . .	25
9.4 Bug Tracking . . . . .	26
9.5 Real-time chat . . . . .	26
9.6 Wiki . . . . .	26
9.7 Build automation . . . . .	27
9.8 Note di rilascio . . . . .	28
9.9 Ambiente di sviluppo . . . . .	29
<b>Ringraziamenti</b>	<b>29</b>
<b>Riferimenti bibliografici</b>	<b>30</b>
<b>Storico</b>	<b>32</b>

## Prefazione

Il presente documento nasce da una relazione redatta dall'autore per sostenere l'esame di Tecnologie Open Source, corso inserito nel Corso di Laurea in Informatica dell'Università degli Studi di Padova, nell'Anno Accademico 2009/2010. Allo stato attuale il documento si presenta nella sua forma originaria, tranne per la pagina di apertura, ma non è detto che in futuro non ci possano essere degli aggiornamenti.

### Licenza

Il documento è rilasciato sotto licenza Creative Commons *Attribuzione - Non commerciale - Non opere derivate (BY-NC-ND) 2.5* valida per la legislazione italiana. Per conoscere i diritti concessi a chi riceve una copia del documento è possibile consultare l'indirizzo <http://creativecommons.org/licenses/by-nc-nd/2.5/it>.

## 1 Introduction

MAME è un progetto che nasce con lo scopo di realizzare un emulatore di sistemi elettronici videoludici da sala giochi, detti comunemente *giochi arcade*. MAME è l'acronimo di Multiple Arcade Machine Emulator. La pronuncia corretta deve seguire le regole fonetiche della lingua italiana.

In grande sintesi, si tratta di un programma che riproduce l'intero comparto hardware su cui funzionano i giochi arcade negli attuali computer general purpose. Se vengono fornite le immagini originali dei giochi, cioè l'estrapolazione dei dati (dump) contenuti nelle ROM e in altri sistemi di storage (CD-ROM, GD-ROM, Hard Disk, ...), MAME è in grado di eseguirli nel modo più fedele possibile.

Nell'ultima versione rilasciata, la 0.140, MAME gestisce oltre 1000 sistemi hardware diversi con i rispettivi giochi che ammontano a 9012, di cui 4510 unici.

Esistono diverse versioni derivate di MAME chiamate comunemente "build", ciascuna con le sue peculiarità. L'unica versione ufficiale gestita dal core team, i *MAME Dev*, è quella a riga di comando, scaricabile dal sito ufficiale [1].

## 2 Vision

MAME è un progetto strettamente non-profit. Il suo scopo principale è quello di essere un punto di riferimento della conoscenza nell'ambito dei videogiochi arcade e di esplicitare il funzionamento delle architetture hardware e software su cui sono basati. Tale scopo è perseguito sia per fini educativi che preservativi, in modo che queste informazioni non vengano perdute per sempre una volta che l'hardware su cui operano i giochi cessa di funzionare o esistere. Le informazioni da preservare riguardano il funzionamento sia dei

giochi arcade, precursori della storia dei videogiochi, sia quello di tutte le componenti elettroniche che formano la piattaforma hardware su cui operano.

L'importanza di preservare queste informazioni è dovuta al fatto che i videogiochi arcade sono a pieno titolo parte della nostra cultura popolare. Per capirne l'importanza basta chiedersi: *“Chi non ha mai giocato, socializzato, inveito . . . davanti ad un gioco da bar?”*, dalla risposta si potrà capire come questa tipologia di giochi ha segnato un'era ed è entrata nella vita di molte persone, più o meno giovani.

Il modo più ovvio per preservare i giochi è collezionarne e mantenerne in efficienza le schede originali. Sono allo stesso tempo importanti e devono quindi essere preservati anche i manuali, le decorazioni dei cabinati<sup>1</sup>, gli opuscoli pubblicitari e tutto ciò che può essere attinente. Quest'ultimi, seppur sembrino di second'ordine, sono in realtà più critici e difficili da preservare, richiedono quindi una maggiore attenzione ed una tempestiva attivazione. Infatti, mentre le schede originali hanno ancora un buon mercato tra i collezionisti, gli altri artefatti stanno via via scomparendo, soprattutto quelli più “intimi” alle case produttrici dei giochi, dato che molte di esse sono fallite, quindi il loro materiale rischia di andare inesorabilmente perduto.

Quanto visto, è lo scopo che MAME si prefigge. Esistono anche altri progetti che nascono da iniziative volte alla conservazione di tali artefatti, tra cui i più importanti sono:

- International Arcade Museum® [2]: il più grande museo mondiale inerente alle arti, alle invenzioni e alla storia delle macchine per l'intrattenimento.
- KLOV® (Killer List of Videogames™) [3]: divisione di IAM che si occupa di videogiochi.
- VIDEOTOPIA™ [4]: un museo mobile che viaggia in tutto il mondo e mostra la storia dei primi media interattivi.
- TOSEC (The Old School Emulation Center) [5]: gruppo dedicato alla preservazione di giochi e programmi per home computer e console di gioco. Lo scopo principale del progetto è fornire un catalogo coerente per la nomenclatura di tutti i software esistenti per queste architetture. Il catalogo è esteso anche a riviste e manuali per alcuni dei sistemi preservati.
- MESS (Multi Emulator Super System) [6]: progetto che mira alla preservazione degli artefatti di console, home computer, pocket computer, PDA, calcolatrici e decoder TV. MESS poggia le proprie basi sul core di MAME quindi utilizza la strategia dell'emulazione per fare preservazione.

---

<sup>1</sup>Struttura in legno e metallo che contiene la scheda originale di un gioco, la plancia con i comandi e il monitor.

Ritornando a MAME, per dimostrare che il comportamento emulato corrisponde a quello originale, è necessario che i giochi possano essere eseguiti. L'aspetto puramente ludico non è tuttavia considerato lo scopo principale, infatti non viene ufficialmente perseguito. La possibilità di giocare con MAME viene considerata, citando Nicola Salmoria, l'autore originario del progetto, solamente un "piacevole effetto collaterale" dell'opera di documentazione. Sulla base di questo principio sono presenti in MAME anche giochi non funzionanti, oppure la cui emulazione, benché sufficientemente accurata, è lontana dal rendere il gioco giocabile.

L'obiettivo finale del progetto è di emulare correttamente tutti i videogiochi arcade prodotti nella storia. Vista la grande ambizione dell'obiettivo, probabilmente non verrà mai conseguito totalmente. Per questo motivo MAME è considerato *work in progress*: un progetto in continuo sviluppo, che si estenderà fino a quando ci saranno persone interessate a portarne avanti lo sviluppo.

Per capire quanto MAME abbia avuto successo, viene di seguito citato un passo della tesi di Nicola Salmoria [7]:

MAME è qualcosa di unico nel suo genere. Ci sono molti altri emulatori di videogiochi, ma nessuno di essi è paragonabile a MAME per la quantità di giochi supportati, per l'attenzione posta nella correttezza dell'emulazione, per la continuità e la durata dell'impegno profuso e per il numero di collaboratori, che sono centinaia sparsi in tutto il mondo.

Un punto fondamentale su cui è importante soffermarsi riguarda la fedeltà dell'emulazione. MAME focalizza gran parte dei suoi sforzi nell'emulare nel modo più fedele e accurato possibile le varie componenti hardware impiegate nei giochi arcade. Allo stato attuale, in cui la maggior parte dei giochi è stata aggiunta, tutto lo sviluppo ruota attorno alla fedeltà emulativa. Molto del codice presente in MAME viene spesso riscritto per renderlo più accurato, senza preoccuparsi se tali modifiche possa determinare un calo delle performance dei giochi quando vengono eseguiti. Si prospetta che anche nello sviluppo futuro del progetto questo obiettivo sarà considerato primario.

Bisogna fare attenzione a non confondere, come spesso accade, emulazione con simulazione. MAME, come già spiegato, è un emulatore. Un emulatore riproduce con estremo dettaglio il comportamento di un'altra macchina, ricreando completamente l'ambiente in cui questa macchina lavora. Invece per un simulatore è importante solo il risultato finale ottenuto. Un simulatore ambisce ad ottenere gli stessi risultati della macchina simulata quando vengono forniti gli stessi input, ma non considera come la macchina lavori internamente per produrre tale output. Una modellazione matematica dei due concetti appena esposti può essere trovata nella tesi di laurea di Nicola Salmoria [7].

## 3 Story

MAME nasce il 5 febbraio 1997, con il rilascio della versione 0.1, realizzato da Nicola Salmoria che ne aveva iniziato lo sviluppo nel dicembre dell'anno precedente. Salmoria è un brillante programmatore di Siena che lavorava quotidianamente in ambiente Commodore Amiga, per il quale ha realizzato anche un sistema di gestione delle icone divenuto così famoso da essere supportato ufficialmente. Passato all'architettura PC, non riusciva più a trovare gli stessi stimoli che gli offriva Amiga.

Un giorno, senza specifici motivi, scaricò da internet un emulatore preliminare del famoso gioco arcade *Pac-Man*. L'emulatore funzionava male, presentava infatti molte anomalie rispetto al gioco originale da sala: i colori era sbagliati, non aveva il sonoro, ecc... Assieme all'emulatore scaricò anche il codice sorgente, liberamente disponibile, e dalla volontà di farlo funzionare correttamente nacque l'idea, poi concretizzata, di realizzare un emulatore universale, il MAME.

