



# Projeto P1/LP1

Bombers

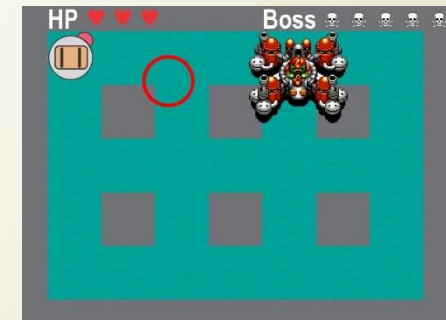
# Objetivos:

- ▶ Criar um jogo que não fosse 100% de chance de zerar de primeira tentativa.
- ▶ Colocar gráficos simples, para que se tenha uma maior harmonia.



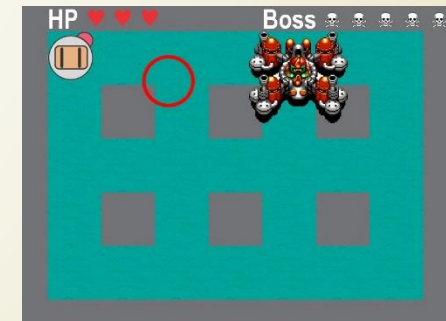
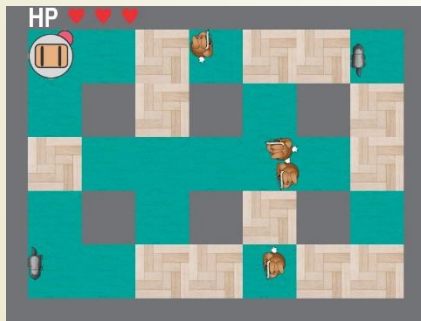
# O jogo : Menu

- ▶ Novo Jogo
  - ▶ Reinicia tudo e começa do level 1.
- ▶ Carregar
  - ▶ A cada fim de mapa é dada uma opção para salvar (utilizamos arquivo), onde pode ser acessada por aqui.
- ▶ Sair

