

```
1 # Fantasma.s
2 # Estructura que representa el personaje enemigo del juego.
3 #
4 # Autores: Ka Fung & Christopher Gomez
5 # Fecha: 10-ene-2022
6
7     .data
8
9     .text
10
11 # Funcion: Crea un Fantasma con su posicion y color.
12 # Entrada:  $a0: Coordenada x.
13 #           $a1: Coordenada y.
14 #           $a2: Color del fantasma.
15 # Salida:   $v0: Fantasma (negativo si no se pudo crear).
16 #           ($v0): Direccion del Bitmap Display.
17 #           4($v0): Color del Fantasma.
18 #           8($v0): Color de la capa de fondo.
19 #           12($v0): Dir. de movimiento del fantasma.
20 #           (0: Izquierda, 1: Arriba, 2: Abajo, 3: Derecha)
21 # Planificacion de registros:
22 # $s0: Color del fantasma.
23 # $t0: Coordenada x del fantasma.
24 # $t1: Auxiliar.
25 Fantasma_crear:
26     # Prologo
27     sw    $fp,    ($sp)
28     sw    $ra, -4($sp)
29     sw    $s0, -8($sp)
30     move  $fp,    $sp
31     addi  $sp,    $sp, -12
32
33     move  $s0, $a2
34
35     # Obtiene direccion del fantasma
36     jal   coord_a_dir_bitmap
37     move  $t0, $v0
38
39     # Reserva memoria para el fantasma
40     li    $v0, 9
41     li    $a0, 13
42     syscall
43     bltz  $v0, Fantasma_crear_fin
44
45     # Inicializa el fantasma
46     sw    $t0,    ($v0)      # Direccion
47     sw    $s0,    4($v0)     # Color del fantasma
48     lw    $t1, colorComida
49     sw    $t1,    8($v0)     # Color capa de fondo
50     li    $t1,    3
51     sb    $t1, 12($v0)      # Direccion de movimiento
52
53 Fantasma_crear_fin:
54     # Epilogo
55     move  $sp,    $fp
56     lw    $fp,    ($sp)
57     lw    $ra, -4($sp)
58     lw    $s0, -8($sp)
59
60     jr    $ra
61
62 # Funcion: Reinicia el fantasma con su posicion inicial.
63 # Entrada:  $a0: Fantasma
64 #           $a1: Coordenada x
65 #           $a2: Coordenada y
66 # Planificacion de registros:
67 # $s0: Fantasma.
68 # $s1: Dir. del fantasma actual en el Bitmap Display
69 # $t0: Auxiliar.
70 # $t1: Dir. del Bitmap Display del fantasma.
71 Fantasma_reiniciar:
72     # Prologo
73     sw    $fp,    ($sp)
74     sw    $ra, -4($sp)
75     sw    $s0, -8($sp)
76     sw    $s1, -12($sp)
77     sw    $s2, -16($sp)
78     move  $fp,    $sp
79     addi  $sp,    $sp, -20
80
81     move  $s0,    $a0
82     lw    $s1,    ($s0)
83     lw    $s2, 8($s0)
84
85     # Obtiene y actualiza direccion del fantasma
86     move  $a0, $a1
87     move  $a1, $a2
88     jal   coord_a_dir_bitmap
89     sw    $v0, ($s0)
90
91     # Actualiza el color del fondo en la dir. nueva del fantasma
```

```
92     lw $t0, colorComida
93     sw $t0, 8($s0)
94
95     # Si en la anterior partida, Pac-Man fue comido
96     # Pintar de color del fondo donde estaba el fantasma
97     lb $t1, fueComido
98     beqz $t1, Fantasma_reiniciar_actualizar_dir
99
100    beq $s2, $t0, Fantasma_reiniciar_pintar_fondo_anterior
101    lw $t0, colorFondo
102
103    Fantasma_reiniciar_pintar_fondo_anterior:
104        sw $s2, ($s1)
105
106        # Actualiza el color del fondo en la dir. nueva del fantasma
107        lw $t0, ($v0)
108        lw $t1, colorComida
109        beq $t0, $t1, Fantasma_reiniciar_pintar_fondo_actual
110
111        # Evita tomar como fondo algun color de fantasma
112        lw $t0, colorFondo
113
114    Fantasma_reiniciar_pintar_fondo_actual:
115        sw $t0, 8($s0)
116
117    Fantasma_reiniciar_actualizar_dir:
118        # Pinta el fantasma en la nueva direccion
119        lw $t0, 4($s0) # Color del fantasma
120        sw $t0, ($v0)
121
122    Fantasma_reiniciar_fin:
123        # Epilogo
124        move $sp, $fp
125        lw $fp, ($sp)
126        lw $ra, -4($sp)
127        lw $s0, -8($sp)
128        lw $s1, -12($sp)
129        lw $s2, -16($sp)
130
131        jr $ra
132
133    # Funcion: Se encarga del movimiento del fantasma por el tablero, tomando
134    #          en cuenta las intersecciones y colisiones.
