

# Muriel, a Tamagotchi

**Desenvolvido por:** Raquel Cristina Schaly Behrens

**Número de matrícula:** 20100544

Os pais de Muriel resolveram tirar férias para passar um tempo juntos, a fim de fortalecer a relação, a qual estava à beira do precipício desde que Muriel nasceu. Isso só foi possível porque você se voluntariou para cuidar dela enquanto eles estiverem fora! Obrigada!

Agora que o objetivo está traçado, você precisa saber de algumas coisas:

- Se você deixar Muriel acordada, com fome ou doente por muito tempo, ela morre!
- Quando você chega na casa dela, descobre que seus pais acabaram de dar-lhe um banho e de limpar seu cocô, mas esqueceram de alimentá-la! Por isso, Muriel está com um pouco de fome;
- A saúde da tamagotchi é imprevisível! A qualquer momento ela pode ficar doente! Mas mesmo assim, não fica doente com muita frequência;
- Os pais de Muriel vão ficar fora por cinco dias.

## Informações técnicas iniciais:

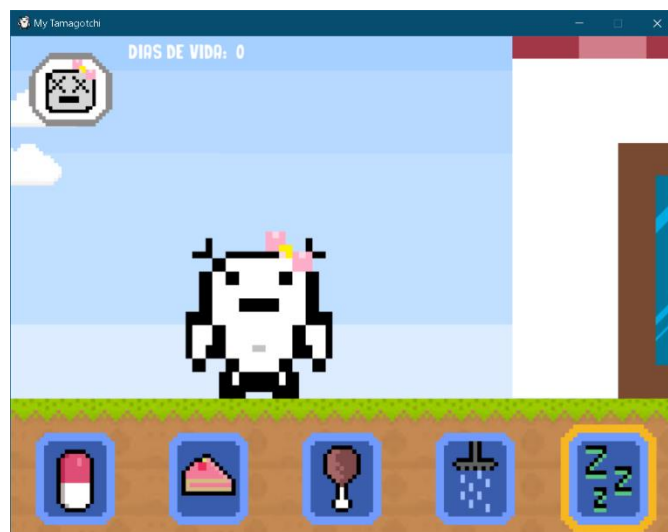
- Um dia no nosso mundo corresponde a um minuto no mundo da família de Muriel;
- A fome, o sono, a vontade de ir ao banheiro, a necessidade de tomar banho, e a quantidade de tempo por qual Muriel está doente (uma vez que ela fica doente) vão de 0 a 100% - todas essas necessidades aumentam em 1% a cada segundo;
- Se você deixar que a fome, ou que o sono, ou que a quantidade de tempo por qual ela está doente cheguem a 100%, Muriel morre.
- A vontade de ir ao banheiro e a necessidade de tomar banho não influenciam na saúde da tamagotchi;
- No início do jogo as necessidades estão assim:
  - Fome: 30%;
  - Sono: 0%;
  - Vontade de ir ao banheiro: 0%;
  - Necessidade de tomar banho: 0%;
  - Muriel não está doente no início, portanto a quantidade de tempo por qual ela está doente está em 0% e ficará assim até ela adoecer;
  - Há uma chance de 0,02% de ela ficar doente para cada meio segundo.
- Quando ela está apenas com fome (saudável e sem sono), Muriel fica com vontade ou de comer bolo, ou de comer frango. Dessa forma, ela pensará em qual deseja mais no momento, e, se você a alimentar com a comida dos seus pensamentos, ela ficará mais satisfeita que se você a alimentar com a outra opção.

## Jogando Muriel, a Tamagotchi

Para começar, clique em “Jogar”, mas se você deseja sair sem começar o jogo, apenas clique em “Sair”:



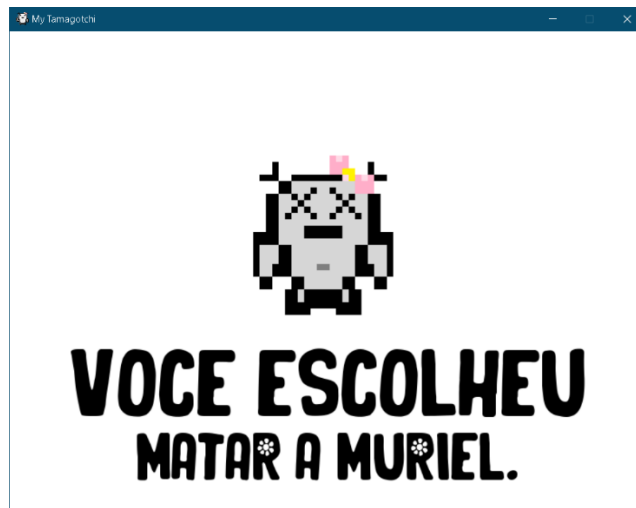
Agora que você começou o jogo, ele aparecerá assim para você:



