Passo a passo — Instalar Pygame e executar o Jogo da Forca (Windows)

Este documento mostra **passo a passo** (com comandos) para: instalar o Python, instalar o Pygame, preparar um projeto no Windows (ou no PyCharm) e executar o arquivo `JogoDaForca.py` com o jogo da forca que você tem.

---

0) Pré-requisitos

- Windows (7/8/10/11).
- Acesso de usuário normal (não é necessário administrador, salvo para instalar certas dependências do sistema).

---

1) Instalar o Python (se ainda não tem)

1. Abra o navegador e acesse: `https://www.python.org/downloads/` e baixe a versão **3.10+**.
2. Execute o instalador e **marque** a opção **Add Python to PATH** antes de clicar em "Install Now".

Verificar instalação: abra o Prompt de Comando (cmd) e rode:

python --version

Você deve ver algo como `Python 3.x.x`.

Se `python` não for reconhecido, tente:

py -3 --version

---

2) Recomendações: criar e ativar um ambiente virtual (opcional, recomendado)

Trabalhar com `venv` evita conflitos entre pacotes de projetos diferentes.

No Prompt de Comando, entre na pasta onde quer criar o projeto e rode:

mkdir forca_game
cd forca_game
python -m venv venv **veja mais sobre a função do “-m” no fim deste documento

Ativar o ambiente virtual:

venv\Scripts\activate

Se estiver usando `PowerShell`:

venv\Scripts\Activate.ps1

Quando o venv está ativo, o prompt mostra algo como `(venv)` no início.

> Para desativar: `deactivate`. (só desative após rodar o jogo e não for mais jogar)

---

3) Atualizar pip (ajuda a evitar erros ao instalar pacotes)

Com o virtualenv ativado (ou mesmo sem ele), rode:

python -m pip install --upgrade pip setuptools wheel

---

4) Instalar o Pygame

No terminal (com o venv ativo, se estiver usando) rode:

python -m pip install pygame

Se `pip` reclamar ou der erro, tente estes passos:

- Atualize `pip` como no passo anterior.
- Rode `python -m pip install pygame --upgrade`.
- Se tiver várias versões do Python, use `py -3 -m pip install pygame`.

Para checar se o pygame foi instalado corretamente:

python -c "import pygame; print(pygame.__version__)"

Se importar sem erros e mostrar a versão — ok.

---

5) Criar o arquivo do jogo (`JogoDaForca.py`)

