

# TIC-80 workshop

## Space Shooter

So Koide - Jan 26th, 2024 - Lightning Talks Special Event

# AGENDA

Everyone) Intro



1. Install TIC-80 at <https://github.com/nesbox/TIC-80/releases>
2. Download **TIC-80 cheat sheet.pdf** at <https://github.com/sokoide/tic80-cheatsheet>
3. Quick first game (help, folder, new, save load)

Optional) Step-by-step space shooter tutorial or make your own game

# INSTALL

- <https://github.com/nesbox/TIC-80> → Releases or build it

▼ Assets	12
<a href="#">tic80-v1.1-3ds.zip</a>	2.22 MB    2 weeks ago
<a href="#">tic80-v1.1-android.apk</a>	12.9 MB    2 weeks ago
<a href="#">tic80-v1.1-html.zip</a>	1.95 MB    2 weeks ago
<a href="#">tic80-v1.1-linux.deb</a>	3.49 MB    2 weeks ago
<a href="#">tic80-v1.1-mac.dmg</a>	2.99 MB    2 weeks ago
<a href="#">tic80-v1.1-rpi-baremetal.zip</a>	4.08 MB    2 weeks ago
<a href="#">tic80-v1.1-rpi.deb</a>	4.23 MB    2 weeks ago
<a href="#">tic80-v1.1-rpi4-baremetal.zip</a>	3.72 MB    2 weeks ago
<a href="#">tic80-v1.1-win.zip</a>	2.89 MB    2 weeks ago
<a href="#">tic80-v1.1-winxp.zip</a>	2.41 MB    2 weeks ago
<a href="#">Source code (zip)</a>	2 weeks ago
<a href="#">Source code (tar.gz)</a>	2 weeks ago

