

```
1 # Utilidades.s
2 # Archivo con distintas funciones utilizadas en Main.s
3 #
4 # Autores: Ka Fung & Christopher Gomez
5 # Fecha: 10-ene-2022
6
7 # Funcion: Convierte una coordenada (x, y) en una
8 #           direccion del Bitmap Display.
9 #           Toma como origen (0, 0) la esquina inferior izquierda.
10 # Entrada: $a0: Coordenada x.
11 #          $a1: Coordenada y.
12 # Salida:  $v0: Direccion del Bitmap Display
13 #          correspondiente a (x, y).
14 # Planificacion de registros:
15 # $t0: Auxiliar
16 coord_a_dir_bitmap:
17     # Prologo
18     sw    $fp, ($sp)
19     move  $fp, $sp
20     addi  $sp, $sp, -4
21
22     # Formula: (32*(31 - y) + x)*4 + MAT
23     li    $t0, 31
24     sub   $v0, $t0, $a1    # a = 31 - y
25     sll   $v0, $v0, 5      # a = a * 32
26     add   $v0, $v0, $a0    # a = a + x
27     sll   $v0, $v0, 2      # a = a * 4
28     lw    $t0, MAT
29     add   $v0, $v0, $t0    # a = a + MAT
30
31     # Epilogo
32     move  $sp, $fp
33     lw    $fp, ($sp)
34
35     jr    $ra
36
37 # Funcion: Retorna la dirección del Bitmap Display correspondiente
38 #          al pixel de arriba en el tablero de 32x32.
39 # Entrada: $a0: Direccion del Bitmap Display.
40 # Salida:  $v0: Direccion del Bitmap Display.
41 obtener_dir_arriba:
42     # Prologo
43     sw    $fp, ($sp)
44     move  $fp, $sp
45     addi  $sp, $sp, -4
46
47     add   $v0, $a0, -128 # DIRRETORNO = DIRACTUAL - 32*4
48
49     # Epilogo
50     move  $sp, $fp
51     lw    $fp, ($sp)
52
53     jr    $ra
54
55 # Funcion: Retorna la dirección del Bitmap Display correspondiente
56 #          al pixel de noreste en el tablero de 32x32.
57 # Entrada: $a0: Direccion del Bitmap Display.
58 # Salida:  $v0: Direccion del Bitmap Display.
59 obtener_dir_noreste:
60     # Prologo
61     sw    $fp, ($sp)
62     move  $fp, $sp
63     addi  $sp, $sp, -4
64
65     add   $v0, $a0, -124 # DIRRETORNO = DIRACTUAL - 32*4 + 4
66
67     # Epilogo
68     move  $sp, $fp
69     lw    $fp, ($sp)
70
71     jr    $ra
72
73 # Funcion: Retorna la dirección del Bitmap Display correspondiente
74 #          al pixel de derecha en el tablero de 32x32.
75 # Entrada: $a0: Direccion del Bitmap Display.
76 # Salida:  $v0: Direccion del Bitmap Display.
77 obtener_dir_derecha:
78     # Prologo
79     sw    $fp, ($sp)
80     move  $fp, $sp
81     addi  $sp, $sp, -4
82
83     add   $v0, $a0, 4 # DIRRETORNO = DIRACTUAL + 4
84
85     # Epilogo
86     move  $sp, $fp
87     lw    $fp, ($sp)
88
89     jr    $ra
90
91 # Funcion: Retorna la dirección del Bitmap Display correspondiente
```

```
92 #           al pixel de sureste en el tablero de 32x32.
93 # Entrada: $a0: Direccion del Bitmap Display.
94 # Salida:  $v0: Direccion del Bitmap Display.
95 obtener_dir_sureste:
96     # Prologo
97     sw    $fp, ($sp)
98     move  $fp, $sp
99     addi  $sp, $sp, -4
100
101     add  $v0, $a0, 132 # DIRRETORNO = DIRACTUAL + 32*4 + 4
102
103     # Epilogo
104     move  $sp, $fp
105     lw    $fp, ($sp)
106
107     jr  $ra
108
109 # Funcion: Retorna la dirección del Bitmap Display correspondiente
110 #           al pixel de abajo en el tablero de 32x32.
111 # Entrada: $a0: Direccion del Bitmap Display.
112 # Salida:  $v0: Direccion del Bitmap Display.
113 obtener_dir_abajo:
114     # Prologo
115     sw    $fp, ($sp)
116     move  $fp, $sp
117     addi  $sp, $sp, -4
118
119     add  $v0, $a0, 128 # DIRRETORNO = DIRACTUAL + 32*4
120
121     # Epilogo
122     move  $sp, $fp
123     lw    $fp, ($sp)
124
125     jr  $ra
126
127 # Funcion: Retorna la dirección del Bitmap Display correspondiente
128 #           al pixel del suroeste en el tablero de 32x32.
