

## Titolo: *InterArtes*

ISSN 2785-3136

Periodicità: annuale

Anno di creazione: 2021

Editore: Dipartimento di Studi Umanistici – Università IULM - via Carlo Bo 1 - 20143 Milano

**Direzione:** Laura Brignoli - Silvia T. Zangrandi

### **Comitato di direzione**

Gianni Canova, Mauro Ceruti, Paolo Proietti,  
Giovanna Rocca, Vincenzo Trione

### **Comitato editoriale**

Maria Cristina Assumma; Matteo Bittanti;  
Mara Logaldo; Stefano Lombardi Vallauri;  
Marta Muscariello

### **Comitato scientifico**

Daniele Agiman (Conservatorio Giuseppe Verdi Milano); Maurizio Ascari (Università di Bologna); Sergio Raúl Arroyo García (Già Direttore Generale del Instituto Nacional de Antropología e Historia); Claude Cazalé Bérard (Université Paris X); Gabor Dobo (Università di Budapest); Felice Gambin (Università di Verona); Maria Teresa Giaveri (Accademia delle Scienze di Torino); Maria Chiara Gnocchi (Università di Bologna); Augusto Guarino (Università L'Orientale di Napoli); Rizwan Kahn (AMU University, Aligarh); Anna Lazzarini (Università di Bergamo); Massimo Lucarelli (Université de Caen); Elisa María Martínez Garrido (Universidad Complutense de Madrid); Martínez Falero (Universidad Complutense de Madrid); Donata Meneghelli (Università di Bologna); Giampiero Moretti (Università Orientale di Napoli); Raquel Navarro Castillo (Escuela Nacional de Antropología y Historia, Mexico); Francesco Pigozzo (Università e-campus); Richard Saint-Gelais (Université Laval, Canada); Massimo Scotti (Università di Verona); Chiara Simonigh (Università di Torino); Evangelia Stead (Université Versailles Saint Quentin); Andrea Valle (Università di Torino); Cristina Vignali (Université de Savoie-Mont Blanc); Frank Wagner (Université de Rennes 2); Anna Wegener (Università di Firenze); Haun Saussy (University of Chicago); Susanna Zinato (Università di Verona).

### **Segreteria di redazione**

Caterina Bocchi

Laura Brignoli, Silvia Zangrandi – Introduzione.

ARTICOLI

Francesco Pigozzo, Daniela Martinelli - Médiatisations de l'inconscient et écritures de l'expérience: six «monographies» de la pédagogie institutionnelle entre analyse littéraire et hypothèses épistémologiques.

Nicola Tallarini – Risvolti e quarte di copertina per le collane d'autore: un genere ibrido tra editoria e letteratura.

Lucia Pasini – *Hello, World*. Musica a programma per il terzo millennio.

Giovanni Favata – Altre lingue nell'italiano scritto di studenti universitari stranieri: il ruolo del repertorio linguistico.

Bénédicte Van Gysel – Typologie des textes à traduire : l'éclairage de l'hybridité.

Benedetta Bartolini – Le caractère hybride de *Psyché*: une collaboration au service d'un spectacle «sans égal».

Ilaria Ottria – Paolo Barbieri illustra Dante. Ibridismo e tessere ovidiane in *Inferno* XXIV-XXV.

Greta Gribaudo – Ibridare le parole e le immagini con le forme del mondo. Il gioco di Italo Calvino col labile confine tra mondo-scritto e non-scritto.

Simone Bacci – Ibridazione e ridondanza. L'effetto stroboscopico di Saviano.

Marie Cécile Bouguia Fodjo - Hybridation culturelle et transmutations identitaires dans *L'Africain* de Jean-Marie Gustave Le Clézio.

Marco Ottaiano - Derive digitali, distopie iperreali e ibridazioni narrative in *Kentukis* di Samanta Schweblin.

Mauro Distefano – Ibridazioni artistiche: il caso de *Le lune di Hvar* di Lalla Romano.

Francesco Patrucco – Il mito del trickster: una figura ibrida e le sue trasposizioni eterogenee.

RECENSIONI

Céline Powell – Le dialogue entre la norme et l'hybridité dans la littérature italienne (BARBARA KUHN, DIETRICH SCHOLLER (éds.), *Italienische Literatur im Spannungsfeld von Norm und Hybridität: Übergänge – Graduierungen – Aushandlungen*, Peter Lang, 2021).

Fabrice De Poli – LibRidinose permutazioni: esercizio di riscrittura pascoliana (LUCA CHITI, *Canti di Castellaccio. Philologica pascoliana*, I Quaderni dell'Oplepo, n.13, 2022).

Florjer Gjepali – Nell'esperienza estetica: corpo e disposizione (EDWARD SLOPEK, *Bodies of Art: The Shaping of Aesthetic Experience*, Quodlibet, 2021).

## ***Hello, World. Musica a programma per il terzo millennio***

Lucia PASINI

Université Sorbonne Nouvelle, Università degli Studi di Torino

**Abstract:**

Some practices of artistic creation are founded on the juxtaposition of two languages or two different media, such as setting a text to music. As a producer of hybrid artefacts, this practice poses multiple problems related to its aesthetic and semantic status. In this article, I will focus on a piece (*Hello, World* by Nicola Campogrande) that allows me to reflect on these issues and adds a further complication: the texts to be set to music are written in source code. After a brief discussion of the features of source code and of their reaction to being set to music, I will turn specifically to Campogrande's piece to show that source code can be considered as an ideal material for reflecting on the far more widespread practice of setting natural language texts to music. In addition, *Hello, World* equally questions current practices of reading the code itself.

**Keywords:**

Source code, musical setting, Campogrande, hybridity, *Lieder*

Il *Nuovo De Mauro*<sup>1</sup> attesta la presenza della parola “ibrido” nella lingua italiana a partire dal 1498, quando questo lemma indicava la provenienza «da due razze differenti». Nei secoli successivi, i significati dell’aggettivo si sono diversificati in funzione del dominio tecnico in cui il vocabolo viene impiegato, ma la sua accezione più comune resta legata ai caratteri di eterogeneità e discordanza: si può così parlare di uno *stile ibrido*, ma non solo. Alcune pratiche di creazione artistica sono fondate proprio sull’accostamento di due linguaggi o, volendo, di due *media* discordanti ed eterogenei, come ad esempio la messa in musica di un testo. Questa pratica, in quanto produttrice di artefatti ibridi, pone un’enorme quantità di problemi legati al suo statuto estetico e semantico. Infatti, il rapporto tra l’istanza che produce il significato (in primo luogo il poeta, ma anche l’io lirico, e in questo caso anche il compositore e l’interprete) e l’istanza che lo riceve (il lettore o lo spettatore o l’ascoltatore) è problematizzato dalla moltiplicazione dei fattori coinvolti – per dirla con Michel Imberty: «l’interpretazione è infatti una complessa interazione tra l’opera scritta, il progetto del compositore e la comprensione personale dell’interprete, senza contare ciò che l’ascoltatore può aggiungere a questa impalcatura già affollata»<sup>2</sup> (Imberty, 1992: 41, traduzione mia).