135    # Entrada: $a0: Fantasma.
136    # Planificacion de registros:
137    # $s0: Fantasma
138    # $t0: Auxiliar
139    Fantasma_mover:
140        # Prologo
141        sw $fp, ($sp)
142        sw $ra, -4($sp)
143        sw $s0, -8($sp)
144        move $fp, $sp
145        addi $sp, $sp, -12
146
147        # Guarda fantasma
148        move $s0, $a0
149
150        # Verifica si se encuentra en una interseccion
151        jal Fantasma_chequear_interseccion
152        beqz $v0, Fantasma_mover_revisar_sig
153
154        # Escoge direccion aleatoria si es una interseccion
155        move $a0, $s0
156        jal Fantasma_cambiar_dir
157
158        j Fantasma_mover_ejecutar
159
160    Fantasma_mover_revisar_sig:
161        # Guarda la direccion del Bitmap Display del fantasma
162        lw $a0, ($s0)
163
164        # Revisa la direccion actual del fantasma
165        lb $t0, 12($s0)
166        beqz $t0, Fantasma_mover_revisar_sig_izq
167        beq $t0, 1, Fantasma_mover_revisar_sig_arriba
168        beq $t0, 2, Fantasma_mover_revisar_sig_abajo
169
170        # Revisa siguiente direccion derecha
171        jal obtener_dir_derecha
172        j Fantasma_mover_chequear_colision
173
174    Fantasma_mover_revisar_sig_izq:
175        jal obtener_dir_izquierda
176        j Fantasma_mover_chequear_colision
177
178    Fantasma_mover_revisar_sig_arriba:
179        jal obtener_dir_arriba
180        j Fantasma_mover_chequear_colision
181
182    Fantasma_mover_revisar_sig_abajo:
```

```
183         jal obtener_dir_abajo
184
185 Fantasma_mover_chequear_colision:
186     move $a0, $v0
187     jal Fantasma_es_camino
188     beqz $v0, Fantasma_mover_ejecutar
189
190     # Si hay alguna colision, se busca una nueva dir. aleatoria
191     move $a0, $s0
192     jal Fantasma_cambiar_dir
193
194 Fantasma_mover_ejecutar:
195     move $a0, $s0
196     jal Fantasma_ejecutar_mov
197
198 Fantasma_mover_fin:
199     # Epilogo
200     move $sp, $fp
201     lw $fp, ($sp)
202     lw $ra, -4($sp)
203     lw $s0, -8($sp)
204
205     jr $ra
206
207 # Funcion: Mueve el pixel del fantasma a la direccion especificada
208 #         en la estructura.
209 # Entrada: $a0: Fantasma.
210 # Planificacion de registros:
211 # $s0: Fantasma.
212 # $s1: Direccion del Bitmap Display siguiente del fantasma.
213 # $t0: Auxiliar.