129 # Entrada: $a0: Direccion del Bitmap Display.
130 # Salida:  $v0: Direccion del Bitmap Display.
131 obtener_dir_suroeste:
132     # Prologo
133     sw    $fp, ($sp)
134     move  $fp, $sp
135     addi  $sp, $sp, -4
136
137     add  $v0, $a0, 124 # DIRRETORNO = DIRACTUAL + 32*4 - 4
138
139     # Epilogo
140     move  $sp, $fp
141     lw    $fp, ($sp)
142
143     jr  $ra
144
145 # Funcion: Retorna la dirección del Bitmap Display correspondiente
146 #           al pixel de izquierda en el tablero de 32x32.
147 # Entrada: $a0: Direccion del Bitmap Display.
148 # Salida:  $v0: Direccion del Bitmap Display.
149 obtener_dir_izquierda:
150     # Prologo
151     sw    $fp, ($sp)
152     move  $fp, $sp
153     addi  $sp, $sp, -4
154
155     add  $v0, $a0, -4 # DIRRETORNO = DIRACTUAL - 4
156
157     # Epilogo
158     move  $sp, $fp
159     lw    $fp, ($sp)
160
161     jr  $ra
162
163 # Funcion: Retorna la dirección del Bitmap Display correspondiente
164 #           al pixel de noroeste en el tablero de 32x32.
165 # Entrada: $a0: Direccion del Bitmap Display.
166 # Salida:  $v0: Direccion del Bitmap Display.
167 obtener_dir_noroeste:
168     # Prologo
169     sw    $fp, ($sp)
170     move  $fp, $sp
171     addi  $sp, $sp, -4
172
173     add  $v0, $a0, -132 # DIRRETORNO = DIRACTUAL - 32*4 - 4
174
175     # Epilogo
176     move  $sp, $fp
177     lw    $fp, ($sp)
178
179     jr  $ra
180
181 # Funcion: Pinta un tablero 32x32 desde una cadena de caracteres terminada
182 #           en nulo, donde cada caracter representa una celda, con el siguiente
```

```
183 #         formato:
184 #             El tamaño de la cadena es de 1025 bytes (32*32 + 1).
185 #             'G' representa gris oscuro (pared).
186 #             ' ' representa blanco (alimento).
187 #             'N' representa naranja (portal).
188 #             'P' representa amarillo (Pac-Man).
189 #             'R' representa rojo (Blinky).
190 #             'M' representa marron (Pinky).
191 #             'A' representa azul (Inky).
192 #             'V' representa verde (Clyde).
193 #         Usa los colores definidos en Main.s
194 #         Cuenta y actualiza a su vez la cantidad de alimento en el mapa.
195 # Entrada: $a0: Direccion de la cadena que contiene el tablero.
196 #         $a1: Direccion de contador de alimentos restantes.
197 #         $a2: Direccion de contador de alimentos totales.
198 # Salida: $v0: negativo si ocurrio algun error.
199 # Planificacion de registros:
200 # $t0: Auxiliar.
201 # $t1: Direccion a escribir en el Bitmap Display.
202 # $t2: Color del pixel a pintar.
203 # $t3: colorPared.
204 # $t4: colorComida.
205 pintar_tablero:
206     # Prologo
207     sw    $fp,    ($sp)
208     move  $fp,    $sp
209     addi  $sp,    $sp, -4
210
211     lw    $t1, MAT
212     lw    $t3, colorPared
213     lw    $t4, colorComida
214
215     pintar_tablero_for_pixel:
216         lb    $t0, ($a0)
217         beq   $t0, $zero, pintar_tablero_fin
218         add   $a0, $a0, 1
219
220         # Pintar tablero
221         beq   $t0, ' ', pintar_tablero_blanco
222         beq   $t0, 'G', pintar_tablero_gris
223         beq   $t0, 'N', pintar_tablero_naranja
224         beq   $t0, 'P', pintar_tablero_amarillo
225         beq   $t0, 'R', pintar_tablero_rojo
226         beq   $t0, 'M', pintar_tablero_marron
227         beq   $t0, 'A', pintar_tablero_azul
228
229         # Si no es ninguno de los demas es verde
230         lw    $t2, colorClyde
231         sw    $t2, ($t1)
232         j     pintar_tablero_aumentar_contador
233
234     pintar_tablero_gris:
235         sw    $t3, ($t1)
236         j     pintar_tablero_for_pixel_sig
237
238     pintar_tablero_blanco:
239         sw    $t4, ($t1)
240         j     pintar_tablero_aumentar_contador
241
242     pintar_tablero_naranja:
243         lw    $t2, colorPortal
244         sw    $t2, ($t1)
245         j     pintar_tablero_for_pixel_sig
246
247     pintar_tablero_amarillo:
248         lw    $t2, colorPacman
249         sw    $t2, ($t1)
250         j     pintar_tablero_for_pixel_sig
251
252     pintar_tablero_rojo:
253         lw    $t2, colorBlinky
254         sw    $t2, ($t1)
255         j     pintar_tablero_aumentar_contador
256
257     pintar_tablero_marron:
258         lw    $t2, colorPinky
259         sw    $t2, ($t1)
260         j     pintar_tablero_aumentar_contador
261
262     pintar_tablero_azul:
263         lw    $t2, colorInky
264         sw    $t2, ($t1)
265         j     pintar_tablero_aumentar_contador
266
267     pintar_tablero_aumentar_contador:
268         # Aumenta contador de alimentos restantes
269         lw    $t0, ($a1)
270         add   $t0, $t0, 1
271         sw    $t0, ($a1)
272
273     pintar_tablero_for_pixel_sig:
```

```
274         add $t1, $t1, 4
275         j    pintar_tablero_for_pixel
276
277 pintar_tablero_fin:
278     # Copia al contador de alimentos total
279     lw $t0, ($a1)
280     sw $t0, ($a2)
281
282     # Epilogo
283     move $sp,    $fp
284     lw    $fp,    ($sp)
285
286     jr $ra
287
288 # Funcion: Escoge una palabra pseudo-aleatoriamente del arreglo de entrada.