Fin da subito, l'interesse che nutriva Salmoria verso il MAME era soprattutto rivolto al suo aspetto documentativo: era infatti molto interessato alle tecnologie adottate dai sistemi arcade e dal loro funzionamento, più che ai giochi. Il suo interesse lo portò ad adottare tecniche di *reverse engineering* per poter scrivere i moduli che riproducevano fedelmente il comportamento delle componenti hardware analizzate. Unendo il lavoro di codifica e documentazione del codice il risultato è una grande "enciclopedia" sulle tecnologie del mondo dei giochi arcade, che è appunto lo scopo principale di MAME, come visto nel paragrafo [2 a pagina 3](#).

I primi rilasci si susseguirono molto rapidamente e dopo poco tempo venne alla luce il primo porting per un sistema non MS-DOS: MacMAME per MAC, la cui prima versione è stata rilasciata il 18 febbraio 1997, in corrispondenza con MAME 0.4. Poco dopo, il 20 aprile 1997, nacque anche il porting per X11: X/MAME, in corrispondenza con la versione 0.18.

Nel periodo che va dal 26 aprile 1997 (0.19) al 10 agosto dello stesso anno (0.27), Mirko Buffoni sostituì Nicola Salmoria alla guida del progetto, ma solo perché Salmoria era in quel periodo occupato per il servizio militare. Salmoria, liberatosi dall'impegno, proseguì il coordinamento del progetto fino a maggio 2003.

Nel frattempo, il 18 luglio 1997, con la versione 0.26.1, nacque il MAME32, il "porting" per Windows del progetto: viene aggiunta a MAME una comoda GUI che permette la gestione delle opzioni dell'emulatore, la visualizzazione la lista di tutti i giochi supportati e la possibilità di eseguirli. Senza dubbio, il MAME32 è stata (e lo è tuttora) la versione derivata più famosa di MAME, nonostante ciò rimane un progetto esterno e quasi sicuramente non verrà mai incluso nel ramo ufficiale.

La frequenza di rilasci cominciò ad abbassarsi già dalla fine del 1997, così, per non far aspettare troppo gli utenti, dalla versione 0.33b1 del 3 maggio 1998 cominciarono ad

essere rilasciate anche delle versioni in stato di beta. Questa politica, morì con la versione 0.37b16 in quanto la frequenza di release era tornata ad assestarsi in circa una al mese, non aveva quindi più senso considerare i singoli rilasci come delle beta visto il tempo trascorso e le modifiche effettuate tra una release e la successiva. Si ripartì quindi, il 12 agosto 2001, con la versione 0.53.

L'introduzione dei primi giochi basati sull'architettura *Neo Geo*, nella versione 0.34b1, scatenò molte polemiche sulla legalità di MAME ed in questa disputa scesero in campo direttamente anche le case produttrici di questi giochi in quanto alcuni erano ancora presenti nelle sale, quindi vedevano MAME come un "concorrente" che faceva diminuire i guadagni. La causa principale di questa disputa è che furono sviluppati giochi su hardware Neo Geo per molti anni, senza mai aggiungere nuove protezioni. Essendo già stato sviluppato il modulo per questa piattaforma, era sufficiente avere le immagini dei nuovi giochi, usciti anche il giorno prima, per poterli giocare con MAME. La conseguenza della scesa in campo delle case produttrici fu la rimozione dei giochi più recenti e il congelamento dello sviluppo del modulo. Inoltre, ci fu maggiore attenzione alle regole, stilate già da tempo, che riguardavano l'aggiunta dei giochi, i quali dovevano avere un certo grado di "anzianità" e una scarsa presenza nelle sale.

Il 15 maggio 2003, data del rilascio della versione 0.68, è un giorno importante per tutto il mondo MAME: Nicola Salmoria abbandona il coordinamento del progetto ed al suo posto subentra David "Haze" Haywood. Una delle novità introdotte dal nuovo coordinatore è il ritorno dei rilasci di versioni beta, questa volta chiamate "U versions", dove la lettera *U* sta per update. La politica di queste versioni intermedie è più coerente con la definizione di beta release, infatti possono essere introdotte anche grandi modifiche che potenzialmente potrebbero rendere non funzionanti parti del progetto. Al contrario le versioni stabili sono solitamente dei punti di arrivo che chiudono un ciclo di sviluppo, iniziato con la prima "u release" dopo la precedente versione stabile, in cui vengono risolti tutti i problemi portati dall'inserimento di nuove caratteristiche. La gestione attuale dei rilasci è discussa in dettaglio nel paragrafo [7 a pagina 17](#).

Un'ulteriore novità introdotta dal nuovo coordinatore è la rimozione del concetto di *test drivers*, ora tutti i moduli vengono rilasciati anche nelle versioni binarie di MAME, dando quindi accesso pure a giochi la cui emulazione è in stato preliminare. La scelta è stata sicuramente lungimirante in quanto segue la filosofia del progetto. Questa aggiunta scatenò molte proteste da parte degli utenti che erano interessati esclusivamente a giocare con MAME in quanto da quel momento si trovarono nella lista tanti titoli a cui non potevano giocare.

Il 18 aprile 2005, MAME cambiò nuovamente coordinatore: Haywood decise di lasciare le redini del progetto in quanto aspramente e, secondo me, ingiustamente criticato per il suo modo di condurre il progetto. Al suo posto si insediò Aaron Giles che è tuttora project leader del progetto. Giles vanta di una importante carriera che lo vede inizialmente

dipendente di LucasArts, nota casa editrice di videogiochi, con il compito di effettuare il porting di alcuni giochi su piattaforma MAC. Venne successivamente a conoscenza del mondo MAME, dedicando molto tempo allo sviluppo di MacMAME. La sua passione per l'emulazione attirò l'attenzione di un programmatore di Connectix che gli offrì un posto di lavoro. Il prodotto di punta di Connectix era VirtualPC, un emulatore che permetteva di eseguire Windows su computer MAC. Attualmente, dopo l'acquisizione di Connectix da parte di Microsoft, Giles lavora ancora nel campo dell'emulazione come dipendente dell'azienda di Redmond.

Il 3 giugno 2006, in corrispondenza con la versione 0.106u1, nasce SDLMAME, port che sfrutta le librerie grafiche SDL. L'utilizzo di SDL permette lo sviluppo e l'utilizzo di MAME anche su sistemi Unix e Mac OS X. Fin dalla nascita, fu l'unica soluzione per gli utenti di questi sistemi in quanto i port originari, X/MAME e MacMAME, erano stati nel frattempo abbandonati.

Un riassunto grafico che mostra la storia di MAME rispetto al numero di giochi supportati è presente in figura 1 [nella pagina seguente](#). La figura mostra 2 superfici: quella di colore blu indica il numero di giochi totali, mentre quella rossa indica il numero di giochi unici. Con giochi "non unici" si intendono versioni alternative di uno stesso gioco, ad esempio aggiornamenti o versioni destinate alla distribuzione in continenti diversi.

Come ultima prova della grande evoluzione di MAME, viene riportato un estratto di un'intervista fatta a Mirko Buffoni [8] nel febbraio 1998, in forma di **D**omanda, **R**isposta e **C**osa è successo.

**D:** Credi che MAME arriverà a 1000 giochi? Se sì, qual è la tua stima? 1000 giochi emulati nel 2000? :)

**R:** Chi lo sa!? :)

**C:** MAME attualmente supporta 9012 giochi, di cui 4510 giochi unici. La prima versione uscita nel 2000, la 0.36b14, conteneva 1913 giochi.

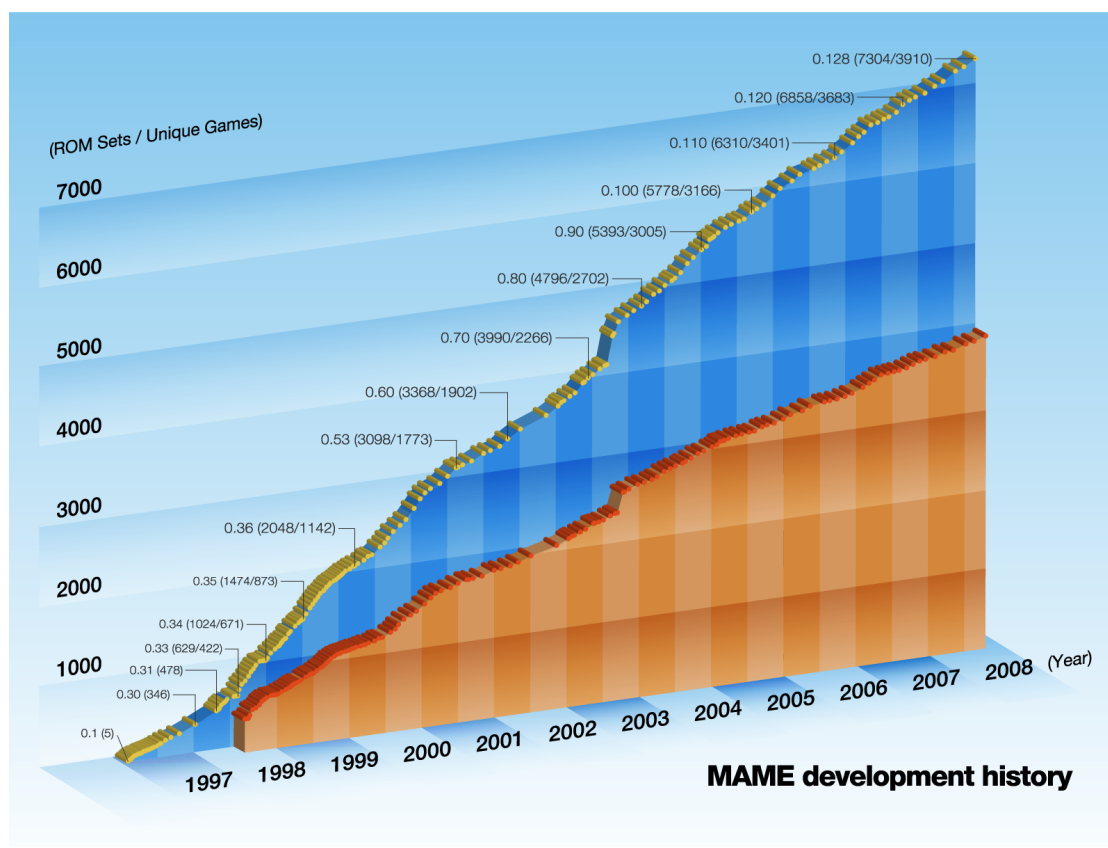
**D:** C'è qualche gioco che non è ancora stato emulato da MAME e che vorresti vedere?

