Start Creating a Game From Scratch

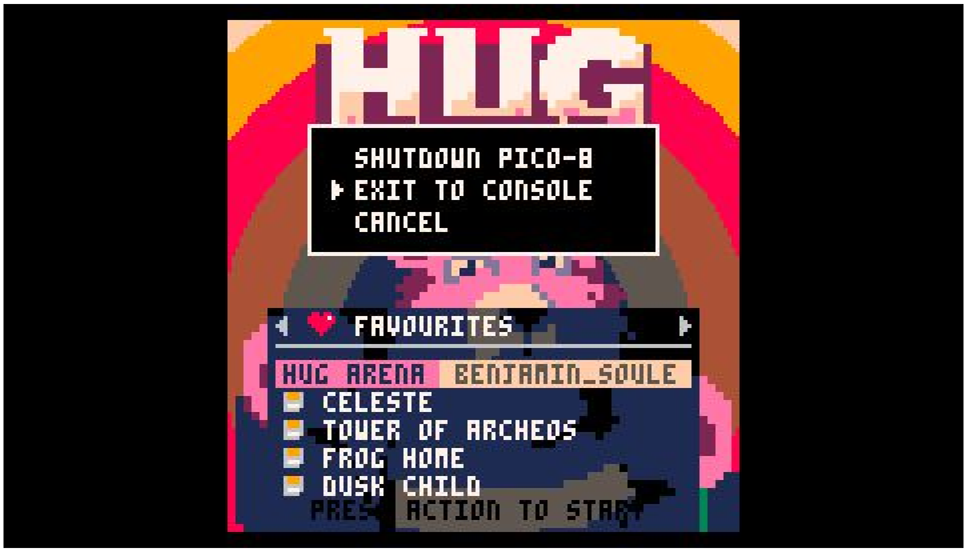
Duration: 1 hour
This workshop starts shows the very beginnings of creating a game from scratch in Pico-8. It explores how to create a sprite character and how to move it using the D-pad.

Introduction: 10 mins

In this short example we will create a new project, draw a sprite, write some code to move our sprite around the screen and save our work. The overarching process can be applied to create a full game that includes a background, music and multiple sprites.

Create & Save New Project: 5 mins

The first thing to do is to create a new project that can be saved as the game is built. From the ‘**splore menu screen**, press **ESC** and choose “**EXIT TO CONSOLE**”.



Once in the console reboot Pico-8:

```
REBOOT
```

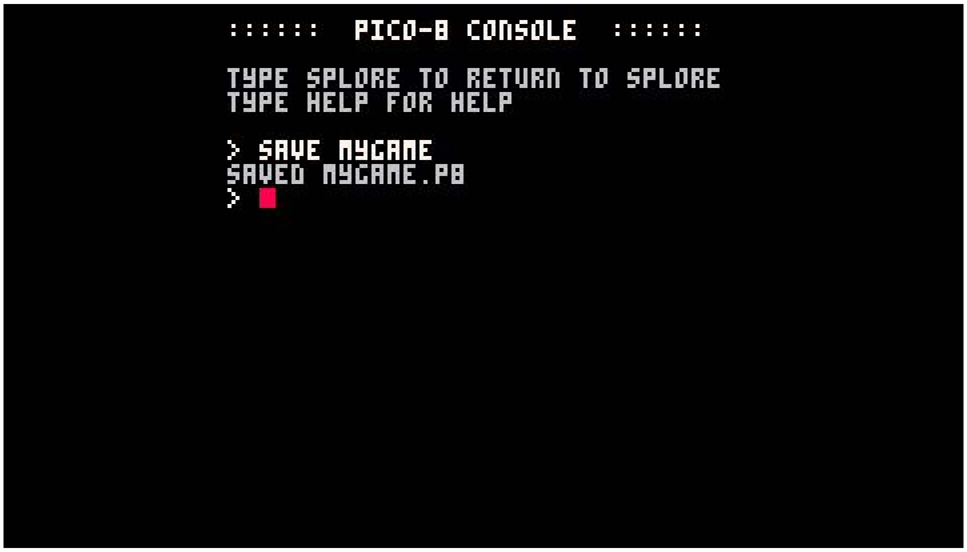
Tip! The reboot (restart) will clear Pico-8's memory which essentially ejects a game that you were playing beforehand. If you were playing or editing a game and do not reboot the next SAVE command will rename and save the code file of that game instead of creating a new blank file.

