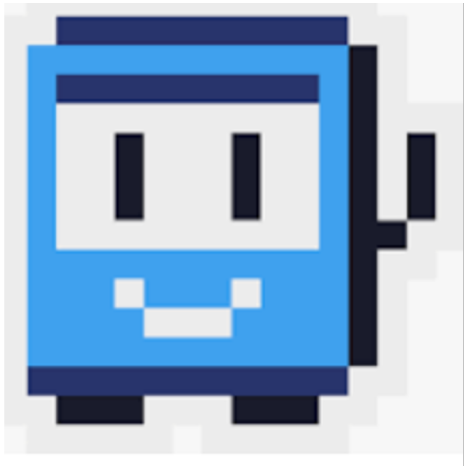


Programming Is Magic

Learn to code making a videogame with TIC-80

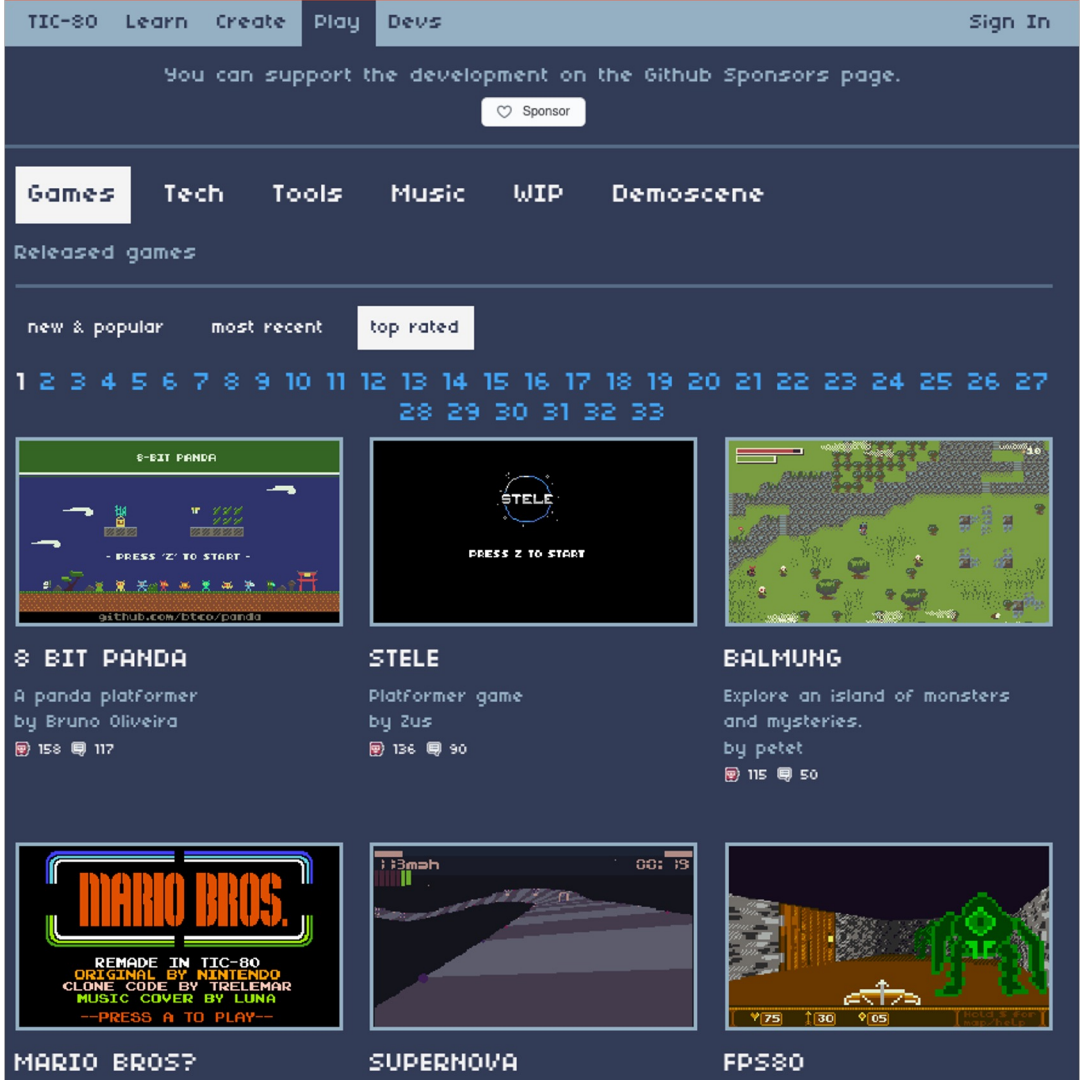
Uberto Barbini

8-bit consoles videogames

What is a virtual console and why TIC80

https://tic80.com/

- Simple to code
- All included
