

```

1 # Pacman.s
2 # Personaje principal del juego
3 #
4 # Autores: Ka Fung & Christopher Gomez
5 # Fecha: 10-ene-2022
6     .data
7
8     .text
9
10 # Funcion: Crea un Pacman con su posicion y color (variable global
11 #           definida como colorPacman en myexception.s).
12 # Salida:  $v0: Pacman (negativo si no se pudo crear).
13 #           ($v0): Direccion del Bitmap Display.
14 #           4($v0): Color del Pacman.
15 # Planificacion de registros:
16 # $s0: Direccion de memoria asignada para el Pacman.
17 # $t0: Auxiliar.
18 Pacman_crear:
19     # Prologo
20     sw    $fp,    ($sp)
21     sw    $ra, -4($sp)
22     sw    $s0, -8($sp)
23     move  $fp,    $sp
24     addi  $sp,    $sp, -12
25
26     # Reserva memoria para el Pacman
27     li    $v0, 9
28     li    $a0, 8
29     syscall
30     bltz  $v0, Pacman_crear_fin
31     move  $s0, $v0
32
33     # Inicializa el Pac-Man amarillo en (14, 11)
34     li    $a0, 14
35     li    $a1, 11
36     jal   coord_a_dir_bitmap
37
38     sw    $v0,    ($s0)
39     lw    $t0, colorPacman
40     sw    $t0, 4($s0)
41
42     move  $v0, $s0
43
44 Pacman_crear_fin:
45     # Epilogo
46     move  $sp,    $fp
47     lw    $fp,    ($sp)
48     lw    $ra, -4($sp)
49     lw    $s0, -8($sp)
50
51     jr    $ra
52
53 # Funcion: Reinicia Pacman con su posicion inicial.
54 # Entrada: $a0: Pacman
55 # Planificacion de registros:
56 # $s0: Pacman.
57 # $t0: Color de Pac-Man.
58 Pacman_reiniciar:
59     # Prologo
60     sw    $fp,    ($sp)
61     sw    $ra, -4($sp)
62     sw    $s0, -8($sp)
63     move  $fp,    $sp
64     addi  $sp,    $sp, -12
65
66     move  $s0, $a0
67
68     # Inicializa nuevamente el Pac-Man amarillo en (14, 11)
69     li    $a0, 14
70     li    $a1, 11
71     jal   coord_a_dir_bitmap
72     sw    $v0, ($s0)
73
74     # Pinta el Pac-Man
75     lw    $t0, colorPacman
76     sw    $t0, ($v0)
77
78     # Epilogo
79     move  $sp,    $fp
80     lw    $fp,    ($sp)
81     lw    $ra, -4($sp)
82     lw    $s0, -8($sp)
83
84     jr    $ra
85
86 # Funcion: Se encarga del movimiento del Pac-Man y su interaccion
87 #           con el entorno (segun la variable global D).
88 # Entrada: $a0: Pacman.
89 #           $a1: Direccion de contador de alimentos restantes.
90 # Planificacion de registros:
91 # $s0: Pacman.

```

```
92 # $s1: Dir. contador de alimentos restantes.
93 # $s2: Direccion actual del Pacman en el Bitmat Display.
94 # $s3: Direccion siguiente del Pacman en el Bitmat Display.
95 # $t0: Color del pixel siguiente.
96 # $t1: Auxiliar.