To check what game is loaded and will be effected press **ESC** to toggle in between the **editors** and **console**. If the editors are empty you are good to go. If not, execute the **REBOOT** command in the console.

Choose a project name and save it by typing:

```
SAVE [PROJECTNAME]
```

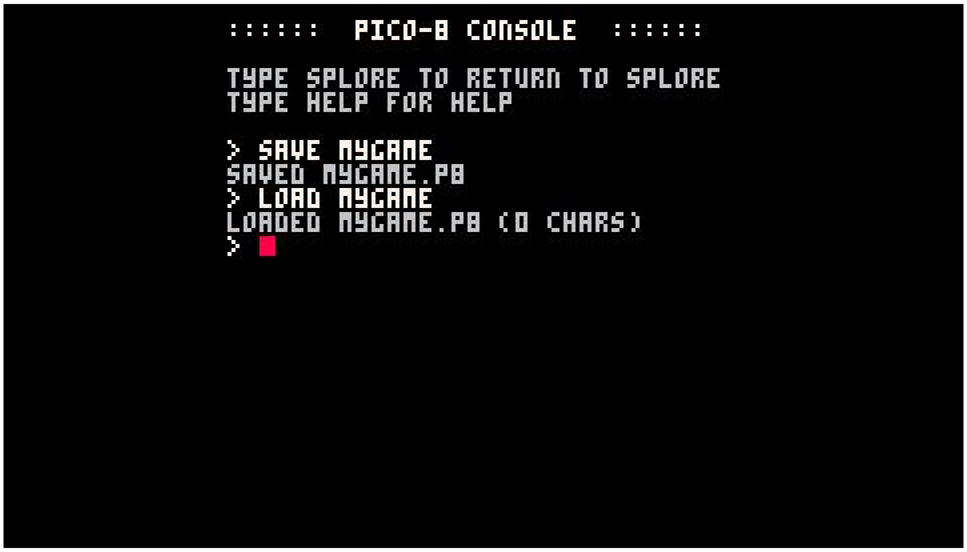
In my case I called it "MYGAME" (Pico-8 automatically capitalizes all commands since Lua syntax calls for it and is case-sensitive).



```
:::::: PICO-8 CONSOLE ::::::

TYPE SPLORE TO RETURN TO SPLORE
TYPE HELP FOR HELP

> SAVE MYGAME
SAVED MYGAME.P8
> 
```

Next, load the newly created project with:

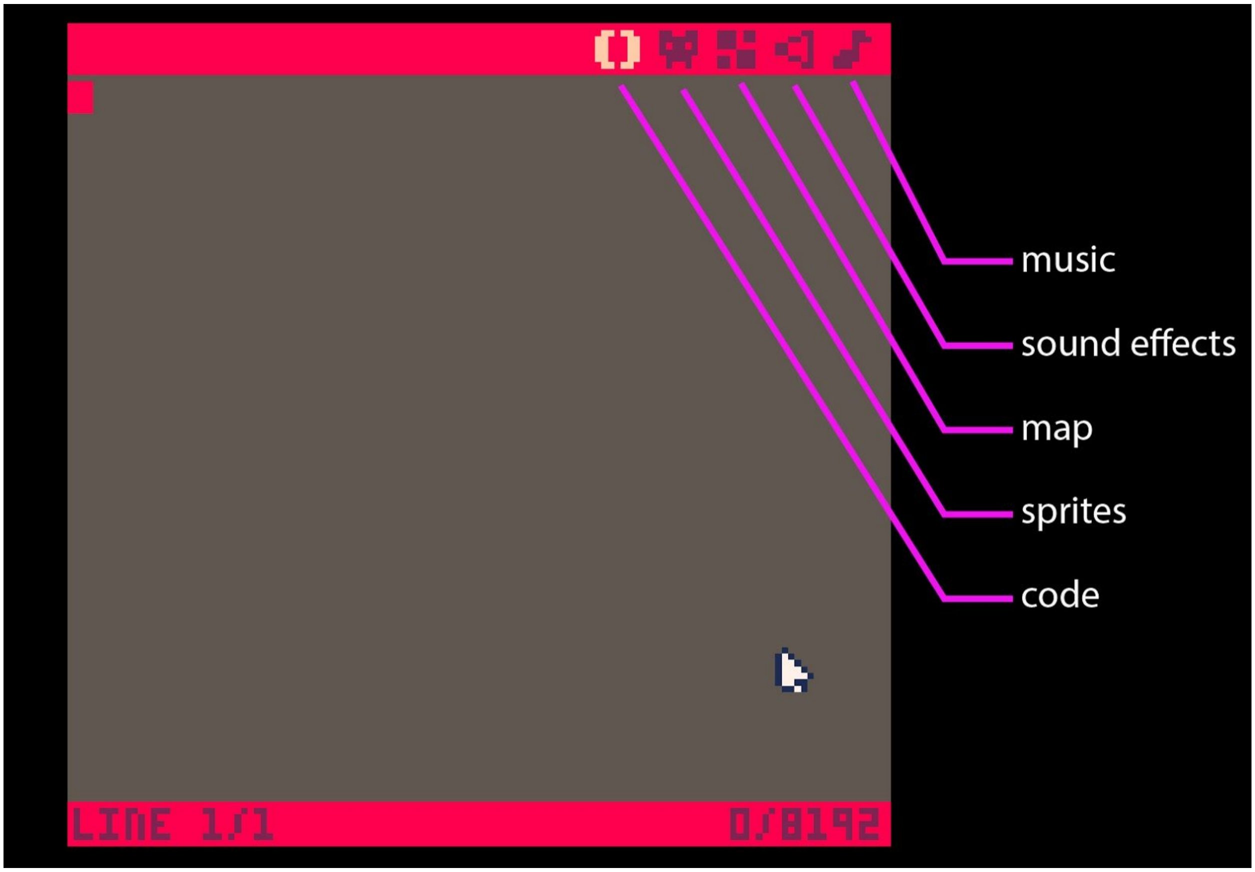
```
LOAD [PROJECTNAME]
```



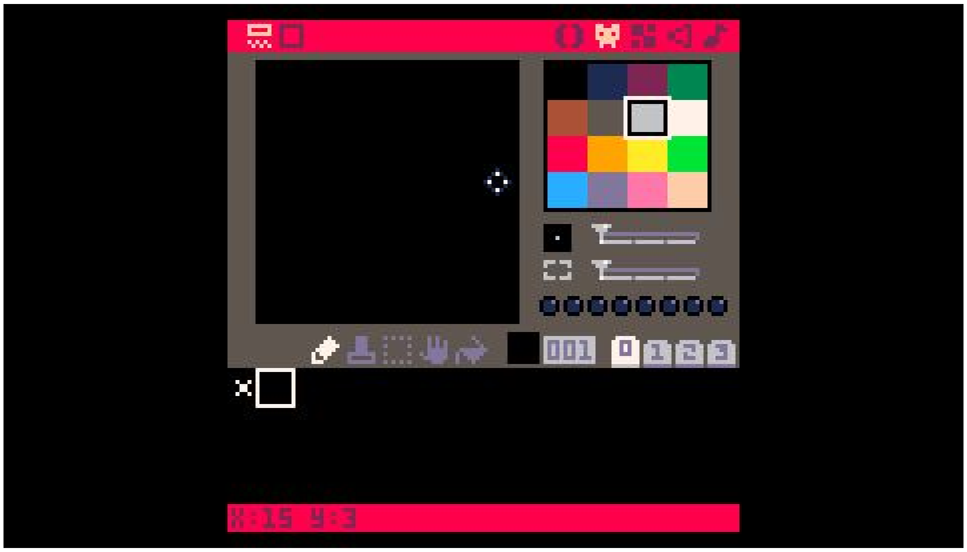
```
:::::: PICO-8 CONSOLE ::::::

TYPE SPLORE TO RETURN TO SPLORE
TYPE HELP FOR HELP

> SAVE MYGAME
SAVED MYGAME.P8
> LOAD MYGAME
LOADED MYGAME.P8 (0 CHARS)
> 
```

Create Sprite: 20 mins

Now that a new project has been created and loaded it’s time to start building. To create a sprite press **ESC** which will drop you into the **Code Editor**.