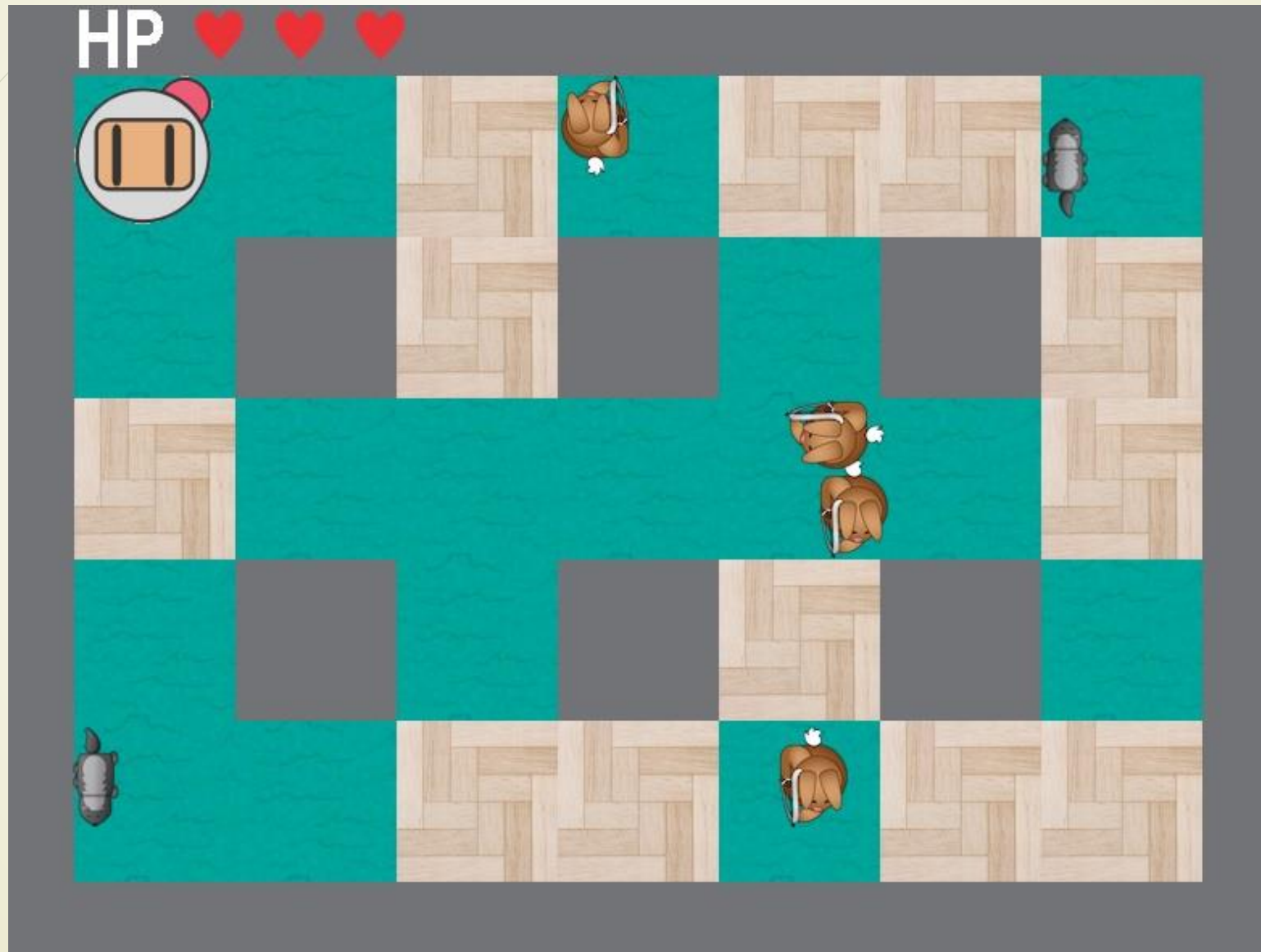


# O jogo :

- **Mapa 1:** Apenas 6 monstros, onde é preciso destruir todos para passar de fase.
- **Mapa 2:** 5 monstros, um deles é o CHEFE, que possui a habilidade de criar explosões ao redor do mapa.
- **Mapa 3:** 1 Monstro, que possui a habilidade do CHEFE do mapa 2 e lá uma arruela de fogo.