# CHEAT SHEET

- <https://github.com/sokoide/tic80-cheatsheet>

TIC-80 Cheat Sheet																																																																																																																																																																	
<ul style="list-style-type: none"> <li>DISPLAY 240x136 pixels, 16 color palette</li> <li>INPUT 4 gamepads with 8 btns/mouse/kbd</li> <li>SPRITES 256 8x8 tiles &amp; 256 8x8 sprites</li> <li>MAP 240x136 cells, 1920x1088 pixels</li> <li>SOUND waveforms x 4 channels</li> </ul>																																																																																																																																																																	
<b>Memory Map, Key &amp; Button Codes</b>																																																																																																																																																																	
<table border="1"> <thead> <tr> <th>ADDR</th><th>INFO</th><th>BYTES</th><th>COMMENTS</th><th></th><th></th></tr> </thead> <tbody> <tr><td>0</td><td>Screen</td><td>16320</td><td>240x136 x 4bit cl</td><td></td><td></td></tr> <tr><td>3FC0</td><td>Palette</td><td>48</td><td>3(RGB) x 16 cls</td><td></td><td></td></tr> <tr><td>3FF0</td><td>Palette Map</td><td>16</td><td>Palette map</td><td></td><td></td></tr> <tr><td>3FF8</td><td>Border Color</td><td></td><td></td><td></td><td></td></tr> <tr><td>3FF9</td><td>Scrn Offset</td><td>2</td><td>X, Y offset</td><td></td><td></td></tr> <tr><td>3FFC</td><td>Mouse Sprite</td><td>1</td><td></td><td></td><td></td></tr> <tr><td>3FFC</td><td>RESERVED</td><td>4</td><td></td><td></td><td></td></tr> <tr><td>4000</td><td>Tiles</td><td>8192</td><td>256 8x8 4bit BGs</td><td></td><td></td></tr> <tr><td>6000</td><td>Sprites</td><td>8192</td><td>256 8x8 4bit SPRs</td><td></td><td></td></tr> <tr><td>8000</td><td>Map</td><td>32640</td><td>240x136 map chips</td><td></td><td></td></tr> <tr><td>FF80</td><td>Gamepads</td><td>4</td><td>Btn state</td><td></td><td></td></tr> <tr><td>FF84</td><td>Mouse</td><td>4</td><td>Mouse state</td><td></td><td></td></tr> <tr><td>FF88</td><td>Keyboard</td><td>4</td><td>Kbd state</td><td></td><td></td></tr> <tr><td>FF8C</td><td>SFX State</td><td>16</td><td></td><td></td><td></td></tr> <tr><td>FF9C</td><td>Sound Reg</td><td>72</td><td></td><td></td><td></td></tr> <tr><td>FFE4</td><td>Waveforms</td><td>256</td><td>16 waves</td><td></td><td></td></tr> <tr><td>100E4</td><td>SFX</td><td>4224</td><td></td><td></td><td></td></tr> <tr><td>11164</td><td>Music Paths</td><td>11520</td><td></td><td></td><td></td></tr> <tr><td>13E64</td><td>Music Tracks</td><td>408</td><td></td><td></td><td></td></tr> <tr><td>13FFC</td><td>Sound St</td><td>4</td><td></td><td></td><td></td></tr> <tr><td>14000</td><td>Stereo Vol</td><td>4</td><td></td><td></td><td></td></tr> <tr><td>14004</td><td>Persist Mem</td><td>1024</td><td>Persistent RAM</td><td></td><td></td></tr> <tr><td>14404</td><td>Sprite Flags</td><td>512</td><td></td><td></td><td></td></tr> <tr><td>14604</td><td>System Font</td><td>2048</td><td>256 8x8 1bit font</td><td></td><td></td></tr> <tr><td>14E04</td><td>RESERVED</td><td>12796</td><td></td><td></td><td></td></tr> </tbody> </table>						ADDR	INFO	BYTES	COMMENTS			0	Screen	16320	240x136 x 4bit cl			3FC0	Palette	48	3(RGB) x 16 cls			3FF0	Palette Map	16	Palette map			3FF8	Border Color					3FF9	Scrn Offset	2	X, Y offset			3FFC	Mouse Sprite	1				3FFC	RESERVED	4				4000	Tiles	8192	256 8x8 4bit BGs			6000	Sprites	8192	256 8x8 4bit SPRs			8000	Map	32640	240x136 map chips			FF80	Gamepads	4	Btn state			FF84	Mouse	4	Mouse state			FF88	Keyboard	4	Kbd state			FF8C	SFX State	16				FF9C	Sound Reg	72				FFE4	Waveforms	256	16 waves			100E4	SFX	4224				11164	Music Paths	11520				13E64	Music Tracks	408				13FFC	Sound St	4				14000	Stereo Vol	4				14004	Persist Mem	1024	Persistent RAM			14404	Sprite Flags	512				14604	System Font	2048	256 8x8 1bit font			14E04	RESERVED	12796			
ADDR	INFO	BYTES	COMMENTS																																																																																																																																																														
0	Screen	16320	240x136 x 4bit cl																																																																																																																																																														
3FC0	Palette	48	3(RGB) x 16 cls																																																																																																																																																														
3FF0	Palette Map	16	Palette map																																																																																																																																																														
3FF8	Border Color																																																																																																																																																																
3FF9	Scrn Offset	2	X, Y offset																																																																																																																																																														
3FFC	Mouse Sprite	1																																																																																																																																																															
3FFC	RESERVED	4																																																																																																																																																															
4000	Tiles	8192	256 8x8 4bit BGs																																																																																																																																																														
6000	Sprites	8192	256 8x8 4bit SPRs																																																																																																																																																														
8000	Map	32640	240x136 map chips																																																																																																																																																														
FF80	Gamepads	4	Btn state																																																																																																																																																														
FF84	Mouse	4	Mouse state																																																																																																																																																														
FF88	Keyboard	4	Kbd state																																																																																																																																																														
FF8C	SFX State	16																																																																																																																																																															
FF9C	Sound Reg	72																																																																																																																																																															
FFE4	Waveforms	256	16 waves																																																																																																																																																														