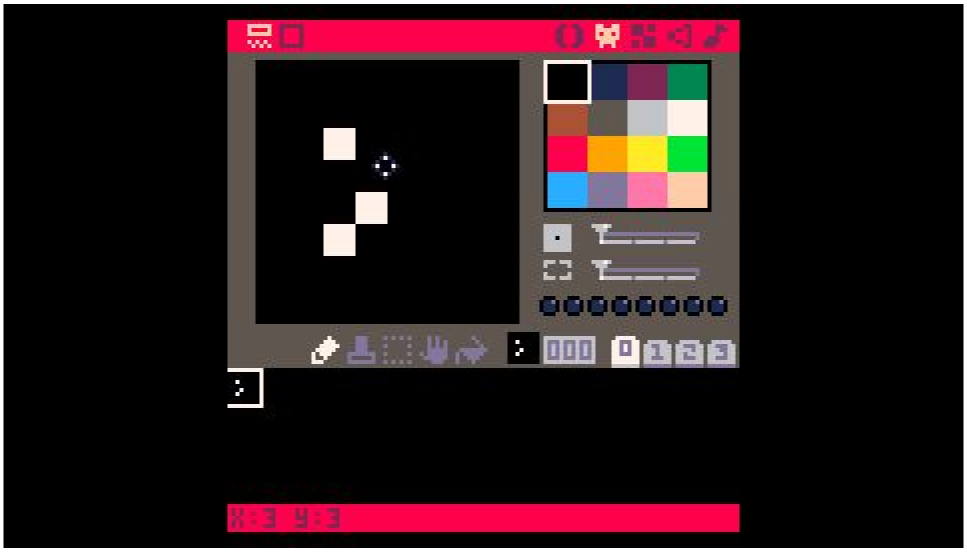


From the Code editor, choose the **Sprite editor** by tapping the appropriate icon in the upper right corner.

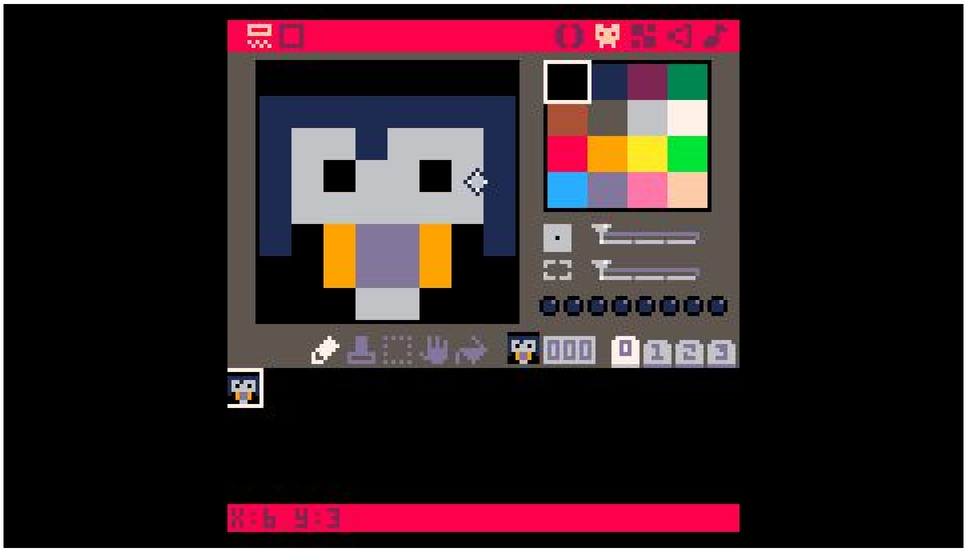


In the Sprite Editor a default “X” design of white pixels can be seen in the upper left corner of the sprite sheet. Each “cell” in the sprite sheet has a corresponding number, this is the sprite’s index number which is used in code to reference that specific design. The “X” is in cell 000 and from left to right it goes up: cell 001, cell 002, etc.

To start creating a custom sprite, **choose an empty cell** or **erase the “X”** in cell 000. To erase, select the cell by tapping to display it in the editor window. Erase it by selecting black and then drawing over it with the Draw tool. For this example, I am erasing and drawing my sprite in cell 000.



Draw a new sprite.



When you are done creating your character save your work by pressing **CTRL+S**. A message of “SAVED [PROJECTNAME.P8]” will appear at the bottom.