**R:** Dragon's Lair and Space Ace :0. Va bè, teniamo i piedi per terra! [...]

**C:** Sono 2 giochi molto particolari simili ad un cartone animato interattivo che utilizzano speciali supporti ottici per memorizzare il video. All'epoca era un'utopia riuscire ad emularli, attualmente il modulo è presente e alcuni di essi sono funzionanti. Rimane un grosso ostacolo per l'emulazione fedele per questa tipologia di giochi ed è dovuto al supporto ottico (Laserdisc) che è analogico, come i vinili a 33 o 45 giri. Riuscire ad estrarre i dati da tale supporto in modo accurato è quasi impossibile oltre a richiedere attrezzature ad-hoc.

**D:** Pensi sia possibile emulare giochi come Tekken, Virtua Fighter o Killer Instinct a una velocità accettabile (non per forza in MAME)?





**Figura 1:** Evoluzione del numero di giochi supportati nella storia di MAME

Fonte: sito ufficiale [1]

**R:** [...] in genere non c'è documentazione disponibile [...]. Perciò credo che non vedremo nessuno di questi giochi emulato, almeno nel breve periodo.

**C:** I giochi citati sono per lo più emulati correttamente. La velocità di esecuzione invece può essere limitata su computer poco performanti in quanto non viene utilizzata l'accelerazione hardware fornita dalle schede video. Questo avviene per garantire massima fedeltà emulativa, non essendo predicibile la resa video se viene sfruttata l'accelerazione 3D.

## 4 Market

Il numero di potenziali utenti a cui è rivolto MAME è molto ampio. Si possono delineare alcune specifiche categorie di utenza, ciascuna interessata ad aspetti diversi del progetto.

## 4.1 Giocatori

Senza dubbio la maggior parte degli utenti vengono a conoscenza di MAME e lo utilizzano per la possibilità di giocare con i titoli emulati. La fascia di età interessata varia dai più giovani, dai 13 anni in su, fino ad arrivare a persone più adulte di anche 40 anni. L'approccio al progetto è quindi molto diversificato, proprio in base alla fascia d'età.

La quasi totalità dei giovani lo utilizzano solo per il gioco fine a se stesso, per passare dei momenti con qualche "gioco vecchio ma divertente". Solitamente non conoscono a fondo lo spirito insito in MAME e lo utilizzano passivamente solo per divertirsi. Questo modo di approcciarsi al progetto porta spesso a lunghe discussioni perché c'è chi rimane stupito del fatto che giochi così vestiti non funzionino o vadano molto lenti anche su PC moderni e molto performanti. Il loro ragionamento deriva da una scarsa conoscenza del funzionamento interno dell'emulatore e gli obiettivi che persegue.

Il pubblico più adulto, invece, approccia l'emulatore poiché ha vissuto in prima persona gli anni di massima espansione delle sale giochi e quindi dei giochi arcade. Questi utenti, probabilmente per una maggiore maturità, riescono meglio ad abbracciare la filosofia di MAME e incentivarne lo sviluppo, anche economicamente, affinché i titoli a loro più cari possano venir meglio emulati.

Quanto detto non vuole assolutamente sentenziare che ci siano giocatori "buoni" e "cattivi", ma solo dare una personale considerazione ad alto livello dall'esperienza vissuta dalle varie comunità.

Un problema importante rispetto alla possibilità di giocare è che i titoli emulati, tranne alcune eccezioni, sono ancora sottoposti a copyright da parte degli sviluppatori originali. MAME non vuole in alcun modo infrangere questi diritti e per questo non fornisce le immagini originali dei giochi, che devono essere possedute dall'utente. Inoltre, nel sorgente dell'emulatore non sono presenti porzioni del codice originale dei giochi.

Una ulteriore nicchia di utenti che hanno tratto grandi benefici da MAME sono coloro che desiderano costruirsi un cabinato per uso "domestico". Lo scopo di questi utenti è ricreare nel modo più fedele possibile le sensazioni del gioco in sala, anche a livello hardware, utilizzando i controlli ed i monitor dei videogiochi originali. Grazie all'utilizzo di MAME, questi utenti sono in grado di costruire una postazione di gioco che contiene molti titoli utilizzando un solo cabinato. Non ci sono alternative nel panorama, altre soluzioni richiederebbero di utilizzare emulatori distinti oppure utilizzare direttamente le schede originali dei giochi.

## 4.2 Gestori di sale giochi

Un'attenzione particolare al MAME è rivolta dai gestori di sale giochi e costruttori di cabinati da gioco. La possibilità di avere una postazione con tutti i giochi più famosi della

storia al solo costo dell'hardware è sicuramente molto appetibile e potrebbe portare facili guadagni. Proprio per evitare che qualcuno possa guadagnare sopra il MAME, il suo impiego commerciale è proibito dalla licenza, come si vedrà in dettaglio nel paragrafo 5 [nella pagina successiva](#). Il vincolo imposto ha come scopo principale quello di impedire l'utilizzo attraverso MAME di immagini di giochi recuperate illegalmente. Anche se MAME permettesse l'uso commerciale, ogni distributore di soluzioni basate su di esso, dovrebbe comunque chiedere il permesso a tutti i produttori dei giochi per avere il diritto di utilizzare le immagini originali. Visto che non è facile ed è molto costoso ottenere questi diritti, è molto probabile che se ne farebbe un uso non legale, procurando danni all'immagine di MAME che verrebbe associato ad un prodotto per delinquere. Purtroppo, nonostante l'espresso divieto, sono state trovate diverse apparecchiature che emulavano giochi tramite MAME sparse per il mondo, violando palesemente la licenza. Una di queste è stata a sua volta emulata in MAME<sup>2</sup> ☺.

### 4.3 Collezionisti

Un'ulteriore categoria di utenti che si avvicina a MAME sono i collezionisti, persone che per pura passione, sfiorando a volte la maniacalità, cercano di trovare, conservare e catalogare tutti gli artefatti che ruotano attorno ad un dato universo di interesse. Nel caso di MAME, questo universo comprende svariati oggetti, dalle schede originali dei giochi ai cabinati, alle immagini in-game, ecc... Grazie all'ambizione di voler diventare "l'enciclopedia" dei giochi arcade, MAME è un ottimo candidato per l'aggregazione di tutti questi appassionati. Un uso concreto delle informazioni presenti viene fatto da coloro che intendono riparare le schede originali in loro possesso, grazie all'accuratezza delle informazioni c'è chi riesce a riportare in vita schede che altrimenti andrebbero cestinate.

Molto spesso la collezione più ambita dagli utenti è quella delle immagini dei giochi, che è illegale. Lo stato di illegalità permane per la legislazione italiana anche se si possiede la scheda originale del gioco. Personalmente, non mi sento però di colpevolizzare chi fa questa raccolta se l'utilizzo rimane strettamente personale, senza scopo di lucro e senza coinvolgere direttamente gli sviluppatori del progetto.

### 4.4 Programmatori

Tra gli utenti di MAME c'è spazio anche per chi non è direttamente interessato ai videogiochi. MAME, dal punto di vista software, è supportato da un'architettura che si presenta molto complessa ma ordinata e molto efficiente; riesce infatti ad emulare migliaia di piattaforme senza richiedere strutture ad alto livello specializzate per ogni singola piattaforma. In sostanza, MAME può essere considerato un framework per l'emulazione.

Altre sfide che MAME offre sono ad esempio:

---

<sup>2</sup>Il "gioco" è stato denominato *39 in 1 MAME Bootleg*.

- Lo studio dei sofisticati sistemi di cifratura utilizzati per proteggere i giochi.
- La comprensione approfondita del funzionamento delle architetture emulate tramite tecniche di reverse engineering.
- L'emulazione delle CPU non realizzando interpreti ma tramite tecniche di compilazione just-in-time, per costruire dei "dynamic recompiler".

## 5 License

La licenza di MAME è cambiata più volte durante la sua storia. Inizialmente fu rilasciato sotto licenza GPLv2. Già dalla versione 0.27 del 10 agosto 1997 la licenza cambiò e ne venne creata una ad-hoc, allo scopo principale di regolare la distribuzione sotto pagamento di prodotti derivati. Infine, con la versione 0.96 del 3 maggio 2005, è stata introdotta una nuova licenza basata sullo schema BSD, modificata per apporre restrizioni sull'uso commerciale del prodotto. La natura non commerciale della licenza di MAME fa sì che non possa essere considerato un prodotto né "open source" secondo la definizione della Open Source Initiative né "free software" secondo la definizione della Free Software Foundation.

Volendo riassumere la licenza di MAME in poche parole, la migliore sintesi è:

MAME is free. Its source code is free. Selling either is not allowed.