- Fast results
- Runs everywhere
- Free



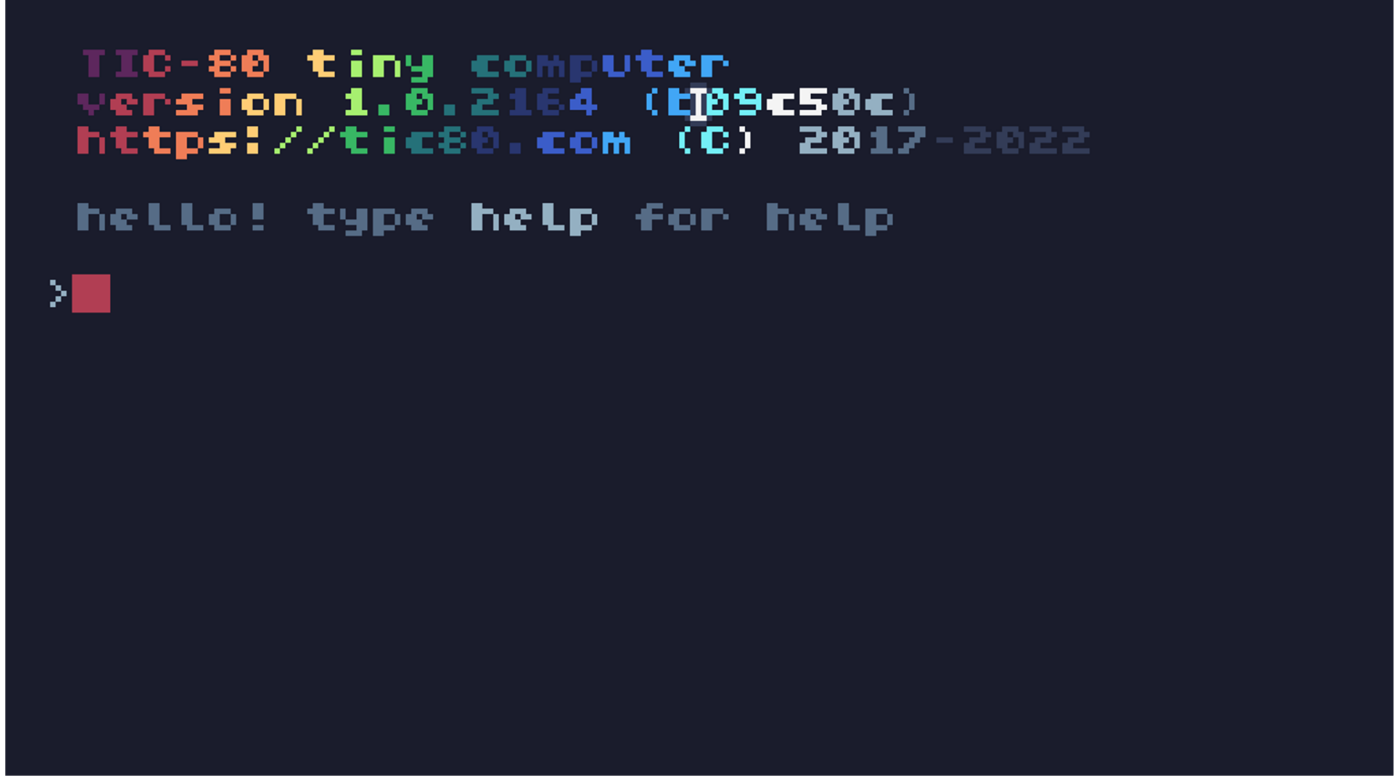
Enough said, let's start

Install TIC-80 in your PC

Run it (double click)
you should see this:

Type **`folder`** to open a file windows and copy **`PIM-initial.tic`** from the usb-stick.