1. Abra o **Notepad**, **Notepad++**, **VS Code**, **PyCharm** ou outro editor.
2. Crie um arquivo chamado `forca.py` dentro da pasta do projeto.
3. Cole o código do jogo dentro desse arquivo (abaixo há uma versão pronta, testada no Pycharm). Salve.

```
import pygame as pg

import random


# --- CORES ---

branco = (255, 255, 255)

preto = (0, 0, 0)

vermelho = (200, 0, 0)

verde = (0, 150, 0)


# --- SETUP DA JANELA ---

pg.init()
```

```python
window = pg.display.set_mode((1000, 600))

pg.display.set_caption("Jogo da Forca")


# --- FONTES ---

font = pg.font.SysFont('courier new', 50)

font_rb = pg.font.SysFont('courier new', 30)

font_small = pg.font.SysFont('courier new', 24)


# --- PALAVRAS ---

palavras = ['paralelepipedo', 'ornintorinco', 'apartamento',
'xicara de cha']


# --- VARIÁVEIS DE ESTADO ---

tentativas_de_letras = []

palavra_escolhida = ''

palavra_camuflada = ''

chance = 0

game_over = False

mensagem_final = ""

entrada_usuario = ""

input_active = False


# posição da caixa de input

INPUT_X, INPUT_Y, INPUT_W, INPUT_H = 700, 400, 250, 40



def resetar_jogo():
```

```
    """Reseta o estado do jogo."""

    global tentativas_de_letras, palavra_escolhida,
palavra_camuflada

    global chance, game_over, mensagem_final, entrada_usuario,
input_active



    tentativas_de_letras = []

    palavra_escolhida = random.choice(palavras).upper()  # sorteia
palavra

    palavra_camuflada = camuflar(palavra_escolhida,
tentativas_de_letras)

    chance = 0

    game_over = False

    mensagem_final = ""

    entrada_usuario = ""

    input_active = False




def desenhar_forca(chance):

    """Desenha a forca e partes do boneco."""

    pg.draw.rect(window, branco, (0, 0, 1000, 600))

    pg.draw.line(window, preto, (100, 500), (100, 100), 10)

    pg.draw.line(window, preto, (50, 500), (150, 500), 10)

    pg.draw.line(window, preto, (100, 100), (300, 100), 10)

    pg.draw.line(window, preto, (300, 100), (300, 150), 10)



    if chance >= 1:

        pg.draw.circle(window, preto, (300, 200), 50, 10)
```

```
    if chance >= 2:
        pg.draw.line(window, preto, (300, 250), (300, 350), 10)
    if chance >= 3:
        pg.draw.line(window, preto, (300, 260), (225, 350), 10)
    if chance >= 4:
        pg.draw.line(window, preto, (300, 260), (375, 350), 10)
    if chance >= 5:
        pg.draw.line(window, preto, (300, 350), (375, 450), 10)
    if chance >= 6:
        pg.draw.line(window, preto, (300, 350), (225, 450), 10)



def desenhar_restart():
    """Desenha botão de restart."""
    pg.draw.rect(window, preto, (700, 100, 200, 65))
    texto = font_rb.render("Restart(1)", True, branco)
    window.blit(texto, (710, 120))



def camuflar(palavra, tentativas):
    """Retorna palavra mascarada com #"""
    return "".join([letra if letra in tentativas or letra in [' ',
'-'] else "#" for letra in palavra])



def mostrar_palavra(palavra_camuflada):
    texto = font.render(palavra_camuflada, True, preto)
```

```
    window.blit(texto, (200, 500))




def mostrar_mensagem(msg, cor):

    texto = font.render(msg, True, cor)

    rect = texto.get_rect(center=(500, 300))

    window.blit(texto, rect)




def mostrar_tentativas():

    certas = [l for l in tentativas_de_letras if l in
palavra_escolhida]

    erradas = [l for l in tentativas_de_letras if l not in
palavra_escolhida]



    titulo = font_rb.render("Tentativas:", True, preto)

    window.blit(titulo, (700, 200))



    certas_txt = font_small.render("Certas: " + " ".join(certas),
True, verde)

    window.blit(certas_txt, (700, 240))



    erradas_txt = font_small.render("Erradas: " + "
".join(erradas), True, vermelho)

    window.blit(erradas_txt, (700, 270))



    chances_restantes = 6 - chance

    cor = verde if chances_restantes > 2 else vermelho
```

```
    chances_txt = font_small.render(f"Chances:
{chances_restantes}/6", True, cor)

    window.blit(chances_txt, (700, 310))




def mostrar_input():

    titulo = font_rb.render("Digite a palavra (ENTER):", True,
preto)

    window.blit(titulo, (700, 360))

    pg.draw.rect(window, preto, (INPUT_X, INPUT_Y, INPUT_W,
INPUT_H), 2)



    mostrar = entrada_usuario[-20:]

    caixa = font_small.render(mostrar, True, preto)

    window.blit(caixa, (INPUT_X + 10, INPUT_Y + 6))




# --- INÍCIO ---

resetar_jogo()



# --- LOOP PRINCIPAL ---

while True:

    for event in pg.event.get():

        if event.type == pg.QUIT:

            pg.quit()

            quit()



        if event.type == pg.MOUSEBUTTONDOWN:
```

```
            x, y = event.pos
            if 700 <= x <= 900 and 100 <= y <= 165:
                resetar_jogo()
            elif INPUT_X <= x <= INPUT_X + INPUT_W and INPUT_Y <= y
<= INPUT_Y + INPUT_H:
                input_active = True
                entrada_usuario = ""
            else:
                input_active = False

        if event.type == pg.KEYDOWN:
            if event.key == pg.K_1:
                resetar_jogo()

            if not game_over:
                if not input_active:  # tentando letra
                    if event.unicode.isalpha():
                        letra = event.unicode.upper()
                        if letra not in tentativas_de_letras:
                            tentativas_de_letras.append(letra)
                            if letra not in palavra_escolhida:
                                chance += 1
                else:  # digitando palavra inteira
                    if event.key == pg.K_RETURN:
                        if entrada_usuario.upper() ==
palavra_escolhida:
                            mensagem_final = "Parabéns, você
venceu!"
```

```
                        else:
                            mensagem_final = f'Você perdeu! A
palavra era:\n {palavra_escolhida}'
                            chance = 6
                        game_over = True
                    elif event.key == pg.K_BACKSPACE:
                        entrada_usuario = entrada_usuario[:-1]
                    else:
                        entrada_usuario += event.unicode.upper()


    # --- DESENHO ---
    desenhar_forca(chance)
    desenhar_restart()
    palavra_camuflada = camuflar(palavra_escolhida,
tentativas_de_letras)
    mostrar_palavra(palavra_camuflada)
    mostrar_tentativas()
    mostrar_input()


    if not game_over:
        if chance >= 6:
            mensagem_final = f"Você perdeu! A palavra era:
{palavra_escolhida}"
            game_over = True
        elif palavra_camuflada == palavra_escolhida:
            mensagem_final = "Parabéns, você venceu!"
            game_over = True
```

```
    if game_over and mensagem_final:
        cor = verde if "venceu" in mensagem_final else vermelho
        mostrar_mensagem(mensagem_final, cor)


    pg.display.update()
```