Più in dettaglio, la licenza attuale prevede che la distribuzione e l'uso del codice di MAME o di un suo derivato sia consentito solo se le seguenti condizioni sono soddisfatte:

- La distribuzione non può avvenire a pagamento e non può essere utilizzato per prodotti o attività commerciali.
- I prodotti derivati distribuiti devono includere il codice sorgente completo del prodotto di base assieme ai sorgenti delle modifiche che sono state effettuate. Inoltre, deve essere distribuito anche il codice sorgente di tutte le librerie che sono state utilizzate per produrre il formato binario del prodotto derivato. È fatta eccezione per le librerie considerate di *sistema* (kernel, compilatori, ...), i cui sorgenti sono normalmente distribuiti presso altri canali. Normalmente questa clausola viene chiamata *system library exception*.
- I prodotti derivati devono includere le informazioni sul copyright, le restrizioni qui descritte e il seguente disclaimer nella documentazione e/o altri materiali forniti con la distribuzione.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 5.1 Lavori derivati

I prodotti derivati sono permessi ma devono anch'essi sottostare alla MAME license. In ogni caso, gli autori scoraggiano la creazione di derivati contenenti specifiche funzionalità che vanno contro la filosofia di MAME. Di fatto, se tali funzionalità vengono aggiunte, i lavori vengono considerati “build illegali” perché non sono conformi con la licenza.

In particolare, è richiesto di:

- Non aggiungere giochi che abbiamo meno di 3 anni o che siano ancora venduti dalle case che li producono. MAME non è inteso per essere una piattaforma che vuole competere con i giochi che sono ancora in commercio.
- Non fornire un sistema per generare liste di giochi che identificano specificatamente i giochi di cui l'utente non possiede l'immagine. Se venisse fatto, questo incoraggerebbe gli utenti a cercare strade, anche illegali, per il solo gusto di avere la “collezione completa”.
- Non rimuovere le schermate informative che appaiono all'avvio dei giochi per avvertire che un certo gioco non funziona. Questo è necessario per non inondare gli sviluppatori di email da parte di utenti che chiedono perché un gioco, marcato come non funzionante, non esegua correttamente.

Essendo che il nome MAME è un marchio registrato, è necessario che i derivati rispettino le regole sull'uso del marchio se desiderano includere “MAME” come parte del loro nome. In generale, questo significa che bisogna prima chiedere il permesso al core team, il quale deciderà se accordarlo nei termini di rispetto della licenza e di uso del marchio.

Infine, il numero di versione dei lavori derivati deve riflettere il numero di versione del rilascio di MAME dal quale derivano.

## 5.2 Trademark

MAME® è un marchio registrato da Nicola Salmoria. Il nome “MAME” ed il suo logo non possono quindi essere utilizzati senza il permesso esplicito da parte del detentore del marchio.

Suona abbastanza strano che un prodotto la cui licenza ne vieta l’utilizzo commerciale in ogni forma abbia la necessità di registrare il proprio marchio. La richiesta di registrazione del marchio è infatti avvenuta come risposta ad un tentativo di registrare il nome MAME da parte di una persona esterna al team di sviluppo, David R. Foley.

Foley, fondatore di UltraCade [9], l’11 gennaio 2005 fece richiesta presso l’USPTO (United States Patent and Trademark Office) di registrare il nome MAME [10]. Valutando le dichiarazioni rilasciate a fronte di questa decisione, le intenzioni di Foley sembravano buone: voleva proteggere l’utilizzo del nome MAME da parte dei costruttori di cabinati che permettevano di far giocare a molteplici titoli. Il nome MAME veniva infatti spesso utilizzato per sollecitare maggiore attenzione su questi cabinati, anche se non utilizzavano effettivamente MAME per eseguire i giochi. Inoltre, Foley era disposto a cedere tutti i diritti acquisiti agli sviluppatori.

La reazione della comunità alla notizia fu di totale sdegno nei confronti di Foley, accusato di voler acquisire i diritti d’uso del nome MAME per utilizzarlo nei cabinati multi gioco costruiti dalla sua azienda, vietarne l’uso ad altre società ed infine far cambiare nome all’emulatore originale. I MAME Dev dovettero quindi far fronte a questa iniziativa e cercarono fin da subito di contattare Foley per dissuaderlo dalla sua idea. Visto che era imminente la registrazione del marchio, trovarono un accordo in modo che decadesse la richiesta originale e ne fu avanzata una nuova in modo che fosse Salmoria il detentore del marchio, sicuramente la persona più adatta. La registrazione venne completata con successo l’1 agosto 2006 [11].

Come conseguenza della registrazione del marchio MAME, ci fu una maggiore attenzione da parte del team di sviluppo verso l’uso del nome. Ad esempio, ha permesso di far chiudere molte inserzioni in siti di compravendita online che offrivano illegalmente schede basate su MAME o DVD contenenti migliaia ROM.

Vennero inoltre create delle regole che dovevano essere rispettate da chi volesse utilizzare il nome MAME, il quale doveva esplicitamente richiederne il permesso ed attenderne l’accettazione da parte del team. In particolare, viene richiesto di non avere alcun collegamento, in qualsiasi forma (link, download, ...) con ROM illegali associando a queste il nome MAME [12].

Rimane comunque possibile parlare genericamente di MAME senza la necessità di apporre affianco al suo nome il simbolo di trademark ™.

## 6 Business Model

La natura strettamente non commerciale di MAME e la volontà di perseguire questa filosofia fa sì che non presenti un business model definito. Sono gli stessi sviluppatori ad accettare tale situazione e non sembra ci siano stimoli per cambiare sotto questo aspetto.

Durante l'analisi del progetto non sono state trovate iniziative presenti o passate che avessero come scopo finanziare economicamente gli sviluppatori. In linea generale, gli sviluppatori di MAME lavorano per passione e non dimostrano interesse a trovare sistemi per recuperare denaro. Per alcuni di essi questa è una condizione necessaria in quanto il loro contratto lavorativo gli vieta di contribuire a progetti con sfruttamento commerciale.

Si può definire la scelta di MAME un "*anti business model*", alla cui base sta un reciproco "accordo" tra sviluppatori ed utenti il quale sancisce che nessuno possa lucrare sopra il lavoro svolto, cioè nessuno possa ricavare denaro utilizzando un prodotto frutto di sola passione e pensato come un bene della cultura collettiva. La scelta è sicuramente in linea con la filosofia generale del progetto e non penso che cambierà in futuro, a meno che il progetto non passi in mano a qualcuno con interessi commerciali, il che sarebbe decisamente mal visto dalla comunità, quindi una ipotesi abbastanza remota.

La scelta di MAME porta sicuramente anche degli svantaggi. Attualmente, l'attrazione per il progetto da parte di nuovi sviluppatori non è molto alta, dato che le architetture più facili sono già state emulate ed il lavoro da svolgere ora è più complesso. Uno sviluppatore che volesse iniziare a contribuire al progetto potrebbe rinunciare a priori, viste le difficoltà per portare a termine il suo compito ed il mancato supporto economico.

Il mondo dei giochi arcade sta vivendo una grave crisi, è infatti da molti anni in declino e l'unico mercato che permane è quello giapponese. L'entrata di MAME nel mercato arcade sarebbe accolta con grande entusiasmo da parte di produttori ed utilizzatori, con un giro d'affari decisamente cospicuo ma senza alcun ritorno per gli sviluppatori. Non si prospettano guadagni in quanto l'utilizzo che ne verrebbe fatto sarebbe puramente ludico e svincolato dalla filosofia e dall'identità stessa dell'emulatore. Inoltre, il numero di titoli emulati è talmente ampio che difficilmente verrebbero finanziati nuovi sviluppi per l'emulatore.

Valutando la situazione del mercato in cui MAME andrebbe ad entrare, trovo quindi corretta la scelta di non volerne far parte.

La scelta effettuata da MAME non è l'unica possibile, ci sono infatti altri modelli di business che potrebbero adattarsi al progetto. Si possono immaginare almeno 3 opzioni: cambiare la licenza del prodotto, collaborare con la sezione *dumping* e creare una fondazione.

### Cambio di licenza

La soluzione più facile per la creazione di un "mercato" attorno a MAME, in modo che il progetto possa avere dei guadagni, è quella di cambiare la licenza attuale. Volendo

mantenere vincoli abbastanza forti sui derivati del progetto, la scelta migliore è adottare uno schema LGPL. In questo modo la redistribuzione di MAME sarebbe maggiormente libera e potrebbero nascere prodotti commerciali che utilizzano MAME come nucleo interno. A supporto di tale tesi, si ricorda che MAME non è solo dedicato a far funzionare giochi, ma il suo sviluppo ha portato alla realizzazione di un vero e proprio framework per l'emulazione, che potrebbe esser adottato per l'emulazione di altri sistemi come ad esempio architetture non PC utilizzate in ambito industriale. Se questi ipotetici prodotti basati su MAME avessero successo, richiederebbero un costante miglioramento del core, con la conseguente possibilità che vengano stanziati dei finanziamenti.

MESS [6] è una prova che MAME può essere utilizzato come core per altri progetti emulativi. Come visto nel paragrafo 2 a pagina 3, questo progetto aggiunge nuovi moduli permettendo l'emulazione di sistemi di gioco non arcade come le console. Il rapporto tra MAME e MESS è in realtà più stretto, dimostrando come un approccio open source, nel senso lasco del termine, possa giovare sia al nucleo come al derivato. Infatti, alcuni moduli scritti originariamente per MESS sono entrati a far parte nel core di MAME anche se questi non erano ancora necessari per emulare giochi arcade, ad esempio il modulo per la CPU del Game Boy<sup>3</sup>. Questa scelta si è dimostrata lungimirante poiché in molte occasioni MAME ha successivamente sfruttato i moduli che erano stati introdotti per MESS.