Write Code: 15 mins

Now, let’s write some code that will move our sprite across the screen. Go to the **Code editor** by tapping the () icon. To draw the new sprite on the screen and map the D-pad keys to movement, type:

```
X = 64
Y = 64
FUNCTION _UPDATE()
        IF (BTN(0)) THEN X=X-1 END
        IF (BTN(1)) THEN X=X+1 END
        IF (BTN(2)) THEN Y=Y-1 END
        IF (BTN(3)) THEN Y=Y+1 END
END

FUNTION _DRAW()
        RECTFILL(0,0,127,127,5)
        SPR(000,X,Y)
 END
```

Lua isn’t white space sensitive, so formatting the code with tabs and spaces is up to you.

What’s going on in the code.

Create variables to hold an X and Y value.

```
X = 64
Y = 64
```

Pico-8 has a game loop that is used in the source code of every game. The game loop updates the display 30 times per second making each game play at 30 frames per second. The first part of the game loop is FUNCTION _UPDATE(). In this function lives all the logic for the next frame including user input which is what we are using it for. Read more about the FUNCTION _UPDATE() here.

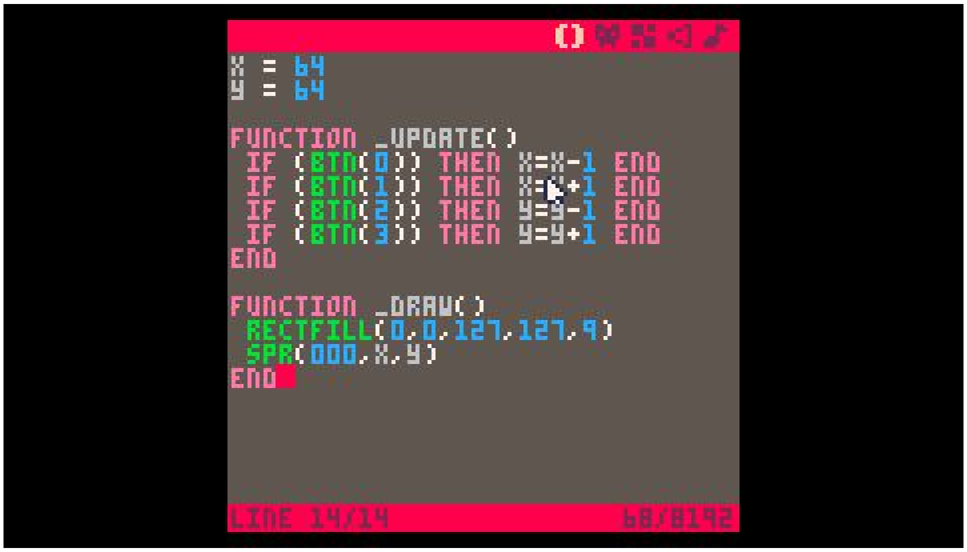
```
FUNCTION _UPDATE()
       IF (BTN(0)) THEN X=X-1 END
       IF (BTN(1)) THEN X=X+1 END
       IF (BTN(2)) THEN Y=Y-1 END
       IF (BTN(3)) THEN Y=Y+1 END
END
```

Within the function we are saying that if **Left** is pressed take whatever is drawn at X and move it one space to the Left. This is repeated for **Right**, **Up** and **Down** all moving one in their respective directions instead of Left. Each button is assigned an **button number** which can be found here along with documentation for the **btn() function**.

The _DRAW() function is where all the visuals are drawn to the screen each frame. Here we draw a rectangle that fills the screen (upper left corner at 0,0 and lower right corner at 127,127) and fills it with the color orange (5). The color number references the Pico-8 color palette. The numbers can also be learned from the manual under Graphics.

Next, we draw a sprite in the center of the screen (64,64 because the screen is 128 x 128 pixels) using the 000 cell to reference it.

```
FUNTION _DRAW()
       RECTFILL(0,0,127,127,5)
       SPR(000,X,Y)
 END
```