214 # $t1: Color de Pac-Man / Fondo
215 Fantasma_ejecutar_mov:
216     # Prologo
217     sw $fp, ($sp)
218     sw $ra, -4($sp)
219     sw $s0, -8($sp)
220     sw $s1, -12($sp)
221     move $fp, $sp
222     addi $sp, $sp, -16
223
224     move $s0, $a0      # Fantasma
225     lw $a0, ($s0)      # Dir. del fantasma
226
227     # Revisa la direccion actual del fantasma
228     lb $t0, 12($s0)
229     beqz $t0, Fantasma_ejecutar_mov_izq
230     beq $t0, 1, Fantasma_ejecutar_mov_arriba
231     beq $t0, 2, Fantasma_ejecutar_mov_abajo
232
233     # Direccion derecha
234     jal obtener_dir_derecha
235     j Fantasma_ejecutar_mov_verif_portal
236
237 Fantasma_ejecutar_mov_izq:
238     jal obtener_dir_izquierda
239     j Fantasma_ejecutar_mov_verif_portal
240
241 Fantasma_ejecutar_mov_arriba:
242     jal obtener_dir_arriba
243     j Fantasma_ejecutar_mov_verif_portal
244
245 Fantasma_ejecutar_mov_abajo:
246     jal obtener_dir_abajo
247
248 Fantasma_ejecutar_mov_verif_portal:
249     # Guarda la direccion siguiente del fantasma
250     move $s1, $v0
251
252     # Verifica si hay un portal
253     lw $t0, colorPortal
254     lw $t1, ($v0)
255     bne $t0, $t1, Fantasma_ejecutar_mov_pintar
256
257     # Portal 6 (0, 18)
258     li $a0, 0
259     li $a1, 18
260     jal coord_a_dir_bitmap
261     beq $v0, $s1, Fantasma_ejecutar_mov_portal_der
262
263     # Portal 6 (0, 17)
264     add $t1, $v0, 128
265     beq $t1, $s1, Fantasma_ejecutar_mov_portal_der
266
267     # De otra forma, se trata del portal 5 (31, 17) o (31, 18)
268     # Mueve el Fantasma al portal izquierdo
269     add $s1, $s1, -120      # DIRSIGUIENTE = DIRACTUAL - 30*4
270     j Fantasma_ejecutar_mov_pintar
271
272 Fantasma_ejecutar_mov_portal_der:
273     # Mueve el Fantasma al portal derecho
```

```
274         add $s1, $s1, 120 # DIRSIGUIENTE = DIRACTUAL + 30*4
275
276 Fantasma_ejecutar_mov_pintar:
277     # Verificamos que la siguiente posicion es camino
278     # Si el camino no es valido, no se mueve el fantasma
279     move $a0, $s1
280     jal Fantasma_es_camino
281     bnez $v0, Fantasma_ejecutar_mov_fin
282
283     # En cambio, se mueve fantasma
284     # Pinta (x, y) del color de la capa de fondo del fantasma
285     lw $t0, ($s0) # Dir. actual del fantasma
286     lw $t1, 8($s0) # Color del fondo
287     sw $t1, ($t0)
288
289     # Actualiza fondo del fantasma
290     lw $t0, ($s1) # Color del fondo de la dir. siguiente
291     sw $t0, 8($s0)
292
293     # Actualiza nueva direccion del fantasma
294     sw $s1, ($s0)
295
296     # Revisa si el fantasma se ha comido a Pac-Man
297     lw $t0, ($s1)
298     lw $t1, colorPacman
299     bne $t0, $t1, Fantasma_ejecutar_mov_pintar_nuevo
300
301     lw $t0, colorFondo
302     sw $t0, 8($s0)
303
304     li $t1, 1
305     sb $t1, fueComido
306
307 Fantasma_ejecutar_mov_pintar_nuevo:
308     # Pinta el fantasma en la nueva direccion
309     lw $t1, 4($s0) # Color del fantasma
310     sw $t1, ($s1)
311
312 Fantasma_ejecutar_mov_fin:
313     # Epilogo
314     move $sp, $fp
315     lw $fp, ($sp)
316     lw $ra, -4($sp)
317     lw $s0, -8($sp)
318     lw $s1, -12($sp)
319
320     jr $ra
321
322 # Funcion: Verifica si un fantasma se encuentra en una interseccion.
323 # Entrada: $a0: Fantasma.
324 # Salida: $v0: 1 si se encuentra en una interseccion,
325 #          0 de otra forma.
