

 Fabrizio-Caruso Simplify and update docs

acc27ba · 6 months ago



65 lines (42 loc) · 3.42 KB

Preview

Code

Blame

Raw











HARDWARE-AGNOSTIC CODE

Thanks to Cross-Lib APIs, the game code can be *hardware-agnostic*. Using only Cross-Lib APIs is not enough, though, as Cross-Lib covers targets that have different screen size, screen shape and targets that lack some hardware features such as graphics, colors, sounds.

ANSI C (A sub-set of)

The code has to be compiled by several different compilers that do not necessarily support the latest version of the C standard. Therefore it should be written in a sub-set of ANSI C89 that is common among all supported compilers.

This sub-set can be described as (mostly) ANSI C89 without:

- `float` types;
- `struct` copies, assignments, parameters (use pointers to `struct` instead);
- dynamically allocated memory (`malloc` , `free`).

Moreover for performance reasons it is better to avoid recursion (as this has an extreme high cost on MOS 6502 targets).

Some C99 features are available:

- Modern `//` comments are allowed.
- `stdint.h` is not available but `uint8_t` and `uint16_t` are (with no need for extra include directive).

Screen size

Using Cross-Lib APIs is not enough as the screen size and shape may be different. Cross-Lib exposes a constant `XSize` for the width of the screen and a constant `YSize` for the height of the screen. Both these macros are measured in terms of numbers of *tiles* (see next section). The game code should rely on fractions of `XSize` and of `YSize` and never use hard-coded sizes or use conditional directives to cover all sizes. By doing so, upon compilation, the game will auto-adapt to the target's screen size.

Tile-based Graphics

Graphics in Cross-Lib is tile-based as this is the only possible graphics that can be supported by all targets. For targets with graphics, the shapes of tiles are only defined at compilation time and cannot be re-defined. So, for targets with graphics, smoothly moving sprites can only be implemented as software sprites through pre-shifted tiles.

Most targets with graphics have 8x8 pixel tiles but some other targets with graphics have different shapes. So do not implement logic that only assumes 8x8 pixel tiles (e.g. when implementing software sprites with pre-shifted tiles). If you want to support targets with no graphics (i.e., tiles are just mapped to ASCII characters), you either avoid code that depends on tile shapes or if you, you have to implement an alternative version for non-graphics targets.

For example for the game Cross Snake you can see how it is rendered on the MSX 1 (graphics, sounds, colors) and on the Game Boy (graphics, sounds but no colors):



For more snapshots we refer to:

<https://github.com/Fabrizio-Caruso/CROSS-LIB/blob/master/docs/SNAPSHOTS.md>

Colors, Sounds and Graphics

Some targets have no colors or no color on text, or no sounds, or no graphics. So do not write any logic that relies only on the presence of colors/sounds/graphics; or if you do, use conditional directives to implement an alternative logic for targets with no colors/sounds/graphics.

Learn from the built-in games and tests

Cross-Lib comes with games and tests whose code can be used to learn how to code universal games.

The code of the games is in:

<https://github.com/Fabrizio-Caruso/CROSS-LIB/tree/master/src/games>

The code of the examples is in:

<https://github.com/Fabrizio-Caruso/CROSS-LIB/tree/master/src/examples>