I rischi presentati ad inizio paragrafo, cioè che MAME entri nel circolo vizioso del commercio arcade con il solo scopo di spillare soldi alla gente, sono molto seri ed aspramente combattuti dal team di sviluppo.

In conclusione, il cambio di licenza non sembra una soluzione di facile adozione.

### **Avvicinare i fondi del dumping**

Il settore del dumping è costituito da persone molto esperte in grado di estrapolare i dati contenuti nelle ROM e nei dispositivi di storage utilizzati dalle schede originali dei giochi. Questo compito è molto complesso e richiede conoscenze e strumentazione specifiche ma allo stesso tempo è di fondamentale importanza per MAME, senza le immagini dei giochi non si potrebbero vedere i risultati dell'emulazione, quindi nemmeno testarne l'affidabilità.

Contrariamente a MAME, il settore del dumping dispone di molti finanziamenti da parte della comunità che è molto attiva con frequenti donazioni. Tutti i proventi delle donazioni sono utilizzati per l'acquisto di nuove schede non ancora emulate in MAME o che necessitano di verifica. È abbastanza paradossale il fatto che ci si mobiliti economicamente per l'acquisto delle schede, ma poi non si supporti chi effettivamente le “riporta in vita” tramite l'emulazione, cioè gli sviluppatori di MAME. Un'iniziativa in tal senso è nata perfino dal project leader attuale: nel gennaio 2008 [13] ha stanziato fondi per il gruppo “ufficiale” di dumping, il Dumping Union Team, donando una data somma per ogni bug che fosse stato risolto in quel mese. L'iniziativa fu accolta positivamente dagli sviluppatori

---

<sup>3</sup>Console portatile di Nintendo commercializzata nei primi anni 90.



che in quel mese lavorarono molto per risolvere diversi bug. La cifra complessiva donata fu di 1080 dollari.

Una soluzione adottabile per venire incontro anche agli sviluppatori potrebbe essere quella di trovare maggiori accordi con i dumper promuovendo collette in una cui parte dei proventi sia destinata anche a chi le schede riesce ad emularle.

### **Fondazione**

Un'interessante ed ambiziosa prospettiva per la gestione del business attorno a MAME è la realizzazione di una fondazione. La fondazione si focalizzerebbe sulla preservazione dei giochi arcade e/o potrebbe espandersi anche ad altre tipologie di macchine per l'intrattenimento. Il numero di persone che ruotano attorno a MAME è elevato ed in generale l'interesse per questi prodotti richiama persone da tutto il mondo quindi numericamente potrebbe essere fattibile gestire una comunità organizzata come una fondazione.

Il limite sulla fattibilità della soluzione è dovuto alla natura hobbistica dell'interesse verso il mondo arcade. Non ci sono più aziende e persone disposte a basare il loro business in questo mondo, morto all'inizio degli'anni 90, quindi sarebbe difficile trovare qualcuno disposto ad organizzare e gestire la struttura che si andrebbe a creare, soprattutto a livello economico.

## **7 Development Process**

Il gruppo degli sviluppatori principali di MAME è composto da circa 25 persone, la cui attività sul progetto non è sempre costante ma dettata soprattutto dagli impegni della vita reale. Questi sviluppatori formano il gruppo dei *MAME Dev* e sono gli unici che hanno accesso completo a tutti gli strumenti per la gestione dello sviluppo. Non esistono ulteriori gruppi, i rimanenti sviluppatori sono considerati semplici *contributors*.

Il gruppo dei MAME Dev non è molto aperto al pubblico, per questo non sono disponibili molte informazioni sulle metodologie di lavoro interne al progetto. Essendo un gruppo molto ristretto, il principio che regola l'organizzazione del lavoro è la *fiducia*: quando uno sviluppatore si prende in carico un lavoro questo viene portato a termine senza grandi controlli da parte degli'altri Dev. Non viene quindi usata, almeno per quanto noto, alcuna metodologia standard per la gestione dello sviluppo. Tra le poche informazioni disponibili, sono note le *code conventions* che si cercano di adottare internamente al progetto, reperibili nell'area wiki ufficiale [14].

Le modalità di sviluppo impiegate riescono comunque a garantire rilasci continui con una buona dose di modifiche e miglioramenti. È infatti poco probabile che più sviluppatori possano riscontrare conflitti nelle loro modifiche visto che allo stato attuale la quantità codice è molto elevata rispetto al numero degli sviluppatori. Inoltre, vista la complessità dei sistemi che vengono emulati, solo i pochi sviluppatori che decidono di lavorarci sono

effettivamente in grado di poter apporre modifiche significative. Per quanto riguarda invece l'architettura generale del software, quasi tutte le modifiche sono fatte dal project leader attuale, Aaron Giles, quindi anche in questo caso non ci sono conflitti nello sviluppo.

Ultimamente, sta emergendo una metodologia di sviluppo associabile alla tecnica di *pair programming*: lo sviluppo di una parte del progetto avviene in collaborazione tra 2 sviluppatori. Questo accade soprattutto tra gli sviluppatori più attivi, che hanno interesse a completare in breve tempo un certo compito.

La politica dei rilasci di MAME segue uno schema preciso che si è assestato da un paio di anni. Ogni versione stabile è numerata secondo lo schema 0.xxx in cui xxx corrisponde ad un numero progressivo. Non è mai stata rilasciata una versione 1.x, per volontà di indicare che il progetto è sempre in un continuo stato di *work in progress* e non ha senso rilasciare una versione “finale”. Le versioni stabili vengono rilasciate normalmente con una cadenza mensile o bimensile.

Tra una versione stabile e la successiva vengono rilasciate diverse versioni intermedie, chiamate *update versions* o *U versions*. La nomenclatura di queste versioni segue lo schema 0.xxxUyy in cui xxx identifica la versione stabile a cui un update fa riferimento e yy è un numero progressivo. Il numero di versioni intermedie rilasciate cambia ad ogni release, solitamente variano da 4 a 8, con una cadenza settimanale o bisettimanale.

Nelle prime versioni intermedie viene concessa l'introduzione di modifiche molto consistenti, che possono anche provocare il malfunzionamento di diverse componenti del sistema. Nei successivi update diminuisce gradualmente la complessità delle modifiche introdotte e vengono risolte molte anomalie sorte dalle modifiche delle prime versioni. Questo schema prosegue finché non si raggiunge una nuova stabilità dell'intero sistema, a questo punto viene rilasciata una nuova versione stabile con affinamenti minimali rispetto l'ultima versione intermedia.

La gestione dei contributi esterni è molto rigida e strettamente regolamentata. Affinché i contributi vengano accettati, è necessario seguire alcune linee guida. In particolare, vengono limitati gli ambiti su cui gli esterni possono lavorare e descrivono le metodologie per “interfacciarsi” con gli sviluppatori ufficiali. Le linee guida impongono le seguenti restrizioni:

- Non effettuare modifiche o migliorie che puntino esclusivamente a rendere il titolo più giocabile, aumentandone la velocità di emulazione o rendendolo più bello da vedere. Migliori risultati sull'aspetto di un gioco devono essere solamente frutto di una più corretta emulazione globale di un sistema e non ottenuti per il solo gusto di poter giocare meglio ad un certo gioco.
- Non effettuare modifiche ai dati estratti dalle ROM o alla loro lettura e nemmeno “hack” ai giochi. Tali modifiche sono accettate solamente se permettono una migliore comprensione del funzionamento di un'architettura non ancora pienamente emulata.

In ogni caso lo sforzo deve essere sempre nella direzione della massima fedeltà emulativa.

- Non inviare modifiche rispetto la nomenclatura dei ROM set (l'insieme delle ROM che compongono un gioco) presenti nel database di MAME. Questi nomi sono per lo più arbitrari, contrariamente alle singole ROM che di solito riportano nel nome la dicitura marchiata sul chip da cui i dati sono stati estratti.
- Se si riscontrano problemi nei nomi dei giochi, cioè se non corrispondono ai titoli originali, o nella loro categorizzazione è consigliato inviare una email agli sviluppatori piuttosto che effettuare modifiche al codice sorgente.
- Le modifiche volte solamente a migliorare il lato estetico del codice devono essere lasciate agli sviluppatori ufficiali. Le modifiche devono quindi portare effettivi cambiamenti alle funzionalità del prodotto.

Ulteriori indicazioni definiscono delle regole da seguire prima di inviare le modifiche effettuate, ai fini di garantire una modalità comune di invio per poter gestire in modo maggiormente automatizzato i contributi esterni degli utenti.

- Prima di effettuare l'invio, bisogna accertarsi che il codice sorgente con le modifiche apportate compili correttamente sia con la modalità *DEBUG* attiva sia non attiva. Devono essere utilizzati i tool ufficiali per la compilazione e devono essere corretti anche eventuali *warnings* prodotti nella fase di compilazione.
- Tutte le modifiche devono essere inviate in formato *diff* basandosi sull'ultima versione intermedia disponibile.
- Una volta ottenuto il file diff, deve essere compresso in formato ZIP e inviato all'indirizzo [submit@mamedev.org](mailto:submit@mamedev.org).

Per modifiche di un certo rilievo viene fornito un feedback all'autore, mentre quelle minori sono spesso accettate o rifiutate senza fornirlo. Per capire se sono state accettate bisogna attendere un paio di versioni intermedie e vedere se l'aggiunta è segnalata nelle note di rilascio. Se non si dovessero avere notizie delle proprie modifiche dopo questo lasso di tempo è permesso richiederne lo stato, sempre tramite email.

