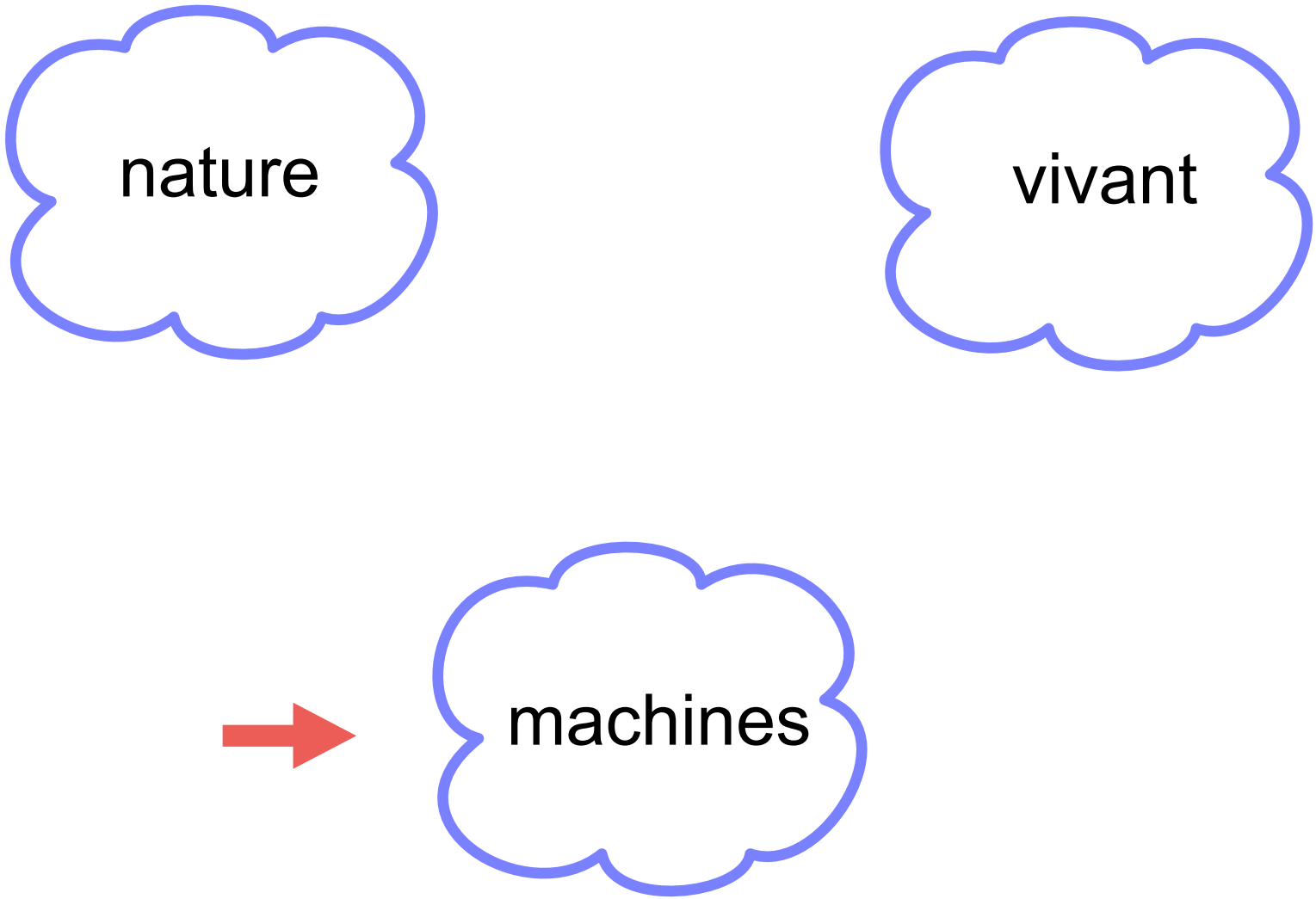
Informatique

Jean-Jacques Lévy
INRIA

SCIENCE ET RÉALITÉ

La Science



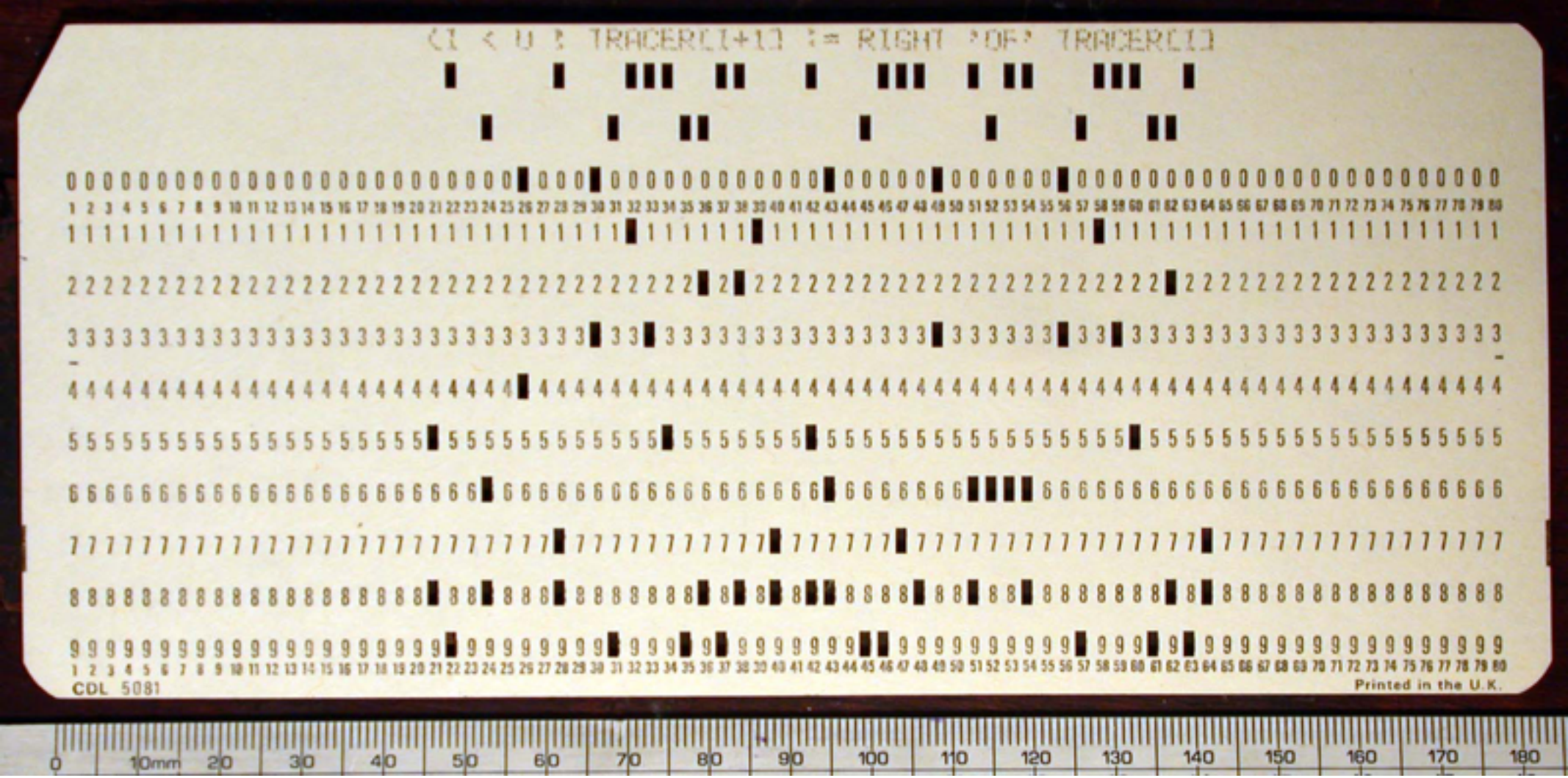
DES

MACHINES

(1945)

1 Méga-octets = 2 livres

1970



80 caractères

1 ligne d’un livre

512000 livres

150000 livres

6000000 livres

150000 livres

2015

Apple II

Lisa

Apollo

Sun 1

Macintosh

PerQ

Blit 5620

1978

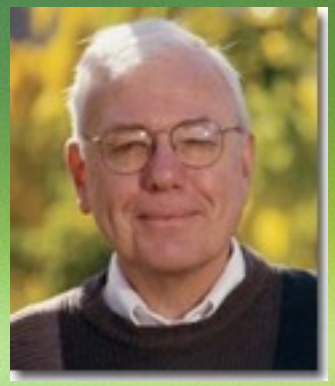
1976

Le garage

- intel 4004
- Xerox PARC (alto, dorado)
- le garage Apple (apple II, lisa, macintosh)
- IBM PC (ms-dos)