Then type
`load PIM-initial`



load abc -> load the abc.tic game

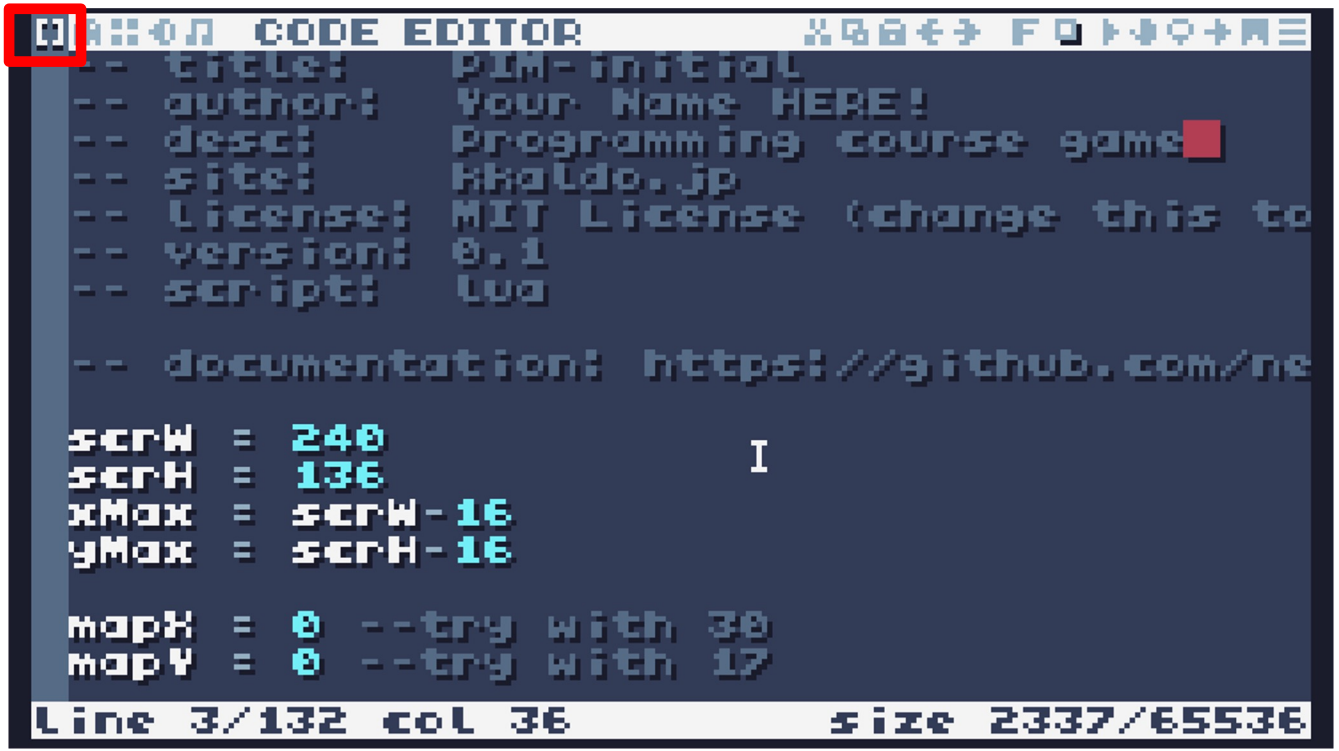
save abc -> save current game

run -> run the current game

surf -> select a game

new lua -> create a new empty game

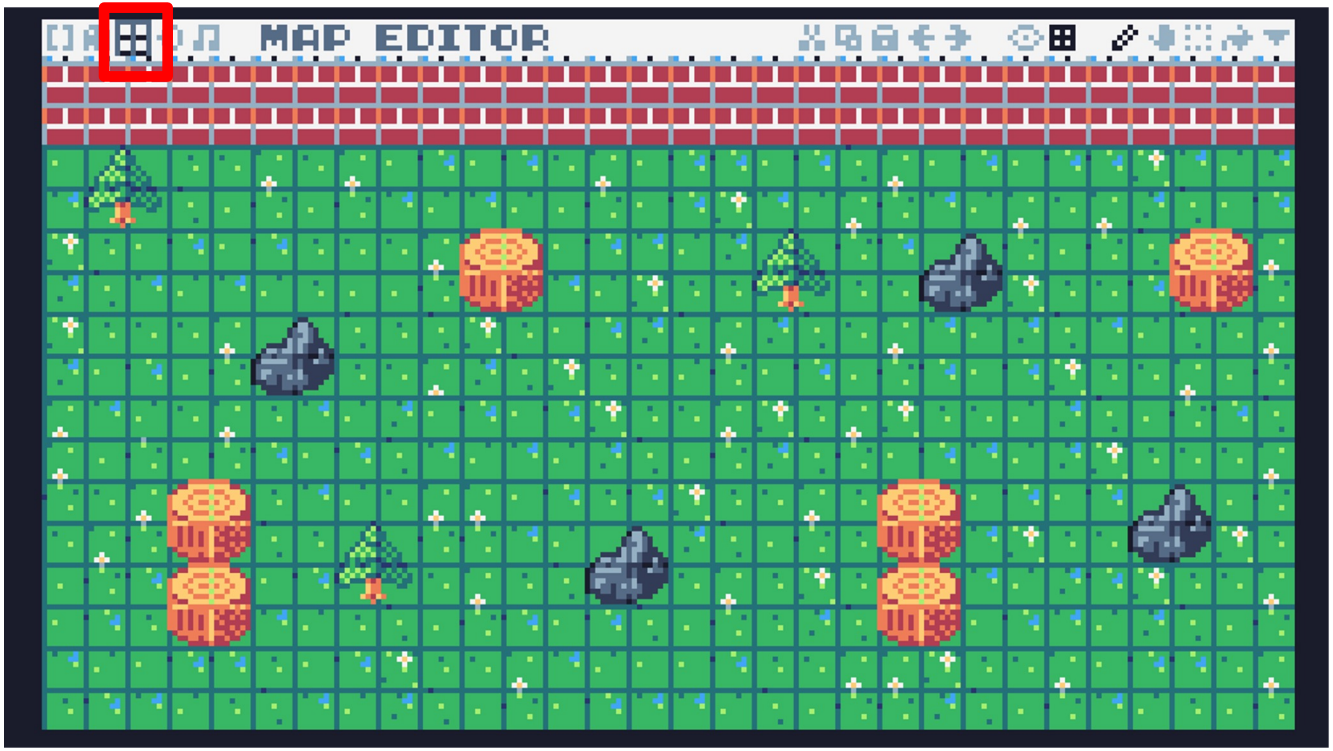
help -> show the help



To show the code editor press F1 or click on the brackets [ ] top left with mouse.
You can change the font, copy-paste, undo-redo, etc.
Everything is self-contained.

F2 or clicking on the ghost top left will show the sprite editor.

Sprites are little images used in games. You can edit, zoom, rotate etc.