6) Executar o jogo

No **PyCharm**:

- Clique com o botão direito no arquivo `JogoDaForca.py` → **Run**.

No **Prompt de Comando** (fora do PyCharm):

Entre na pasta onde você salvou o `JogoDaForca.py` usando `cd caminho/da/pasta`.

Exemplo:

```
cd C:\Users\SeuNome\Desktop\forca_game
```

Rode o jogo:

```
python JogoDaForca.py
```

A janela do jogo da forca deve abrir!

---

7) Como jogar

- Digite **letras no teclado** → aparecem na palavra ou contam como erro.
- Clique na **caixa de texto** (lado direito) → digite a **palavra inteira** → ENTER confirma.
- Clique em **Restart** ou aperte **“1”** para reiniciar a partida.

** Sobre o uso de “-m”

No Windows (e em qualquer sistema com Python), a opção `-m` do comando `python` significa **“run library module as a script”** — ou seja, **executar um módulo da biblioteca Python como se fosse um programa**.

Exemplos práticos:

- `pip install pygame`

Aqui você está chamando diretamente o **executável** `pip.exe` (no Windows) que foi instalado junto com o Python.
Se você tiver vários Pythons no sistema, às vezes esse `pip` não corresponde ao Python que você está usando.

- **2. Com `-m`:**

```
python -m pip install pygame
```

Aqui você diz explicitamente:
“Use o **módulo** `pip` do **mesmo Python** que eu chamei (`python`)”.

Isso evita erros do tipo: instalar num Python, mas tentar rodar em outro.
É o jeito **mais confiável** de garantir que o pacote vá para a versão certa do Python.

- **3. Outros usos do `-m`**

Rodar o servidor HTTP embutido do Python:

```
python -m http.server
```

Verificar versão do `venv` e criar ambientes virtuais:

```
python -m venv venv
```

Rodar testes com unittest:

```
python -m unittest
```

Guia rápido — Python `-m`

Comando	Para que serve
`python -m pip install pygame`	Instala um pacote (ex.: `pygame`) usando o mesmo Python que você chamou. Evita confusão em PCs com vários Pythons.
`python -m pip list`	Lista todos os pacotes instalados nesse Python.
`python -m venv venv`	Cria um ambiente virtual chamado `venv` (pasta isolada com seu próprio Python + pacotes).
`python -m http.server`	Sobe um servidor web local na porta 8000. Você pode abrir `http://localhost:8000` no navegador.
`python -m unittest`	Executa testes unitários automaticamente nos arquivos do projeto.
`python -m idlelib.idle`	Abre o **IDLE**, o editor oficial simples do Python.
`python -m timeit "código"`	Mede o tempo de execução de um trecho de código. Ex.: `python -m timeit "x=2+2"`.

`python -m site`	Mostra os caminhos onde o Python procura bibliotecas instaladas.
`python -m ensurepip`	Garante que o `pip` esteja instalado no seu Python.
`python -m pydoc módulo`	Mostra documentação de um módulo. Ex.: `python -m pydoc math`.