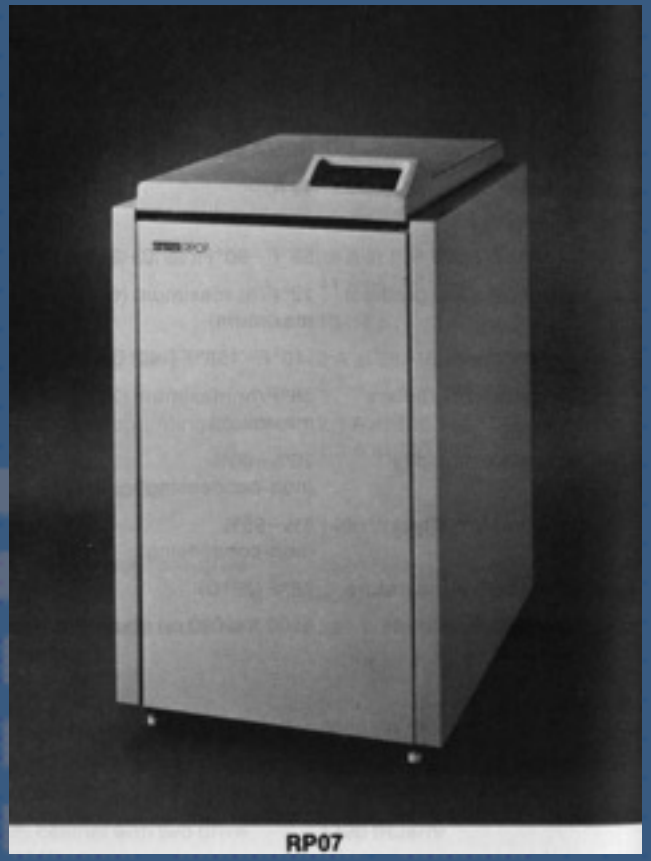
Alto

- vision égoiste
- tout le monde a son ordinateur
- seul l'interface compte



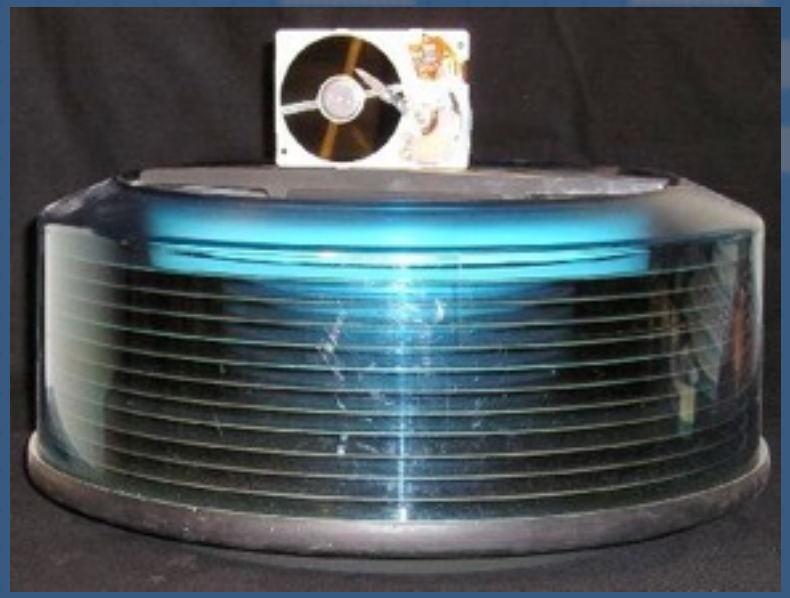
Chuck Thacker

vax 11/750

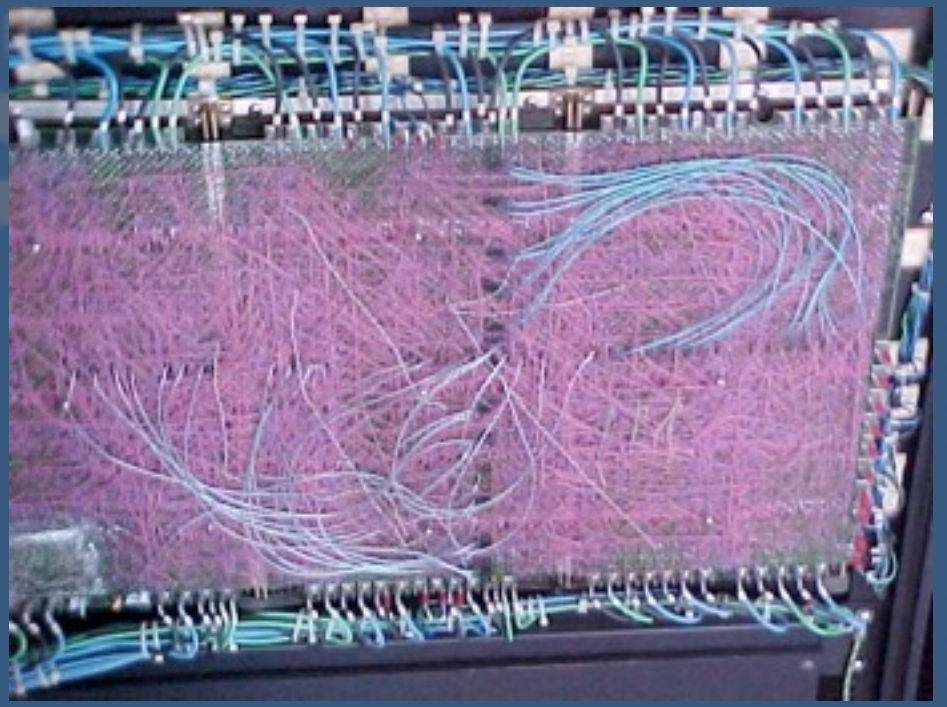


RP07
(700MO)

1975



RM05 (256 MO)