---

<sup>1</sup> Il dizionario è consultabile sul sito di *Internazionale*, <<https://dizionario.internazionale.it/>>. Ultimo accesso: 1° giugno 2022.

<sup>2</sup> Tuttavia, poco più avanti nello stesso testo (p. 42-43), Imberty racconta un esperimento condotto sulla melodia *Mandoline* di Debussy su una poesia di Verlaine: «Ho quindi chiesto agli esecutori di registrare due versioni di *Mandoline*, una in un tempo due volte più lento dell’altra, la quale rispettava le indicazioni di Debussy.

In altre parole, il compositore apporta delle modifiche all'istanza all'origine del significato, che si tratti del poeta stesso o di un narratore fittizio. In questo senso, è possibile considerare la trasposizione musicale come una pratica invasiva di per sé, poiché consiste nell'esportazione di un testo poetico verso qualcos'altro<sup>3</sup>. La questione del valore estetico del nuovo oggetto artistico è irrilevante in questo contesto: ciò che conta è che la poesia cessa di essere tale e diventa *Lied*, *song*, *mélodie*. Inoltre, per quanto riguarda la percezione immediata del significato del testo poetico, questa viene certamente alterata dalla messa in musica: se nella poesia il suono è all'origine del significato, il significato cambia quando cambia il suono. Più precisamente, se l'immaginazione è il luogo in cui il significato prende forma, le immagini evocate dal testo non sono le stesse che evoca la musica, forse anche quando la seconda imita il primo. Infine, quando l'attenzione del destinatario si sposta dalla parola, al centro della poesia, ai suoni, al centro del brano musicale, il risultato è l'obliterazione del significato ulteriore che l'immaginazione accorda alle parole situate in un contesto poetico a favore del loro status ordinario, a cui sono relegate dalla coesistenza con un altro mezzo di espressione artistica che diventa preponderante nel contesto musicale. In questo senso, qualsiasi procedimento di messa in musica può essere visto come un'espropriazione del testo dal suo contesto, dal suo *medium*, dalla sua intenzione originale. Ovviamente, questo modo di intendere l'atto di musicare un testo, concepito indipendentemente dal possibile intervento di un compositore, è alquanto provocatorio, ma pone dei problemi rilevanti nell'ambito di un'interrogazione sull'ibridazione reciproca di questi due mezzi di espressione artistica, soprattutto se questo fenomeno viene considerato attraverso il prisma della teoria della ricezione (Fish, 1980; Iser, 1994; Jauß, 1982).

Quest'ultima è ampiamente conosciuta nel contesto contemporaneo e, secondo questa concezione del rapporto estetico tra l'artista e il destinatario, questi svolge un ruolo attivo nella costituzione dell'oggetto artistico in questione. Così, il processo di messa in musica di un testo poetico può essere visto come un lavoro ermeneutico da parte del compositore, più o meno consapevole, più o meno approfondito, ma che merita di essere preso in considerazione nel campo degli studi letterari e che si esprime nella creazione di un artefatto artistico

---

Il risultato essenziale di questo esperimento è stato che questo rallentamento del tempo non ha alterato in modo apprezzabile il clima emotivo ed espressivo del brano, che è dovuto a molti altri fattori specifici della scrittura e indipendenti dall'esecuzione stessa». Pertanto, sarebbe prudente qualificare la misura in cui l'esecuzione può influenzare la percezione degli ascoltatori; d'altra parte, l'esperimento di Imberty indaga solo sui cambiamenti di tempo, e si può immaginare che la modifica di altri parametri salienti possa portare a risultati diversi.

<sup>3</sup> Questa è anche la direzione delineata in Barral, Seretti e Paplomata (2021: 11).

inedito, il brano musicale. Inoltre, lo studio della ricezione delle poesie e dei brani stessi porta a una comprensione raffinata delle relazioni comunicative tra i poli del poeta, del compositore, degli interpreti e del pubblico – in breve, tra le istanze appena menzionate come responsabili, nella loro interazione, della negoziazione del significato. Va tenuto presente, tuttavia, che questa negoziazione è determinata in modo specifico dallo spazio sociale e/o geografico, nonché dal momento nel corso della storia, in cui essa ha luogo.

In ciò che segue, mi concentrerò su un brano che permette di riflettere su queste questioni, e aggiunge anche un'ulteriore complicazione: i testi musicati non sono in linguaggio naturale, ma in codice sorgente<sup>4</sup>. Si tratta di *Hello, World*<sup>5</sup>, ciclo di Lieder composto da Nicola Campogrande su commissione di CodeFest, Codexpo e l'Università di Torino nell'ambito del primo festival al mondo dedicato al codice sorgente, ed eseguito in prima assoluta il 9 giugno 2021 dall'Orchestra Sinfonica di Milano Giuseppe Verdi, con il concorso della soprano Vittoriana De Amicis e la direzione di Alpesh Chauhan (Campogrande, 2021). In questo brano, il compositore ha scelto quattro programmi, espressi in quattro linguaggi di programmazione diversi, che fanno tutti la stessa cosa: scrivono sul monitor del computer la frase «hello, world». Questi programmi si chiamano abitualmente proprio in questo modo, «Hello, world», e costituiscono il primo esempio che viene insegnato quando si comincia a imparare un nuovo linguaggio di programmazione, o quando se ne inventa uno. La scelta del compositore è stata orientata dai committenti che gli hanno proposto diversi programmi espressi in una quantità di linguaggi di programmazione e tratti dalla collezione presente su <<http://helloworldcollection.de/>><sup>6</sup>. Durante l'evento di apertura del Codefest, *Singing the Code*, tenutosi il 28 settembre 2021, Campogrande ha sottolineato come il suo approccio alla lettura del codice prima della fase di composizione sia stato all'insegna dell'intuito e della sensibilità, invece che informato da uno studio erudito<sup>7</sup>.

