

marcel duchamp, a Trojan Horse, even is a Trojan horse with virus-like characteristics all aimed towards producing what appears to be a practical joke. It is a machine expression of practical joking, which was a personal characteristic of Marcel Duchamp, and sometimes a characteristic of his art.

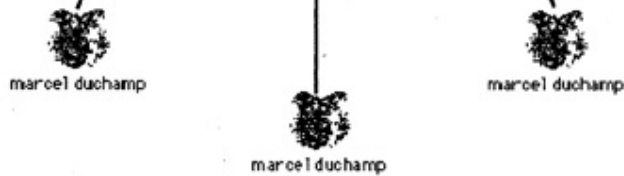
The icon for this software artwork is an excellent example of Duchamp's love of jokes. Based on a photographic portrait done by Man Ray, the icon is a scanned image of Duchamp, whose face and hair are covered with shaving cream.



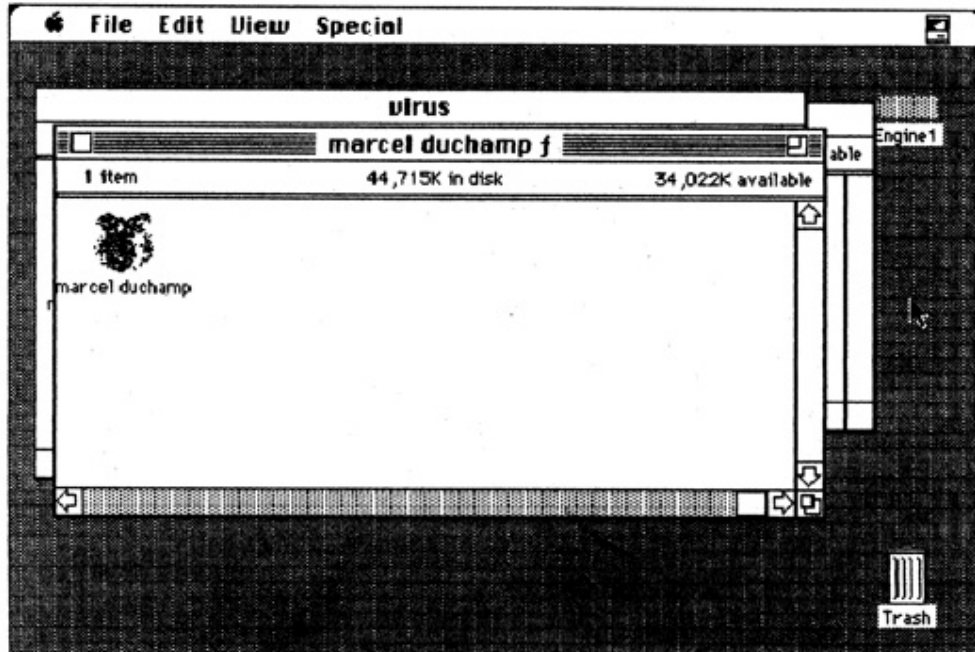
in his Monte Carlo Bond the bond to investors as a way cracking scheme — an with games and chance. The yet it is recognizable.

This photo was used by Duchamp in his Monte Carlo Bond lithograph of 1924. Duchamp sold the bond to investors as a way of raising capital for his blackjack interesting example of his fascination icon is much reduced and detail is lost,

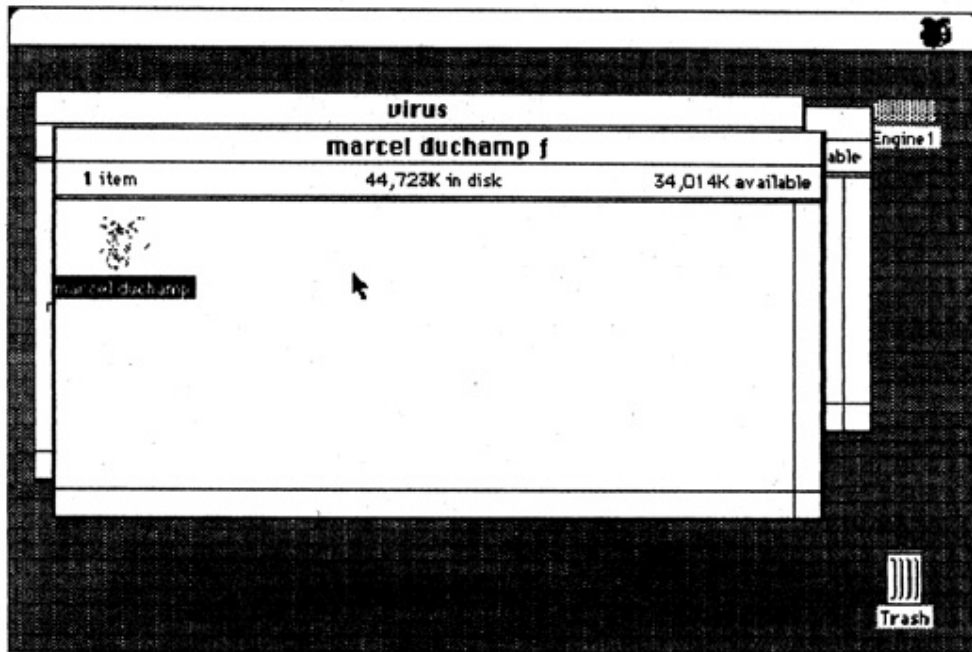
in his Monte Carlo Bond the bond to investors as a way cracking scheme — an with games and chance. The yet it is recognizable.



