

# TikZ vs curve2e

## Confronto sul disegno di un logo

Claudio Beccari, Gianluca Pignalberi

### Sommario

Quale pacchetto conviene usare tra `curve2e` e `TikZ` quando il compito è disegnare un semplice logo alfabetico?

### Abstract

Which package is more convenient to use when the task is to draw a simple, alphabetic logo: `curve2e` or `TikZ`?

### La sfida

L'intera tiratura di *ArXiv* del 2019, consegnata in occasione del *CTeX meeting 2019*, ci è stata regalata da una casa editrice universitaria torinese: la CLUT — Cooperativa Libreria Universitaria di Torino ([www.clut.it](http://www.clut.it)). Tale casa editrice, avente sede all'interno del Politecnico di Torino, è nota tra gli addetti ai lavori per essere una delle poche italiane a preferire la ricezione dei manoscritti in  $\LaTeX$ . La sua classe di composizione è opera di Claudio Beccari.

Naturalmente avevamo bisogno di un loro logo da stampare sulla rivista. Ne abbiamo avuti due: uno vettoriale coi soli contorni delle lettere (opera di Claudio Beccari) e uno raster con le lettere colorate, ma troppo piccolo. Visto lo scarso tempo per andare in stampa, la redazione di *ArXiv* ha realizzato una versione vettoriale, imprecisa ma passabile, con Inkscape: ogni lettera è stata disegnata unendo una serie di rettangoli e semicerchi.

Quest'anno la CLUT ha affidato a Gianluca Pignalberi e a Massimiliano Dominici la composizione del libro BARBARA BONELLI, *Appunti di chimica* — Terza edizione e, a tal proposito, i compositori hanno deciso di realizzare un logo “filologicamente corretto” con l'aiuto di uno dei pacchetti di  $\LaTeX$  che lo permettono. Ogni lettera del logo è contenuta in un quadrato di lato 3,4 cm e ognuna è distanziata dall'altra di 5 mm. La scelta è caduta su `TikZ` (TANTAU, 2020).

Questo è il codice (le righe di ogni tracciato sono state qui suddivise su più righe solo per motivi di spazio):

```
\begin{tikzpicture}
\draw [fill] (1.7,0)%
arc [start angle=270, end angle=90,%
radius=1.7] -- ++(1.7,0)%
```

```
-- ++(0,-1.5) -- ++(-1.7,0)%
arc [start angle=90, end angle=270,%
radius=.2] -- ++(1.7,0)%
-- ++(0,-1.5) -- cycle; % C
\draw [fill,xshift=3.9cm] (0,0)%
-- ++(0,3.4) -- ++(1.5,0)%
-- ++(0,-1.9) -- ++(1.9,0)%
-- ++(0,-1.5) -- cycle; % L
\draw [fill,xshift=7.8cm] (0,1.7)%
-- ++(0,1.7) -- ++(1.5,0)%
-- ++(0,-1.7)%
arc [start angle=180, end angle=360,%
radius=.2] -- ++(0,1.7) -- ++(1.5,0)%
-- ++(0,-1.7) arc [start angle=0,%
end angle=-180,radius=1.7]; % U
\draw [fill,xshift=11.7cm] (.95,0)%
-- ++(0,1.9) -- ++(-.95,0)%
-- ++(0,1.5) -- ++(3.4,0)%
-- ++(0,-1.5) -- ++(-.95,0)%
-- ++(0,-1.9) -- cycle; % T
\end{tikzpicture}
```

Il metodo usato è banale: ogni lettera è disegnata in un quadrato con lato di 3,4 cm e con l'angolo in basso a sinistra posto nell'origine e poi traslata orizzontalmente alla distanza giusta. Il disegno è un banale inseguimento in senso orario del contorno con la prima coordinata assoluta (posta il più possibile vicino all'origine) e tutte le altre relative — lo stesso metodo del linguaggio Logo (LOGO FOUNDATION, 2020) — e il poligono mistilineo così creato viene poi riempito di nero.

Informato dell'intenzione di creare il logo con `TikZ`, Claudio Beccari ne ha realizzata una versione disegnata con `curve2e`, (BECCARI, 2020), il cui codice è il seguente (anche in questo caso le righe troppo lunghe sono suddivise su più righe per la stampa):

```
\newcommand\CLUTlogo[2][nero]{%
\bggroup
\def\tempA{#2}\def\tempB{nero}%
\def\tempC{#1}%
\setbox0\hbox{C}\edef\CLUTsize{%
\ifx\tempA\empty\ht0\else#2\fi}%
\unitlength=\dimexpr\CLUTsize/34\relax
\picture(154,34)
\unless\ifx\tempB\tempC\color{#1}\fi
\buttcap \linethickness{15\unitlength}
\Curve(34,27.5)<-1,0>(17.5,27.5)<-1,0>%
(17.5,7.5)<1,0>(34,7.5)<1,0>
\put(40,0){\polyline[\miterjoin](7.5,34)%
```

TABELLA 1: Confronto di quattro proprietà dei documenti prodotti.

Misura	TikZ (contorno)	TikZ (“scrittura”)	curve2e
Lunghezza <sup>a</sup> (byte)	785	477	699
Numero di righe <sup>b</sup>	4	4	7
Dimensione PDF (byte)	1323	1225	1086
Compilazione (s)	0,369	0,229	0,196
Compilazione <sup>c</sup> (s)	0,204	0,200	0,180

<sup>a</sup> Dell’intero documento standalone.

<sup>b</sup> Del solo codice negli ambienti (tikz)picture, escluse le definizioni di comando e di apertura e chiusura ambiente.