arrière
d'un
dec 10

1975

vax 11/780

1970

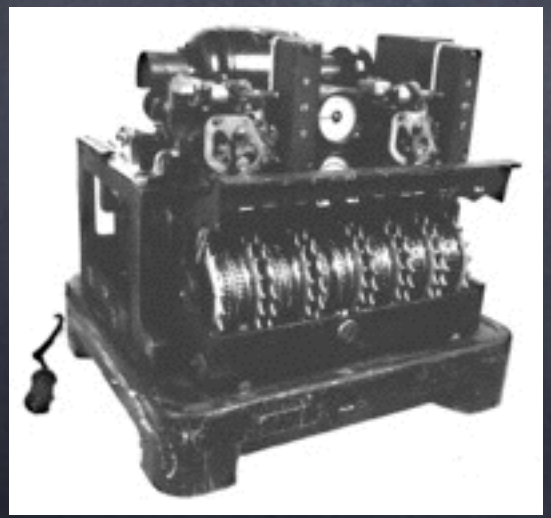
Multics

- temps partagé
- 10 à 100 utilisateurs / ordinateur
- courrier électronique

- IBM 704, 360/370
- SDS 940, Butler Lampson
- GE 645, Multics; MIT, Bull

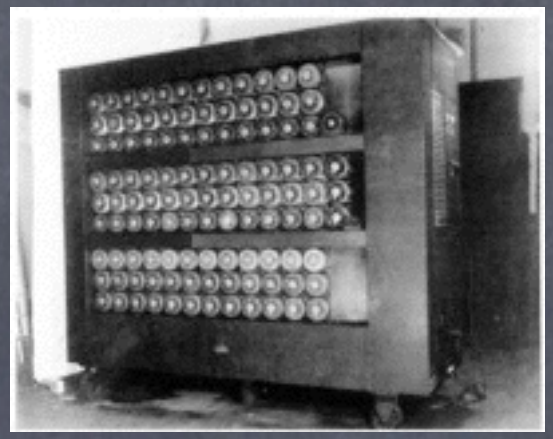
Allemagne

Enigma



Lorenz

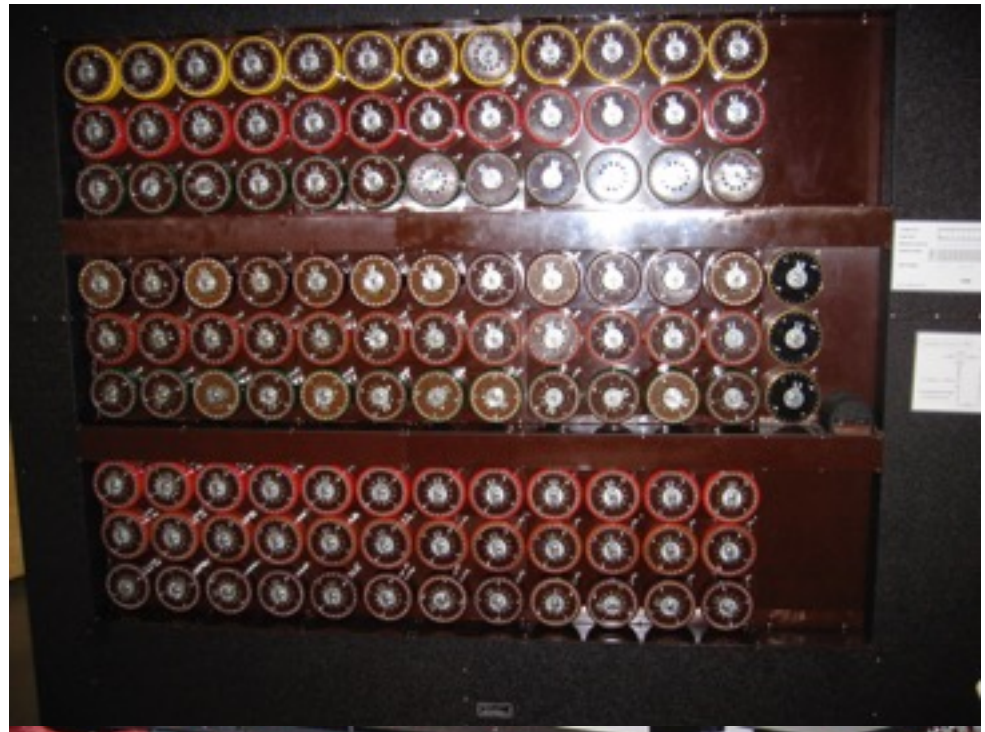
1940

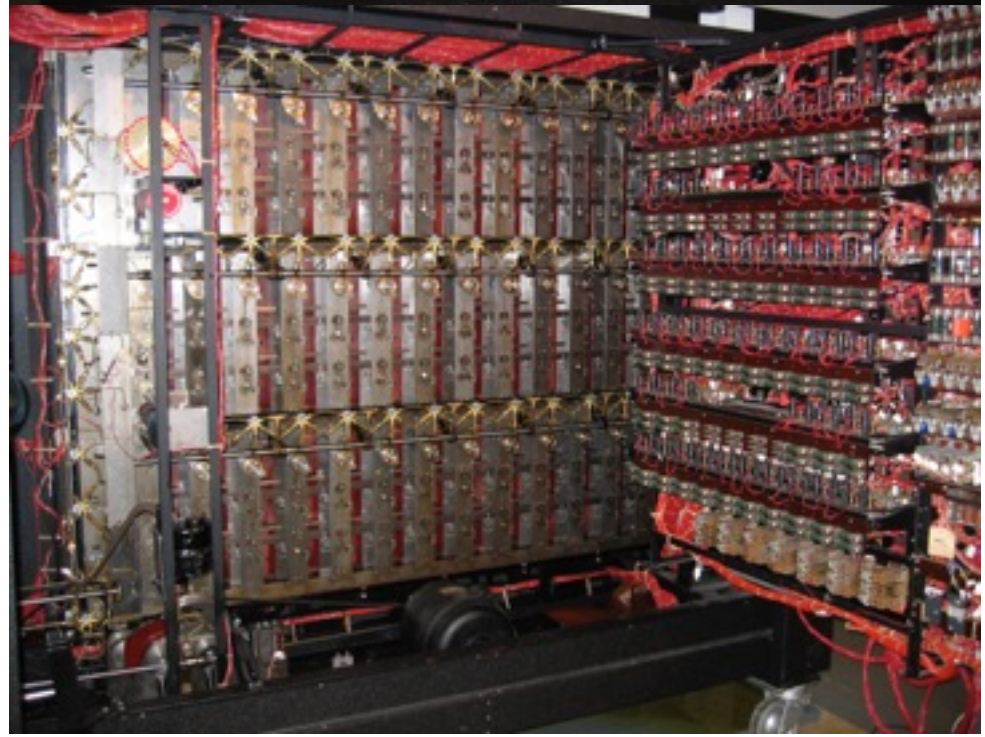


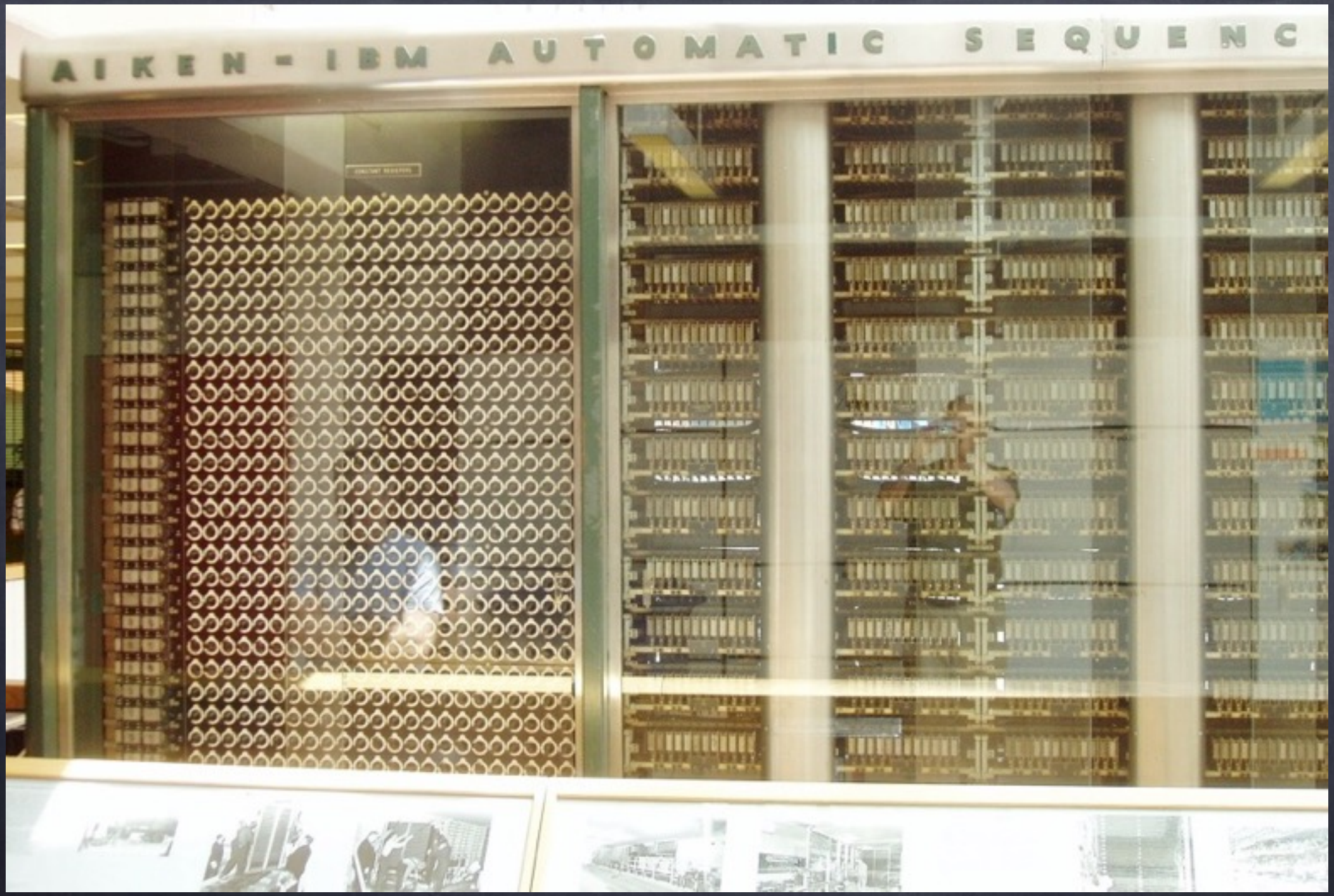