The Trojan Horse comes with its own folder, *marcel duchamp f*. When the folder is opened, the desktop of the Macintosh appears this way:

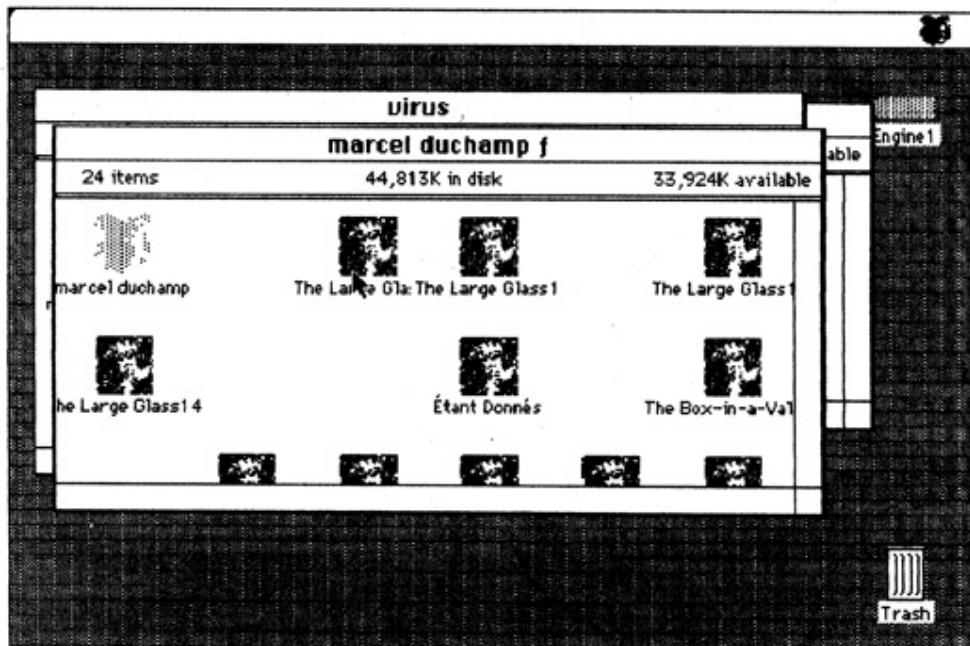


When the Trojan Horse is started with a double-click,...



... it can be seen that this horse is not an application, but an application fragment. There is no menubar, so `marcel duchamp...` can't be quit. There are no windows, as in a typical Macintosh application, just `Front Window(s)` and `marcel duchamp...` runs its course.

Icons begin appearing. Sometimes there are ten as in this instance.



Then they stop accumulating, but continue moving around, as replacements with the same name are generated.

When the run ended, if you examined the entire *marcel duchamp f* and its contents, you would discover a mosaic like this:



marcel duchamp... is quite random. Another sequence might begin with three icons all named THE LARGE GLASSx like this..

**Why are there so many Large Glasses?**

During his lifetime, Duchamp authorized, either before or after the fact, one replica of the LARGE GLASS made by Ulf Linde, and one by Richard Hamilton. Additionally, he made, or supervised making LARGE GLASS miniature replicas for The Box-in-a-valise. The Box-in-a-valise edition numbered 300 copies, so the icons named The Large Glass<sup>0</sup> through The Large Glass<sup>23</sup> are pointers to a potential infinity of copies of this seminal work.

**Why are all the icons of the generated files of *étant Donnés*, when all but one have names that refer to other works?**

The standard way to view Duchamp's art is to view it all of a piece. As Arturo Schwarz remarks:

Duchamp's opus is a single continuum, each item being a page of a work in progress. ...  
The full impact of each work can be appreciated only if it is seen in its proper chronological and thematic order.  
text for MULTIPLES, *Neues Beiliner Kunstverein*, May 8-June 15, 1974, p. 47.

The icon for the generated files is not actually from *étant Donnés*. Rather it is a scanned version of Given the Illuminating gas and the waterfall (preparatory study) from the years 1948-49. It is pointing to *étant Donnés* and it is alluding specifically to the fixed point perspective this diorama demands of the viewer. Instead of considering Duchamp's works chronologically, view them experientially. First, the powerful and actual visual experience of what is, then the great abstractions of The Large Glass. In this reverse chronology, you, the viewer, are glued to the peephole, seeing another man's experience of time. As an astronomer searching the heavens, the more you can see and identify, the further back in time you are going.





```

(else
  (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
  (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
  (write-file "Étant Donnés" 0 (i) "SALT" "TEXT"))
(if (even? (random 2))
  (begin
    (write-file "The Green Box" 0 (i) "SALT" "TEXT")
    (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
    (write-file (string-append "The Large Glass"
                               (number->string (random 24)))
              0 (i) "SALT" "TEXT"))
  (if (even? (random 2))
    (begin
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT"))
    (begin
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "The White Box" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT")
      (write-file "Étant Donnés" 0 (i) "SALT" "TEXT"))
    (if (even? (random 2))
      (begin
        (write-file (string-append "The Large Glass"
                                   (number->string (random 24)))
                  0 (i) "SALT" "TEXT")
        (write-file "The Green Box" 0 (i) "SALT" "TEXT")
        (write-file (string-append "The Large Glass"
                                   (number->string (random 24)))
                  0 (i) "SALT" "TEXT")
        (write-file "The Box-in-a-Valise" 0 (i) "SALT" "TEXT")
        (write-file (string-append "The Large Glass"
                                   (number->string (random 24)))
                  0 (i) "SALT" "TEXT"))
      (set! a (+ a 1)))
    (exit))))

```

Otherwise, do these.

Do this if (random 2) produces an odd number.

Do these if (random 2) produces an odd number.

Exit from the application-builder when compiling, or from the application when conditions of the while command are fulfilled.

Add one to the counter and loop back to while.