<sup>c</sup> Esclusa la prima compilazione.

```
(7.5,7.5)(34,7.5)}
\put(80,0){%
  \Curve(7.5,34)<0,-1>(7.5,17.5)<0,-1>%
  (27.5,17.5)<0,1>(27.5,34)<0,1>}
\put(120,0){\segment(0,27.5)(34,27.5)
  \segment(17,0)(17,34)}
\endpicture
\egroup}
```

Il codice realizzato consente di specificare la grandezza e il colore del logo; per ottenere il logo delle stesse dimensioni di quello ottenuto con TikZ, basta specificare `\CLUTlogo{34mm}`; se non si specifica nessuna dimensione (`\CLUTlogo{}`), il logo ha la stessa altezza delle maiuscole nel font corrente. A parte questa variante, il codice vero e proprio per disegnare il logo è compreso fra `\picture` e `\endpicture`. Esso si basa sul tracciamento di spezzate mistilinee di spessore pari a 15 unità di disegno.

Ispirato dalla soluzione proposta da Beccari e come suggerito da un anonimo *referee*, Pignalberi ha realizzato una seconda versione TikZ in cui i tracciati sono disegnati esattamente come si scriverebbero le singole lettere. Anche l’ordine e la direzione ricalcano le sue mosse quando scrive. Ecco il nuovo codice:

```
\begin{tikzpicture} [line width=1.5cm]
\draw (3.4,2.65) -- ++(-1.7,0)%
arc [start angle=90,end angle=270,%
radius=.95] -- ++(1.7,0); % C
\draw [xshift=3.9cm] (.75,3.4)%
-- ++(0,-2.65) -- ++(2.65,0); % L
\draw [xshift=7.8cm] (.75,3.4)%
-- ++(0,-1.7) arc [start angle=180,%
end angle=360,radius=.95]%
-- ++(0,1.7); % U
\draw [xshift=11.7cm] (1.7,3.4)%
-- ++(0,-3.4) (0,2.65) -- ++(3.4,0); % T
\end{tikzpicture}
```

Il risultato dei tre disegni, visualmente identico, è mostrato nella figura 1.

A un’analisi “visuale”, la prima versione TikZ è molto prolissa; il codice di Beccari è senz’altro



FIGURA 1: Logo della casa editrice CLUT realizzato sia con TikZ, sia con curve2e.

TABELLA 2: Tempi rilevati per le compilazioni dei logo. Da notare il valore più elevato delle prime compilazioni, quando per la prima volta si carica l’intero codice (anche di `time` nella memoria cache nel caso della prima compilazione).

TikZ (c.)	TikZ (s.)	curve2e
1,861	0,493	0,337
0,205	0,197	0,179
0,201	0,203	0,179
0,206	0,200	0,181
0,204	0,199	0,181
0,202	0,202	0,181
0,202	0,201	0,179
0,202	0,197	0,183
0,204	0,199	0,179
0,207	0,199	0,178

più compatto e ancor più stringata è la seconda versione TikZ. La sintassi di curve2e sembra meno comprensibile di quella di TikZ agli occhi di chi non conosca le basi dei linguaggi dei due pacchetti esaminati ma TikZ richiede che sia specificato se un percorso sarà rettilineo o meno. Ovviamente il codice di Beccari ha anche il vantaggio di essere “parametrico”, cioè di permettere la variazione delle dimensioni e del colore, cosa fattibile anche in TikZ grazie alle funzioni matematiche offerte da PGF. Nonostante le differenze progettuali, abbiamo proceduto a una comparazione più significativa e riportato nella tabella 1 il risultato di alcune misurazioni oggettive. Tali misurazioni riguardano la lunghezza del codice (ogni ambiente di disegno è stato incluso in un documento standalone identico), il suo numero di righe, la dimensione del PDF generato e la velocità di compilazione (media di dieci compilazioni con PDFLATEX cronometrate con `time` di `bash`; i singoli valori rilevati sono riportati nella tabella 2).

Alla fine del confronto, possiamo affermare che `curve2e` è preferibile rispetto a `TikZ` (il cui codice pure riesce a essere piú compatto) almeno nei casi come quello oggetto della sfida, in cui non si debba fare ricorso a strumenti esclusivi di `TikZ` e a maggior ragione se tali disegni sono molti all'interno del documento. Forse imparare il linguaggio di `picture`, il “capostipite” di `pict2e` di cui `curve2e` è un'estensione, non è banale, ma nella sua apparente semplicità la sintassi di `TikZ` non è meno facile.

## Ringraziamenti

Gli autori della comunicazione ringraziano la CLUT nella persona dell'architetto Michele Ruffino per l'amichevole concessione a usare il logo di proprietà della casa editrice. Ringraziano inoltre gli anonimi *referee* per le puntuali osservazioni e suggerimenti che hanno migliorato la qualità dell'articolo e l'uniformità dei risultati mostrati.

## Riferimenti bibliografici

BECCARI, Claudio (2020). «The `curve2e` manual». G<sub>T</sub>T. Leggibile col comando `texdoc curve2e-manual`.

LOGO FOUNDATION (2020). «Logo programming language». [https://el.media.mit.edu/logo-foundation/what\\_is\\_logo/logo\\_programming.html](https://el.media.mit.edu/logo-foundation/what_is_logo/logo_programming.html). Ultima visita 16-5-2020.

TANTAU, Till (2020). *TikZ & PGF Manual for Version 3.1.5b*. Leggibile col comando `texdoc tikz`.

- ▷ Claudio Beccari  
claudio dot beccari at gmail dot com
- ▷ Gianluca Pignalberi  
g dot pignalberi at gmail dot com