289 # Entrada: $a0: Numero de elementos del arreglo (1-3).
290 #          $a1: Direccion de arreglo de $a0 palabras
291 # Salida:  $v0: Opcion elegida pseudo-aleatoriamente.
292 # Planificacion de registros:
293 # $t0: Auxiliar.
294 # $t1: Opcion 1.
295 escoger_aleatorio:
296     # Prologo
297     sw    $fp, ($sp)
298     move $fp, $sp
299     addi $sp, $sp, -4
300
301     move $t0, $a0
302     move $t1, $a1
303
304     beq  $a0, 1, escoger_aleatorio_primer0
305
306     # (tiempo del sistema) mod (numero de opciones)
307     li   $v0, 30
308     syscall
309     abs  $a0, $a0
310     div  $a0, $t0
311     mfhi $t0
312
313     beqz $t0, escoger_aleatorio_primer0
314     beq  $t0, 1, escoger_aleatorio_segundo
315
316     # Si no es 0 o 1, es 2
317     lw $v0, 8($t1)
318     j   escoger_aleatorio_fin
319
320     escoger_aleatorio_primer0:
321         lw $v0, ($t1)
322         j   escoger_aleatorio_fin
323
324     escoger_aleatorio_segundo:
325         lw $v0, 4($t1)
326
327 escoger_aleatorio_fin:
328     # Epilogo
329     move $sp, $fp
330     lw    $fp, ($sp)
331
332     jr $ra
333
334 # Funcion: Imprime la puntuacion actual del juego.
335 # Entrada: $a0: Titulo de la puntuacion (Victoria, Derrota, Juego Finalizado)
336 # Planificacion de registros:
337 # $t0: Auxiliar.
338 # $t1: Auxiliar.
339 imprimir_puntuacion:
340     # Prologo
341     sw    $fp,    ($sp)
342     move $fp,    $sp
343     addi $sp,    $sp, -4
344
345     # Imprime titulo
346     li $v0, 4
347     syscall
348
349     # Imprime vidas
350     la $a0, msgVidas
351     li $v0, 4
352     syscall
353
354     lw $a0, V
355     li $v0, 1
356     syscall
357
358     la $a0, nuevaLinea
359     li $v0, 4
360     syscall
361
362     # Calcula puntuacion (porcentaje alimento)
363     la $a0, msgComida
364     li $v0, 4
```

```
365      syscall
366
367      lw    $t0, alimRestante
368      lw    $t1, alimTotal
369      sub   $t0, $t1, $t0
370      mul   $t0, $t0, 100
371      div   $t0, $t0, $t1
372
373      move  $a0, $t0
374      li    $v0, 1
375      syscall
376
377      la    $a0, msgComida2
378      li    $v0, 4
379      syscall
380
381      la    $a0, nuevaLinea
382      li    $v0, 4
383      syscall
384
385      # Calcula tiempo del juego
386      la    $a0, msgTiempo
387      li    $v0, 4
388      syscall
389
390      # Tiempo actual - tiempo inicial
391      li    $v0, 30
392      syscall
393      lw    $t0, tiempo
394      sub   $t0, $a0, $t0
395
396      # Convertir de ms a segs
397      li    $t1, 1000
398      divu  $t0, $t1
399      mflo  $a0
400      li    $v0, 1
401      syscall
402
403      la    $a0, msgTiempo2
404      li    $v0, 4
405      syscall
406
407      # Imprime puntos
408      la    $a0, puntos
409      li    $v0, 4
410      syscall
411
412      # Epilogo
413      move  $sp, $fp
414      lw    $fp, ($sp)
415
416      jr    $ra
```