Se deseja sair, a qualquer momento do jogo, você pode apertar no botão no canto superior esquerdo da tela:



Mas saiba que, uma vez que você clicar nesse botão, você abandonará Muriel, e ela vai ficar sozinha por tempo demais. Dessa forma, Muriel vai certamente morrer:



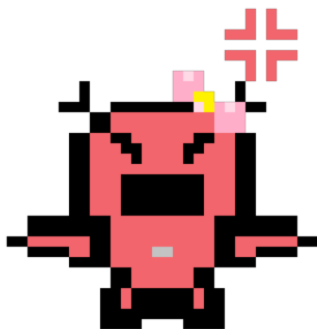
### Conheça um pouco das emoções e jeitos de Muriel



Está feliz! Sua fome e seu sono estão muito baixos, e ela não está doente!



Suas necessidades estão OK, mas poderiam melhorar... Tente alimentá-la ou colocá-la pra dormir!



Está com muita fome! Por favor, dê algo a ela!



Muriel quer bolo!



Muriel quer frango!



Está com sono, precisa dormir!



Está doente, precisa de remédio!



Está faminta e com sono!



Está com sono e doente!



Está com fome e doente!



Está com fome, doente e com sono!  
Rápido, ajude-a!



A Muriel morreu!

## Botões

Portanto, para cuidar de Muriel, e não deixa-la morrer, você precisa saber o que que os botões do jogo fazem:



Nesse botão, ao dar remédio para a tamagotchi, você vai curá-la de sua doença



Nesse botão, você alimenta Muriel com bolo



Nesse botão, você alimenta Muriel com frango



Nesse botão, você faz Muriel tomar banho



Nesse botão, você faz Muriel dormir

### Emoções que não influenciam na saúde dela:

Se Muriel ficar muito tempo sem tomar banho, ela fica suja:



Se você alimentar Muriel mais de três vezes, ela vai fazer cocô no chão. **Para limpá-lo, basta clicar uma vez em cima.**



### Ações

Se você der a Muriel, a comida que ela deseja, ela vai comer feliz:



Senão, ela comerá normalmente:



Quando você a faz dormir, ela fica assim:

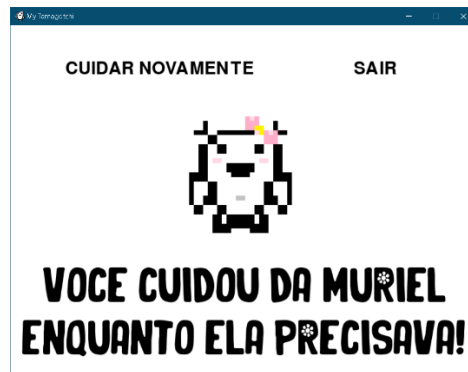


## Final do jogo

Caso você pare de cuidar de Muriel por muito tempo, ela morre! Você pode voltar no tempo para cuidar dela direitinho, ou você pode deixá-la morta e sair do jogo.



Caso você cuide dela pelo tempo esperado, sem deixa-la morrer, o jogo acaba! Você pode escolher cuidar dela por mais cinco dias e fornecer férias mais longas aos pais dela!



## **Como executar o jogo**

Para jogar, você deve primeiramente ter o python instalado em seu computador. Caso você não tenha, você pode instalar em '<https://www.python.org/downloads/>'. É importante que você anote o diretório de instalação.

Também instale a biblioteca de desenvolvimento de jogos (a qual foi usada para o desenvolvimento desse jogo) em '<https://www.pygame.org/download.shtml>'.

Agora você precisa ir no terminal do seu computador (clique em windows + R, e digite 'cmd'). Vá para a pasta do seu diretório. No meu computador, por exemplo, eu teria que digitar: 'cd AppData\Local\Programs\Python\Python37'.

Agora digite 'python -m pip install pygame' para instalar o pygame corretamente.

