

Preply Pizza!

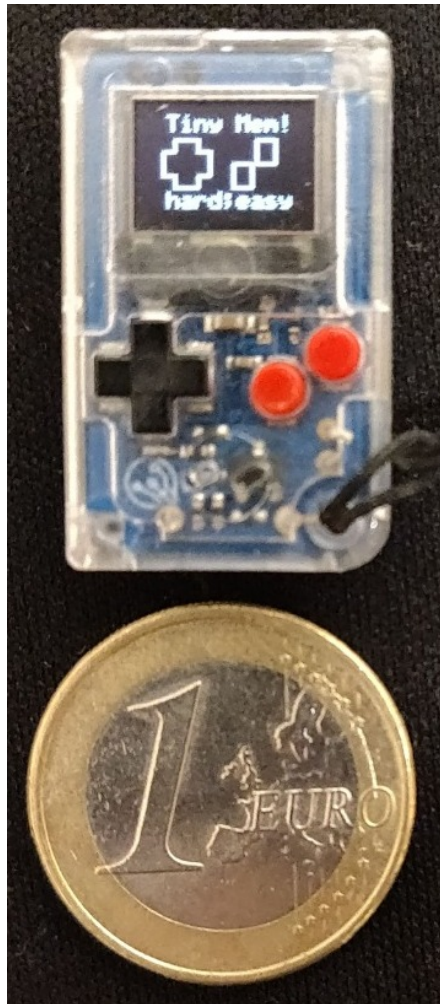


preply.com/en/about-us

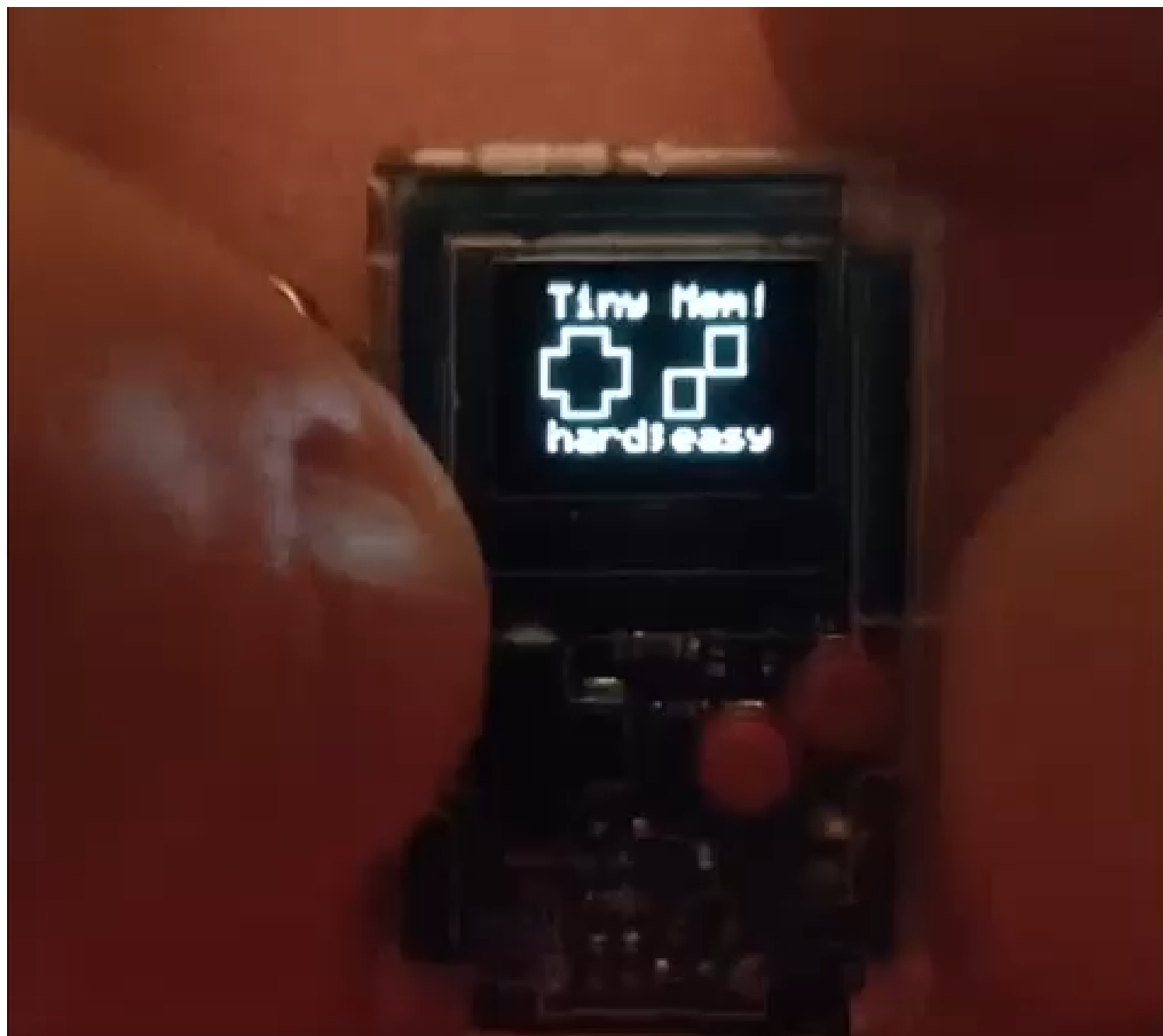
#PyDayBCN2023

github.com/isaacbernat/tinymem

Tinymem: a game under 40 lines/30mm



```
1 import random, time, thumby, collections
2 controls_map = bytearray([224,32,32,32,32,32,63,1,1,1,1,1,1,1,1,1,63,32,32,32,32,224,
3 controls_sprite = thumby.Sprite(49, 21, controls_map, 10, 10) # sprite based on L
4 Button = collections.namedtuple("Button", "letter freq x y")
5 KEYS = [Button("", 20, 0, 0), Button("A", 7458, 52, 12), Button("B", 7902, 42, 22)]
6
7 def show(val=0, text=["", "", "", "", ""]):
8     thumby.display.fill(0)
9     for index, content in enumerate(text):
10         thumby.display.drawText(content, 0, 8 * index, 1)
11     thumby.display.drawSprite(controls_sprite)
12     thumby.display.drawText(KEYS[val].letter, KEYS[val].x, KEYS[val].y, 1)
13     thumby.display.update()
14     thumby.audio.playBlocking(KEYS[val].freq, val == 0 or 1000)
15 def start():
16     show(text=[" Tiny Mem!", "", "", "", " hard;easy"])
17     value_range = (1, 2) if wait_press() < 3 else (3, 6)
18     random.seed(time.ticks_ms())
19     return 0, [random.randint(*value_range) for i in range(100)]
20 def wait_press(c=None):
21     while(c is None):
22         c = (thumby.buttonL.justPressed() and 6) or (thumby.buttonD.justPressed()
23     return c
24 def turn(max_pos, sequence, current_pos=0):
25     for index, val in enumerate(sequence[:max_pos + 1]): # show sequence
26         show(val=val, text=[f" key={KEYS[val].letter}", "", "", "", f" num={inde
27     show(text=[" your turn", "", "", "", " repeat"]) # ask sequence
28     while (current_pos <= max_pos):
29         if sequence[current_pos] != wait_press(): # GAME OVER
30             show(text=[" your mem=", "", "", "", f" {str(max_pos*(min(sequence)
31             wait_press()
32             return start()
33             show(val=sequence[current_pos], text=[f" {current_pos + 1} done", "", "",
34             current_pos += 1
35     return max_pos + 1, sequence
36
37 max_pos, sequence = start()
38 while(True):
39     max_pos, sequence = turn(max_pos, sequence)
```



Code pt1: imports, constants and game loop

```

1 import random, time, thumby, collections
2
3 controls_map = bytearray([224,32,32,32,32,63,1,1,1,1,1,1,1,1,1,1])
4 controls_sprite = thumby.Sprite(49, 21, controls_map, 10, 10)