Per uno sviluppatore occasionale, che vuole dare un contributo al progetto, è spesso molto difficile trovare un compito da svolgere e inizializzare il suo lavoro perché la curva di apprendimento per programmare MAME è molto ripida. Sono infatti richieste conoscenze approfondite del linguaggio C, dell'hardware dei giochi arcade, dell'architettura dei processori, del funzionamento di sistemi audio/video e infine di tecniche di reverse engineering. I MAME Dev, consci di queste problematiche, suggeriscono di iniziare con modifiche che non richiedano una conoscenza completa di MAME, alcuni esempi sono:

- Aggiungere il supporto ai *savestate*, che permettono di salvare e caricare automaticamente lo stato di esecuzione di un gioco quando viene terminato e nuovamente eseguito. La gestione dei *savestate* è piuttosto semplice, inoltre ci sono già molti giochi che li supportano. L'estensione del supporto ai *savestate* è un buon inizio per chi vuole contribuire e al contempo apprendere la struttura di MAME.
- Capire il funzionamento dei DIP switch<sup>4</sup> ancora ignoti. Lo studio consiste nell'impostare una certa configurazione e poi scoprire quali aspetti del gioco sono stati modificati (difficoltà, lingua, gettoni necessari, ...) in modo da capire la funzione di un certo switch.
- Cercare nell'issue tracker dei bug da risolvere. Spesso sono presenti bug abbastanza banali che non sono ancora stati risolti perché nessuno ha voglia di prendersene carico.

In prospettiva futura è molto probabile che le metodologie di gestione del processo di sviluppo rimangano inalterate.

Un importante cambiamento in corso di esecuzione è il passaggio all'utilizzo del linguaggio di programmazione C++. Questa fase è iniziata nel gennaio 2010 e molto gradualmente vengono introdotte funzionalità al core che sfruttano caratteristiche specifiche di C++. Anche se è stato espresso esplicitamente che non c'è interesse a trasformare l'intero progetto in C++, viene in ogni modo garantita la possibilità di utilizzare un compilatore per tale linguaggio, seppur la maggior parte del codice rimanga tuttavia in linguaggio C. A questo scopo sono stati di recente introdotti nuovi strumenti di sviluppo ufficiali basati sulla versione 4.4.3 di MinGW, il porting per Windows del celebre GCC.

## 8 Community

L'ecosistema che si è creato attorno a MAME è molto ampio e richiama persone in tutto il mondo. Questo è sicuramente dovuto all'ambito in cui MAME opera, i giochi arcade, culla di grandi passioni per molte persone. Tuttavia, tale diffusione è anche un chiaro risultato del successo di MAME e della sua filosofia, abbracciata e portata avanti dagli utenti.

Il sito ufficiale [1], contrariamente a molti progetti opensource, non fornisce modalità di aggregazione per la community di MAME. La forma principale di espressione sono quindi i forum di discussione, ne esistono svariati in giro per la rete, alcuni di essi trattano esclusivamente di MAME mentre altri trattano di giochi arcade o emulatori più in generale. Molto spesso i forum di discussione fanno parte di più ampi portali che hanno lo scopo di raccogliere svariate risorse su MAME, oltre a presentare le news dell'intero panorama. Il

---

<sup>4</sup>Componenti elettroniche che aprono o chiudono un circuito, usate nei giochi arcade per gestire alcune configurazioni.

più importante portale non ufficiale è *MAME World* [15], indicato come fonte principale dagli stessi MAME Dev. MAME World offre anche un servizio di web hosting in cui sono ospitati diversi siti collegati al mondo MAME come le pagine di progetti correlati e quelle di alcuni sviluppatori e collaboratori.

I principali prodotti della comunità MAME sono le build derivate, che verranno presentate nel paragrafo 8.1 a pagina 23.

Attorno a MAME sono stati creati anche altri numerosi progetti con l'obiettivo di fornire ulteriore documentazione o aggiungere nuove caratteristiche. Di seguito verranno elencati i progetti più importanti che sono opera della comunità ma non direttamente degli sviluppatori.

- MAME Info Dat [16]: una serie di documenti che raccolgono svariate informazioni sui giochi emulati da MAME. Le informazioni presenti, per ogni gioco, sono:
  - La versione in cui è stato introdotto.
  - Le modifiche apportate nei vari rilasci.
  - Lavori futuri pianificati (TODO).
  - Bug ancora aperti.
  - Il numero di livelli.
  - La tipologia di gioco e altri giochi simili.
  - Altri emulatori che lo supportano.
  - La dimensione del ROM set e il numero di ROM in esso contenute.
  - ... e molto altro ...

Sono inoltre contenute tante informazioni per ogni architettura emulata in MAME e la storia che ne ha portato allo stato attuale. Senza dilungarsi ulteriormente, si può concludere che MAME Info rappresenta una miniera inesauribile ed inestimabile di informazioni.

- MAME History [17]: un documento che contiene per ogni gioco le informazioni relative alla sua storia, alla trama, ai trucchi, ecc... Le informazioni sono estrapolate da un database più ampio che contiene le stesse informazioni per quasi tutte le macchine arcade esistenti, attualmente sono descritte oltre 38 000 macchine.
- MAWS [18]: un database che estrapola le informazioni dai rilasci di MAME e da altri documentazioni correlati, permettendone la navigazione attraverso una comoda interfaccia web. È molto utile per ottenere informazioni sui giochi supportati in modo semplice e veloce, attraverso il sofisticato sistema di query implementato.
- Cat32 [19]: una categorizzazione di tutti i giochi presenti in MAME secondo diversi criteri.

- HighScore [20]: aggiunge il supporto per il salvataggio dei punteggi raggiunti dagli utenti. Non essendo più supportato dalla versione ufficiale di MAME devono essere usate delle versioni derivate che lo supportano.
- Cheat [21]: aggiunge il supporto per i trucchi. Periodicamente questa estensione viene criticata perché considerata non conforme con l'obiettivo della massima fedeltà emulativa. Il supporto ai trucchi è mantenuto in vita perché sono necessari per gli sviluppatori per poter avanzare nel gioco facilmente. Ad esempio, se un gioco presenta un bug al livello 100, uno sviluppatore deve poterci arrivare senza giocare fino a quel punto.
- clrmamepro [22]: è IL rom manager per eccellenza, ossia un tool che gestisce tutte le ROM a disposizione di un utente, aiutandolo a mantenerle sempre allineate con le modifiche introdotte da MAME ad ogni nuovo rilascio.

Una nutrita comunità che ruota attorno a MAME è costituita da utenti italiani. Il grande numero di questi utenti è probabilmente dovuto alle origini italiane del progetto ed alla grande presenza di sale giochi nel nostro paese, soprattutto nelle località balneari. Il contributo portato dalla comunità italiana è molto importante e riconosciuto dagli stessi sviluppatori, di cui fanno parte alcuni connazionali. Molte schede di giochi sono infatti state acquistate grazie al contributo economico di utenti italiani, di cui alcuni si sono anche presi il gravoso compito di estrapolare i dati contenuti nelle ROM, lavoro che fino a quel momento era per lo più in mano ad un unica persona.

Alcune tra le risorse più importanti della comunità italiana presenti nella rete sono qui di seguito elencate:

- MAME Italian Forum [23]: il forum di discussione ufficiale della comunità italiana di utilizzatori di MAME. Il forum riesce a conciliare ottimamente il supporto ai nuovi utenti con discussioni prettamente tecniche. Ha un ottima reputazione a livello globale, tanto da esser frequentato anche da diversi MAME Dev.
- MameChannel [24]: un portale che contiene molte informazioni sull'universo MAME e fornisce i collegamenti per molti progetti della comunità.
- EMULAB [25]: simile a MAME Channel ma parla più in generale di emulatori. Offre inoltre un servizio di web hosting per le pagine di alcuni MAME Dev e di alcuni progetti correlati.
- Progetto EMMA [26]: clone di MAWS [18] completamente in italiano. Fornisce anche una serie di altri documenti tradotti.

## 8.1 Le “build” di MAME

MAME supporta ufficialmente un’unica versione, distribuita presso il sito ufficiale [1]. Fino a poco tempo fa questa versione consisteva solo nell’interfaccia command-line per Windows, storicamente disponibile invece per MS-DOS. Con la versione 0.136u1 è stato introdotto nel ramo ufficiale di MAME anche il port che sfrutta le librerie SDL, permettendo quindi di compilare nativamente MAME anche su sistemi non Windows, come Linux e Mac OS X. Il supporto ufficiale, fornito per i binari generati dai sorgenti ufficiali, è quindi stato esteso anche a questi sistemi.

Esistono 4 versioni binarie ufficiali, tutte compilate per Windows:

- mame0xxx\_b: versione “liscia”.
- mame0xxx\_b\_i686: ottimizzata per processori i686.
- mame0xxx\_b\_64bit: per processori a 64 bit.
- mame0xxx\_b\_debug: versione rivolta agli sviluppatori, compilata con supporto esteso alle funzionalità di debug.

Molto presto sono nate molte altre versioni derivate di MAME, non ufficialmente supportate. Lo scopo principale di queste nuove versioni è quello di fornire un’interfaccia più user-friendly e con caratteristiche aggiuntive. Le “build” di MAME ne hanno sicuramente aiutato la diffusione verso un pubblico più ampio, in quanto l’uso della versione command-line non è molto confortevole per un utente che desidera principalmente giocare con l’emulatore. Tra queste versioni, MameUI (originariamente Mame32) [27], è stata la prima con interfaccia grafica per sistemi Windows. Attualmente è senza dubbio la versione più utilizzata e che ha ottenuto maggior consenso da parte degli utenti.

Nel portale MameChannel, nell’apposita sezione [24], è possibile trovare una lista dettagliata di molte versioni di MAME, ciascuna accompagnata con una descrizione che ne descrive le peculiarità.