Innanzitutto, dunque, prenderò in considerazione la possibilità di concettualizzare il codice sorgente come un artefatto testuale, e quindi di trattarlo come qualsiasi altro artefatto testuale, di leggerlo o di metterlo in musica. A sua volta, questo permetterà di sottolinearne le differenze e le similitudini con il linguaggio naturale, sia nei loro contesti di impiego

---

<sup>4</sup> Le caratteristiche di questo mezzo di comunicazione saranno discusse nel dettaglio più avanti, ma per il momento lo si può definire come «la forma testuale del codice di programmazione che viene scritto e modificato dai programmatori informatici» (Berry, 2011: 29, traduzione mia).

<sup>5</sup> I brani sono disponibili sul sito del compositore: <<http://www.campogrande.it/portfolios/hello-world/>>. Ultimo accesso: 29 maggio 2022.

<sup>6</sup> Ultimo accesso: 29 maggio 2022.

<sup>7</sup> La registrazione dell'evento è disponibile su: <<https://www.youtube.com/watch?v=ppRbblstOvs>>. Ultimo accesso: 31 maggio 2022.

corrente sia nelle rispettive reazioni quando vengono musicati, e quindi soggetti all'espropriazione descritta più in alto. Una volta discusse queste problematiche di ambito più generale, sarà possibile concentrarsi sul ciclo *Hello, World* al fine di toccare con mano le tematiche trattate precedentemente, anche se i riferimenti puntuali alla composizione saranno presenti nell'intera argomentazione fornendo una controparte sensibile alla riflessione teorica.

### 1. Interruzioni e perturbazioni

Prima di entrare nel vivo della discussione intorno agli effetti che la presenza della musica sortisce sul codice sorgente, è bene cominciare da un'osservazione che permette di pensare la sua specificità rispetto alle altre componenti delle tecnologie che fanno ormai parte integrante della vita quotidiana. Infatti, il codice sorgente non è che una fase della scrittura e dell'esecuzione di un programma: il livello umano, in cui è il programmatore umano che scrive ed eventualmente legge, rilegge e modifica il testo. Tuttavia, la macchina non è in grado di eseguire direttamente le istruzioni articolate in un linguaggio di programmazione, ed è invece necessario che queste siano tradotte in linguaggio macchina, cioè in un linguaggio comprensibile per un certo elaboratore<sup>8</sup>. Prendendo proprio «Hello, World», il programma si presenta così in linguaggio C:

```
#include "stdio.h"

int main() {
    printf("Hello World!");
    return 0;
}
```

Ed ecco invece alcune righe della versione compilata:

```
000060: a090 0408 0e00 0000 0e00 0000 0600 0000
000070: 0010 0000 0000 0000 0000 0000 0000 0000
000080: ba0e 0000 00b9 a090 0408 bb01 0000 00b8
```

---

<sup>8</sup> Questo è solo un passaggio all'interno di un processo molto più complesso che può essere articolato in diversi modi a seconda delle condizioni locali. Descriverlo in maniera esaustiva è fuori dagli scopi di questo contributo.

Non è impossibile pensare di musicare questa versione, ma questo testo è molto più difficile da decifrare per l'umano che la sua realizzazione in linguaggio C, sebbene anch'essa richieda una competenza specifica per essere intelligibile. Campogrande ha fatto qualcosa di simile scegliendo un programma in Malbolge<sup>9</sup>, ma è necessario sottolineare che lo scopo dell'invenzione di Malbolge è proprio la creazione di un linguaggio incomprensibile, mentre la compilazione automatica produce un testo estremamente chiaro per la macchina, sebbene esso presenti qualche difficoltà alla lettura umana.

Detto questo, si possono individuare cinque aspetti caratteristici del codice sorgente sui quali il procedimento di messa in musica causa delle perturbazioni, se non delle interruzioni vere e proprie.

Il primo elemento che subisce una modifica sostanziale è la funzione conativa tipica del codice sorgente. Questa, secondo la definizione di Roman Jakobson, è orientata verso il destinatario e «trova la sua espressione grammaticale più pura nel vocativo e nell'imperativo» (Jakobson, 1960: 355, traduzione mia). In un certo senso, il codice sorgente è un insieme di istruzioni, espresse all'imperativo per i linguaggi che seguono questo paradigma<sup>10</sup>, che il programmatore rivolge alla macchina. Al tempo stesso, l'azione di scrittura da parte del programmatore può essere descritta, secondo la terminologia di John Langshaw Austin, come uno *speech act*, cioè un momento in cui l'espressione di un'azione è l'azione stessa (Austin, 1975: 6; cfr. anche Coste, 2002: 273-275). Il codice sorgente è quindi essenzialmente performativo<sup>11</sup>, oltre che diretto prevalentemente verso il destinatario.

Com'è ovvio, nell'ambito della messa in musica, il codice continua a essere rivolto verso un destinatario, ma questi non è più la macchina, bensì un pubblico umano. Se la funzione conativa è dunque perturbata ma largamente mantenuta nel nuovo contesto, l'elemento performativo è pienamente interrotto: il potenziale di azione conferito alla macchina è del tutto neutralizzato. Il programma musicato non *fa* nulla. Tuttavia, la dimensione performativa è mantenuta in almeno un'occasione nel brano di Campogrande: il terzo *Lied*, «In Delphi», si conclude con l'intonazione da parte della soprano della parola *end*. Viene quindi

---

<sup>9</sup> Malbolge è un linguaggio di programmazione esoterico, inventato da Ben Olmstead nel 1998. I linguaggi esoterici, o *esolangs*, non sono progettati per offrire una soluzione efficiente o elegante a problemi computazionali, ma per esplorare e potenzialmente sovvertire gli assunti della scienza informatica oppure come scherzo a scopo ludico. Programmare in Malbolge è talmente difficile che il programma «Hello, World» «non è stato scritto a mano, ma da un altro programma, che ha essenzialmente trattato Malbolge come un problema di crittoanalisi» (Temkin, 2017: 85, traduzione mia).

<sup>10</sup> Per uno studio della semiosi di altri paradigmi, specialmente i paradigmi ad oggetti e funzionale, si veda Tanaka-Ishii (2006).