The Bombe

Colossus

Bletchley Park

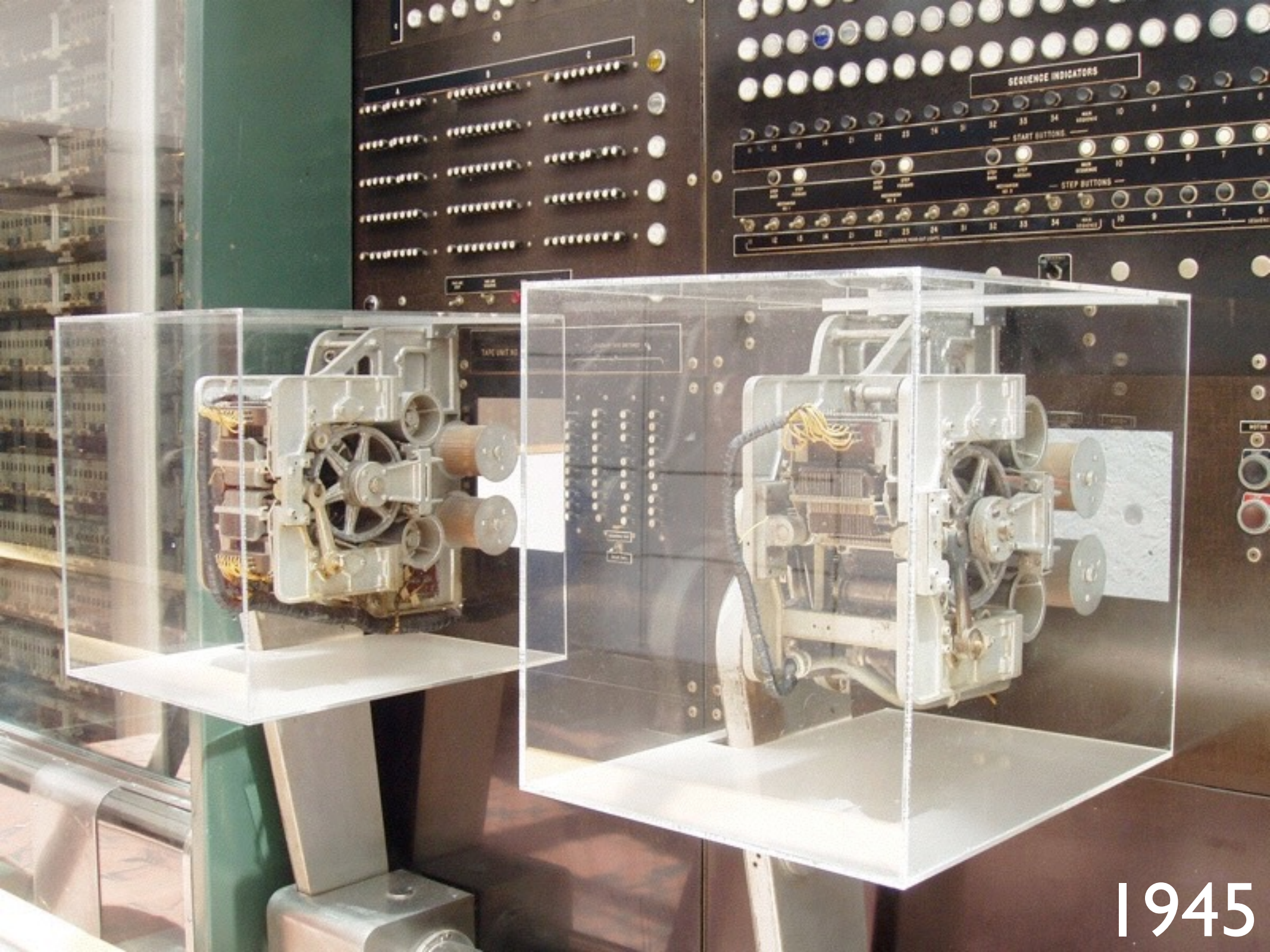






1945 Mark I, Harvard

1945 Mark I, Harvard


SEQUENCE INDICATORS
START BUTTONS
STEP BUTTONS
1945

1930

Les logiciens

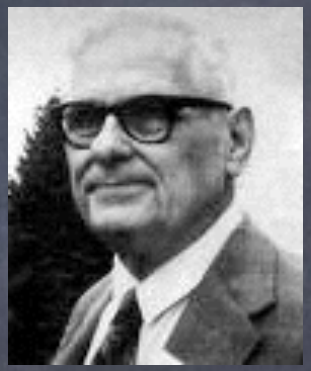
Hilbert → Gödel → Church → Turing

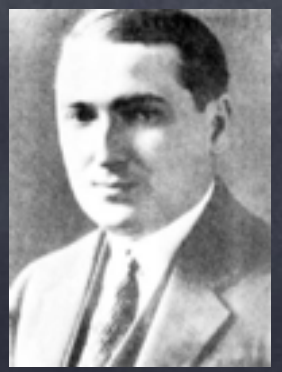
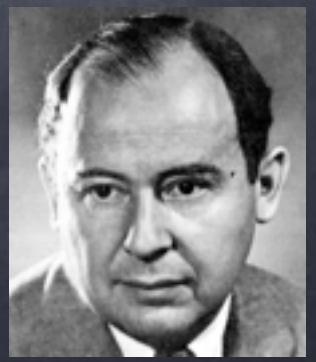
Church → Kleene

Kleene → Post

Post → von Neumann





UN

RÉSEAU

(1990)