100E4	SFX	4224																																																																																																																																																															
11164	Music Paths	11520																																																																																																																																																															
13E64	Music Tracks	408																																																																																																																																																															
13FFC	Sound St	4																																																																																																																																																															
14000	Stereo Vol	4																																																																																																																																																															
14004	Persist Mem	1024	Persistent RAM																																																																																																																																																														
14404	Sprite Flags	512																																																																																																																																																															
14604	System Font	2048	256 8x8 1bit font																																																																																																																																																														
14E04	RESERVED	12796																																																																																																																																																															
<table border="1"> <thead> <tr> <th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th> </tr> </thead> <tbody> <tr><td>0</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td><td>P</td></tr> <tr><td>1</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>2</td><td>6</td><td>7</td><td>8</td><td>9</td><td>=</td><td>(</td><td>)</td><td>\</td><td>;</td><td>'</td><td>,</td><td>.</td><td>/</td><td>SP</td><td></td><td></td></tr> <tr><td>3</td><td>TA</td><td>RE</td><td>BK</td><td>DE</td><td>IN</td><td>PG</td><td>PG</td><td>HO</td><td>EN</td><td>DO</td><td>LE</td><td>RG</td><td>CA</td><td>CT</td><td>SH</td><td></td></tr> <tr><td></td><td>B</td><td>T</td><td>SP</td><td>L</td><td>S</td><td>UP</td><td>DN</td><td>ME</td><td>D</td><td>UP</td><td>WN</td><td>FT</td><td>HT</td><td>PS</td><td>RL</td><td>FT</td></tr> <tr> <td></td><td>↑</td><td>↓</td><td>←</td><td>→</td><td>↶</td><td>↷</td><td>↶</td><td>↷</td><td>↶</td><td>↷</td><td>↶</td><td>↷</td><td>↶</td><td>↷</td><td>↶</td><td>↷</td></tr> </tbody> </table>							0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	1	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	2	6	7	8	9	=	(	)	\	;	'	,	.	/	SP			3	TA	RE	BK	DE	IN	PG	PG	HO	EN	DO	LE	RG	CA	CT	SH			B	T	SP	L	S	UP	DN	ME	D	UP	WN	FT	HT	PS	RL	FT		↑	↓	←	→	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷																																					
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																																																																																																																																		
0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P																																																																																																																																																	
1	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5																																																																																																																																																	
2	6	7	8	9	=	(	)	\	;	'	,	.	/	SP																																																																																																																																																			
3	TA	RE	BK	DE	IN	PG	PG	HO	EN	DO	LE	RG	CA	CT	SH																																																																																																																																																		
	B	T	SP	L	S	UP	DN	ME	D	UP	WN	FT	HT	PS	RL	FT																																																																																																																																																	
	↑	↓	←	→	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷																																																																																																																																																	
<b>Input</b>																																																																																																																																																																	
<pre>btn(id) → pressed btnc(code,hold=-1,period=-1) → released &amp; pressed since last frame   period key(code) → pressed keyp(code,hold=-1,period=-1) → released &amp; pressed since last frame   period mouse() → x, y, left, mid, right, scrLx, scrLy</pre>																																																																																																																																																																	
<b>Drawing</b>																																																																																																																																																																	
<pre>clip(x,y,w,h) : set clipping region cls(color=0) : clear screen circ(x,y,radian,color) : fill circle circb(x,y,radian,color) : draw circle frame line(x0,y0,x1,y1,color) : draw line pix(x,y,color) : draw dot pix(x,y) → color rect(x,y,w,h,color) : fill rect rectb(x,y,w,h,color) : draw rect frame tri(x1,y1,x2,y2,x3,y3,color) : fill triangle ttri(x1,y1,x2,y2,x3,y3,u1,v1,u2,v2,u3,v3,tx,ty) c=0,chromakey=-1,z1=0,z2=3,z3=0) : texture tri elli(x,y,a,b,color) : fill ellipse ellib(x,y,a,b,color) : draw ellipse frame</pre>																																																																																																																																																																	
<b>Program/Interrupts</b>																																																																																																																																																																	
<pre>exit() : exit app reset() : reset time() → ticks (milliseconds) tstamp() → epoch seconds since 1970/1/1 trace(msg,color=15) : debug print</pre>																																																																																																																																																																	
<b>Sprite/Map</b>																																																																																																																																																																	
<pre>fget(spId,flg) → true if the flag is set fset(spId,flg,bool) : set sprite flg[0..8] map(x=0,y=0,w=30,h=17,sx=0,sy=0,colorkey=-1,scale=1,remap=nil) : # x,y,w,h: rect of map tiles to draw # colorkey: opaque (-1) or color index # scale: scaling drawn tiles? # remap: func(tile,x,y)-&gt;tile,flip,rot mget(x,y) → tileId mset(x,y,tileId) : set tile spr(id,x,y,colorkey=-1,scale=1,flip=0,rotate=0, w=1,h=1) : # flip: 0,1,2,3 -&gt; no,horiz,vert,both # rotate: 0,1,2,3 -&gt; 0, 90, 180, 270</pre>																																																																																																																																																																	