F3 or the squares icon on top left will show the map editor. Here you use sprites to create your world. You can zoom in and out and draw with sprites.

F4 or the speaker icon will show the sound effects editor. Here you can define your sounds, creating a sound wave and using a pitch.

F5 or the note icon will show the music editor, to compose music for your games.

Useful shortcuts

Esc -> exit from a running game or show the command line

Ctrl-R -> start a game
F1,F2,F3,F4,F5 -> show the editors

Console:
Arrows Up, Down -> select previous statement
Tab -> autocomplete file names

Code Editor:
Ctrl + / -> comment some code
Ctrl + f -> find in the source
Alt + Enter -> Fullscreen mode on/off
Ctrl + s -> save the game
Ctrl + o -> code navigation
Ctrl + Tab -> indent line of code

Lua language

TIC-80 can use many different languages but the default is Lua.

Lua is a small, elegant language easy to use.

The full tic80 documentation is here: https://github.com/nesbox/TIC-80/wiki

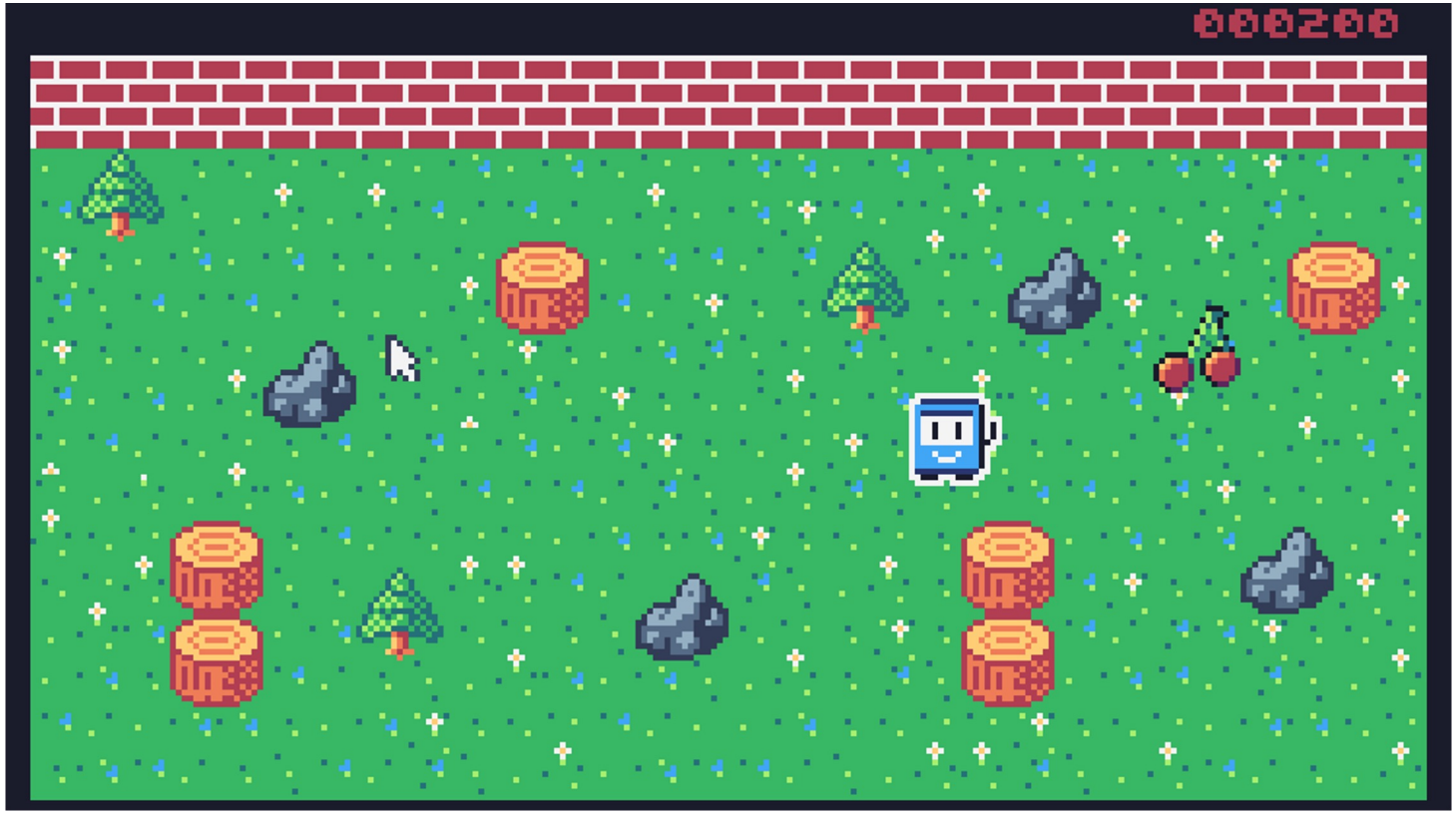
You don't need to understand Lua for the exercises, play around to make it work.