internet

75000000000 appareils en 2020

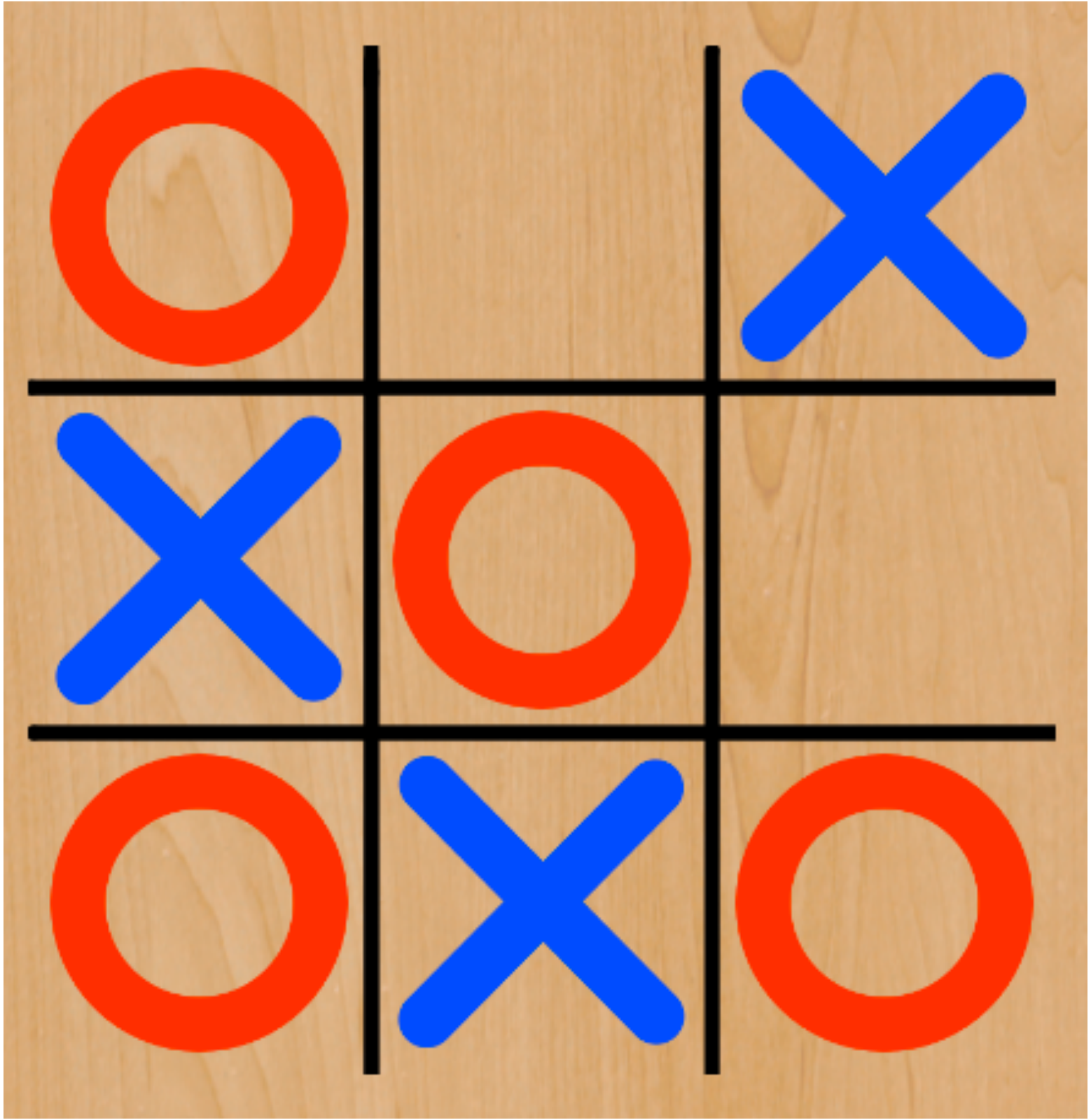
DES

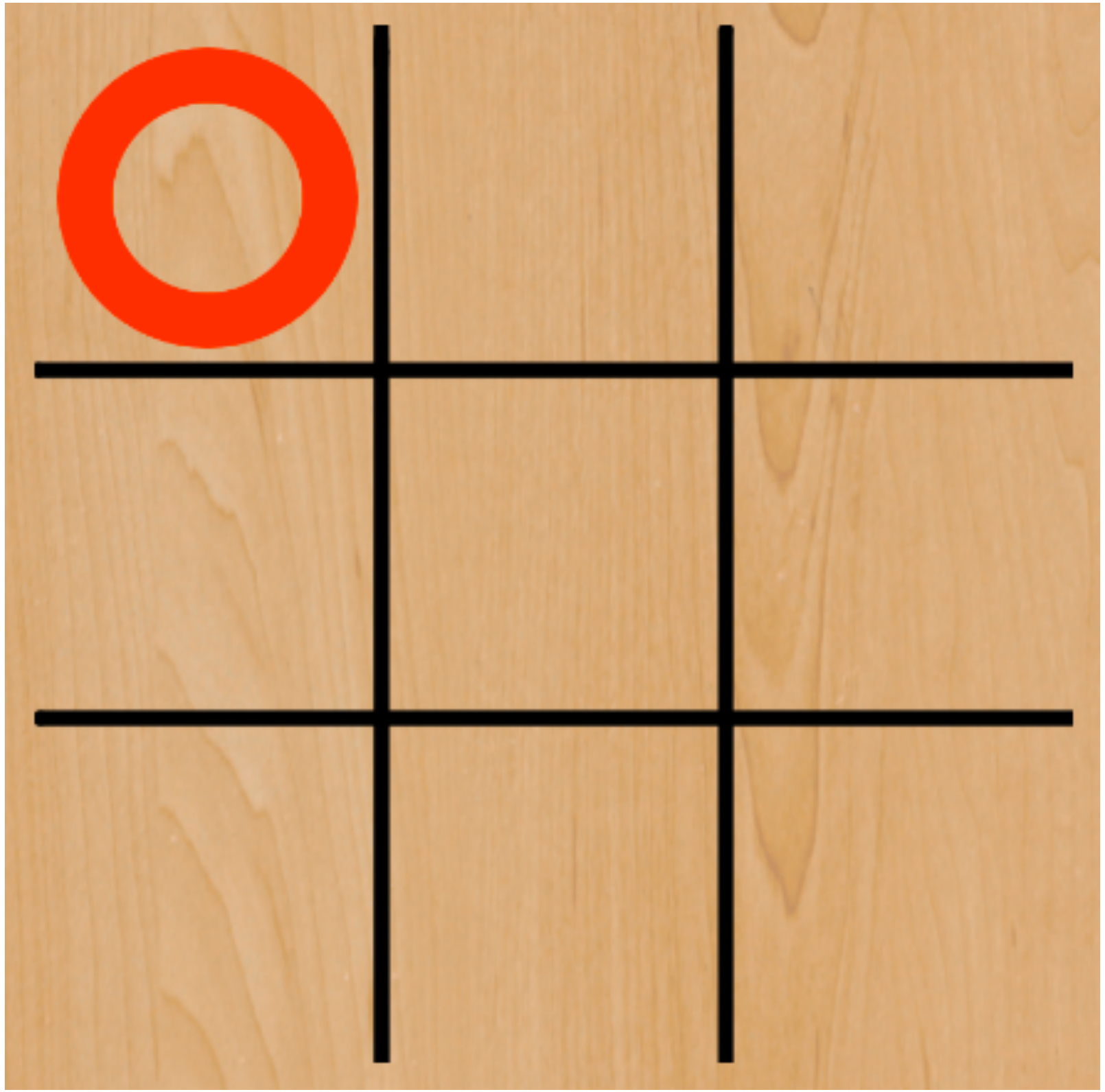
PROGRAMMES

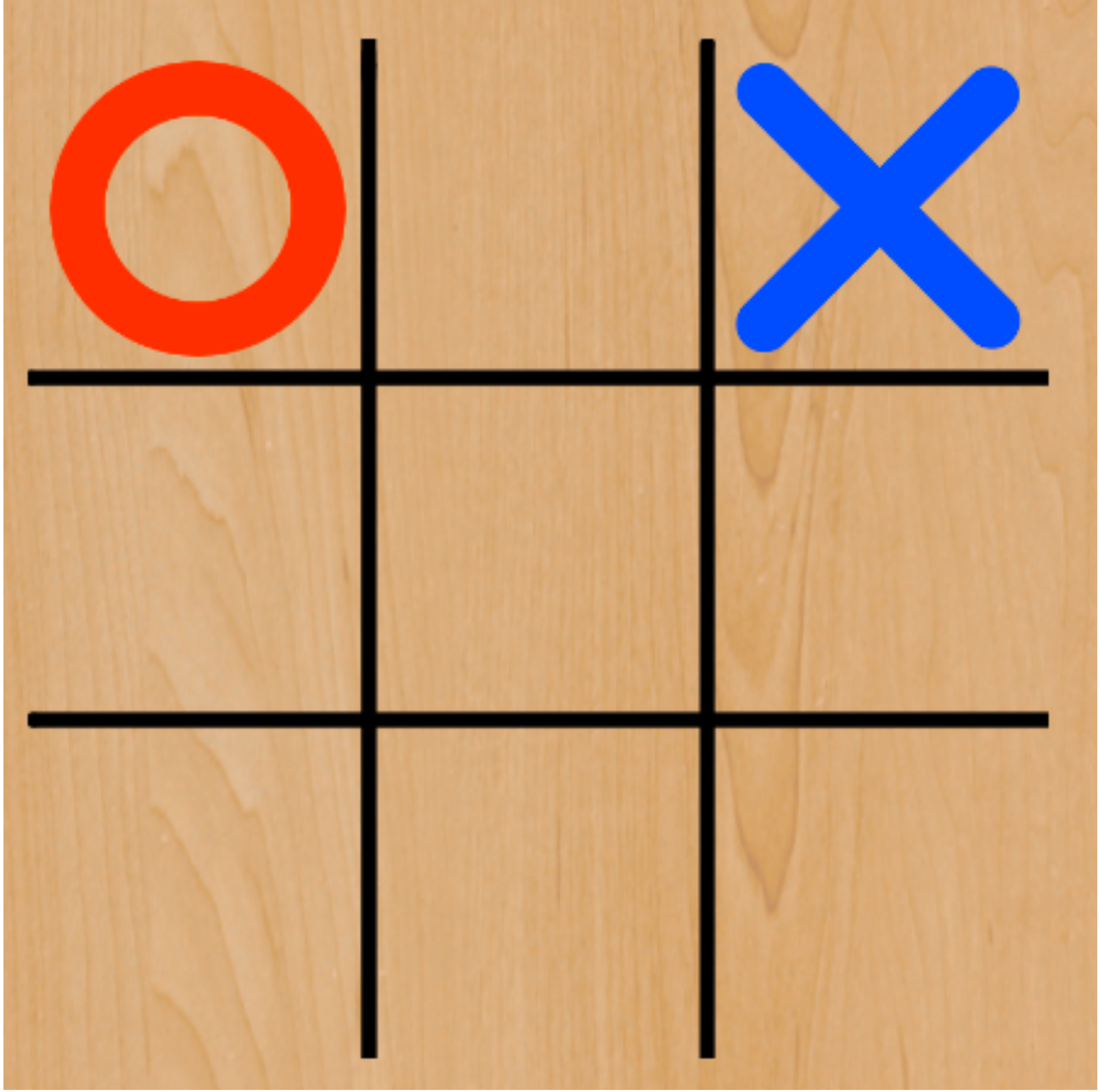
(1960)

