



HELLO WORLD

Game Development Using TIC-80

BY CHAITANYA KOPARKAR

TIC-80 (<https://tic80.com>) is a fantasy computer for making, playing, and sharing tiny games. It is referred to as a “fantasy” computer because it is essentially a software program that emulates a hardware device with its own display, sound, memory, storage, and operating system. TIC-80 is modeled after gaming consoles from the 1980s and can be used to make and play retro-styled games. For example, its display only has 240x136 pixels and supports only 16 colors, the amount of memory available is very limited compared to modern standards, and so on.

While these hardware specifications may seem very limiting, they’re actually very freeing since the device itself limits the size of the game that can be created using TIC-80. This lets us go back to the basics by creating small and fun games, without getting bogged down by the size, complexity, and effort required to create competent games using modern game development frameworks. Plus, TIC-80 is sufficiently capable of running some incredible games like Witch’em Up, a clone of Mario Bros, and much more.

TIC-80 makes it especially easy to

get started with game development since it lets one do every aspect of the development work within the computer itself. It provides built-in tools to write code, draw sprites,¹ create music and sound effects, etc. The end result is a “cartridge file” for the game that can be played on a computer or even on a web browser.

Let’s briefly look at some of these tools and look at the code of a small game, akin to a “Hello World” program for other languages. For more details, please refer to the TIC-80 website and the wiki page.²

HOW IT WORKS

TIC-80 can be downloaded or used in the browser directly. Once TIC-80 boots up, it displays a command-line interpreter as shown in Figure 1. This is similar to a Linux shell, and has commands such as `cd`, `ls`, etc. One can type “help” or “help commands” to list all the available commands. For our purposes, we’ll explore the Hello World cartridge that

can be created using the “new” command followed by the name of the programming language needed to write the code.

First type “new js” to create a Javascript cartridge followed by “run” to run the game, as shown in Figure 2. This game is intentionally simple and only lets us move the TIC-80 logo around the screen using arrow keys. If we press ESC and close the game and press ESC again, we should see the Code Editor as shown in Figure 3. This editor can be used to write the game code³ directly.

The complete source code for the Hello World game is shown in Listing 1. Let us go over some aspects of it. The code starts with special comments that are required to store metadata information, most important of which is to identify the programming language used. Next, the function `TIC()` is the main game loop. This function runs 60 times a second and is used to update the “game state” and draw objects on the screen as necessary. In this example, this function starts by detecting a button press and updates the x and y coordinates of the TIC-80 logo accordingly. The functions `cls` clears (or re-colors) the screen and `spr` draws a specific sprite on the screen. The first argument to `spr` is a bit complicated but serves the purpose of doing a blinking animation. Essentially, it switches between two different sprites at certain time intervals.

This is the crux of developing a game on TIC-80. The game loop is the heart of the game and generally follows a similar template:

1. Check for user input
2. Update game state
3. Re-draw game graphics

Using this basic idea and the other TIC-80 functionality, it is possible to develop a variety of interesting and fun games.

³ The PRO version of TIC-80 also lets one use an external text editor to write code, but we won’t go over that process in this article.

Listing 1. Code for the Hello World game on TIC-80.

```
// title:   game title
// author:  game developer, email, etc.
// desc: short description
// site: website link
// license: MIT License (change this to your license of choice)
// version: 0.1
// script: js

var t=0
var x=96
var y=24

function TIC()
{
    if(btn(0))y--
    if(btn(1))y++
    if(btn(2))x--
    if(btn(3))x++

    cls(13)
    spr(1+((t%60)/30|0)*2,x,y,14,3,0,0,2,2)
    print("HELLO WORLD!",84,84)
    t++
}
```

Figure 1. TIC-80 command prompt.

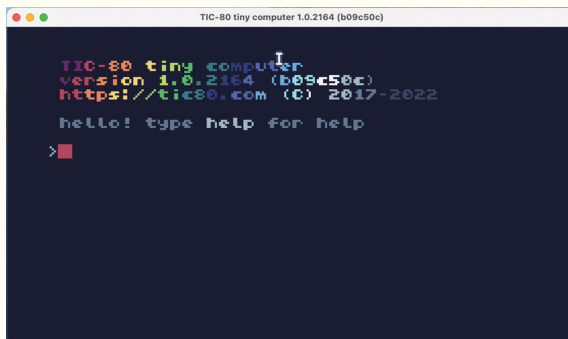


Figure 2. The Hello World game on TIC-80.

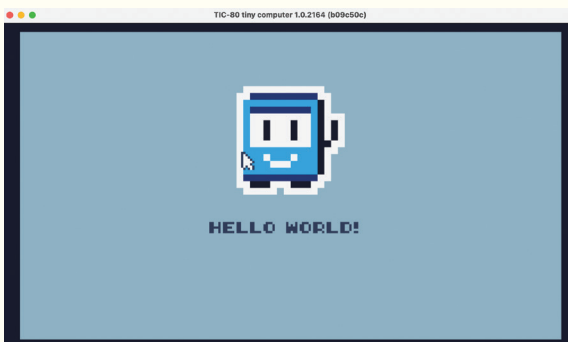


Figure 3: Code for the Hello World game.



FURTHER READING

This article only scratches the surface of what's possible with TIC-80; its purpose is to serve as an introduction. There is much more to know to effectively write games for TIC-80 such as its RAM and VRAM layout, creating sprites and game maps, doing animations, other available TIC-80 API functions, and so on. You can find more information by visiting the TIC-80 wiki page. The API section describes all the

core TIC-80 functions and what they do. There is a tutorials section,⁴ which contains many posts that review the code of different games, and some delve into specific aspects such as game mechanics, physics, and other advanced techniques.

⁴ <https://github.com/nnesbox/TIC-80/wiki/tutorials>

DOI: 10.1145/3611690
Copyright held by the author.



Digital Threats: Research and Practice

Digital Threats: Research and Practice (DTRAP) is a peer-reviewed journal that targets the prevention, identification, mitigation, and elimination of digital threats. DTRAP aims to bridge the gap between academic research and industry practice. Accordingly, the journal welcomes manuscripts that address extant digital threats, rather than laboratory models of potential threats, and presents reproducible results pertaining to real-world threats.



For further information
and to submit your
manuscript,
visit dtrap.acm.org