

```

world.pngHelloWorld
map]9.2), comprobaremos que la posición $8000 hace referencia a la V RAM interna. Y precisamente es de aquí de donde la consola
El mto de que no estén en memoria que crearemos es el equivalente al dir de Amstrad, con el que podremos insertar filas de bytes de un origen a
mem]COPY_MEM :
lda, [hl]; cargamos el dato en Ald[de], a; copiamos el dato al destino de cbc; un o menos por copiar
COPY_MEM
sprites]7.4, lo que debemos hacer es indicar el sitio en el que lo queramos y un número de tile mostrar (de la tributa no nos vamos a pre
Estos datos los vamos a guardar en cualquier lugar de la ROM (según vea conveniente el ensamblador), para una vez se ejecute el
oam]MAKE_LETTER :
ldhl, letter_Hldde, V RAMldbc,
COPY_MEM
DB5, 5, 7, 0
scroll.pngSpriteAnimado en NO$GMB

```

[anexo_input]9.4 puede hallar más información). Una vez comprobada la dirección en la que el jugador nos indica, moveremos en esa dirección el P AD lo tendremos que leer varias veces seguidas, ya que debido al efecto "bouncing" se pueden producir resultados no concordes.

```

[caption=Lectura
del
PAD,
label=code:read_pad] =====
Valores correspondientes =====
A_BUTTON EQU B_BUTTON EQU SELECT EQU START EQU RIGHT_J EQU LEFT_J EQU UP_J EQU DOWN_J
=====

```

Función de lectura

```

=====
READ_PAD :
ld
a,
P1F5; (cruzeta activada, botones no) ld[rP1], a
ld
a,
[rP1]
ld
a,
[rP1]
ld
a,
[rP1]
ld
a,
[rP1]
; leemos
varias
veces
and
0F; solo queremos los 4 bits bajos swapa; cambiamos los valores altos por los bajos ld b, a; insertamos el resultado en el registro b
ld
a,
P1F4; (botones activados, cruzeta no) ld[rP1], a
ld
a,
[rP1]
ld
a,
[rP1]
ld
a,
[rP1]
ld
a,
[rP1]
and
0F; nuevamente solo queremos los bits bajos or b; hacemos una operación con b para obtener los 8 bits.
cpl
; comentario

```

PAD
y
hacer
una
op-
eración
AND
con
la
con-
stante
que
quer-
amos.
Si
el
re-
sul-
tado
no
da
valor
nulo,
quiere
de-
cir
que
esta
pul-
sada
en
ese
in-
stante.

Ahora
lo
que
queda
es
saber
cómo
quer-
e-
mos
mover
al
per-
son-
aje.
En
mi
caso,
sim-
ple-
mente
decre-
mento
o
in-
cre-
mento
las
co-
or-
de-
nadas
Y
o
X
del
scroll,
menos
en
el
caso
de

guardarme

ya

desde

un

prin-

ci-

pio

el

tile

pre-

cal-

cu-

lado

en

el

que

se

en-

con-

traba

mi

per-

son-

aje.

Dig-

amos

que

si

mi

posi-

ción

en

co-

or-

de-

nadas

era

la

(80,80),

siendo

el

valor

del

scroll

(0,0),

el

tile

del

per-

son-

aje

es

el

(9,9).

Los

cál-

cu-

los

son

sen-

cil-

los:

di-

vidi-

mos

las

co-

or-

de-

nadas

del

per-

son-

aje

en-

tre

8

[caption=Comprobación
de
Col-
i-
siones
del
En-
e-
migo,
label=code:check_{tile_n}]CM_{CHECK}_{NEXT}_{ILE_EN} :
ld
c,
a
call
CM_{CHECK}_{TILE}; ReturnsonHLthememoryaddressofthetile.ModifiesHL, A, B, DE.
ld
a,
c
or
a
jr
z,
cnt_{top}; Moveenemyupdecajrz, cnt_{bot}; Moveenemydowndecajrz, cnt_{left}; Moveenemyleftjrcnt_{right}; Moveenemyright
¿Qué
in-
put
le
pasamos?
Es
lo
”único”
que
nos
queda
por
cal-
cu-
lar.
Real-
mente
pro-
gramé
varias
fun-
ciones
que
cal-
cu-
la-
ban
la
di-
rec-
ción
a
seguir
de
man-
era
dis-
tinta.
El
mayor
prob-
lema
de
la
GB
es
el
del
tiempo,
en
el
sen-
tido
de

os-
curo
al
más
claro.
Hay
que
ll-
e-
var
cuidado
con
el
color
00
en
los
sprites,
ya
que
se
con-
vierte
en
traspar-
ente.

Por
otro
lado,
en
el
caso
de
la
GBC,
la
cosa
se
com-
plica.
Nos
ten-
emos
que
hacer
dos
fun-
ciones,
una
para
la
paleta
de
fondo
y
otra
para
los
sprites.
La
idea
es
la
de
que
la
GB
posee
dos
di-
rec-
ciones
de
memo-
ria
que

nece-
si-
ta-
mos
vari-
ables
guardadas
en
la
OAM,
las
cuales
más
tarde
la
con-
sola
in-
ter-
preta
y
mues-
tra.
El
prob-
lema
con
el
que
nos
va-
mos
a
topar
en
cuanto
teng-
amos
dis-
tin-
tas
en-
ti-
dades
ac-
tu-
al-
izán-
dose
si-
multánea-
mente,
es
el
de
que
para
hac-
erlo
nece-
si-
ta-
mos
es-
tar
den-
tro
del
rango
V-
Blank
(ya
que,
de
otra
forma,
la

a
la
con-
clusión
de
que
hay
que
re-
hac-
erlo.
Aún
así,
no
está
de
más
de-
jar
por
es-
crito
el
pro-
ceso
que
se
ha
seguido:

^[h]
[width=0.6]include/images/desarrollo/hud1.png

Primera
It-
eración
del
HUD

Lo
primero
que
hace-
mos,
al
igual
que
con
los
ma-
pas
de
fondo,
es
hac-
er-
nos
nue-
stros
pro-
pios
tilemaps
con
las
ven-
tan-
as
que
quer-
e-
mos
mostrar.
El
prob-
lema
con
el
que
nos
va-

etc.
Por
ello,
a
la
hora
de
ju-
gar,
se
podía
ver
cómo
el
en-
e-
migo
so-
la-
mente
tomaba
de-
ci-
siones
en
acabar
la
ronda
del
ju-
gador,
lo
que
además
provo-
caba
una
ex-
pe-
ri-
en-
cia
lenta.

Con
el
mod-
elo
ECS,
sin
em-
bargo,
he
po-
dido
in-
de-
pen-
dizar
to-
das
las
tar-
eas
en
sis-
temas
y
mane-
jadores
de
datos,
con
lo
que
puedo
ac-

sprites
jun-
tos),
cal-
cu-
lar
las
posi-
ciones
de
pan-
talla
de
los
cu-
a-
tro.
Tam-
bién,
con
el
número
de
tile,
de-
bía
in-
cre-
men-
tarlo
y
ponérselo
al
sprite
que
to-
case.
Esto
se
tradujo
en
un
trozo
de
código
poco
leg-
i-
ble
y
que
a
la
larga
puede
suponer
prob-
le-
mas,
pero
de
mo-
mento
me
sirve.
Con
el
pun-
tero
de
la
úl-
tima
posi-
ción
donde
puedo