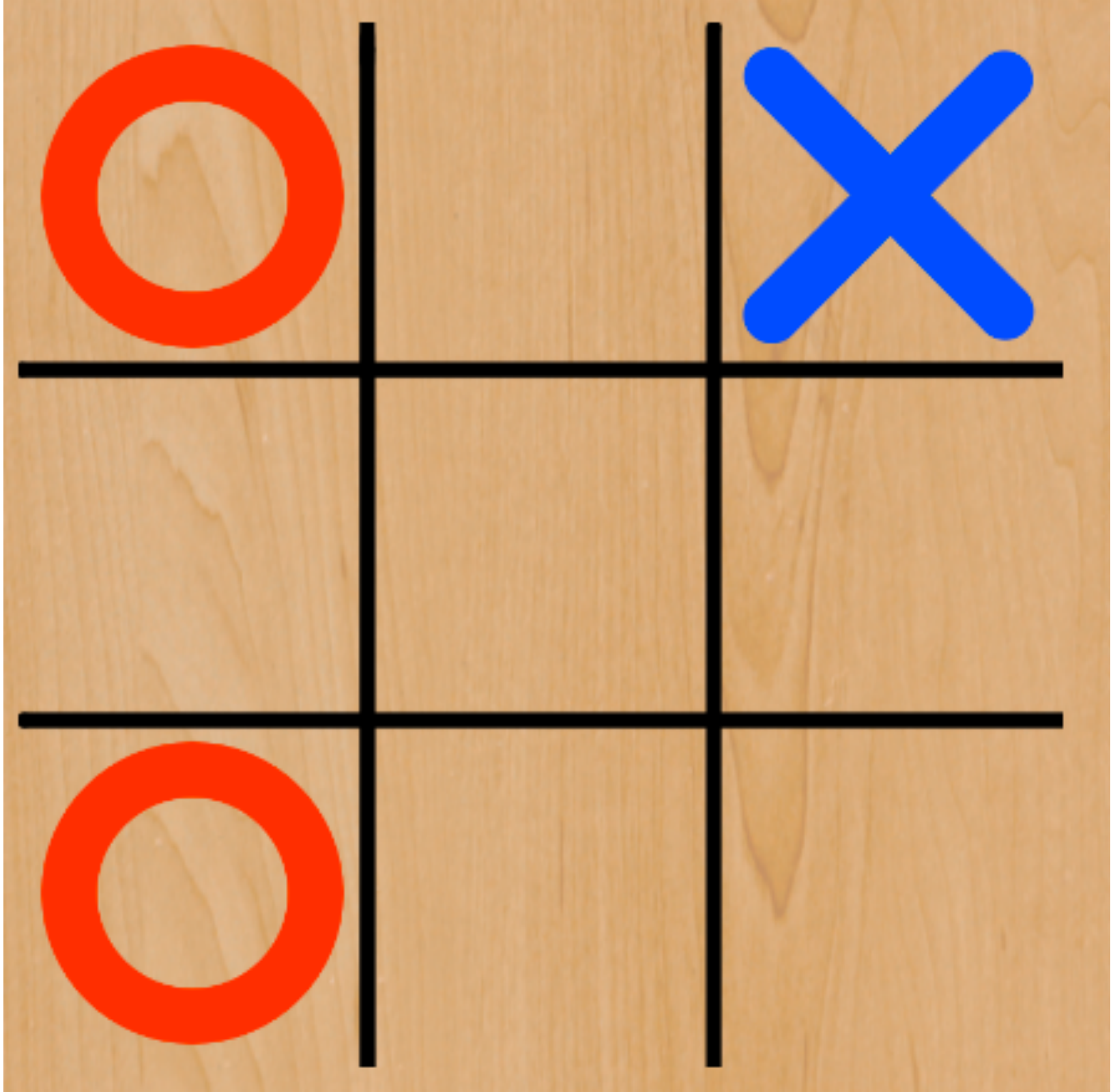
parler aux machines

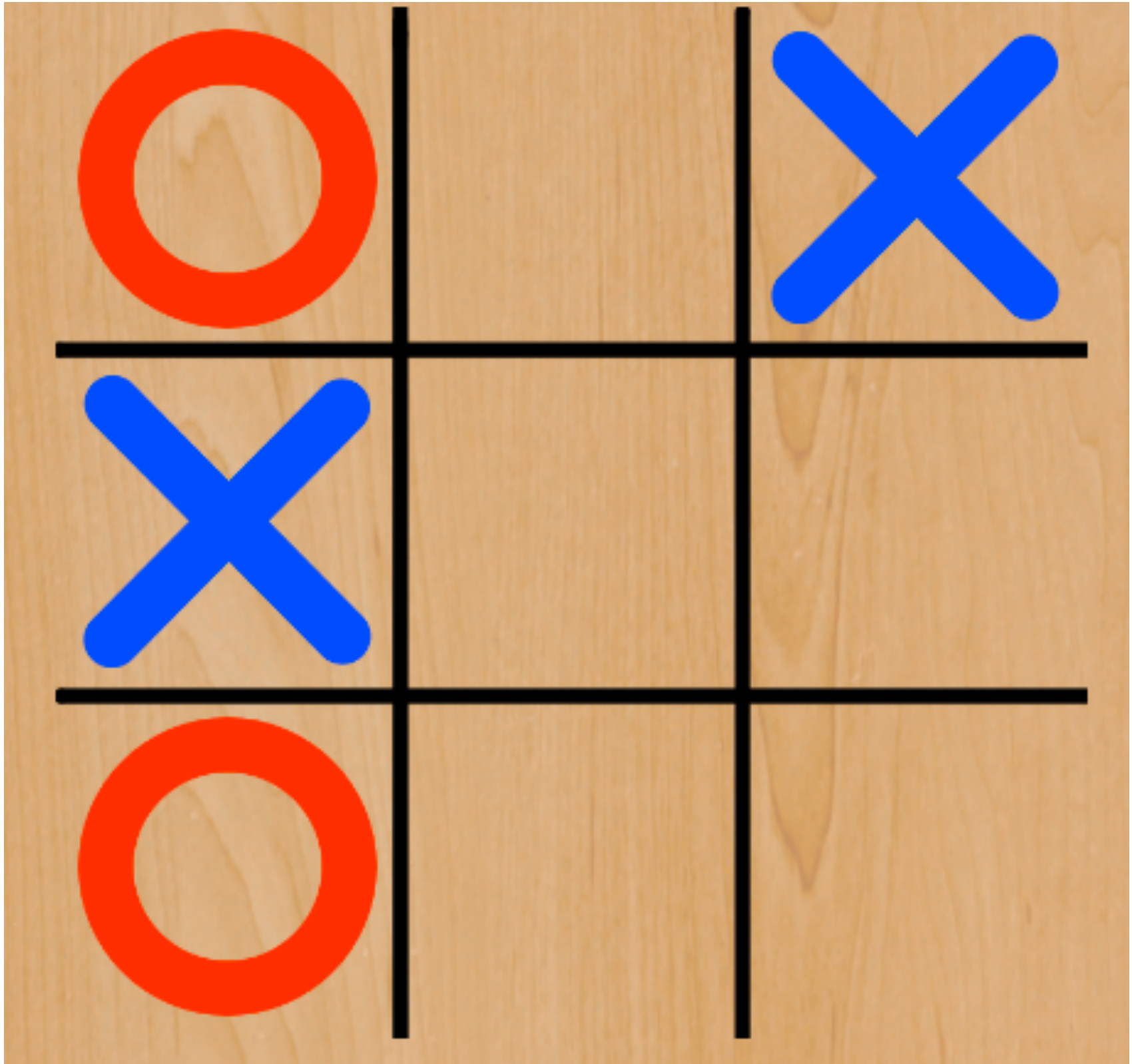
TIC-TAC-TOE

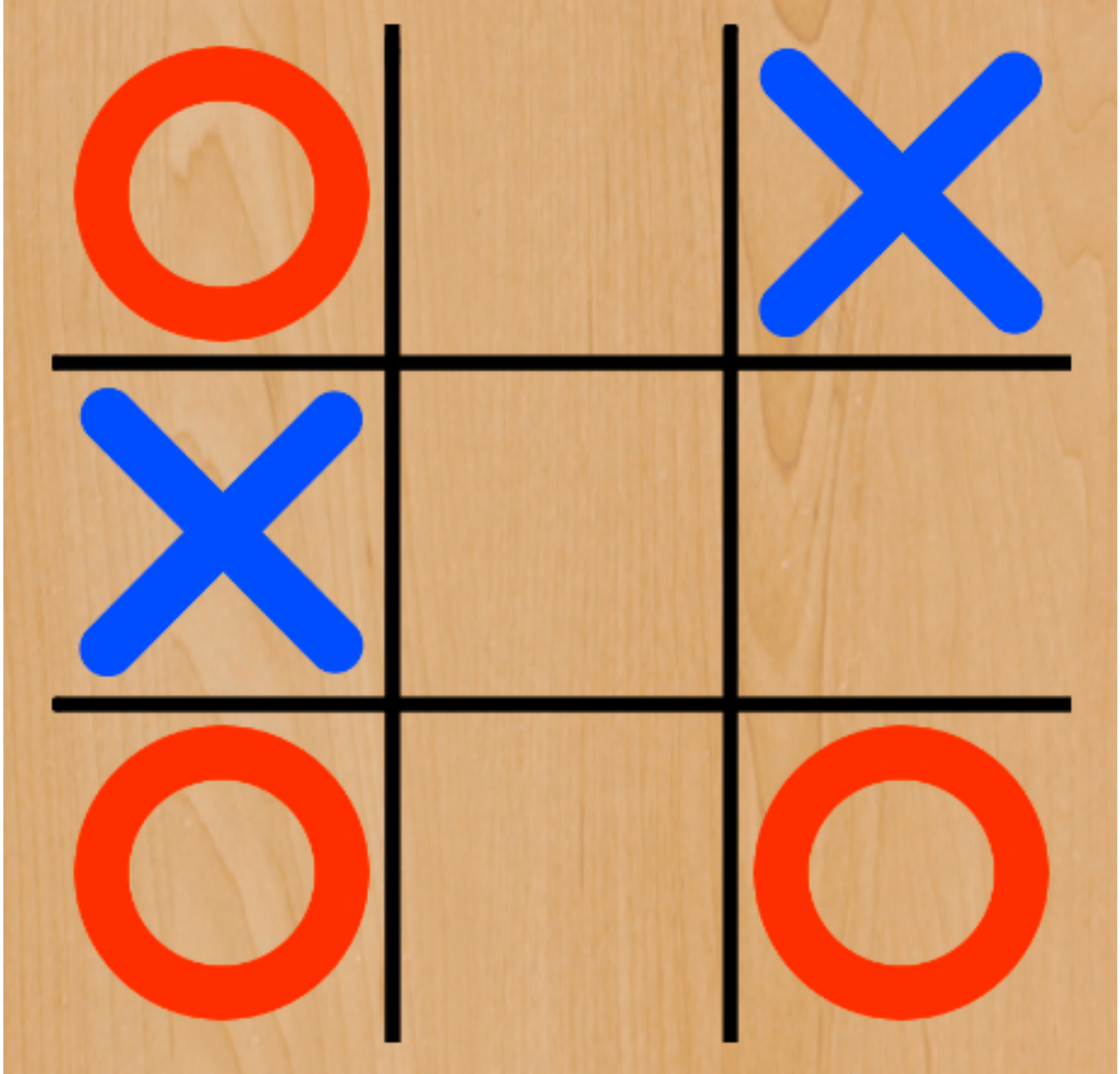


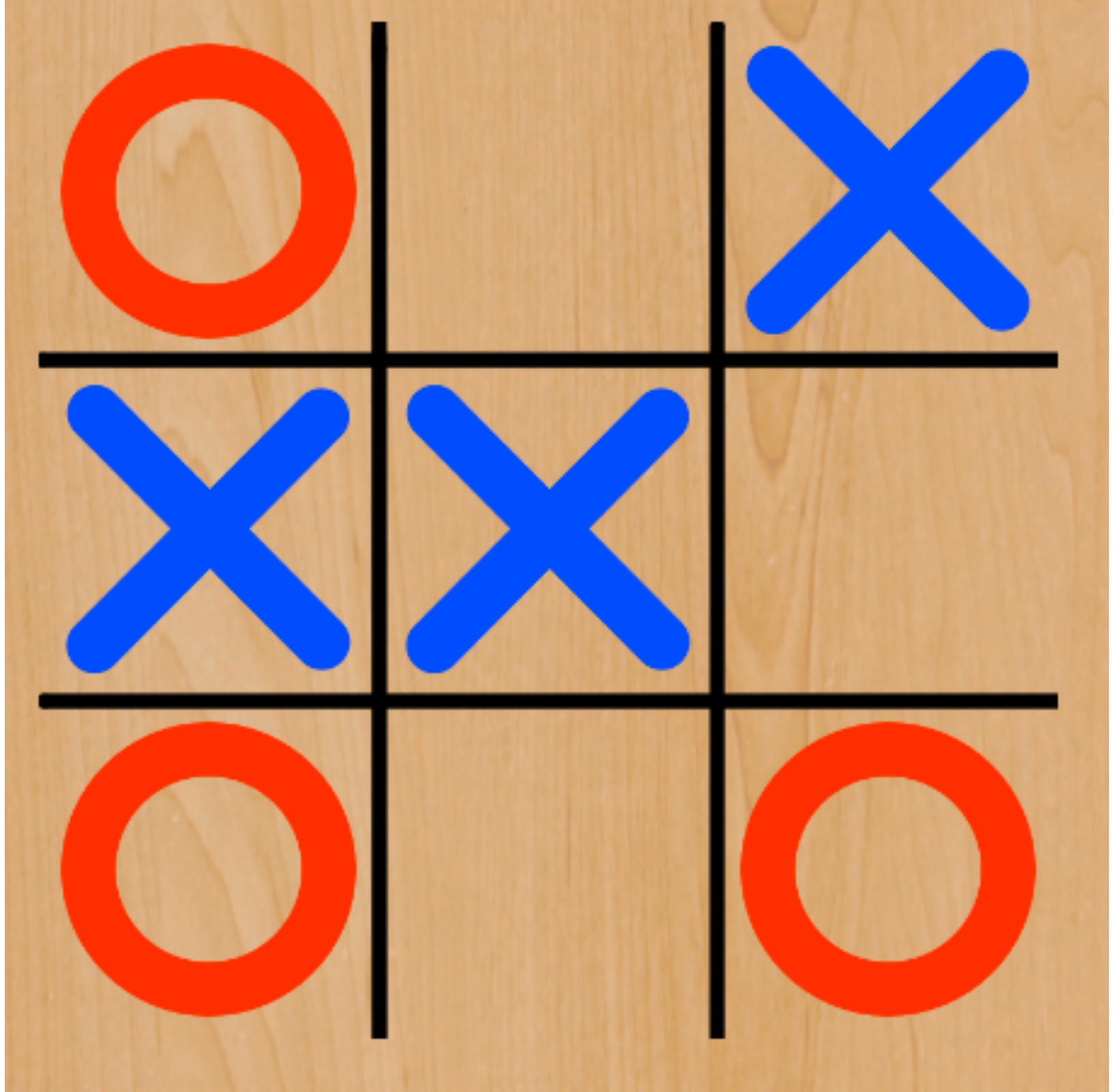


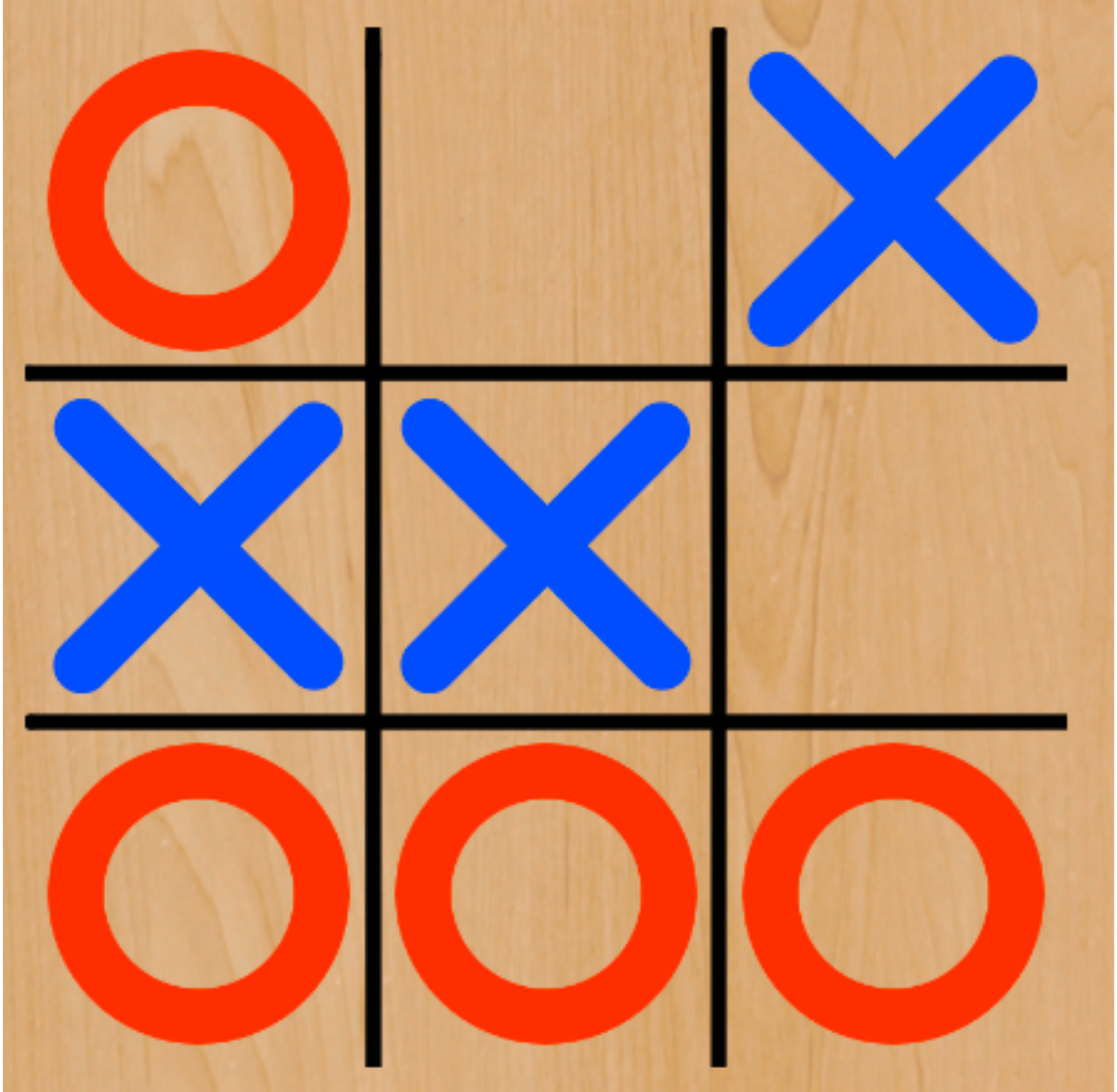












```
type brick = Empty | Computer | User

let brickToChar x = match x with
 | Empty -> ' '
 | Computer -> 'X'
 | User -> 'O' ;;

let emptyBoard = Array.make_matrix 3 3 Empty ;;

let copyBoard b = let b' = Array.make_matrix 3 3 Empty in
  for i=0 to 2 do for j=0 to 2 do b'.(i).(j) <- b.(i).(j) done done;
  b';;

let get b (i,j) = b.(i).(j) ;;
let set player b (i,j) =
  let b' = copyBoard b in b'.(i).(j) <- player; b' ;;

let winningPositions = [
 [(0,0) ; (1,1); (2,2)];
 [(2,0) ; (1,1); (0,2)];
 [(0,0) ; (0,1); (0,2)];
 [(1,0) ; (1,1); (1,2)];
 [(2,0) ; (2,1); (2,2)];
 [(0,0) ; (1,0); (2,0)];
 [(0,1) ; (1,1); (2,1)];
 [(0,2) ; (1,2); (2,2)]
];;

let allPositions = [
 (0,0); (0,1); (0,2);
 (1,0); (1,1); (1,2);
 (2,0); (2,1); (2,2) ] ;;

let hasWon player board =
  List.exists (function ligne ->
      List.for_all (function (i,j) -> player = board.(i).(j)) ligne)
   winningPositions ;;
```

```
let freeSpace board =
  List.filter (function (i,j) -> board.(i).(j) = Empty) allPositions

(******************** strategy ******************************)
type evaluation =  Win | Draw | Lose

let rec evaluate board move =
  let b2 = set Computer board move in
  if hasWon Computer b2 then Win
  else
    match freeSpace b2 with
    | [ ] -> Draw
    | userChoices ->
       let b3s = List.map (set User b2) userChoices in
       if List.exists (hasWon User) b3s then Lose
       else if List.exists (fun b3 -> bestOutcome b3 = Lose) b3s
       then Lose
       else if List.exists (fun b3 -> bestOutcome b3 = Draw) b3s
       then Draw
       else Win

and findBestChoice b =
  match freeSpace b with
  | [ ] -> ((-1,-1), Draw)
  | choices ->
    try let c = List.find (fun c -> evaluate b c = Win) choices in
