

# Documentacao do Jogo - BASED

## 1. VISAO GERAL

Jogo educativo sobre alimentacao saudavel com 4 fases diferentes. Desenvolvido em Python com Pygame.

## 2. ITEM.PY - Sistema de Itens

Representa todos os itens coletaveis e obstaculos do jogo.

### Tipos de Itens:

- hamburger, refrigerante, sorvete = Comidas ruins (prejudicam)
- maca, alface, banana = Comidas boas (coletaveis)
- pedra, cacto = Obstaculos mortais (game over)

### Metodos:

- \_\_init\_\_(): Cria item, carrega imagem
- \_draw\_rock(): Desenha pedra
- \_draw\_cactus(): Desenha cacto
- is\_good\_item(): Verifica se e comida boa
- update(): Faz item flutuar

## 3. PLAYER.PY - Jogador

Controla o personagem principal.

### Atributos:

- direction: Vetor de movimento (x, y)
- speed: Velocidade
- gravity: Forca da gravidade
- on\_ground: Se pode pular
- good/bad\_items\_collected: Contadores

### Metodos:

- apply\_gravity(): Aplica gravidade
- jump(): Faz o jogador pular
- collect\_item(): Coleta item

## 4. LEVEL.PY - Fase 1 (Runner)

Fase estilo endless runner. Jogador corre automaticamente.

### Mecanicas:

- Scroll automatico que acelera com o tempo
- Obstaculos: pedras e cactos (game over ao tocar)
- Comidas ruins no caminho
- A cada 7 obstaculos: 2 comidas boas no ar

### Dificuldade Progressiva:

```
# A cada 5 segundos:
scroll_speed += 0.5 # Mais rapido
obstacle_interval -= 5 # Mais obstaculos
```

### Vitoria/Derrota:

- Vitoria: Coletar 9 comidas boas

## Documentacao do Jogo - BASED

- Derrota: Tocar em pedra ou cacto

# Documentacao do Jogo - BASED

## 5. LEVEL2.PY - Fase 2 (Agua)

Fase aquatica com movimento livre.

### Mecanicas:

- Movimento: setas para todas direcoes
- Tubaroes que se movem horizontalmente
- Comidas caem do topo
- Colisao com tubarao = game over

### Vitoria/Derrota:

- Vitoria: Coletar 9 comidas boas
- Derrota: Tocar em tubarao ou 8 comidas ruins

## 6. LEVEL3.PY - Fase 3 (Labirinto)

Labirinto com canhoes que atiram comida ruim.

### Mecanicas:

- Layout em grid (X = parede, espaco = passagem)
- 29 canhoes posicionados pelo mapa
- Canhoes atiram comidas ruins periodicamente
- Movimento: setas + pulo

### Vitoria/Derrota:

- Vitoria: Sair pelo topo do mapa
- Derrota: Ser atingido 3 vezes

## 7. LEVEL4.PY - Fase 4 (Boss)

Batalha final contra o Boss.

### Boss:

- Vida: 1800 HP
- Move em diagonal pela tela
- Atira comidas ruins em todas direcoes
- Rage Mode: 50% HP - fica mais rapido

### Jogador:

- Tecla F: Atira laser (20 dano)
- PowerUps: Armadura + Laser forte (40 dano)

### Cutscenes:

- Inicio: Dialogo de apresentacao
- Rage Mode: Boss fala 'Eu vou vencer!'

# Documentacao do Jogo - BASED

## 8. CONCEITOS PYGAME

### Sprites e Groups:

```
class Item(pygame.sprite.Sprite):
    self.image = pygame.Surface(size) # Visual
    self.rect = self.image.get_rect() # Posicao

self.items = pygame.sprite.Group()
self.items.add(item) # Adiciona
self.items.draw(screen) # Desenha todos
```

### Colisoes:

```
# Sprite vs Group
pygame.sprite.spritecollide(player, items, True)

# Rect vs Rect
player.rect.colliderect(tile.rect)
```

### Desenho:

```
pygame.draw.rect(surface, cor, rect)
pygame.draw.circle(surface, cor, centro, raio)
pygame.draw.polygon(surface, cor, pontos)
```

### Teclas:

```
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]: # Esquerda
if keys[pygame.K_SPACE]: # Pular
if keys[pygame.K_f]: # Atirar
```

### Audio:

```
pygame.mixer.music.load('musica.mp3')
pygame.mixer.music.play(-1) # Loop
pygame.mixer.music.stop()
```

## 9. FUNCOES PYTHON USADAS

### random:

```
random.choice(lista) # Item aleatorio
random.randint(min, max) # Inteiro aleatorio
random.shuffle(lista) # Embaralha
```

### math:

```
math.cos(angulo) # Cosseno
math.sin(angulo) # Seno
math.pi # 3.14159...
```

### getattr / setattr:

```
# Pega atributo ou valor padrao
getattr(self, 'timer', 0)

# Verifica se existe
if hasattr(item, 'is_falling'):
```

# Documentacao do Jogo - BASED

## 10. FLUXO DO JOGO

1. Menu inicial
2. Fase 1: Runner - Coletar 9 comidas, desviar obstaculos
3. Fase 2: Agua - Coletar 9 comidas, fugir tubaroes
4. Fase 3: Labirinto - Subir ao topo, desviar tiros
5. Fase 4: Boss - Derrotar com lasers
6. Vitoria!

## 11. ESTRUTURA DE ARQUIVOS

```
jog/  
  main.py      # Loop principal  
  player.py    # Classe jogador  
  item.py      # Itens/obstaculos  
  level.py     # Fase 1  
  level2.py    # Fase 2  
  level3.py    # Fase 3  
  level4.py    # Fase 4  
  assets/      # Imagens e sons
```

## 12. DICAS DE PROGRAMACAO

1. Sempre herde de pygame.sprite.Sprite para objetos visuais
2. Separe logica (update) de visual (draw)
3. Use constantes no topo (SCREEN\_WIDTH, TILE\_SIZE)
4. Use Groups para gerenciar multiplos sprites
5. hasattr() e getattr() evitam erros de atributos inexistentes