# QUICK FIRST GAME

CTRL|⌘ + R → ESC

help → help commands

```
>help
usage: help [<text>] |version| welcome|
spec|ram|vram|commands|api|keys|buttons|
startup|terms|license|
type help commands to show commands
press ESC to switch editor/console

>help commands
Console commands:
cd cls config del demo dir edit eval
exit export folder help import load
menu mkdir new resume run save surf
>
```

new python → edit

```
>new python
new cart has been created
>edit
```



HELLO WORLD!

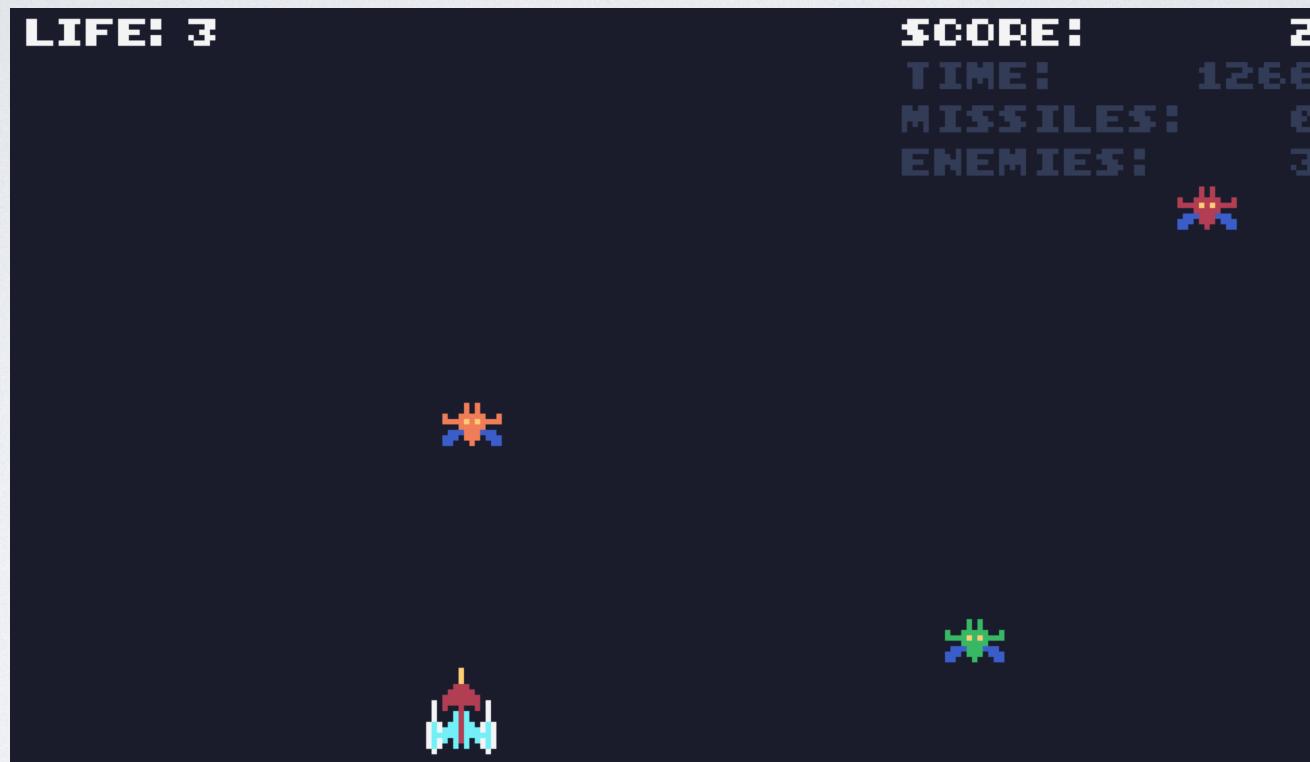
save, load, folder

```
>save first
cart first.tic saved!

>load first
cart first.tic loaded!
use RUN command to run it

>folder
Storage path:
/Users/scott/Library/Application
Support/com.nesbox.tic/TIC-80/
>
```