326 # Planificacion de registros:
327 # $s0: Direccion del Bitmap Display
328 # $s3: colorPared
329 # $t0: Color del pixel que se chequea
330 # $t1: Booleanos auxiliares
331 # $t2: Booleanos auxiliares
332 Fantasma_chequear_interseccion:
333     # Prologo
334     sw $fp, ($sp)
335     sw $ra, -4($sp)
336     sw $s0, -8($sp)
337     sw $s1, -12($sp)
338     move $fp, $sp
339     addi $sp, $sp, -16
340
341     # Guarda la direccion del fantasma
342     lw $s0, ($a0)
343
344     # Guarda los colores a comparar
345     lw $s1, colorPared
346
347     # La condicion de interseccion consiste en chequear que
348     # las adyacencias sean caminos y la diagonal una pared
349
350     # Verifica si (x+1, y+1) es pared / Noreste
351     move $a0, $s0
352     jal obtener_dir_noreste
353     lw $t0, ($v0)
354     bne $t0, $s1, Fantasma_chequear_interseccion_der_abajo
355
356     # Verifica si (x, y+1) es camino / Arriba
357     move $a0, $s0
358     jal obtener_dir_arriba
359
360     move $a0, $v0
361     jal Fantasma_es_camino
362     bnez $v0, Fantasma_chequear_interseccion_der_abajo
363
364     # Verifica si (x+1, y) es camino / Derecha
```

```
365  move $a0, $s0
366  jal  obtener_dir_derecha
367
368  move $a0, $v0
369  jal  Fantasma_es_camino
370  bnez $v0, Fantasma_chequear_interseccion_abajo_izq
371
372  j  Fantasma_chequear_interseccion_retornar_cierto
373
374  Fantasma_chequear_interseccion_der_abajo:
375      # Verifica si (x+1, y-1) es pared / Sureste
376      move $a0, $s0
377      jal  obtener_dir_sureste
378      lw  $t0, ($v0)
379      bne $t0, $s1, Fantasma_chequear_interseccion_abajo_izq
380
381      # Verifica si (x+1, y) es camino / Derecha
382      move $a0, $s0
383      jal  obtener_dir_derecha
384
385      move $a0, $v0
386      jal  Fantasma_es_camino
387      bnez $v0, Fantasma_chequear_interseccion_abajo_izq
388
389      # Verifica si (x, y-1) es camino / Abajo
390      move $a0, $s0
391      jal  obtener_dir_abajo
392
393      move $a0, $v0
394      jal  Fantasma_es_camino
395      bnez $t1, Fantasma_chequear_interseccion_izq_arriba
396
397      j  Fantasma_chequear_interseccion_retornar_cierto
398
399  Fantasma_chequear_interseccion_abajo_izq:
400      # Verifica si (x-1, y-1) es pared / Suroeste
401      move $a0, $s0
402      jal  obtener_dir_suroeste
403      lw  $t0, ($v0)
404      bne $t0, $s1, Fantasma_chequear_interseccion_izq_arriba
405
406      # Verifica si (x, y-1) es camino / Abajo
407      move $a0, $s0
408      jal  obtener_dir_abajo
409
410      move $a0, $v0
411      jal  Fantasma_es_camino
412      bnez $v0, Fantasma_chequear_interseccion_izq_arriba
413
414      # Verifica si (x-1, y) es camino / Izquierda
415      move $a0, $s0
416      jal  obtener_dir_izquierda
417
418      move $a0, $v0
419      jal  Fantasma_es_camino
420      bnez $v0, Fantasma_chequear_interseccion_retornar_falso
421
422      j  Fantasma_chequear_interseccion_retornar_cierto
423
424  Fantasma_chequear_interseccion_izq_arriba:
425      # Verifica si (x-1, y+1) es pared / Noroeste
426      move $a0, $s0
427      jal  obtener_dir_noroeste
428      lw  $t0, ($v0)
429      bne $t0, $s1, Fantasma_chequear_interseccion_retornar_falso
430
431      # Verifica si (x-1, y) es camino / Izquierda
432      move $a0, $s0
433      jal  obtener_dir_izquierda
434
435      move $a0, $v0
436      jal  Fantasma_es_camino
437      bnez $v0, Fantasma_chequear_interseccion_retornar_falso
438
439      # Verifica si (x, y+1) es camino / Arriba
440      move $a0, $s0
441      jal  obtener_dir_arriba
442
443      move $a0, $v0
444      jal  Fantasma_es_camino
445      bnez $v0, Fantasma_chequear_interseccion_retornar_falso
446
447      j  Fantasma_chequear_interseccion_retornar_cierto
448
449  Fantasma_chequear_interseccion_retornar_falso:
450      li $v0, 0
451      j  Fantasma_chequear_interseccion_fin
452
453  Fantasma_chequear_interseccion_retornar_cierto:
454      li $v0, 1
455
```



```
456 Fantasma_chequear_interseccion_fin:
457     # Epilogo
458     move $sp,      $fp
459     lw  $fp,      ($sp)
460     lw  $ra,      -4($sp)
461     lw  $s0,      -8($sp)
462     lw  $s1,     -12($sp)
463
464     jr $ra
465
466 # Funcion: Escoge aleatoriamente una direccion valida para el siguiente
467 #          movimiento del fantasma. Solo retorna la direccion contraria
468 #          a la actual si es la unica disponible
469 # Entrada: $a0: Fantasma.
