

**ANO**  
**2023**



# **UNINTER**

**CADERNO DE RESPOSTAS DA  
ATIVIDADE PRÁTICA DE:**

**LINGUAGEM DE PROGRAMAÇÃO  
APLICADA**

**ALUNO: (ISABELY MARQUES, 4390631)**

**Caderno de Resposta Elaborado por:  
Prof. MSc. Renan Portela Jorge**

## Questão 01 – Aula Prática 01

Dado o seguinte trecho de código da aula prática 01, indique as chamadas de método em que é feito o desenho na tela e qual a necessidade do método `pygame.display.flip()` para visualização das superfícies.

```
# Carregar musica e deixar ela tocando
pygame.mixer_music.load('./asset/fase1.mp3')
pygame.mixer_music.play(-1)
pygame.mixer_music.set_volume(0.3)
print('setup end')
print('loop start')
while True:
    clock.tick(60)
    window.blit(source=bg_surf, dest=bg_rect)
    window.blit(source=player1_surf, dest=player1_rect)
    pygame.display.flip()
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            print('loop end')
            pygame.quit()
            quit()
    pressed_key = pygame.key.get_pressed()
    if pressed_key[pygame.K_w]:
        player1_rect.centery -= 1
    if pressed_key[pygame.K_s]:
        player1_rect.centery += 1
    if pressed_key[pygame.K_d]:
        player1_rect.centerx += 1
    if pressed_key[pygame.K_a]:
        player1_rect.centerx -= 1
    pass
```

RESPOSTA: (INSERIR RESPOSTA AQUI)

os métodos responsáveis por desenhar na tela são as chamadas `window.blit`

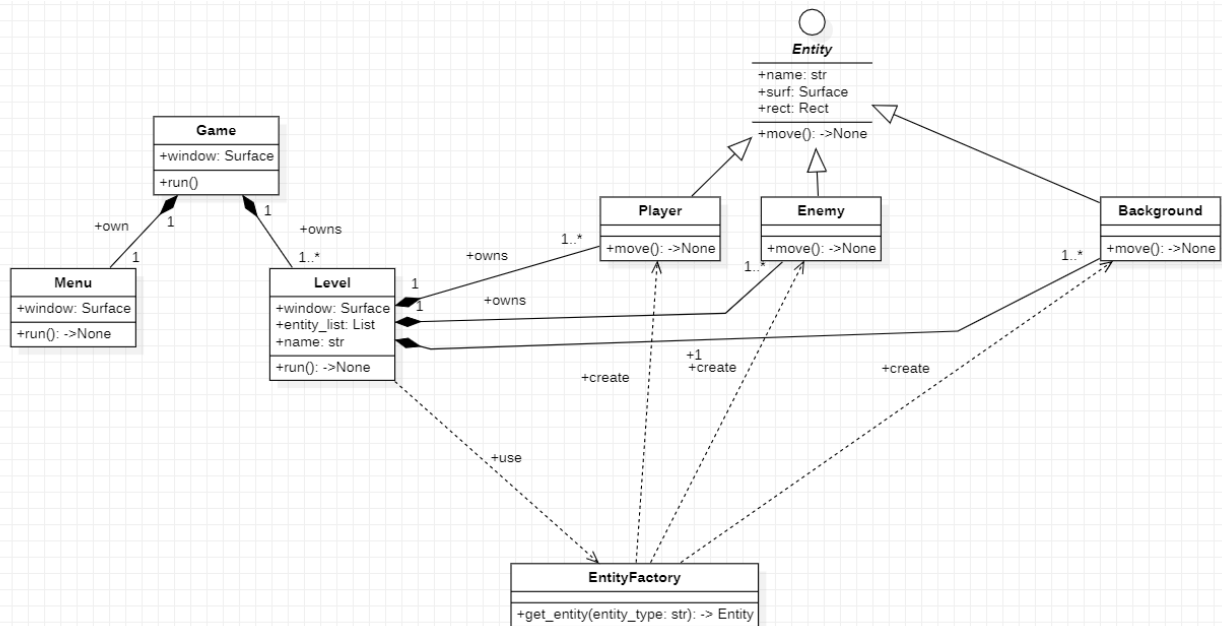
-> `window.blit(source=bg_surf, dest=bg_rect)` - desenha a superfície de fundo (`bg_surf`) na posição especificada por `bg_rect`.

-> `window.blit(source=player1_surf, dest=player1_rect)` - desenha a superfície do jogador 1 (`player1_surf`) na posição especificada por `player1_rect`.

método `pygame.display.flip()` se torna importante para a visualização das superfícies na tela ele atualiza a tela inteira para refletir tudo o que foi desenhado desde a última chamada que basicamente ele troca o buffer de exibição, ou seja tudo o que foi desenhado até o momento será mostrado na tela

## Questão 02 – Aula Prática 02

**Na aula prática 02, começamos a construir o diagrama UML do nosso jogo (figura abaixo). A partir desse digrama explique a relação entre Level, Player e EntityFactory e de que maneira objetos da classe player não compõe EntityFactory, mas sim Level.**



RESPOSTA: (INSERIR RESPOSTA AQUI)

o level é como o chefe dos jogadores player ele não só possui os jogadores mas também é responsável por gerenciar e controlar tudo que eles fazem para criar os jogadores o level pede ajuda para a entityfactory a entityfactory faz o trabalho de criar os jogadores quando o level pede mas depois disso ela não se envolve mais então basicamente a entityfactory só faz os jogadores enquanto o level cuida e controla o que eles fazem os jogadores pertencem ao level não a entityfactory



## Questão 03 – Aula Prática 03

**Na aula prática 03, instanciou-se 14 objetos da classe background que foram inseridos dentro de uma lista de entidades que compõe o objeto Level1. Explica com suas próprias palavras como se alcançou o efeito de profundidade no cenário. Na sua explicação deve apresentar o motivo das 14 instâncias do objeto background.**

RESPOSTA: (INSERIR RESPOSTA AQUI)

pra alcançar o efeito de profundidade no cenário foi utilizada a técnica de parallax scrolling o efeito de profundidade é alcançado ao mover cada uma das 14 camadas do cenário a diferentes velocidades, com as camadas mais distantes movendo mais lentamente do que as camadas mais próximas e isso cria uma ilusão de profundidade, assim o ambiente do jogo fica mais realista e imersivo.



## Questão 04 – Aula Prática 04

**Na aula prática 04, utilizou-se da estratégia de instanciação de inimigos via evento temporal dentro da classe Level. De que maneira, os eventos são verificados? Apresente também é em que momento se atrela o intervalo temporal com o id do evento**

RESPOSTA: (INSERIR RESPOSTA AQUI)

os eventos são verificados dentro do loop `for event in pygame.event.get():`, ele roda sem parar enquanto o jogo está funcionando o loop principal do jogo verifica tudo o que aconteceu desde a última vez que ele rodou e o tempo do intervalo é configurado com a linha `pygame.time.set_timer(EVENT_ENEMY, 2000)`