

```

1 ; Gameplay algorithm for the Nintendo Game&Watch MC-25 "Mickey Mouse" / EG-26 "Egg"
2 ; Reversed from the original ROM program. All rights reserved to original author(s).
3 ; For study and testing purposes. Not for commercial use!
4 ; Transcribed by Milan Galcik - milan.galcik@gmail.com
5 ; Version of transcription: 1.1 (25 Jan 2024)
6
7
8
9 GLOBAL VARIABLES
10     current_mode      ; enumeration: demo | game_A | game_B
11
12     wait_loop_counter ; 0..31
13     wait_loop_max     ; 0..31
14
15     current_hen       ; enumeration: left_up | right_up | left_down | right_down
16     left_up_eggs      ; array [1..5] of 0 | 1
17     right_up_eggs     ; array [1..5] of 0 | 1
18     left_down_eggs    ; array [1..5] of 0 | 1
19     right_down_eggs   ; array [1..5] of 0 | 1
20     egg_in_danger     ; enumeration: none | left_up | right_up | left_down | right_down
21     crushed_egg_position ; enumeration: none | left | right
22     basket_position   ; enumeration: left_up | right_up | left_down | right_down
23
24     spawned_egg_counter ; 0..15
25     max_eggs           ; 0..15
26     miss_counter       ; 0..3
27     half_miss          ; true | false
28     idle_hen_counter  ; 0..3
29
30     current_score     ; 0..999
31     max_score_A       ; 0..999
32     max_score_B       ; 0..999
33
34     clock_hours       ; 1..12
35     clock_minutes    ; 0..59
36     clock_seconds     ; 0..59
37     alarm_hours       ; 1..12
38     alarm_minutes    ; 0..59
39
40     random_seed       ; 0..15
41
42
43
44 SUB Reset
45     clock_hours = 12
46     clock_minutes = 0
47     clock_seconds = 0
48     alarm_hours = 12
49     alarm_minutes = 0
50
51     random_seed = 0
52
53     current_mode = demo
54
55     wait_loop_counter = 0
56     wait_loop_max = 0
57
58     current_hen = right_up
59     left_up_eggs = 00000
60     right_up_eggs = 00000
61     left_down_eggs = 00000
62     right_down_eggs = 00000
63     egg_in_danger = none
64     crushed_egg_position = none
65     basket_position = right_down
66
67     spawned_egg_counter = 0
68     max_eggs = 0
69     miss_counter = 0
70     half_miss = false

```



```

141         idle_hen_counter = 0
142         call Check_Egg_In_Danger_vs_Basket_Position
143     else
144         increment idle_hen_counter
145     endif
146
147         call Prepare_Next_Hen
148     until (idle_hen_counter = 0) or (idle_hen_counter >= 3)
149     endif
150 endif
151 else
152     decrement empty_loop_countdown
153
154     if (empty_loop_countdown = 0)
155         if (crushed_egg_position <> none)
156             crushed_egg_position = none
157
158             if (rooster_Minnie_was_visible = true)
159                 if (half_miss = true)
160                     half_miss = false
161                 else
162                     increment miss_counter
163                     half_miss = true
164                 endif
165             else
166                 if (miss_counter >= 3) and (half_miss = true)
167                     half_miss = false
168                 else
169                     increment miss_counter
170                 endif
171             endif
172
173             if (miss_counter >= 3) and (half_miss = false)
174                 make_sound_of_game_over
175                 empty_loop_countdown = ???           ; tweak this value according to speed of your system
176             endif
177
178             elseif (miss_counter >= 3) and (half_miss = false)
179                 miss_counter = 0
180                 current_mode = demo                 ; back to demo mode
181             endif
182         endif
183     endif
184
185     call Scan_Input_Controls
186     call Update_Clock
187     call Update_Screen
188
189     ; if your system is too fast, add some delay here to achieve realistic game speed
190     until (current_mode = demo)
191     endif
192 END SUB
193
194
195
196 SUB Prepare_for_Game_A_or_Game_B
197     current_mode = game_A | game_B
198
199     call Clear_All_Eggs
200     crushed_egg_position = none
201
202     current_score = 0
203     miss_counter = 0
204     half_miss = false
205
206     call Compute_Max_Eggs
207
208     current_hen = call Get_Random_Hen_Position
209
210     wait_loop_counter = 0

