

Adjusting Ultima V SAVED.GAM

Richard Todd – 2014-01-01

I don't like keeping track of my food and stuff like that. So, I want to make a simple script to adjust my U5 saved game so that I'm never hungry or stuck in the dark, or whatever.

Inventory

My food, gold, keys, gems, and torches are at 0x202, 0x204, 0x206, 0x207, and 0x208, respectively. The food and gold max out at 9999 (0x270F), while the others max out at 99 (0x63).

```
% file: u5sav.pl
% vim : filetype=prolog :
write_9999( Strm ) :-
    put_byte( Strm, 0x0f ),          % little endian..
    put_byte( Strm, 0x27 ).

write_99( Strm ) :-
    put_byte( Strm, 0x63 ).

fixup_inv( Strm ) :-
    seek( Strm, 0x202, bof, _ ),
    write_9999( Strm ),              % food
    write_9999( Strm ),              % gold
    write_99( Strm ),                % keys
    write_99( Strm ),                % gems
    write_99( Strm ).                % torches
```

Health

It would be nice to be of fine health, would it not? For this, I need to open up and read everyone's maximum HP, then write to the save game file everyone's current HP to match.

```
fixup_hp( Strm ) :-
    open( "C:\\GOG Games\\Ultima456\\Ultima 5\\SAVED.BAK",
        read, Bak, [ type( binary ) ] ),
    read_hps( Bak, Hps ),
    close( Bak ),
    write_hps( Strm, Hps ).
```

So, to read the maximum HPs for everyone, I just need to go to the correct offsets in the file. Character records are at the front of the file (position 0x02), and each one is 32 (0x20) bytes long. The maximum HP is at offset 0x12.

```
read_hps( Strm, Hps ) :-
    seek( Strm, 0x14, bof, _ ),
    read_n_hps( Strm, 16, Hps ).
```

```

read_n_hps( Strm, N, [ Max | Rest ] ) :-
    N > 0,
    get_byte( Strm, LowByte ),
    get_byte( Strm, HighByte ),
    Max = word( LowByte, HighByte ),
    N1 is N - 1,
    seek( Strm, 0x1E, current, _ ),
    read_n_hps( Strm, N1, Rest ).
read_n_hps( _, 0, [ ] ).

```

Writing out the max HPs is pretty much the same process, only the offset is 0x10 within each character record. Also, we're going to set their status to 0'G for Good. The status is at offset 0x0B in every character record.

```

write_hps( Strm, Hps ) :-
    seek( Strm, 0x0D, bof, _ ),
    write_n_hps( Strm, 16, Hps ).

write_n_hps( Strm, N, [ word( LowByte, HighByte ) | Rest ] ) :-
    N > 0,
    put_byte( Strm, 0'G ),
    seek( Strm, 0x04, current, _ ),
    put_byte( Strm, LowByte ),
    put_byte( Strm, HighByte ),
    N1 is N - 1,
    seek( Strm, 0x19, current, _ ),
    write_n_hps( Strm, N1, Rest ).
write_n_hps( _, 0, [ ] ).

```

Spells and Potions

At offset 0x24A we have 64 bytes representing our spells and potions. Let's max them out!

```

fixup_spells( Strm ) :-
    seek( Strm, 0x24A, bof, _ ),
    write_n_99s( Strm, 64 ).

write_n_99s( Strm, N ) :-
    N > 0,
    N1 is N - 1,
    write_99( Strm ),
    write_n_99s( Strm, N1 ).
write_n_99s( _, 0 ).

```

Skull Keys

I'd also like to never run out of skull keys... These are at 0x20B.

```

fixup_keys( Strm ) :-
    seek( Strm, 0x20B, bof, _ ),
    write_99( Strm ).

```

The Script Main

Now, we just need to copy the file, open it, and open the output file:

```
:- use_module( library( filesex ) ). % for copy_file
main :-
    copy_file( "C:\\GOG Games\\Ultima456\\Ultima 5\\SAVED.GAM",
               "C:\\GOG Games\\Ultima456\\Ultima 5\\SAVED.BAK" ),
    open( "C:\\GOG Games\\Ultima456\\Ultima 5\\SAVED.GAM",
          update, File, [ type( binary ) ] ),
    fixup_inv( File ),
    fixup_hp( File ),
    fixup_spells( File ),
    fixup_keys( File ),
    close( File ),
    halt.
```