Learn Lua in 15 minutes: https://tylerneylon.com/a/learn-lua/

Cheatsheet: https://devhints.io/lua

Book Programming in Lua: https://www.lua.org/pil/contents.html

The Videogame: eat fruits in the garden



The Exercises

Here's a list of exercises for this course.

First, load the game PIM-initial, then you can get started!

Remember:

- Press F1 to view the code
- Press Ctrl+R to run it
- Press Esc+F1 to stop it

Have fun!

1. **Instruct the Computer Using Functions:**
   Look at the `TIC()` function. Your task is to modify it to change the background color to green (or any color you prefer).
2. **Screen Coordinates X and Y:**
   Update the `TIC()` function to display your name at the center of the screen. Feel free to add multiple lines of text.
3. **Drawing Sprites:**
   Design your sprite (or modify the current one) in the Sprite editor (F2). Then, add this line to the `TIC()` function to display it: `draw(user)`. You can select a different sprite by changing the `sprNum` in the `user` variable.

4. **Animations:**
   An animation is simply an alternation between two sprites. Add this line to your code and adjust your animation as necessary: `anim(user)`.
5. **Moving:**
   Let's make the sprite move when you press the arrow keys. Add `move(user)` to the TIC function. Oops, an error occurred! Can you spot and fix it how to move in the right direction?
   Inspect the code inside the `move()` function to rectify the error. Feel free to modify the speed as well.

6. **Design the World:**
   To draw the background from the map, add `map(mapX, mapY)` to the TIC() function.
   Now you can modify the map in the map editor (F3) and create a new one. Also, experiment with different values for the `mapX` and `mapY` variables to change the background.
7. **Add a Goal:**
   Draw some food in a particular location by adding `draw(food)` to the TIC function.
   However, the `baseFood` variable is pointing to an wrong value!
   Please correct it to display fruits. Feel free to design new food items and change the `sprNum` in the `food` variable.

8. **Eat the Food:**
   Add the ability to eat the food by incorporating `eat(food)` into the TIC function.
   But it's currently silent! Add some sound effects using the `sfx()` command in the `gulp()` function. Start by copying from the `boink()` function and add your own sound effects. You can use more than one sound.
9. **Add a Score:**
   To display the score, add
   `print(string.format("%06d", score),200,0,txtCol)`
   inside the TIC function.
   You can adjust the score for each food by modifying `100` in the following line:
   `score = score + (food.sprNum-baseFood)*100`.

Continuation

You can continue from now on your own, these are just some ideas on how to continue:

- Add multiple foods at same time. Create new variable food2 and call eat(food2) in TIC.
- Add a timer that counts down and show it on the top left. To do this create a variables and decrease by one in TIC(). When time go to zero the game ends.
- Earn more time by eating fruits, change it in the eat() function.
- Implement a title screen and a main menu to start a new game.
- Upload your game online and play it on your phone.