

```

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: 1.2 (12 Apr 2025)
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

```

```

71     idle_hen_counter = 0
72
73     current_score = 0
74     max_score_A = 0
75     max_score_B = 0
76
77     call Update_Screen
78 END SUB
79
80
81
82 SUB Start_Game_A
83     if (current_mode = demo)
84         call Prepare_Game_A      ; see SUB Prepare_for Game_A or Game_B
85         call Update_Screen
86         call Game_Loop
87     endif
88 END SUB
89
90
91
92 SUB Start_Game_B
93     if (current_mode = demo)
94         call Prepare_Game_B      ; see SUB Prepare_for Game_A or Game_B
95         call Update_Screen
96         call Game_Loop
97     endif
98 END SUB
99
100
101
102 SUB Game_Loop
103     if (current_mode <> demo)
104         localvar empty_loop_countdown = 0
105         localvar rooster_Minnie_was_visible = false
106
107     repeat
108         if (empty_loop_countdown = 0)
109             increment wait_loop_counter
110
111         if (wait_loop_counter > wait_loop_max)
112             wait_loop_counter = 0
113
114         if (egg_in_danger <> none)
115             crushed_egg_position = left | right      ; according to egg_in_danger value
116             egg_in_danger = none
117
118             make sound of crushed egg
119             call Clear_All_Eggs
120
121             if (rooster / Minnie is visible)
122                 rooster_Minnie_was_visible = true
123             else
124                 rooster_Minnie_was_visible = false
125             endif
126
127             empty_loop_countdown = ???                  ; tweak this value according to speed of your system
128
129         elseif (crushed_egg_position = none)
130             idle_hen_counter = 0
131
132             repeat
133                 call Shift_Eggs_Of_Current_Hen
134
135                 if (call Ignore_Current_Hen? = false) and (call Is_There_Starting_Egg_For_Current_Hen? = false) and (spawned_egg_counter < max_eggs) and
136                     ((call Generate_Random_Number = true) or (spawned_egg_counter = 0))
137                     lay egg at first position for current hen
138                     increment spawned_egg_counter
139             endif
140

```

```

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

```

```

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

```

```

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

```

```

351
352 SUB Scan_Input_Controls
353   if (current_mode <> demo)
354     localvar new_basket_position = call Read_Input
355
356   if (new_basket_position <> basket_position)
357     basket_position = new_basket_position
358     call Generate_Random_Number
359     call Check_Egg_In_Danger_vs_Basket_Position
360     call Update_Screen
361   endif
362 endif
363 END SUB
364
365
366
367 SUB Update_Score
368   localvar previous_score = current_score
369
370   increment current_score
371   if (current_score > 999)
372     current_score = 0
373   endif
374
375   if (hundreds of previous_score) <> (hundreds of current_score)
376     if (hundreds of current_score) <> 9
377       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)
378     endif
379   elseif (tens of previous_score) <> (tens of current_score)
380     decrement wait_loop_max
381     if wait_loop_max < 6
382       wait_loop_max = 6
383     endif
384   endif
385
386   call Compute_Max_Eggs
387
388   if (current_mode = game_A) and (current_score > max_score_A)
389     max_score_A = current_score
390   elseif (current_mode = game_B) and (current_score > max_score_B)
391     max_score_B = current_score
392   endif
393 END SUB

```