      (c, Win)
    with Not_found ->
    try let c = List.find (fun c -> evaluate b c = Draw) choices in
      (c, Draw)
    with Not_found ->
      (List.hd choices, Lose)

and bestOutcome b = snd (findBestChoice b)
```

```
let bestChoice b = fst (findBestChoice b)

(******************** interface ********************************)
let computerPlay b = set Computer b (bestChoice b)

let printBoard b =
  Printf.printf "   | A | B | C |\n" ;
  Printf.printf "---+---+---+---+\n" ;
  for y=0 to 2 do
   Printf.printf " %d | %c | %c | %c | \n" (y + 1)
    (brickToChar (get b (y,0)))
    (brickToChar (get b (y,1)))
    (brickToChar (get b (y,2)))
  done;
  Printf.printf "---+---+---+---+ \n" ;;

let rec userPlay b =
  Printf.printf "Quelle case jouez-vous? (format: a1)\n" ;
  let input = read_line() in
  if String.length input <> 2
     || input.[0] < 'a' || input.[0] > 'c'
     || input.[1] < '1' || input.[1] > '3' then begin
     Printf.printf "N'importe quoi! \n";
     userPlay b
  end else
     let y = int_of_char(input.[0]) - int_of_char('a')  in
     let x = int_of_char (input.[1]) - int_of_char ('1') in
     if get b (x,y) <> Empty then begin
       Printf.printf "Case pas libre.\n";
       userPlay b
     end else
       set User b (x,y) ;;
```

```
let rec gameLoop b player =
  if freeSpace b = [ ] then
    Printf.printf "Fin. Match nul \n"
  else if player = Computer then begin
    Printf.printf "Ordinateur joue... \n";
    let b2 = computerPlay b in
    printBoard b2;
    if hasWon Computer b2 then
      Printf.printf "Fin. J'ai gagné \n"
    else
      gameLoop b2 User
  end else if player = User then
    let b2 = userPlay b in
    printBoard b2;
    if hasWon User b2 then
      Printf.printf "Fin. Vous avez gagné \n"
    else
      gameLoop b2 Computer
   ;;

let choose positions =
  let n = List.length positions in
  List.nth positions (Random.int n) ;;

let rec beginner () =
  let input = read_line() in