<sup>11</sup> Per una riflessione dettagliata sulle implicazioni della possibilità di concepire il codice sorgente come uno *speech act*, si veda Cox, McLean (2013).

ordinato al brano di terminare ed esso prontamente obbedisce. Ma non bisogna tralasciare che questo episodio costituisce l'eccezione e non la regola, ed è proprio il suo statuto eccezionale a renderlo un momento particolarmente notevole e non privo di ironia: è infatti preceduto di qualche battuta dall'intonazione della parola *begin* – accompagnata dall'indicazione di espressione «sicuro» in partitura –, la quale però non inizia nulla di chiaramente riconoscibile. Quando invece si sente la parola *end*, seguita dal silenzio che segnala la fine del *Lied*, il passaggio a un regime enunciativo diverso diventa evidente, nel momento in cui, seguendo la formulazione di Austin, l'enunciazione dell'azione *costituisce* essa stessa l'azione. Senza dimenticare che, in questo caso, la musica stessa assume il ruolo di allocutario (colui al quale ci si rivolge quando si parla) di una presa di parola di cui non è il vero destinatario, poiché questo ruolo rimane presso il pubblico.

La seconda perturbazione, a cui è stato già accennato, concerne la funzione poetica, cioè «l'attenzione al messaggio in quanto tale» (Jakobson, 1960: 356, traduzione mia), nella misura in cui è possibile distinguere il messaggio dalla sua espressione nell'ambito di uno *speech act*. Difatti, le modalità e i contenuti della comunicazione messi in atto dal codice di programmazione sono limitati in maniera significativa dal fatto che il destinatario è una macchina, la cui comprensione funziona diversamente dall'intelligenza umana. In questo caso, la comunicazione non è interrotta dall'intervento del compositore, ma è considerevolmente alterata, soprattutto a causa della rimozione della macchina da una delle estremità dell'equazione comunicativa.

In seguito, alcune perturbazioni sostanziali interessano la dialettica tra denotazione e connotazione. La denotazione di una parola o di un'espressione è il suo significato diretto, distinto dai concetti ad essa associati o da essa suggeriti. Altrimenti detto, la denotazione di una parola corrisponde a ciò che quella parola rappresenta direttamente. La connotazione, invece, si riferisce a ciò che essa comporta o suggerisce, ovvero ai significati secondari o alle implicazioni che le sono associate. È importante tenere a mente che, se la denotazione di un vocabolo è oggetto di accordo tra i parlanti di una medesima lingua, la connotazione può subire delle variazioni rilevanti da locutore a locutore. Nel caso della messa in musica del codice, cioè un mezzo di espressione in cui la denotazione spadroneggia a scapito della connotazione, quest'ultima è sicuramente accessoria rispetto all'efficacia del programma redatto, ma è resa esplicita quando il codice è accompagnato dalla musica e anzi prende il sopravvento sulla denotazione, ormai offuscata dalla concorrenza dell'espressione musicale, non soltanto perché, come si è detto, il software musicato non è più performativo, ma anche



perché l'intervento della musica permette di creare legami tra elementi che non sono in relazione l'uno con l'altro nella loro iterazione puramente testuale. Ad esempio, nel quarto ed ultimo *Lied* del ciclo *Hello World*, Campogrande riprende per tre volte una breve sezione caratterizzata da un profilo melodico chiaramente riconoscibile nella parte vocale (un'ampia frase ascendente seguita da un frammento discendente, che a sua volta precede un segmento in cui la voce si fa più statica e adotta un movimento cromatico) e nell'accompagnamento degli archi, che seguono la discesa della voce in una serie di accordi semidiminuiti, che a loro volta scendono per semitoni. In corrispondenza di questo passaggio, la voce canta la prima volta: «Aperta parentesi tonda uguale minore di apostrofo inverso», la seconda: «chiusa parentesi tonda virgolette effe», e la terza: «apostrofo inverso uguale asterisco zeta minuscola chiusa parentesi quadra». Si tratta dunque di caratteri che alla lettura non dimostrano in maniera evidente un'affinità reciproca, ma nel brano musicale viene conferita loro una connotazione simile per mezzo di un trattamento musicale analogo. Questa connotazione, del resto, è completamente nelle mani e nella sensibilità del compositore, siccome in questo caso egli non si trova a musicare delle parole, accompagnate dal loro significato denotativo, ma dei segni, il cui significato denotativo è accessibile soltanto a chi possiede determinate competenze tecniche, di cui il compositore è sprovvisto, come del resto ne è sprovvista con tutta probabilità la maggior parte di un pubblico generico.

Successivamente, un aspetto che subisce in questo caso un'interruzione vera e propria è la provvisorietà del codice sorgente<sup>12</sup>. Questo, infatti, è spesso a un passo dall'essere qualcosa di diverso; è spesso parziale ed è spesso in potenza: è sufficiente che un utente apporti una modifica, per quanto minimale, e il programma cessa di essere quello che era e diventa qualcos'altro. Si tratta di un elemento centrale del codice di programmazione: un software che non è mantenuto, il cui codice non viene modificato e aggiornato, è destinato all'abbandono e all'obsolescenza. Eppure, affinché un programma possa essere musicato, è necessario che il suo codice venga fissato in una condizione di stabilità virtualmente perpetua.

E infine, una perturbazione aggiuntiva concerne l'autorialità collettiva tipica delle comunità di programmatori. È raro difatti che un programma che contiene centinaia o migliaia di righe di codice possa essere attribuito a un singolo autore, e la vocazione alla parzialità di cui si è appena parlato rende spesso macchinoso risalire all'artefice di ogni intervento sul

---

<sup>12</sup> Si tratta senza dubbio di una generalizzazione, a cui esistono delle eccezioni.

codice di programmazione<sup>13</sup>. Se quella che si potrebbe definire come una paternità distribuita non è una caratteristica specifica del codice sorgente, ma ad esempio contraddistingue in uguale misura alcune fiabe, i miti dell'antichità o certe melodie popolari, nel caso del codice questa è però accompagnata da una connotazione opposta a quella che viene abitualmente accostata alle produzioni folkloristiche almeno a partire dall'epoca romantica. Queste, infatti, avrebbero la prerogativa di rappresentare una dimensione atavica, ancestrale, autentica, di una comunità spesso qualificata a sua volta da tratti mitici e leggendari, mentre il codice, perlomeno nella sua accezione comune, si situa agli antipodi di questo tipo di espressione immediata e genuina.

Una volta individuati questi elementi di sovversione che l'aggiunta della musica provoca sul codice sorgente, vorrei concentrarmi più da vicino su due aspetti più specifici, cioè, da un lato, la funzionalità e, dall'altro, l'espressività del codice di programmazione. Se la prima viene riconosciuta senza difficoltà come una delle caratteristiche fondanti del codice di programmazione come mezzo comunicativo, la seconda è spesso relegata a un ruolo di secondo piano, nonostante gli sforzi per contrastare proprio questa tendenza da parte di un campo di indagine accademica fondato recentemente in California – i *Critical Code Studies* (Marino, 2014; Marino, 2020)<sup>14</sup>. Sulla base dei lavori di Mark Marino, queste due prospettive saranno considerate in questa sede come equipollenti.