470 # Salida:  $v0: Dir. aleatoria valida de movimiento del fantasma.
471 #          (0: Izquierda, 1: Arriba, 2: Abajo, 3: Derecha)
472 # Planificacion de registros:
473 # $s0: Fantasma
474 # $s1: Direccion del Bitmap Display del fantasma
475 # $s2: Direccion opuesta al movimiento del fantasma
476 # $s3: Contador de direcciones disponibles / Auxiliar
477 # $s4: Dir. inicial de la pila donde se guarda dir. de movimientos validos
478 # $t0: Auxiliar
479 Fantasma_cambiar_dir:
480     # Prologo
481     sw  $fp,      ($sp)
482     sw  $ra,      -4($sp)
483     sw  $s0,      -8($sp)
484     sw  $s1,     -12($sp)
485     sw  $s2,     -16($sp)
486     sw  $s3,     -20($sp)
487     sw  $s4,     -24($sp)
488     move $fp,      $sp
489     addi $sp,      $sp, -28
490
491     move $s0,      $a0          # Fantasma
492     lw  $s1,      ($a0)          # Dir. del fantasma
493     lw  $s2,     12($a0)          # Dir. opuesta al movimiento del fantasma
494     andi $s2,      $s2, 0xF
495     li  $t0,      3
496     sub $s2,      $t0, $s2
497
498     # Si es una interseccion, chequea caminos disponibles
499     # No tomara en cuenta la direccion opuesta como disponible
500     move $s3, $zero          # Contador de dir. disponibles
501
502     la $s4, ($sp)
503
504     # Arriba (salta la direccion si es la opuesta)
505     beq $s2, 1, Fantasma_cambiar_dir_chequear_derecha
506
507     # Verificamos si la dir. de arriba es camino
508     move $a0, $s1
509     jal obtener_dir_arriba
510     move $a0, $v0
511     jal Fantasma_es_camino
512
513     bnez $v0, Fantasma_cambiar_dir_chequear_derecha
514
515     # Empila las direcciones disponibles para moverse
516     li  $t0, 1
517     sw  $t0, ($sp)
518     add $s3, $s3, 1
519     add $sp, $sp, -4
520
521 Fantasma_cambiar_dir_chequear_derecha:
522     beq $s2, 3, Fantasma_cambiar_dir_chequear_abajo
523
524     # Verifica si la dir. derecha es camino
525     move $a0, $s1
526     jal obtener_dir_derecha
527     move $a0, $v0
528     jal Fantasma_es_camino
529
530     bnez $v0, Fantasma_cambiar_dir_chequear_abajo
531
532     li  $t0, 3
533     sw  $t0, ($sp)
534     add $s3, $s3, 1
535     add $sp, $sp, -4
536
537 Fantasma_cambiar_dir_chequear_abajo:
538     beq $s2, 2, Fantasma_cambiar_dir_chequear_izquierda
539
540     # Verifica si la dir. abajo es camino
541     move $a0, $s1
542     jal obtener_dir_abajo
543     move $a0, $v0
544     jal Fantasma_es_camino
545
546     bnez $v0, Fantasma_cambiar_dir_chequear_izquierda
```

```
547
548     li $t0, 2
549     sw $t0, ($sp)
550     add $s3, $s3, 1
551     add $sp, $sp, -4
552
553 Fantasma_cambiar_dir_chequear_izquierda:
554     beqz $s2, Fantasma_cambiar_dir_verificar
555
556     # Verifica si la dir. izquierda es camino
557     move $a0, $s1
558     jal obtener_dir_izquierda
559     move $a0, $v0
560     jal Fantasma_es_camino
561
562     bnez $v0, Fantasma_cambiar_dir_verificar
563