Esistono anche delle versioni derivate non legali, in quanto non rispettano i termini della licenza di MAME, descritti in dettaglio nella [paragrafo 5.1 a pagina 13](#). Non è da nascondere che anche queste versioni hanno comunque portato nuovi utenti verso di MAME, soprattutto MAME Plus! [28] che fin dai suoi albori, grazie all’inclusione di giochi recenti, ha destato l’attenzione di molti utenti. Da qualche anno questa versione ha cambiato atteggiamento e filosofia di fronte alla comunità diventando una versione legale. Inoltre, dal settembre 2009, MAME Plus ha scisso l’interfaccia grafica dal suo core, creando una GUI scritta completamente con il toolkit grafico QT. Vista la natura cross-platform di QT, questa interfaccia è stata adottata da molti utenti che utilizzano sistemi non Windows, infatti le alternative fino a quel momento disponibili avevano molte limitazioni rispetto la GUI per Windows.

## 9 Development and Information Management Tools

Gli strumenti di sviluppo utilizzati dal core team e dalla comunità di MAME sono molteplici, alcuni liberamente accessibili altri aperti esclusivamente agli sviluppatori ufficiali. Nel seguito del paragrafo verranno descritti in dettaglio tutti gli strumenti adottati indicando le informazioni che veicolano e come la comunità può interagire con essi.

Come visto, la maggior parte dei processi di sviluppo sono presi in carico dai MAME Dev che lavorano in un ambiente non esposto alla comunità. Per questo, è probabile che esistano ulteriori strumenti oltre a quelli elencati di cui però non si hanno sufficienti informazioni.

### 9.1 Sito web

Il sito web ufficiale [1] si presenta con un aspetto molto gradevole e contiene gran parte delle informazioni sugli sviluppi di MAME. La gestione del sito è affidata al project leader del progetto. In Home page sono presenti i post relativi ai rilasci di MAME con eventuali informazioni di rilievo come note su eventuali regressioni nell'emulazione di alcune piattaforme. È disponibile anche un feed RSS per tenere traccia delle news sui rilasci.

Nella sezione contenente le informazioni generali sul progetto è presente un'interessante pagina relativa alla storia di MAME, in cui sono presentati anche i milestone che nei 13 anni di vita del progetto si sono via via susseguiti. Sono inoltre presenti le pagine che presentano gli aspetti legali (licenza, trademark), forniscono i collegamenti ad alcune risorse collegate e permettono di contattare i MAME Dev.

La gestione delle comunicazioni provenienti dall'esterno è molto curata. Viene richiesto all'utente di selezionare il tipo di richiesta che vuole effettuare e in base alla scelta vengono fornite delle istruzioni su come realizzarla. È molto probabile che le richieste vengano poi smistate in modo differente in base alla tipologia, facendo in modo che arrivino solo agli sviluppatori preposti a fornire un particolare tipo di supporto. Personalmente ho sfruttato questa funzionalità per richiedere il permesso di utilizzare il logo ufficiale di MAME da inserire in questa relazione, essendo quest'ultimo protetto da trademark e quindi necessario l'esplicito consenso per usarlo. Mi sono stati proposti 3 destinatari: il project leader, il detentore del trademark ed il gruppo preposto alle questioni legali. Purtroppo, allo stato attuale, non mi è stata data alcuna risposta.

Nella sezione downloads è possibile scaricare i sorgenti dell'emulatore, gli aggiornamenti rispetto all'ultima versione stabile, le vecchie versioni stabili e le immagini di alcuni giochi di cui è stata concessa la distribuzione da parte dei rispettivi produttori. Infine, sono presenti le pagine con le FAQ, la wiki, un viewer del codice sorgente e gli strumenti di sviluppo.



Oltre al sito ufficiale, quasi tutti i MAME Dev gestiscono un proprio sito privato in cui forniscono notizie riguardo i lavori che stanno eseguendo sul progetto. Queste pagine vengono chiamate *Dev WIP Page*: le pagine dei “lavori in corso”.

Nel periodo che va dal febbraio 1999 al luglio 2004, Santeri Saarimaa ha gestito una pagina che conteneva tutte le novità, aggiornate giornalmente, sullo sviluppo del progetto. Il frutto del suo lavoro, reperibile dal sito ufficiale, rappresenta ora un’importante archivio storico, una sorta di “museo” della storia di MAME.

## 9.2 Mailing List

La mailing list ufficiale di MAME non è disponibile pubblicamente ma accessibile solo ai MAME Dev. Questa è collegata allo strumento di versioning per ricevere le notifiche dei commit da parte di altri sviluppatori. L’archivio della mailing list viene salvato e reso accessibile tramite il sito ufficiale, ma anch’esso non è reso pubblico.

Esiste inoltre un newsgroup non ufficiale, `alt.games.mame`, che tuttavia giace in uno stato di completo abbandono.

## 9.3 Versioning

Come strumento di versioning viene utilizzato *Subversion (SVN)*. Prima della sua adozione, veniva utilizzato *Concurrent Versions System (CVS)* e prima ancora, agl’albori del progetto, veniva gestito tutto a mano.

Il repository è accessibile solo dai MAME Dev, sia in lettura sia in scrittura. Se un contribuente esterno riesce ad inviare regolarmente modifiche di buona qualità, dopo qualche mese gli viene permesso l’accesso. Seppur questa scelta possa limitare l’avvicinamento di nuovi sviluppatori, la frequenza di rilasci settimanali permette di effettuare modifiche sporadiche in quanto permettono di rimanere sempre ottimamente aggiornati sullo stato del progetto.

Alcuni motivi per cui non sono pubblicamente accessibili il server SVN e la mailing list sono qui elencati:

- A volte si discute di codice non ancora pubblicabile rispetto la regola di non includere giochi troppo recenti e ancora attivi nel mercato.
- La discussione esce qualche volta dal mondo MAME, ad esempio si parla di dumping e questioni legali.
- Capita che venga aggiunto codice che successivamente si preferisce rimuovere, ad esempio perché si vuole trovare un’implementazione migliore o perché vengono introdotte modifiche con cui altri sviluppatori non concordano.

## 9.4 Bug Tracking

Gli strumenti per il tracciamento delle anomalie riscontrate sono gestiti in simbiosi con la comunità, diversamente dagli altri strumenti fin qui analizzati. Questa scelta è dovuta essenzialmente al fatto che gli sviluppatori di MAME non sono tutti grandi giocatori, anzi, sono per lo più “grandi” programmatori. L’identificazione delle anomalie dell’emulazione è quindi molto più semplice per chi ha vissuto o vive tuttora immerso nei videogiochi, conoscendone ogni singolo dettaglio.

Per avvicinare le due comunità, nel 1999, è nato il *MAME Testing Project* [29], con scopo principale proprio quello di assistere gli sviluppatori di MAME nel tracciamento degli errori riscontrati. Il progetto ebbe fin da subito un grande successo, in quanto rappresentava un punto centrale e più controllato per l’identificazione delle anomalie. Il progetto, chiamato ora *MAME Testers*, è diventato ufficialmente l’unico punto in cui si concentra l’intera gestione dei bug.

MAME Testers si avvale di *Mantis* come bugtracking system, a cui sono state effettuate delle modifiche (liberamente disponibili) per adattarlo alle esigenze del progetto.

L’inserimento di nuovi bug da parte degli utenti è strettamente regolamentato, in modo da evitare il proliferare di segnalazioni duplicate o difficilmente tracciabili. Per aiutare gli utenti a compilare correttamente un ticket di segnalazione è fornita un’ampia e dettagliata documentazione che riguarda sia le regole che l’utente deve rispettare sia l’utilizzo dello strumento di tracciamento.

Viene esplicitamente richiesto di non aprire nuovi ticket che riguardano giochi il cui stato di emulazione è indicato come *NOT WORKING*. È un regola molto importante perché previene la segnalazione di anomalie invalide: gli sviluppatori sono già consapevoli che l’emulazione di tali giochi deve esser corretta.

## 9.5 Real-time chat

Non esiste, ne è mai esistito, alcun canale pubblico ufficiale per poter chattare con gli sviluppatori di MAME. In passato, il canale *#MAME* presente nel server *UnitedUsers* veniva spesso frequentato anche da MAME Dev, ma dopo l’arrivo di alcune persone poco gradite, abbandonarono il canale. Attualmente si possono incontrare alcuni MAME Dev in un canale privato che raduna molti sviluppatori di progetti legati all’emulazione. Purtroppo non sono note le coordinate per accedere al canale.

## 9.6 Wiki

Nel sito ufficiale [1] è presente un’area wiki relativa al progetto. Allo stato attuale la wiki è divisa in 7 categorie, qui di seguito elencate:

- Non-Working Drivers: fornisce molti dettagli sulle motivazioni per cui alcune architetture non sono ancora correttamente emulate in MAME. Lo scopo di questa

sezione è tenere traccia di cosa manca nell'emulazione dei giochi, utile sia per uno sviluppatore che voglia portarne avanti lo sviluppo sia per gli utenti che vogliono capire perché il loro gioco preferito non funziona, senza inviare richieste di supporto.

- **How MAME Works:** come si evince dal nome, vengono forniti molti dettagli sull'architettura software di MAME. Sono descritti in dettaglio molti aspetti su come lavora il nucleo di MAME e le tecniche migliori per programmare sopra di esso.
- **Developer WIPs:** sono raccolte le *To-Do list* di alcuni developer utilizzate per comunicare pubblicamente su cosa stanno lavorando e quali sono le nuove funzionalità che hanno intenzione di introdurre in futuro. Vengono inoltre forniti i collegamenti alle pagine personali di tutti gli sviluppatori.
- **Frequently Asked Questions:** contiene le FAQ relative al MAME. Rivolto soprattutto ai nuovi utenti che vogliono avere maggiori informazioni sul progetto.
- **Previous MAME Versions:** per ogni rilascio di MAME sono riportate le modifiche che sono state effettuate.
- **How Arcade Games Work:** insieme di informazioni relative all'architettura dei giochi arcade, che ne determina il loro funzionamento interno.
- **Let's Emulate Space Invaders:** una guida step-by-step, non ancora pubblicata, su come emulare *Space Invaders*, uno dei giochi più famosi di sempre, utilizzando il core di MAME.