Per cominciare, dunque, una prima osservazione a proposito della funzionalità dei linguaggi di programmazione: affinché un software possa essere efficace ed eseguire delle azioni, è necessario che l'oggetto che viene manipolato dal programma sia tradotto in una forma che permette alla macchina di eseguire questa manipolazione. La macchina, infatti, non agisce sull'oggetto vero e proprio, ma sulla sua rappresentazione, o, perlomeno, su alcune componenti di questa rappresentazione che di volta in volta sono ritenute pertinenti rispetto all'obiettivo posto davanti al programmatore. In altre parole, perché la realtà possa essere processata dalla macchina è indispensabile che ne venga fornita una rappresentazione discreta, cioè composta da elementi distinti e discontinui. Ora, la discussione dei problemi di natura epistemologica ed etica che questa considerazione solleva sconfinava oltre

---

<sup>13</sup> Di fatto l'identificazione degli autori di una determinata modifica è possibile, ma queste informazioni restano nelle note e nella documentazione del codice di programmazione, le quali raramente vengono rese pubbliche, se non nel caso di software open-source in cui è previsto che lo siano.

<sup>14</sup> La possibilità di studiare congiuntamente le caratteristiche dei linguaggi di programmazione e del linguaggio umano viene esplorata anche in Tanaka-Ishii (2010) e in Gazoni (2018).

l'ambito ristretto di questo contributo<sup>15</sup>, ma si può tuttavia notare come la presenza della musica riporti prepotentemente il codice di programmazione nella dimensione del *continuum*.

Inoltre, una seconda condizione che spesso caratterizza il codice sorgente è il silenzio. Detto altrimenti, malgrado l'ubiquità del codice di programmazione nella vita quotidiana di tutti, la norma è che questo livello del software non si manifesti mai<sup>16</sup>. Di fatto, il programmatore e il codice che questi redige restano nascosti dietro la visibilità del risultato di un programma funzionante. In questo contesto, la manifestazione sensibile del livello del codice è solitamente il segnale di un problema, di un *bug*, di un malfunzionamento. L'intervento della dimensione musicale sradica il codice da questa condizione, prestandolo alla voce.

Infine, un'ultima caratteristica essenziale affinché il codice sorgente sia funzionale, è l'assenza di ambiguità:

A differenza delle lingue naturali come l'inglese o il cinese, i linguaggi informatici devono essere precisi. Sia la loro forma (sintassi) che il loro significato (semantica) devono essere specificati senza ambiguità, in modo che sia i programmatori che i computer possano capire che cosa deve fare un programma. (Scott, 2000: 31, traduzione mia)

Ciononostante, se l'azione di un programma dev'essere priva di indeterminatezza, l'effetto di questa azione non può che comportare delle implicazioni potenzialmente ambigue e poco trasparenti, e questo è un punto che sottolinea il secondo aspetto individuato più in alto come parte integrante del codice di programmazione, cioè l'espressività. Tanto è vero che è comunemente ammesso che, se esistono molteplici modi di scrivere uno stesso programma nello stesso linguaggio, una versione specifica di uno di questi modi corrisponde a uno stile di programmazione, cioè all'espressione personale dell'autore del codice<sup>17</sup>. Di

---

<sup>15</sup> Queste problematiche sono state toccate brevemente da Leslie Chan durante la sua lezione magistrale intitolata «Is Open Scholarship Possible Without Open Infrastructure?», tenutasi il 7 giugno 2022 al Digital Humanities Summer Institute (URL: <<https://dhsi.org/>>. Ultimo accesso: 7 giugno 2022).

<sup>16</sup> Anche qui si tratta di una generalizzazione, che non esclude l'esistenza di pratiche caratterizzate invece da un'esibizione estetica del codice, come ad esempio il *live coding*. Se questa attività nasce in ambito musicale, si sta ormai espandendo anche in ambito visivo ma, nonostante una diffusione sempre maggiore, è importante sottolineare che questo uso del codice rimane distinto dall'uso che ne fanno quotidianamente il programmatore da un lato e l'utente dall'altro.

<sup>17</sup> Si tratta di una caratteristica particolarmente evidente in alcune competizioni di programmazione, che si basano proprio sulla tensione tra la componente funzionale e la componente espressiva. Si veda ad esempio l'Underhanded C Contest o il IOCCC (The International Obfuscated C Code Contest). Il primo è una gara di programmazione online in cui i concorrenti mirano a creare codice che sembra svolgere una funzione, mentre ne svolge subdolamente un'altra; il secondo è una gara tra programmatori per scrivere il programma C dall'aspetto più complicato possibile, tenendo conto, tuttavia, che l'obiettivo non è scrivere programmi funzionalmente complicati, ma piuttosto che i programmi presentati siano semplici nelle loro funzionalità, ma

conseguenza, è possibile ascrivere a un programma, nella sua forma di testo ordinato e coerente redatto in un linguaggio di programmazione, l'espressione di una dimensione propriamente lirica<sup>18</sup>, che si manifesta nel complesso delle scelte che costituiscono l'impronta peculiare della personalità individuale dell'autore, o della comunità di autori<sup>19</sup>. Nell'ottica della messa in musica, a questa soggettività si sovrappongono svariate altre istanze: come si è detto nell'introduzione, all'espressione del programmatore si aggiungono quella del compositore, dell'interprete e del pubblico, la cui ricezione non è indifferente alla costituzione dell'opera stessa, come si è detto in precedenza, e forse in questo caso è attraente la prospettiva di considerare anche la soggettività del programma, visto che la sua funzione consiste nel dotare la macchina della capacità di parola.