# SPACE SHOOTER WORKSHOP



# LOAD STARTER.TIC

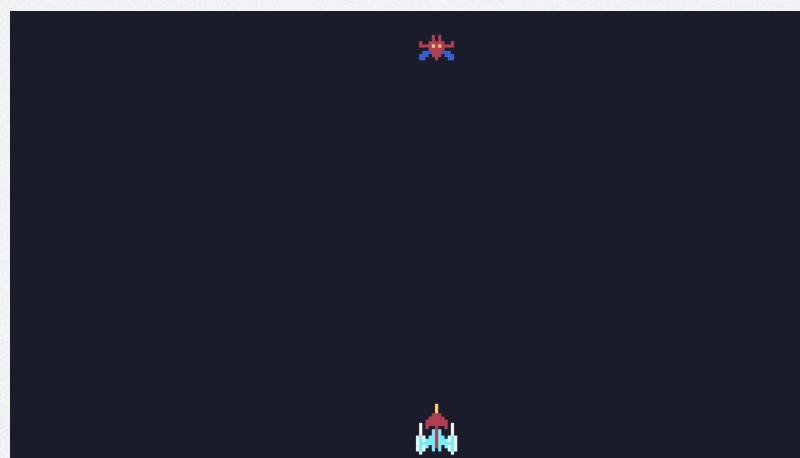
- Download **starter.tic** at <https://github.com/sokoide/tic80-cheatsheet>
- load **starter.tic** → edit
- save **shooter**



# DISPLAY YOUR SHIP

```
def BOOT():
    pass

def TIC():
    cls(0)
    # your ship
    spr(256, 128, 128, w=2, h=2, colorkey=0)
    # enemy
    spr(256+32, 128, 0, w=2, h=2, colorkey=0)
```



# MOVE YOUR SHIP

```
class Sprite:
    def __init__(self, id, x, y):
        self.id=id
        self.x=x
        self.y=y

p=Sprite(256, 128, 128) #player
e=Sprite(256+32, 128, 0) #enemy

def handle_input():
    if btn(2):
        p.x-=1
    if btn(3):
        p.x+=1

def TIC():
    cls(0)
    handle_input()
    spr(p, id, p.x, p.y, w=2, h=2, colorkey=0)
    spr(e, id, e.x, e.y, w=2, h=2, colorkey=0)
```

# FIRE MISSILES

```

p=Sprite(256, 128, 128) #player
e=Sprite(256+32, 128, 8) #enemy
ms=[] # array of missiles

def fire():
    ms.append(Sprite(258, p.x, p.y-8))

def draw_missiles():
    for m in ms:
        spr(m, id, m.x, m.y, w=2, h=2, colorkey=0)

def move_missiles():
    for m in ms:
        m.y-=4