Depois disso, apenas abra a pasta do jogo disponibilizada, e clique diretamente no arquivo 'mytagamotchi.py' para executar com o python.

## **Conceitos de Orientação a Objetos usados:**

Para animar o plano de fundo que representa o dia (com nuvens se mexendo), defini uma classe:



```

#classe para animar as nuvens do plano de fundo
class Nuvem(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.spritesNuvem = [pygame.image.load('frame_dia.png'), pygame

self.current_sprite = 0
self.image = self.spritesNuvem[self.current_sprite]
self.rect = self.image.get_rect()
self.rect.topleft = [0,0]

    def update(self):
        self.current_sprite += 0.1

        if self.current_sprite >= len(self.spritesNuvem):
            self.current_sprite = 0

        self.image = self.spritesNuvem[int(self.current_sprite)]

nuvens_sprites = pygame.sprite.Group()
dianuvem = Nuvem()
nuvens_sprites.add(dianuvem)

```

Método de inicialização

Atributos

Objeto

Essa classe possui um método de inicialização (def `__init__`), e dentro dele obtemos as características (atributos) da Nuvem (todas os seus sprites – imagens da animação –, em qual sprite o programa está – que de início é 0 –, a primeira imagem a ser mostrada – `self.image` –, e o lugar em que a imagem será mostrada – que nesse caso, é na tela inteira: coordenadas (0,0)).

A classe possui também o método 'update', que ao somar 0.1 na variável 'self.current\_sprite', a cada chamada desse método, está mais próximo de mostrar a próxima imagem. Somei 0.1 pois não quis que as imagens passassem muito rápido. Logo após isso, quando 'self.image' recebe 'self.spritesNuvem'[int...]', o int transforma um número não inteiro em inteiro. Dessa forma, os números entre 0 e 1 (exceto o 1), seriam iguais a 0. Portanto, apenas depois de 10 vezes em que o método foi chamado, que a figura a aparecer muda.

O objeto 'dianuvem' é instanciado para pegar o sprite a ser mostrado. Ele é adicionado em 'nuvens\_sprites'. E, mais tarde, na função principal do jogo e em outras secundárias 'nuvens\_sprites' é desenhado na tela, uma por uma, ao chamarmos o método 'update' (instanciar) de 'nuvens\_sprites'.

```

nuvens_sprites.draw(gameDisplay)

if periodo_dia <= 30:
    nuvens_sprites.update()
elif periodo_dia > 30:
    gameDisplay.blit(frame_noite_dentro, (0,0))

```

A mesma lógica foi usada para criar a classe 'Tamagotchi()' que mostra o tamagotchi correto na tela, mas nela também incluímos atributos que precisam ser declarados ao chamarmos essa classe:

```
#classe para mostrar o tamagotchi correto na tela
class Tamagotchi(pygame.sprite.Sprite):
    def __init__(self, pos_x, pos_y):
        super().__init__()
        self.spritesNormal = [pygame.image.load('tamagotchi_r
        self.spritesSono = [pygame.image.load('tamagotchi_sor
        self.spritesFome = [pygame.image.load('tamagotchi_fon
        self.spritesMorrendo = [pygame.image.load('tamagotch:
        self.spritesFomeSono = [pygame.image.load('tamagotch:
        self.spritesFomeDoenca = [pygame.image.load('tamagotc
        self.spritesFeliz = [pygame.image.load('tamagotchi_fe
        self.spritesDormindo = [pygame.image.load('tamagotch:
        self.spritesDoenca = [pygame.image.load('tamagotchi_c
        self.spritesDoencaSono = [pygame.image.load('tamagotc
        self.spritesComendoGostou = [pygame.image.load('tamaç
        self.spritesComendoNaoGostou = [pygame.image.load('te

        self.current_sprite = 0
        self.image = self.spritesNormal[self.current_sprite]
        self.rect = self.image.get_rect()
        self.rect.topleft = [pos_x,pos_y]

    def update(self, x):
        self.current_sprite += 0.20
```

Atributos

Esses atributos são 'pos\_x' e 'pos\_y', que definem em qual posição o tamagotchi vai aparecer. Quando instanciamos a classe mais tarde por um objeto (tamagotchi), informamos os valores x e y (definidos no início do código) a serem atribuídos para 'pos\_x' e 'pos\_y':

```
moving_sprites = pygame.sprite.Group()
tamagotchi = Tamagotchi(x,y)
moving_sprites.add(tamagotchi)
```