

# PSEUDOCODE

FUNCTION count\_neighbors(board, row, column):

    count = 0

    FOR EACH neighbor cell AROUND the current cell (row, column):

        IF AROUND cell HAS a living cell:

            count++

    RETURN count

FUNCTION occupied\_in\_next\_tick(currently\_occupied, neighbor\_count):

    IF currently\_occupied AND (neighbor\_count IS 2 OR neighbor\_count IS 3):

        RETURN true

    ELSE IF NOT currently\_occupied AND neighbor\_count IS 3:

        RETURN true

    ELSE:

        RETURN false

FUNCTION tick(board):

    CREATE a new metaverse called NEW

    RESIZE NEW to match the size of the current board

    FOR EACH cell IN the board:

        COUNT neighbors AROUND the cell

        SET the cell's state in NEW for the next tick based on neighbor count

    RETURN NEW

FUNCTION citizenship\_row\_to\_metaverse\_row(input\_row, row, board):

    FOR EACH character IN input\_row:

        IF character IS '1':

            SET corresponding cell in the metaverse board to alive

        ELSE IF character IS '0':

SET corresponding cell in the metaverse board to dead

ELSE:

RETURN false

RETURN true

FUNCTION initialize\_metaverse\_from\_file(metaverse\_file, metaverse, generations):

READ size AND generations FROM the file

RESIZE the metaverse board to match the size

FOR EACH row IN the file:

CONVERT row INTO metaverse format

RETURN success status

PROCEDURE model\_metaverse(starting\_metaverse, generations):

SET current\_metaverse TO starting\_metaverse

FOR i FROM 0 TO generations - 1:

DISPLAY current\_metaverse

UPDATE current\_metaverse FOR the next tick