Ma, ritornando alla questione principale, nel caso di *Hello, World*, ad esempio, la presenza del compositore è chiaramente percepibile già a partire dalla lettura del codice: se la lingua naturale da cui la maggior parte dei linguaggi di programmazione prende in prestito singoli vocaboli per esprimere comandi puntuali è l'inglese, nel brano di Campogrande i caratteri sono enunciati in italiano e non sempre allo stesso modo. Infatti, alla fine di «In B», il primo *Lied* del ciclo, il carattere «w» è letto come «double-u», mentre alla fine di «In Malbolge», è pronunciato come «doppia vu». Anche se è vero che, in questo caso, la prima «double-u» è il primo carattere nella parola *world*, separato dal resto del vocabolo a causa della limitazione a quattro caratteri per variabile nel linguaggio B, mentre la seconda, in Malbolge, non è che un carattere tra i tanti. Tuttavia, un caso più evidente si trova sempre nell'ultimo *Lied*, in cui la soprano canta «linea verticale di cui non conosco la definizione», dove la prima persona singolare esprime senza ombra di dubbio l'intrusione di un'individualità esterna al codice.

In questo primo momento, ho dunque descritto brevemente alcune caratteristiche distintive del codice di programmazione e alcune delle conseguenze che l'atto di musicare dei software produce su di esse. Prima di passare a una riflessione incentrata più precisamente su *Hello, World*, è doveroso notare che, sebbene le osservazioni avanzate fin qui si prestino a essere generalizzate, esse rimangono informate in larga parte dal brano di Campogrande,

---

difficili da leggere. Per una riflessione dettagliata su alcune proposte particolarmente interessanti nell'ambito delle due competizioni, si veda Berry (2011: 75-93).

<sup>18</sup> Per una discussione della nozione di lirismo, si veda Charles-Wurtz (1998: 35-41).

<sup>19</sup> La questione delle relazioni tra lo stile, sia questi singolare o collettivo, e l'espressione di un'individualità attraverso di esso sono estremamente complesse e dibattute. In questa sede, mi limito a segnalare la riflessione di Dessons (1995) e gli sviluppi in analisi del discorso legati all'introduzione della nozione di *ethos* discorsivo (Amossy, 1999; Amossy, 2010).

se non altro perché, a mia conoscenza, ad oggi il suo è l'unico lavoro in cui sia stato musicato del codice sorgente, benché non sia naturalmente l'unico possibile.

## 2. Hello, World

La prima scelta che si presenta davanti a un compositore in procinto di musicare un testo è la scelta del materiale testuale e musicale. Per quanto riguarda il primo, raramente la preferenza espressa per un testo corrisponde a una preferenza per un linguaggio. Certo, ogni testo è redatto in una lingua naturale specifica (o eventualmente più di una), ma è un caso relativamente poco frequente che vengano musicati nello stesso lavoro diverse traduzioni dello stesso testo. Nel caso di *Hello, World*, invece, la scelta del materiale è equivalsa a una scelta di linguaggio, ma non di linguaggi con la pretesa di esprimere tutti all'incirca la stessa cosa, com'è il caso delle traduzioni di un medesimo testo, bensì di linguaggi che, attraverso percorsi, lessici e grammatiche diverse, condividono un'unica teleologia.

Si pone dunque la questione delle differenze e delle somiglianze tra i linguaggi scelti da Campogrande, a partire dal fatto che lo stesso segno può assumere significati diversi in linguaggi diversi. Si è appena visto l'esempio di «w» nei *Lieder* «in B» e «in Malbolge». Al contrario, lo stesso simbolo mantiene lo stesso significato all'interno di uno stesso linguaggio, e il compositore rende esplicite queste corrispondenze in particolare nel *Lied* «In Delphi», applicando un trattamento equiparabile alle loro varie ripetizioni: è il caso del punto e virgola e dell'apostrofo.

Un ulteriore elemento oggetto di un evidente rapporto di equivalenza attraverso i linguaggi è la locuzione «hello, world», che ritorna nei due *Lieder* «In B» e «In Unix Shell». Come si è detto, tutti i programmi musicati in questo lavoro sortiscono lo stesso effetto, e istruiscono cioè la macchina a far comparire sul monitor la scritta «hello, world», ma vi arrivano secondo delle drammaturgie diverse, come si è espresso Campogrande durante l'evento di apertura del CodeFest. In ogni caso, questa costante fornisce un punto di riferimento. Infatti, se lo stesso segno assume significati diversi in linguaggi diversi, la semiosi di «hello, world» rimane sempre la stessa, indipendentemente dal linguaggio di programmazione. La locuzione compare per la prima volta al termine del primo *Lied*, «In B», benché in una forma frammentata, a causa del fatto che «il massimo numero di caratteri che una costante nel linguaggio B può contenere [è] 4» (Penge, 2021). Il programma si presenta dunque in questo modo:

```
main() {  
    extern a, b, c;  
    putchar(a); putchar(b); putchar(C); putchar('!*n');  
}  
  
a 'hell';  
b 'o, w';  
c 'orld';
```

In realtà, la frammentazione provoca una risemantizzazione seguita da una desemantizzazione. In primo luogo, invece di «hello» si legge «hell», un vocabolo dalla connotazione poco meno che opposta a quella di «hello», che Campogrande musica quasi come un'imprecazione, in *fortissimo* e con l'indicazione «drammatico». E in secondo luogo, la parola *world*, divisa tra due variabili, perde di immediatezza, benché il segmento «-orld» sia identificabile senza difficoltà dall'ascoltatore avvertito del titolo della composizione. Inoltre, lo statuto per così dire anomalo di questa unità lessicale è sottolineato da Campogrande in questa istanza con un ampio vocalizzo, mentre fino a questo momento il canto è rimasto in un regime più vicino al recitativo, con qualche eccezione, è vero, ma di portata decisamente minore.

La locuzione ritorna nel brano «In Unix Shell», dove la sua eccezionalità è chiaramente percepibile: oltre a un'accelerazione e all'indicazione «allegro», la voce apre le virgolette che precedono l'intonazione di «Hello world» nella tonalità di Do maggiore, tradizionalmente la più semplice ma anche la più solare, oltre che la dominante di Fa minore, su cui si apre il *Lied*, e con un movimento di sesta ascendente, ripreso in canone dall'oboe solo, dal flauto solo e infine dal clarinetto solo. Anche in questo caso, la voce intona un ampio vocalizzo, reminiscenze della sua controparte nel brano precedente.

