INITIALIZE_DATA : v0

```
{
  iter ← 0

  // initialize data structures according to GUI

  CA_states [0..M+1, 0..N+1] ← 0
  CA_strat [0..M+1, 0..N+1] ← 0

  CA_actions [0..M+1, 0..N+1] ← 1*
  CA_KD_strat [0..M+1, 0..N+1] ← -1*
  CA_KC_strat [0..M+1, 0..N+1] ← -1*
  CA_KDC_strat [0..M+1, 0..N+1] ← -1*      } ale tylko na granicach tablic

  Encap_8_0s [1..M, 1..N] ← 0
  Encap_8_1s [1..M, 1..N] ← 0

  if debug & read_CA_state_deb & read_CA_strat_deb
  then
    { CA_STATES_DEB [1..M, 1..N] ← CA_states_deb.txt
      CA_STRAT_DEB [1..M, 1..N] ← CA_strat_deb.txt

      CA_states [0..M+1, 1..N+1] ← CA_STATES_DEB [,]
      CA_strat [0..M+1, 1..N+1] ← CA_STRAT_DEB [,]

      CA_KD_strat [,], CA_KC_strat [,], CA_KDC_strat [,] ← UPDATE

    }
  else
  { // initialize states according to GUI

    for i=1 to M
      for j=1 to N
      { x ← rand(0,1)
        if x ≤ GUI: p_init_C
        then
          CA_states [i,j] ← 1
        else
          CA_states [i,j] ← 0
      }

    if debug then print_01
```