## 9.7 Build automation

La compilazione del codice sorgente avviene con l'ausilio del programma *GNU Make*, pilotato da un *Makefile* che permette la compilazione di MAME per le architetture supportate, abilitando eventualmente alcune opzioni specifiche, ad esempio il debugging. Il *Makefile* è strutturato in modo da identificare autonomamente i vari sistemi supportati così che sia sufficiente eseguire il comando `make` senza ulteriori opzioni.

Ogni rilascio di MAME, prima che venga reso pubblico, viene compilato utilizzando 8 schemi di configurazione diversi. Se tutte le prove di compilazione completano con successo, viene pubblicato sul sito ufficiale il nuovo codice sorgente e le 4 distribuzioni binarie, descritte nel paragrafo [8.1 a pagina 23](#). Non è noto come sono gestiti questi cicli di compilazione, l'ipotesi più probabile è che vengano eseguiti mediante un semplice file batch.

## 9.8 Note di rilascio

Ad ogni rilascio di MAME viene fornito insieme alla distribuzione del software anche un file testuale con le note di rilascio per la versione corrente, chiamato *whatsnew\_num-versione*. Le informazioni contenute nelle note di rilascio sono molteplici e ben accurate, in questo modo è possibile avere una visione complessiva dei cambiamenti che sono avvenuti da una versione ad un'altra. Questo è molto comodo vista la scelta di non rendere pubblico il repository SVN, infatti leggendo le note di rilascio un contributore può allineare le sue modifiche con il core. La struttura delle note di rilascio è cambiata nella storia di MAME raggruppando lo "standard" attuale che raccoglie le seguenti informazioni:

- MAMETesters Bugs Fixed: vengono riportati i bug segnalati su MAME Testers [29] che sono stati corretti in questa versione. Per avere una rapida referenza con i rispettivi bug viene indicato il codice che li identifica nello strumento di issue tracking e la categoria di bug in cui appartengono.
- Source Changes: in questa sezione vengono riportati tutti i cambiamenti che sono stati effettuati nel core di MAME e nei driver per i vari dispositivi emulati. Viene data molta enfasi in caso di cambiamenti che toccano il nucleo di MAME cambiando le API rivolte agli sviluppatori di driver.
- New games added or promoted from NOT\_WORKING status: sono segnalati i nuovi giochi funzionanti aggiunti oppure che passano da uno stato di non funzionanti a funzionanti.
- New clones added: come la precedente però in questo caso si tratta dei "cloni" e non nuovi giochi unici.
- New games marked as GAME\_NOT\_WORKING: sono segnalati i nuovi giochi aggiunti i quali non sono però ancora in uno stato giocabile.

Tutte le modifiche segnalate nelle note di rilascio sono minuziosamente attribuite agli sviluppatori che effettivamente ci hanno lavorato. Per i nuovi giochi sono segnalati anche i *dumper* che hanno estratto i dati contenenti nelle ROM dei giochi. L'attribuzione delle modifiche è molto importante per i contributors i quali si sentono considerati e valorizzati nel loro lavoro. Inoltre, così facendo i contributors tendono a lavorare più volentieri al progetto e favorisce l'inserimento di nuovi sviluppatori.

Un problema relativo all'attuale gestione delle note di rilascio è dovuto al fatto che sono scritte e mantenute a mano dal project leader. Sebbene la manutenzione manuale possa portare un buona qualità delle note, sarebbe necessaria maggiore automazione in modo da aumentare il numero di informazioni presenti diminuendo drasticamente i tempi per la loro stesura. Una prima soluzione potrebbe essere quella di interfacciarsi con lo strumenti di bug tracking in modo da estrapolare i bug che vengono via via corretti

generando automaticamente i report. Per automatizzare anche le restanti informazioni sarebbe necessario adottare strumenti avanzati di tracciamento con i quali sia possibile non sol gestire i bug del progetto, ma anche il tracciamento dei vari milestone, rilasci e dei ticket corrispondenti ai lavori che vengono inizializzati, ad esempio a fronte di un bug, e poi completati.

Se si riuscisse ad avviare un sistema simile, la gestione delle informazioni sarebbe nettamente semplificata e priva di errori umani, portando ad esempio alla creazione automatica delle note di rilascio, oltre a molti nuovi orizzonti che andrebbero ad aprirsi. D'altro canto una corretta gestione di questa nuova mole di dati dovrebbe essere rigidamente normata in modo che i contributors seguano degli standard per mantenerle in uno stato consistente, che potrebbe portare eccessiva frustrazione, non essendo dipendenti stipendiati, oltre a tempi abbastanza lunghi per inizializzare un tale sistema.

## 9.9 Ambiente di sviluppo

L'ambiente di sviluppo ufficiale comprende il compilatore MinGW, il porting per Windows di GCC, uniti ad altri tool UNIX-like come *diff*, *patch* e infine alcune librerie aggiuntive per la grafica, DirectX per Windows e SDL per gli altri sistemi operativi. L'intero ambiente, pronto per l'uso, viene distribuito tramite il sito ufficiale. Sono contenute anche le informazioni necessarie per replicare manualmente la configurazione dell'ambiente se ce ne fosse bisogno.

La soluzione adottata è molto potente in quanto sfrutta uno tra i migliori compilatori C/C++ disponibili. Tuttavia, è molto difficile effettuare il debug del codice senza un IDE che lo supporti, utilizzare infatti il solo *GDB (GNU Project Debugger)* è molto scomodo. Per risolvere questo problema, viene fornita una guida per configurare l'ambiente di sviluppo in modo che possa essere sfruttato attraverso *Microsoft Visual Studio*, che offre un'interfaccia molto più comoda per supportare tutte le fasi di sviluppo.

## Ringraziamenti

*Si ringraziano Fabio "etabeta" Priuli, Roberto "robiza" Zandonà e i membri del MAME Italian Forum [23] per avermi fornito preziose informazioni. Si ringrazia in particolar modo Riccardo "ricky74" Cattaneo per aver apprezzato e pubblicato questo lavoro sul portale MAME Channel [24].*

## Riferimenti bibliografici

- [1] <http://mamedev.org> (Citato alle pagine 3, 9, 20, 23, 24 e 26.)
- [2] <http://www.arcade-museum.com> (Citato a pagina 4.)
- [3] [http://www.arcade-museum.com/game\\_list.php?letter=A&type=Videogame](http://www.arcade-museum.com/game_list.php?letter=A&type=Videogame) (Citato a pagina 4.)
- [4] <http://www.videotopia.com> (Citato a pagina 4.)
- [5] <http://www.tosecdev.org> (Citato a pagina 4.)
- [6] <http://www.mess.org> (Citato alle pagine 4 e 16.)
- [7] NICOLA SALMORIA (2002), *Il progetto MAME: reverse engineering e macchine da gioco*, Università degli studi di Siena. (Citato a pagina 5.)
- [8] <http://iep.ath.cx/interviste/mirko.htm> (Citato a pagina 8.)
- [9] <http://web.archive.org/web/20060427082535/http://www.ultracade.com> (Citato a pagina 14.)
- [10] <http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=76627578&action=Request+Status> (Citato a pagina 14.)
- [11] <http://tarr.uspto.gov/servlet/tarr?regser=serial&entry=78578919&action=Request+Status> (Citato a pagina 14.)
- [12] <http://bagofglass.com/?p=91> (Citato a pagina 14.)
- [13] <http://aarongiles.com/?p=221> (Citato a pagina 16.)
- [14] [http://mamedev.org/devwiki/index.php/MAME\\_Coding\\_Conventions](http://mamedev.org/devwiki/index.php/MAME_Coding_Conventions) (Citato a pagina 17.)
- [15] <http://mameworld.info> (Citato a pagina 21.)
- [16] <http://mameinfo.mameworld.info> (Citato a pagina 21.)
- [17] <http://www.arcade-history.com> (Citato a pagina 21.)
- [18] <http://maws.mameworld.info/maws> (Citato alle pagine 21 e 22.)
- [19] <http://catlist.mameworld.info> (Citato a pagina 21.)
- [20] <http://highscore.mameworld.info> (Citato a pagina 22.)
- [21] <http://cheat.retrogames.com> (Citato a pagina 22.)

- [22] <http://mamedev.emulab.it/clrmamepro> (Citato a pagina 22.)
- [23] <http://www.mameitalia.net> (Citato alle pagine 22 e 29.)
- [24] <http://www.mamechannel.it> (Citato alle pagine 22, 23 e 29.)
- [25] <http://www.emulab.it> (Citato a pagina 22.)
- [26] <http://www.progettoemma.net> (Citato a pagina 22.)
- [27] <http://www.mameui.info> (Citato a pagina 23.)
- [28] <http://mameicons.free.fr/mame32p> (Citato a pagina 23.)
- [29] <http://mametesters.org> (Citato alle pagine 26 e 28.)

## **Storico**

**WIP (xx/x/xxxx) :**

- Piccole modifiche allo stile del documento.
- Aggiunto paragrafo [9.8 a pagina 28](#).
- Corrette alcune imprecisioni linguistiche.

**0.138 (20/7/2010) :** primo rilascio pubblico.