Continuando un ipotetico percorso lungo le tappe del processo compositivo, non bisogna dimenticare che non sono i linguaggi a essere musicati, bensì i programmi e si è reso necessario per il compositore introdurre nella sua pratica una fase che d'abitudine non è presente quando si tratta di musicare un testo. Infatti, quando si ha a che fare con un oggetto espresso nei linguaggi naturali, è possibile passare direttamente dalla lettura alla composizione, mentre nel caso del codice sorgente, Campogrande ha introdotto una fase intermedia consistente nel rendere il testo pronunciabile, ancora prima che cantabile. Si tratta di un procedimento inverso a certe pratiche delle seconde avanguardie – si pensi ad esempio alla *Sinfonia* di Luciano Berio, dove i testi musicati diventano spesso inintelligibili a favore della

dimensione puramente sonora e pre-articolatoria dei fonemi. Inoltre, l'operazione che il compositore compie sul codice di *Hello, World* risulta in un fenomeno bizzarro, per cui la voce che canta, e che enuncia ogni segno, si ritrova in un esercizio di dettatura, cioè l'unica attività che prevede l'enunciazione esplicita di tutti i segni che compongono un testo, a partire dalla punteggiatura. Si tratta evidentemente di un'anomalia rispetto alla prassi corrente di composizione sulla base di un testo, sebbene sia possibile avanzare l'osservazione che i segni di interpunzione, benché non vengano enunciati in quanto tali, sono comunque musicati, nel senso che, ad esempio, una frase che termina con un punto interrogativo non sarà compresa prima e musicata poi nello stesso modo di una frase che termina con un punto esclamativo.

Dopodiché, una volta reso pronunciabile il testo dei quattro programmi scelti, la fase di lettura e di riflessione su di essi non ha assunto dei tratti tecnici e specialistici, bensì si è basata sulla ricerca delle dinamiche espressive ed emotive ascrivibili agli artefatti testuali selezionati. Ad esempio, proprio all'inizio del ciclo, la voce intona in modo «dubbioso», come si legge in partitura, «aperta parentesi chiusa parentesi», per poi proseguire «entusiasta» con «aperta parentesi graffa», come se una prima incertezza esternata dall'apertura di una parentesi che viene immediatamente chiusa si risolvesse nella scelta di una parentesi di altro tipo. Allo stesso modo, le due chiusure di parentesi che seguono sono accompagnate dall'indicazione «improvvisamente triste». Uno svolgimento analogo è apprezzabile nei madrigalismi che punteggiano la composizione: è il caso della raffigurazione dell'eco che precede l'arrivo trionfale di «hello, world» nel *Lied* «In Unix Shell» proprio in corrispondenza della parola *echo*, quando la voce, doppiata dai clarinetti, dalle trombe, e dai violoncelli, ribatte la stessa nota man mano che il ritmo aumenta proporzionalmente alla dinamica che diminuisce. Alle volte l'illustrazione di questo tipo di suggestioni sconfinava nell'onomatopea: sempre nello stesso *Lied*, infatti, la parola *slash* è intonata in maniera secca e tagliente dalla soprano, a cui fanno eco i contrabbassi con un pizzicato alla Bartok dal timbro quasi percussivo.

Infine, un terzo momento di questa traiettoria è la scelta dell'organico strumentale, in questo caso della voce accompagnata dall'orchestra. Si tratta di una decisione che va di pari passo con la volontà di descrivere questo lavoro come un ciclo di *Lieder*, e cioè di iscriverlo nella tradizione dei grandi cicli schubertiani prima e dei cicli per voce e orchestra di Gustav Mahler dopo. Non solo, il *Lied* è un genere che rivendica la propria pretesa interpretativa; altrimenti detto, un *Lied* non è affatto un testo cantato su un accompagnamento musicale che gli è largamente indifferente, ma, al contrario, si tratta di un lavoro di

interpretazione e rielaborazione certosino e attento al dettaglio, che tutto è tranne che indipendente dal testo da cui prende le mosse. Ed è proprio questo il motivo delle tensioni tra i due media individuate già nell'introduzione – tensioni che sono presenti in *Hello, World*. Questo ciclo, pertanto, nonostante l'insolito materiale testuale, ribadisce la propria dignità nel contesto di una tradizione illustre e di lunga data, sottolineando la serietà del progetto. Tuttavia, la compostezza dell'opera non impedisce dei momenti di ironia, sottolineata da Elisabetta Fava durante *Singing the Code*, in apertura del CodeFest. L'ironia è una condizione, sia detto *en passant*, preclusa alla macchina, ed è presente già a partire dalle battute di apertura che, nonostante siano state annunciate «In B», attaccano con un accordo di La maggiore.

Così, prestando un tale respiro storico alla propria composizione, Campogrande fa del codice sorgente un materiale esemplare per riflettere sulla pratica ben più diffusa di messa in musica di testi in linguaggio naturale: al di là delle diverse specificità linguistiche, storiche, geografiche e sociali che variano in funzione del corpus preso in considerazione, che si tratti di *Lieder* in lingua tedesca, di *mélodies* in lingua francese, o di *songs* in lingua inglese, è interessante interrogarsi sulla misura in cui le riflessioni avanzate a proposito di *Hello, World* si prestino, o non si prestino, a una possibile generalizzazione a questi repertori. Infatti, è necessario sottolineare che, se una tale generalizzazione è concepibile, essa lo è in primo luogo perché i repertori appena citati sono radicati in una tradizione lirica e, in secondo luogo, perché i testi scelti per le trasposizioni musicali sono stati scritti per la maggior parte in maniera indipendente da qualsiasi possibile elaborazione da parte di un compositore. Ora, da un lato, nonostante il codice sorgente non sia stato ideato nell'ottica di una comunicazione di tipo lirico, si è visto che non ne è incapace e che, anzi, la dimensione propriamente espressiva dei linguaggi di programmazione non è da sottovalutare, anche dal punto di vista estetico. E dall'altro, a mia conoscenza, non esistono software scritti con l'intenzione di fornirne il codice a un compositore perché questi lo musicchi. Inoltre, è altrettanto importante mettere in evidenza che quest'ultimo elemento, la condizione di un testo redatto senza guardare a un'eventuale trasposizione musicale, esclude da queste considerazioni la maggior parte delle opere liriche, il cui libretto è di solito concepito in relazione alla scrittura musicale, per non parlare del fatto che le opere sono messe in scena e che questa caratteristica richiede approcci molto diversi rispetto a quelli applicabili alla musica solitamente non accompagnata da un'ambientazione teatrale.