564     move $t0, $zero
565     sw $t0, ($sp)
566     add $s3, $s3, 1
567     add $sp, $sp, -4
568
569 Fantasma_cambiar_dir_verificar:
570     bnez $s3, Fantasma_cambiar_dir_escoger_direccion
571
572     # Se mueve en la direcciones contraria si es la unica opcion
573     sw $s2, ($sp)
574     add $s3, $s3, 1
575     add $sp, $sp, -4
576
577 Fantasma_cambiar_dir_escoger_direccion:
578     move $a0, $s3
579     add $a1, $sp, 4
580
581     jal escoger_aleatorio
582     sb $v0, 12($s0)
583
584     # Desempila direcciones disponibles
585     move $sp, $s4
586
587 Fantasma_cambiar_dir_fin:
588     # Epilogo
589     move $sp, $fp
590     lw $fp, ($sp)
591     lw $ra, -4($sp)
592     lw $s0, -8($sp)
593     lw $s1, -12($sp)
594     lw $s2, -16($sp)
595     lw $s3, -20($sp)
596     lw $s4, -24($sp)
597
598     jr $ra
599
600
601 # Funcion: Verifica si el pixel en la direccion es un camino
602 #         válido para el fantasma.
603 # Entrada: $a0: Direccion del Bitmap Display del fantasma.
604 # Salida:  $v0: 0 si el pixel es un camino.
605 #         1 de otra forma.
606 # Planificacion de registros:
607 # $t0: Color de comida / fondo / portal / Pac-Man.
608 # $t1: Color del pixel en la direccion.
609 # $s0: Dir. del Bitmap Display del fantasma.
610 Fantasma_es_camino:
611     # Prologo
612     sw $fp, ($sp)
613     sw $ra, -4($sp)
614     sw $s0, -8($sp)
615     move $fp, $sp
616     addi $sp, $sp, -12
617
618     # Color del pixel
619     lw $t1, ($a0)
620     move $s0, $a0
621
622     # Si es un camino (comida, fondo, portal o Pac-Man)
623     move $v0, $zero
624     lw $t0, colorComida
625     beq $t1, $t0, Fantasma_es_camino_fin
626
627     lw $t0, colorFondo
628     beq $t1, $t0, Fantasma_es_camino_fin
629
630     lw $t0, colorPacman
631     beq $t1, $t0, Fantasma_es_camino_fin
632
633     lw $t0, colorPortal
634     beq $t1, $t0, Fantasma_es_camino_verif_portal
635
636     li $v0, 1
637     j Fantasma_es_camino_fin
```

```
638
639 Fantasma_es_camino_verif_portal:
640     # Portal 6 (0, 18)
641     li $a0, 0
642     li $a1, 18
643     jal coord_a_dir_bitmap
644     beq $v0, $s0, Fantasma_es_camino_verif_portal_der
645
646     # Portal 6 (0, 17)
647     add $t1, $v0, 128
648     beq $v0, $s0, Fantasma_es_camino_verif_portal_der
649
650     # De otra forma, se trata del portal 5 (31, 17) o (31, 18)
651     # Verifica recursivamente si el otro lado del portal es camino
652     add $a0, $s0, -120      # DIRSIGUIENTE = DIRACTUAL - 30*4
653     jal Fantasma_es_camino
654     j Fantasma_es_camino_fin
655
656     Fantasma_es_camino_verif_portal_der:
657     # Análogamente, verifica si el otro lado del portal es camino
658     add $a0, $s0, 120      # DIRSIGUIENTE = DIRACTUAL + 30*4
659     jal Fantasma_es_camino
660
661 Fantasma_es_camino_fin:
662     # Epilogo
663     move $sp, $fp
664     lw $fp, ($sp)
665     lw $ra, -4($sp)
666     lw $s0, -8($sp)
667
668     jr $ra
```