97 Pacman_mover:
98     # Prologo
99     sw    $fp,    ($sp)
100     sw    $ra,    -4($sp)
101     sw    $s0,    -8($sp)
102     sw    $s1,    -12($sp)
103     sw    $s2,    -16($sp)
104     sw    $s3,    -20($sp)
105     move  $fp,    $sp
106     addi  $sp,    $sp, -24
107
108     move  $s0, $a0    # Pacman
109     move  $s1, $a1    # Dir. contador de alimentos restantes
110
111     # Movimiento Pac-Man
112     lw    $t1, D
113     lw    $s2, ($s0)
114     move  $a0, $s2
115
116     beq   $t1, 'A', Pacman_mover_arriba    # Arriba
117     beq   $t1, 'b', Pacman_mover_abajo     # Abajo
118     beq   $t1, 'I', Pacman_mover_izq      # Izquierda
119
120     # Si no es ninguna de las anteriores, es derecha 'D'
121     jal   obtener_dir_derecha
122     j     Pacman_mover_siguiente
123
124 Pacman_mover_arriba:
125     jal   obtener_dir_arriba
126     j     Pacman_mover_siguiente
127
128 Pacman_mover_abajo:
129     jal   obtener_dir_abajo
130     j     Pacman_mover_siguiente
131
132 Pacman_mover_izq:
133     jal   obtener_dir_izquierda
134
135 Pacman_mover_siguiente:
136     # Obtiene el color del pixel.
137     move  $s3, $v0
138     lw    $t0, ($s3)
139
140     # Si se trata de una pared (no hace nada)
141     lw    $t1, colorPared
142     beq   $t0, $t1, Pacman_mover_fin
143
144     # Pinta de negro el pixel actual
145     lw    $t1, colorFondo
146     sw    $t1, ($s2)
147
148     # Si se trata de un camino (comida o fondo)
149     lw    $t1, colorComida
150     beq   $t0, $t1, Pacman_mover_actualizar_comida
151     lw    $t1, colorFondo
152     beq   $t0, $t1, Pacman_mover_pintar_pacman
153
154     # Si se trata de un portal
155     lw    $t1, colorPortal
156     beq   $t0, $t1, Pacman_mover_siguiente_portal
157
158     # En cambio, se trata de un fantasma
159     li    $t1, 1
160     sb    $t1, fueComido
161
162     j     Pacman_mover_fin
163
164 Pacman_mover_siguiente_portal:
165     # Portal 6 (0, 18)
166     li    $a0, 0
167     li    $a1, 18
168     jal   coord_a_dir_bitmap
169     beq   $v0, $s3, Pacman_mover_siguiente_portal_der
170
171     # Portal 6 (0, 17)
172     add   $t1, $v0, 128
173     beq   $t1, $s3, Pacman_mover_siguiente_portal_der
174
175     # De otra forma, se trata de los portales 5 (31, 17) y (31, 18)
176     # Mueve el Pac-Man al portal izquierdo
177     add   $s3, $s2, -116    # DIRSIGUIENTE = DIRACTUAL - 29*4
178
179     j     Pacman_mover_siguiente_portal_verif_alim
180
181 Pacman_mover_siguiente_portal_der:
182     # Mueve el Pac-Man al portal derecho
```

```
183         add $s3, $s2, 116 # DIRSIGUIENTE = DIRACTUAL + 29*4
184
185 Pacman_mover_siguiente_portal_verif_alim:
186     # Verificamos si la nueva posicion tiene alimento
187     lw $t0, colorComida
188     lw $t1, ($s3)
189     beq $t0, $t1, Pacman_mover_actualizar_comida
190
191     j Pacman_mover_pintar_pacman
192
193 Pacman_mover_actualizar_comida:
194     # Actualiza el contador
195     lw $t1, ($s1)
196     add $t1, $t1, -1
197     sw $t1, ($s1)
198
199 Pacman_mover_pintar_pacman:
200     # Pinta el Pacman
201     lw $t1, colorPacman
202     sw $t1, ($s3)
203
204     # Actualiza la direccion del Pacman en el Bitmap Display
205     sw $s3, ($s0)
206
207 Pacman_mover_fin:
208     # Epilogo
209     move $sp, $fp
210     lw $fp, ($sp)
211     lw $ra, -4($sp)
212     lw $s0, -8($sp)
213     lw $s1, -12($sp)
214     lw $s2, -16($sp)
215     lw $s3, -20($sp)
216
217     jr $ra
```