  if String.length input <> 1 then beginner() else
  if input.[0] = 'O' || input.[0] = 'o' || input.[0] = '0' then
    User
  else if input.[0] = 'X' || input.[0] = 'x' then
    Computer
  else beginner() ;;
```

```
let main () =
  Random.self_init() ;
  Printf.printf "Qui démarre? O/X :: " ;
  let player = beginner() in
  if player = Computer then
    let b = set Computer emptyBoard (choose (freeSpace emptyBoard)) in
    printBoard b ;
    gameLoop b User
  else
    gameLoop emptyBoard User;;


main();;
```

Les logiciens

Mark II,
Harvard

Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 Arctan started

1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 ?95 correct

13" sc (032) MP - MC ~~1.4821442000~~ ~~2.13047764715 (-3)~~ 4.615925059(-2)

(033) PRO 2 2.13047 6415

correct 2.13067 6415

Relays 6-2 in 033 failed special speed test
In relay " 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multy Adder Test.

1545 

Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

~~1525~~ 1630 Arctangent started.

1700 closed down.

Unix

- modularité “small is beautiful”

- AT&T Bell laboratories
- théoriciens ET praticiens
- système de hackers pour hackers
- pdp 11; Vax 780/750

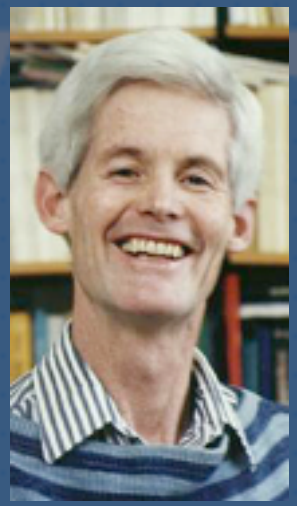
Faire fonctionner les ordinateurs

- langages de programmation
- systèmes d'exploitation

Jean Ichbiah

- correction des programmes
- trouver de bons algorithmes

Don Knuth



Steve Cook

P=NP ?

Microsoft

- ordinateur dans chaque maison
- bureautique (Word, Excel, Powerpoint)
- éditeur de logiciel, pas de matériel
- améliorations du système (NT, 95, XP, Vista)

Charles Simonyi

- Dave Cutler (DEC-VMS, NT)
- éditeurs WYSIWYG (bravo, Word)

Linux et le logiciel libre

- Emacs, éditeur de texte extensible
- gcc, compilateur C de la Free Software Foundation
- Linux = Unix refait par Linus Tordsvald
- tout le monde participe au système
- source public mais invasif
- logiciels de qualité

Richard Stallman

Informatique

c'est

super !