```

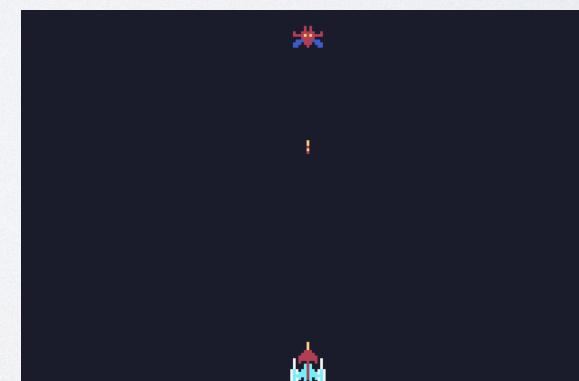
#TODO: we need to delete missiles if m.y<0  
(next page)

```

def handle_input():
    if btn(2): # left
        p.x-=1
    if btn(3): # right
        p.x+=1
    if btnp(4): # z key
        fire()

def TIC():
    cls(0)
    handle_input()
    move_missiles()
    draw_missiles()

```



# DELETE MISSILES IF M.Y<0

```
class Sprite:  
    def __init__(self, id, x, y):  
        self.id=id  
        self.x=x  
        self.y=y  
        self.delme=False
```

```
def TIC():  
    cls(0)  
    handle_input()  
    move_missiles()  
    delete_sprites()  
    draw_missiles()
```

```
def move_missiles():  
    for m in ms:  
        m.y-=4  
        if m.y<0:  
            trace("Delete me "+str(m.y))  
            m.delme=True
```

```
def delete_sprites():  
    global ms  
    ms=[x for x in ms if x.delme==False]
```

# MORE ENEMIES

```
p=Sprite(256, 120, 120) #player  
es=[Sprite(256+32, 60, 0),  
    Sprite(256+32, 120, 0),  
    Sprite(256+32, 180, 0)] #enemies  
m=[ ] # array of missiles
```

```
def move_enemies():  
    for e in es:  
        e.y+=1  
        if e.y>136:  
            e.delme=True
```

```
def TIC():  
    cls(0)  
    handle_input()  
    move_missiles()  
    move_enemies()  
    delete_sprites()  
    draw_missiles()  
    spr(p, id, p.x, p.y, w=2, h=2, colorkey=0)
```

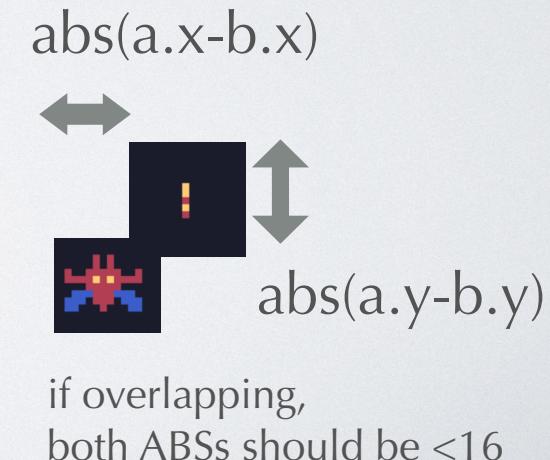
# HIT TEST - YOUR MISSILE & ENEMIES

```
def TIC():
    cls(0)
    handle_input()
    move_missiles()
    hit_test()
    delete_sprites()
    draw_missiles()
```

```
def hit_test():
    for m in ms:
        for e in es:
            if hit(m, e):
                m.delme=True
                e.delme=True
```

```
def hit(a, b):
    if a.delme or b.delme:
        return False
    if abs(a.x-b.x)<16 and abs(a.y-b.y)<16
        return True
    return False
```