```

```

211     if (current_mode = game_A)
212         wait_loop_max = 31
213     elseif (current_mode = game_B)
214         wait_loop_max = 25
215     endif
216 END SUB
217
218
219
220 SUB Compute_Max_Eggs
221     localvar big_score = (hundreds of current_score) + (tens of current_score)
222
223     if (big_score > 16)
224         max_eggs = 12
225     elseif (big_score >= 14)
226         max_eggs = 9
227     elseif (big_score >= 11)
228         max_eggs = 7
229     elseif (big_score >= 9)
230         max_eggs = 5
231     elseif (big_score >= 5)
232         max_eggs = 4
233     elseif (big_score >= 1)
234         max_eggs = 3
235     elseif (current_score >= 5) and (current_score <= 9)
236         max_eggs = 2
237     else
238         max_eggs = 1
239     endif
240 END SUB
241
242
243
244 FUNC Ignore_Current_Hen?
245     if (current_mode = game_A)
246         if (current_hen = left_up) and (miss_counter = 2)
247             return true
248         elseif (current_hen = right_up) and (miss_counter = 3)
249             return true
250         elseif (current_hen = left_down) and (miss_counter = 0)
251             return true
252         elseif (current_hen = right_down) and (miss_counter = 1)
253             return true
254         else
255             return false
256         endif
257     else
258         return false
259     endif
260 END FUNC
261
262
263
264 SUB Prepare_Next_Hen
265     if (current_hen = left_up)
266         current_hen = right_up
267     elseif (current_hen = right_up)
268         current_hen = left_down
269     elseif (current_hen = left_down)
270         current_hen = right_down
271     elseif (current_hen = right_down)
272         current_hen = left_up
273     endif
274 END SUB
275
276
277
278 SUB Clear_All_Eggs
279     left_up_eggs = 00000
280     right_up_eggs = 00000

```

```

281     left_down_eggs = 00000
282     right_down_eggs = 00000
283     egg_in_danger = none
284
285     spawned_egg_counter = 0
286 END SUB
287
288
289
290 SUB Shift_Eggs_Of_Current_Hen
291     if there is an egg from current hen at last (i.e. 5th) position
292         egg_in_danger = current_hen
293     endif
294
295     if there is any egg from current hen
296         make sound of current hen
297     endif
298
299     shift all eggs of current_hen
300 END SUB
301
302
303
304 SUB Check_Egg_In_Danger_vs_Basket_Position
305     if egg_in_danger = basket_position
306         decrement spawned_egg_counter
307         egg_in_danger = none
308         call Update_Score
309     endif
310 END SUB
311
312
313
314 FUNC Generate_Random_Number      ; returns two values simultaneously: random_number in range 0..15, and a boolean value
315     random_number =
316         a random value in range 0..15 +      ; based on hardware timer in original game
317         ones of clock_seconds +
318         tens of clock_seconds +
319         ones of clock_minutes +
320         tens of clock_minutes +
321         ones of clock_hours +
322         tens of clock_hours +
323         idle_hen_counter +
324         hundreds of max_score_A +
325         tens of max_score_A +
326         ones of max_score_A +
327         random_seed
328     random_number = random_number modulo 16
329     return random_number
330
331     increment random_seed by 6
332     if (random_seed >= 16)
333         random_seed = random_seed modulo 16
334         return true
335     else
336         return false
337     endif
338 END FUNC
339
340
341
342 FUNC Get_Random_Hen_Position
343     localvar random_number = call Generate_Random_Number
344     random_number = random_number modulo 4
345
346     return random_number
347 END FUNC
348
349
350

```

```

351 SUB Scan_Input_Controls
352   if (current_mode <> demo)
353     localvar new_basket_position = call Read_Input
354
355     if (new_basket_position <> basket_position)
356       basket_position = new_basket_position
357       call Generate_Random_Number
358       call Check_Egg_In_Danger_vs_Basket_Position
359       call Update_Screen
360     endif
361   endif
362 END SUB
363
364
365
366 SUB Update_Score
367   localvar previous_score = current_score
368
369   increment current_score
370   if (current_score > 999)
371     current_score = 0
372   endif
373
374   if (hundreds of previous_score) <> (hundreds of current_score)
375     if (hundreds of current_score) <> 9
376       increment wait_loop_max by 7 ; i.e. slow down the gameplay when turning around hundreds (except 899 to 900 transition when speed stays the same)
377     endif
378   elseif (tens of previous_score) <> (tens of current_score)
379     decrement wait_loop_max
380     if wait_loop_max < 6
381       wait_loop_max = 6
382     endif
383   endif
384
385   call Compute_Max_Eggs
386
387   if (current_mode = game_A) and (current_score > max_score_A)
388     max_score_A = current_score
389   elseif (current_mode = game_B) and (current_score > max_score_B)
390     max_score_B = current_score
391   endif
392 END SUB

```