Non solo, l'approccio di *Hello, World* al codice di programmazione interroga in uguale misura le pratiche correnti di lettura del codice stesso, il quale viene raramente considerato come un artefatto testuale provvisto di una dimensione estetica e degno di una stima equiparabile a testi poetici o letterari redatti nei linguaggi naturali. Infatti, come evidenzia Marino (2020: 38, traduzione mia), «lo studio del codice informatico non pone attualmente l'accento sull'interpretazione, sulla ricerca e sulla produzione di significato», poiché queste problematiche sono considerate come l'appannaggio delle scienze umane e delle arti liberali, e non dell'informatica. Ma l'appropriazione del codice sorgente da parte della composizione musicale e lo studio delle modalità e delle conseguenze di questa appropriazione dimostrano invece che il codice di programmazione non è la prerogativa esclusiva dei dipartimenti di informatica e che, al contrario, «ci sono questioni di espressione (connotazioni, implicazioni, risonanze), di storia sociale e materiale e di ideologia» (Marino, 2020: 38) che tutte contribuiscono a fare del codice sorgente un oggetto degno di considerazione in ambito prettamente umanistico. Soprattutto,

il codice [...] non esiste solo per i computer, che potrebbero operare solo [...] su segnali elettronici, ma anche per i programmatori. Pertanto, il computer può essere uno dei destinatari del codice, ma ci sono anche il programmatore, altri programmatori, manager e talvolta persino utenti che hanno accesso al suo testo. In effetti, il pubblico del codice si è ampliato: giudici e avvocati, politici e opinionisti e persino poeti includono il codice nelle loro prese di parola. (Marino, 2020: 38)

E a questa lista si possono ora aggiungere i compositori, i musicologi, i direttori d'orchestra, i cantanti, gli strumentisti e il pubblico in sala da concerto.

Infine, dunque, la condizione di ibridità che non può che caratterizzare la fusione di due mezzi espressivi eterogenei e discordanti si rivela feconda non soltanto per la riflessione teorica a cui dà luogo, ma anche per le pratiche che accomuna in un'opera artistica inedita – una musica *a programma*, non nel senso di Berlioz, ma nel senso di Campogrande.

## **Bibliografia**

- AMOSSY, Ruth (1999), *Images de soi dans le discours. La construction de l'ethos*, Lonay, Delachaux et Niestlé.
- AMOSSY, Ruth (2010). *La Présentation de soi: ethos et identité verbale*, Parigi, Presses universitaires de France.
- AUSTIN John Langshaw (1975), *How to Do Things with Words*, Cambridge, MA, Harvard University Press.
- BARRAL Céline, SERETTI Marina, PAPLOMATA Katerina (eds.) (2021), *La Portée musicale du poème. Des paradoxes de l'autonomie à la communauté sonore*, Sampzon, Editions Delatour.

- BERRY David M. (2011), *The Philosophy of Software. Code and Mediation in the Digital Age*, New York, Palgrave Macmillan.
- CAMPOGRANDE Nicola (2021), *Hello, World*, Lipsia, Breitkopf & Härtel.
- CHARLES-WURTZ Ludmila (1998), *Poétique du sujet lyrique dans l'oeuvre de Victor Hugo*, Parigi, Honoré Champion.
- COSTE Daniel (2002), «Des usages pragmatiques à l'action et de l'acquisition à l'appropriation», in CICUREL Francine, VERONIQUE Daniel (eds.), *Discours, action et appropriation des langues*, Parigi, Presses Sorbonne Nouvelle.
- COX Geoff, MCLEAN Alex (2013), *Speaking Code: Coding as Aesthetic and Political Expression*, Cambridge, MA, M.I.T. Press.
- DESSONS, Gérard (1995), «Les Enjeux de la manière», *Langages*, n. 118/29, pp. 53-63.
- GAZONI, Ricardo Maciel (2018). «A Semiotic Analysis of Programming Languages», *Journal of Computer and Communication*, n. 6, pp. 91-101.
- FISH Stanley (1980), *Is There a Text in This Class? The Authority of Interpretive Communities*, Cambridge, MA, Harvard University Press.
- IMBERTY Michel (1992), «De quelques problèmes méthodologiques dans l'approche psychologique expérimentale de l'interprétation musicale», in DALMONTE Rossana, BARONI Mario (eds.), *Atti del Secondo Convegno Europeo di Analisi Musicale*, Trento, 24-27 ottobre 1991, Università degli Studi di Trento, Dipartimento di Storia della Civiltà Europea.
- ISER Wolfgang (1994), *Der Akt des Lesens*, Stuttgart, Utb für Wissenschaft.
- JAKOBSON Roman (1960). «Closing Statement: Linguistics and Poetics», in SEBEOK Thomas Albert (ed.), *Style in Language*, New York, M.I.T. Press, pp. 350-377.
- JAUB Hans Robert (1982). *Ästhetische Erfahrung und literarische Hermeneutik*, Frankfurt am Main, Suhrkamp Verlag.
- MARINO Mark C. (2014), «Field Report for Critical Code Studies, 2014», *Computational Culture. A Journal of Software Studies*, n. 4, URL: <<http://computationalculture.net/field-report-for-critical-code-studies-2014%e2%80%a8/>>. Ultimo accesso: 21 aprile 2022.
- MARINO Mark C. (2020), *Critical Code Studies*, Cambridge, MA, M.I.T. Press.
- PENGE Stefano (2021), «Hello, World!», nel blog *Digi Tales. Apprendimento digitale e dintorni*, URL: <<https://www.stefanopenge.it/wp/hello-world/>>. Ultimo accesso: 6 maggio 2022.
- SCOTT Michael Lee (2000), *Programming Language Pragmatics*, San Francisco, CA, Morgan Kaufmann Publishers.
- TANAKA-ISHII, Kumiko (2006), «Dyadic versus Triadic Sign Models in Functional and Object-Oriented Computer Programming Paradigms», *Semiotica*, n. 1-4/158, pp. 213-231.
- TANAKA-ISHII, Kumiko (2010), *Semiotics of Programming*, Cambridge, Cambridge University Press.
- TEMKIN Daniel (2017), «Language without Code: Intentionally Unusable, Uncomputable, or Conceptual Programming Languages», *Journal of Science and Technology of the Arts*, n. 3/9, pp. 83-91.

### **Come citare l'articolo:**

Lucia Pasini, «Hello, world. Musica a programma per il terzo millennio», *InterArtes* [online], n.2 "Ibrido" (Laura Brignoli, Silvia Zangrandi eds.), novembre 2022, pp. 43-58. <[https://www.iulm.it/wps/wcm/connect/iulm/99b027b3-8662-4141-bb8f-32bf996f35a4/03+vPasini\\_Hello+world+def.pdf?MOD=AJPERES](https://www.iulm.it/wps/wcm/connect/iulm/99b027b3-8662-4141-bb8f-32bf996f35a4/03+vPasini_Hello+world+def.pdf?MOD=AJPERES)>