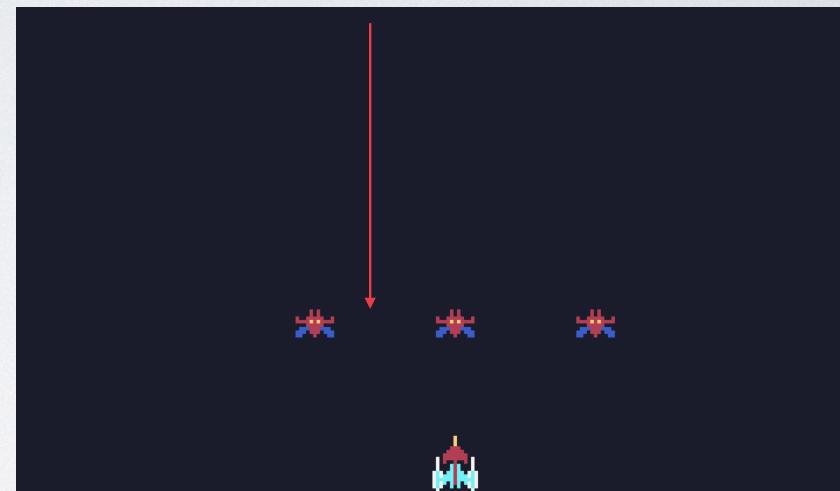
```
def delete_sprites():
    global ms, es
    ms=[x for x in ms if x.delme==False]
    es=[x for x in es if x.delme==False]
```



# MOVE ENEMIES

```
def TIC():
    cls()
    handle_input()
    move_missiles()
    move_enemies()
    hit_test()
```

```
def move_enemies():
    for e in es:
        e.y+=1
        if e.y<0:
            e.delme=True
```



# HIT TEST - ENEMIES & YOUR SHIP

```
# global vars
p=Sprite(256, 128, 128) #player
es=[Sprite(256+32, 88, 0),
    Sprite(256+32, 128, 0),
    Sprite(256+32, 168, 0)] #enemies
m=[] # array of missiles
gameover=-1
```

```
def hit_test():
    global gameover
    for m in ms: # missiles & enemies
        for e in es: I
            if hit(m,e):
                m.delme=True
                e.delme=True
    for e in es: # enemies & your ship
        if hit(e,p):
            e.delme=True
            gameover=180
```

## CONT'D: GAMEOVER

```
def TICK():
    global gameover
    cls(0)
    if gameover==0:
        print("GAME OVER",x=90,y=60,color=12)
        gameover=1
    if gameover==1:
        reset()
    else:
        handle_input()
        move_missiles()
        move_enemies()
        hit_test()
        delete_sprites()
        draw_missiles()
        spr(p,id,p.x,p.y,w=2,h=2,colorkey=0)
        for e in es:
            spr(e,id,e.x,e.y,w=2,h=2,colorkey=0)
```

# SPAWN ENEMIES

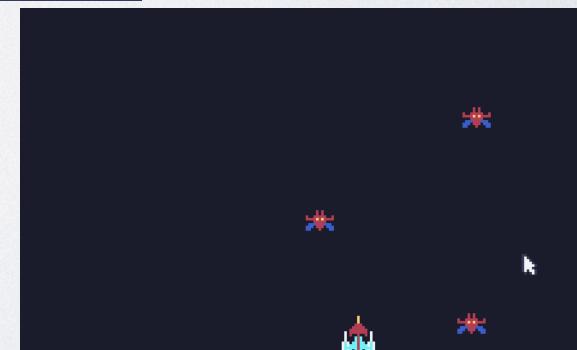
```
# global vars
pSprite(256, 128)
es=[Sprite(256+32, 1,
           Sprite(256+32, 1,
           Sprite(256+32, 1
mS=[] # array of
gameover=-1
t=0
```

```
def TIC():
    global gameover, t
    t+=1
    cls(0)
    if gameover>=0:
        print("GAME OVER")
        gameover=-1
        if gameover<=0:
            reset()
    else:
        spawn_enemies()
        handle_input()
```

```
def spawn_enemies():
    if t%40==0:
        x=random.randint(0, 240-16)
        es.append(Sprite(256+32, x, 0))
```

title:	space sh
author:	sokolide
desc:	space sh
site:	<a href="https://">https://</a>
license:	MIT Licen
version:	0.1
script:	python

```
import random
```



# MORE FEATURES!

- Lives
- Scoring
- Different enemy character
- Enemy animation
- Explosion animation
- Internal information on top-right
- ...
- **final.tic** is available at <https://github.com/sokoide/tic80-cheatsheet> for your reference