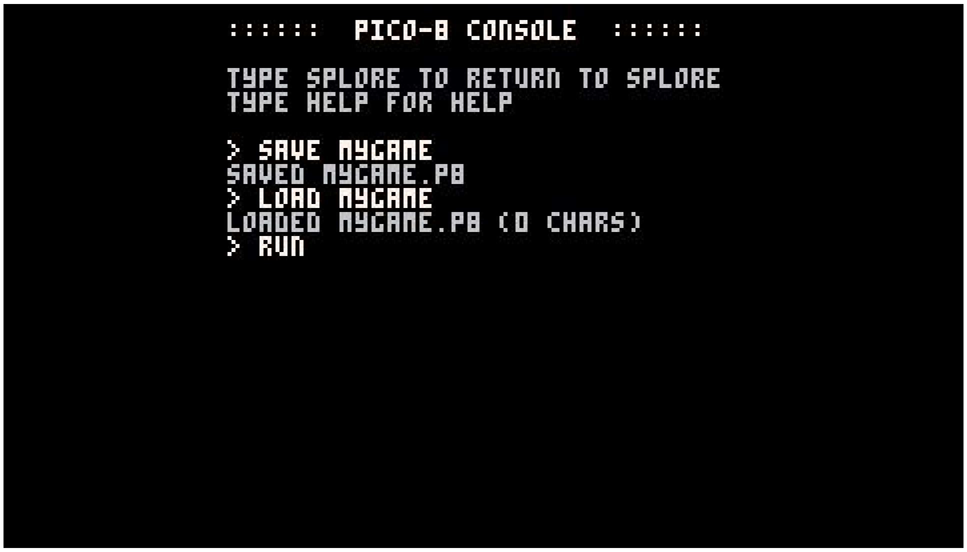


```
X = 64
Y = 64

FUNCTION _UPDATE()
 IF (BTN(0)) THEN X=X-1 END
 IF (BTN(1)) THEN X=X+1 END
 IF (BTN(2)) THEN Y=Y-1 END
 IF (BTN(3)) THEN Y=Y+1 END
END

FUNCTION _DRAW()
 RECTFILL(0,0,127,127,9)
 SPR(000,X,Y)
END
```

```
LINE 14/14                    68/8192
```

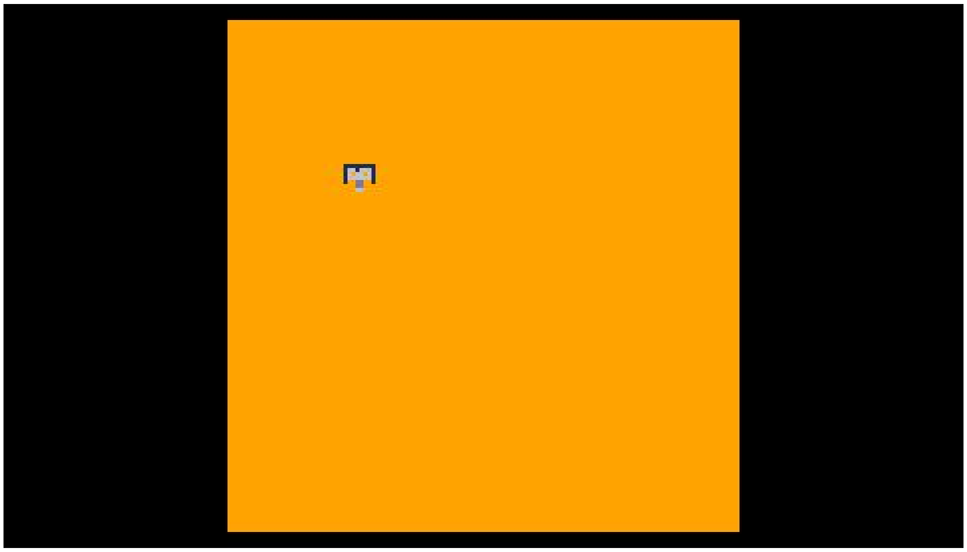
- Save with **CTRL+S** and switch to the **console** by pressing **ESC**.

- Execute the code written in the Code editor by typing “**RUN**” then pressing **return**.



```
:::::: PICO-8 CONSOLE ::::::

TYPE SPLORE TO RETURN TO SPLORE
TYPE HELP FOR HELP

> SAVE MYGAME
SAVED MYGAME.P8
> LOAD MYGAME
LOADED MYGAME.P8 (0 CHARS)
> RUN
```

Move the sprite around using the four directions of the D-pad.



Congrats! The next step would be to create a map and maybe some enemies for your new character. Check out the resource section below for more tutorials.

Pico-8 Resources

Youtube videos

Pico-8 fanzine

Pico-8 manual

Pico-8 website

Pico-8 Wiki

NTC Pico-8 Blog Post