

```
# Pacman.s
# Personaje principal del juego
#
# Autores: Ka Fung & Christopher Gomez
# Fecha: 10-ene-2022
.data

.text

# Funcion: Crea un Pacman con su posicion y color (variable global
#          definida como colorPacman en myexception.s).
# Salida:  $v0: Pacman (negativo si no se pudo crear).
#          ($v0): Direccion del Bitmap Display.
#          4($v0): Color del Pacman.
# Planificacion de registros:
# $s0: Direccion de memoria asignada para el Pacman.
# $t0: Auxiliar.
Pacman_crear:
    # Prologo
    sw    $fp,    ($sp)
    sw    $ra, -4($sp)
    sw    $s0, -8($sp)
    move  $fp,    $sp
    addi  $sp,    $sp, -12

    # Reserva memoria para el Pacman
    li    $v0, 9
    li    $a0, 8
    syscall
    bltz  $v0, Pacman_crear_fin
    move  $s0, $v0

    # Inicializa el Pac-Man amarillo en (14, 11)
    li    $a0, 14
    li    $a1, 11
    jal   coord_a_dir_bitmap

    sw    $v0,    ($s0)
    lw    $t0, colorPacman
    sw    $t0, 4($s0)

    move  $v0, $s0

Pacman_crear_fin:
    # Epilogo
    move  $sp,    $fp
    lw    $fp,    ($sp)
    lw    $ra, -4($sp)
    lw    $s0, -8($sp)

    jr    $ra

# Funcion: Reinicia Pacman con su posicion inicial.
# Entrada: $a0: Pacman
# Planificacion de registros:
# $s0: Pacman.
# $t0: Color de Pac-Man.
Pacman_reiniciar:
    # Prologo
    sw    $fp,    ($sp)
    sw    $ra, -4($sp)
    sw    $s0, -8($sp)
    move  $fp,    $sp
    addi  $sp,    $sp, -12

    move  $s0, $a0

    # Inicializa nuevamente el Pac-Man amarillo en (14, 11)
    li    $a0, 14
    li    $a1, 11
    jal   coord_a_dir_bitmap
    sw    $v0, ($s0)

    # Pinta el Pac-Man
    lw    $t0, colorPacman
    sw    $t0, ($v0)

    # Epilogo
    move  $sp,    $fp
    lw    $fp,    ($sp)
    lw    $ra, -4($sp)
    lw    $s0, -8($sp)

    jr    $ra

# Funcion: Se encarga del movimiento del Pac-Man y su interaccion
#          con el entorno (segun la variable global D).
# Entrada: $a0: Pacman.
#          $a1: Direccion de contador de alimentos restantes.
# Planificacion de registros:
# $s0: Pacman.
# $s1: Dir. contador de alimentos restantes.
# $s2: Direccion actual del Pacman en el Bitmat Display.
# $s3: Direccion siguiente del Pacman en el Bitmat Display.
# $t0: Color del pixel siguiente.
# $t1: Auxiliar.
```

```
Pacman_mover:
# Prologo
sw $fp, ($sp)
sw $ra, -4($sp)
sw $s0, -8($sp)
sw $s1, -12($sp)
sw $s2, -16($sp)
sw $s3, -20($sp)
move $fp, $sp
addi $sp, $sp, -24

move $s0, $a0 # Pacman
move $s1, $a1 # Dir. contador de alimentos restantes

# Movimiento Pac-Man
lw $t1, D
lw $s2, ($s0)
move $a0, $s2

beq $t1, 'A', Pacman_mover_arriba # Arriba
beq $t1, 'b', Pacman_mover_abajo # Abajo
beq $t1, 'I', Pacman_mover_izq # Izquierda

# Si no es ninguna de las anteriores, es derecha 'D'
jal obtener_dir_derecha
j Pacman_mover_siguiente
Pacman_mover_arriba:
jal obtener_dir_arriba
j Pacman_mover_siguiente
Pacman_mover_abajo:
jal obtener_dir_abajo
j Pacman_mover_siguiente
Pacman_mover_izq:
jal obtener_dir_izquierda
Pacman_mover_siguiente:
# Obtiene el color del pixel.
move $s3, $v0
lw $t0, ($s3)

# Si se trata de una pared (no hace nada)
lw $t1, colorPared
beq $t0, $t1, Pacman_mover_fin

# Pinta de negro el pixel actual
lw $t1, colorFondo
sw $t1, ($s2)

# Si se trata de un camino (comida o fondo)
lw $t1, colorComida
beq $t0, $t1, Pacman_mover_actualizar_comida
lw $t1, colorFondo
beq $t0, $t1, Pacman_mover_pintar_pacman

# Si se trata de un portal
lw $t1, colorPortal
beq $t0, $t1, Pacman_mover_siguiente_portal

# En cambio, se trata de un fantasma
li $t1, 1
sb $t1, fueComido

j Pacman_mover_fin
Pacman_mover_siguiente_portal:
# Portal 6 (0, 18)
li $a0, 0
li $a1, 18
jal coord_a_dir_bitmap
beq $v0, $s3, Pacman_mover_siguiente_portal_der

# Portal 6 (0, 17)
add $t1, $v0, 128
beq $t1, $s3, Pacman_mover_siguiente_portal_der

# De otra forma, se trata de los portales 5 (31, 17) y (31, 18)
# Mueve el Pac-Man al portal izquierdo
add $s3, $s2, -116 # DIRSIGUIENTE = DIRACTUAL - 29*4

j Pacman_mover_siguiente_portal_verif_alim

Pacman_mover_siguiente_portal_der:
# Mueve el Pac-Man al portal derecho
add $s3, $s2, 116 # DIRSIGUIENTE = DIRACTUAL + 29*4

Pacman_mover_siguiente_portal_verif_alim:
# Verificamos si la nueva posicion tiene alimento
lw $t0, colorComida
lw $t1, ($s3)
beq $t0, $t1, Pacman_mover_actualizar_comida

j Pacman_mover_pintar_pacman
Pacman_mover_actualizar_comida:
# Actualiza el contador
lw $t1, ($s1)
add $t1, $t1, -1
sw $t1, ($s1)
Pacman_mover_pintar_pacman:
```

```
# Pinta el Pacman
lw $t1, colorPacman
sw $t1, ($s3)

# Actualiza la direccion del Pacman en el Bitmap Display
sw $s3, ($s0)
```

Pacman_mover_fin:

```
# Epilogo
move $sp, $fp
lw $fp, ($sp)
lw $ra, -4($sp)
lw $s0, -8($sp)
lw $s1, -12($sp)
lw $s2, -16($sp)
lw $s3, -20($sp)

jr $ra
```