5 Button = collections.namedtuple("Button", "letter freq x y")
6 KEYS = [Button("", 20, 0, 0), Button("A", 7458, 52, 12),
7         Button("B", 7902, 42, 22), Button("U", 10548, 18, 12),
8         Button("R", 8870, 24, 17), Button("D", 7458, 18, 22),
9         Button("L", 7902, 12, 17)]
10
11 max_pos, sequence = start()
12 while(True):
13     max_pos, sequence = turn(max_pos, sequence)

```

moar info?! \o/ -> github.com/isaacbernat/tinymem

Code pt2: start, show and wait press

```
1 def start():
2     show(text=[" Tiny Mem!", "", "", "", " hard;easy"])
3     value_range = (1, 2) if wait_press() < 3 else (3, 6)
4     random.seed(time.ticks_ms())
5     return 0, [random.randint(*value_range) for i in range(100)]
6
7 def show(val=0, text=["", "", "", "", ""]):
8     thumby.display.fill(0)
9     for index, content in enumerate(text):
10         thumby.display.drawText(content, 0, 8 * index, 1)
11     thumby.display.drawSprite(controls_sprite)
12     thumby.display.drawText(KEYS[val].letter, KEYS[val].x, KEYS[val].y, 1)
13     thumby.display.update()
14     thumby.audio.playBlocking(KEYS[val].freq, val == 0 or 1000)
15
16 def wait_press(c=None):
17     while(c is None):
18         c = (thumby.buttonL.justPressed() and 6) or (thumby.buttonD.justPr
19             (thumby.buttonR.justPressed() and 4) or (thumby.buttonU.justPr
20             (thumby.buttonB.justPressed() and 2) or (thumby.buttonA.justPr
21     return c
```

Code pt3: turn

```
1 def turn(max_pos, sequence, current_pos=0):
2     for index, val in enumerate(sequence[:max_pos + 1]): # show s
3         show(val=val, text=[f" key={KEYS[val].letter}", "", "", "
4             f" num={index + 1}"])
5     show(text=[" your turn", "", "", "", " repeat"]) # ask sequ
6     while (current_pos <= max_pos):
7         if sequence[current_pos] != wait_press(): # GAME OVER
8             show(text=[" your mem=", "", "", "",
9                 f" {str(max_pos*(min(sequence) == 1 or 2))} b
10            wait_press()
11            return start()
12        show(val=sequence[current_pos],
13            text=[f" {current_pos + 1} done", "", "", "",
14                f" {max_pos - current_pos} left"])
15        current_pos += 1
16    return max_pos + 1, sequence
```

tinymem is here too ^_^ -> code.thumby.us

GAME OVER: Thanks for playing. Now check Preply 🍕!



emulator + IDE: code.thumby.us