# Mapa 1

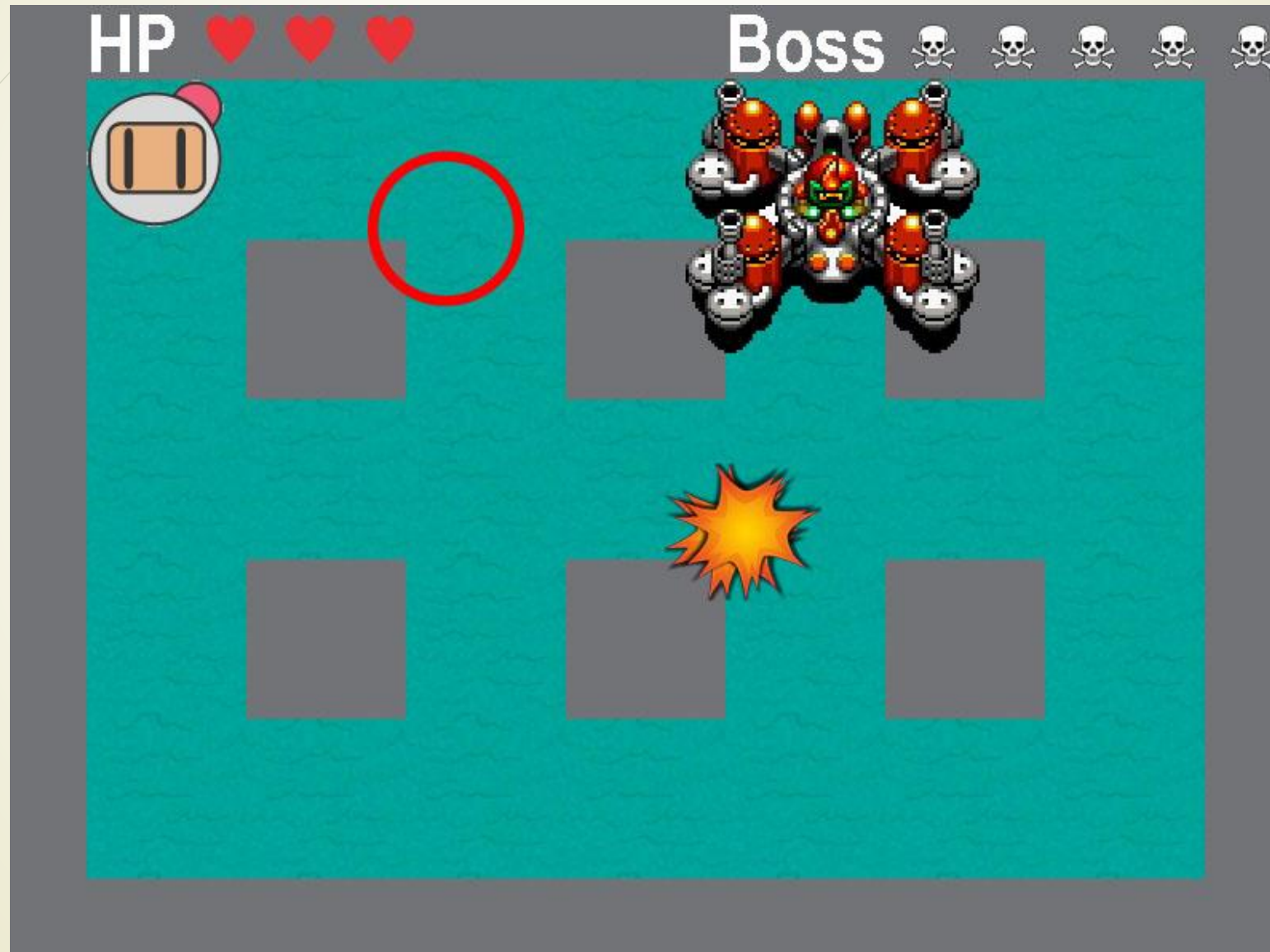




## Mapa 2



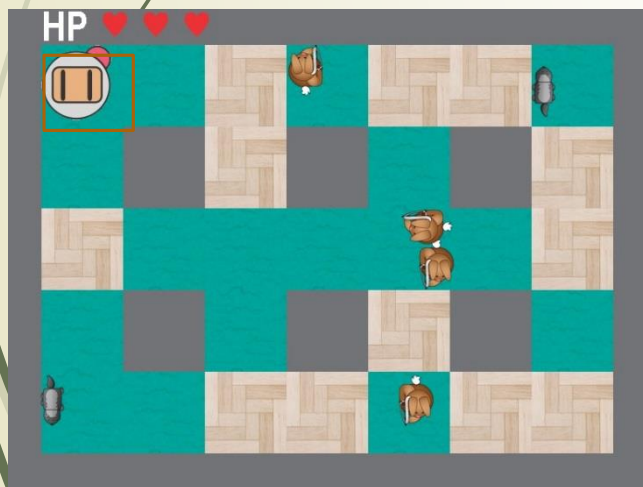
# Mapa 3





# Algumas explicações do código





```
#Carregar Objetos
person = Objetos('p.png', (50,50))
bomba = Objetos('bomba.png', (50,50))

#Carregar Textos
HP = Textos('HP')
texto_chefe = Textos('Boss')
```

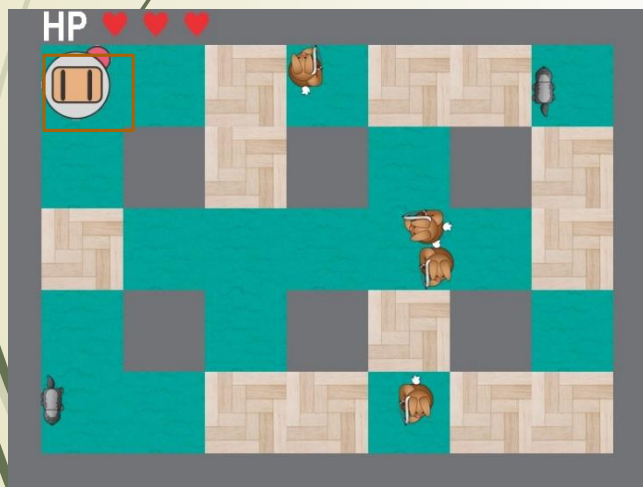
```
class Objetos:
    def __init__( self, imagem, posicao_inicial ):
        self.posicao = list( posicao_inicial )
        ''' Carregar Imagens '''
        diretorio_foto = os.path.join('images', imagem)
        self.imagem = pygame.image.load( diretorio_foto ).convert_alpha()

    def colidiu(self , obj2, obj2Px, obj2Py):
        return self.imagem.get_rect( center= (self.posicao[0] + self.imagem.get_size()[0]/2,

    def colidiu_bloco( self , lista_blocos ):
        for x in lista_blocos: #blocos indestrutíveis
            if self.imagem.get_rect(center= (self.posicao[0] + self.imagem.get_size()[0]/2, :
                if self.imagem == explosao_H.imagem or self.imagem == explosao_V.imagem:
                    blocos_madeira.remove( x )
            return True
        return False

class Textos:
    def __init__( self, mensagem ):
        self.fonte = font.SysFont("ARIAL", 50, True, False)
        self.texto = self.fonte.render(mensagem, 0, (255, 255, 255))

class Audio:
    def __init__( self, audio ):
        self.diretorio_audio = os.path.join('audio', audio)
        self.audio = pygame.mixer.Sound( self.diretorio_audio )
```



```
def eventosMapa():
    global teclas, menu, ativar_bomba, cont_bomba_explosao_v, cont_bomba_explosao_h, qnt_vida, lista_inimigos

    for evento in pygame.event.get():
        if evento.type == pygame.KEYDOWN:
            #andar
            if evento.key == K_w or evento.key == K_UP:
                teclas[0] = True

            elif evento.key == K_s or evento.key == K_DOWN:
                teclas[2] = True

            if evento.key == K_a or evento.key == K_LEFT:
                teclas[1] = True

            elif evento.key == K_d or evento.key == K_RIGHT:
                teclas[3] = True

            #bomba
            if evento.key == K_SPACE and not ativar_bomba :
                ativar_bomba = True
                bomba.posicao = person.posicao[0] + 10, person.posicao[1] + 10

                cont_bomba_explosao_v = 0
                cont_bomba_explosao_h = 0

                som_clicar.audio.play()

            if evento.key == K_PAGEUP:
                qnt_vida = 10
                lista_inimigos = []

            if evento.key == K_ESCAPE:
                som_lvl1.audio.stop()
                som_lvl2.audio.stop()
                nivel_fase = 1
                carregarFuncionalidadesMapa1()
                menu = True

        if evento.type == pygame.KEYUP: #parar_andar
            if evento.key == pygame.K_w or evento.key == K_UP:
                teclas[0] = False

            elif evento.key == pygame.K_s or evento.key == K_DOWN:
                teclas[2] = False

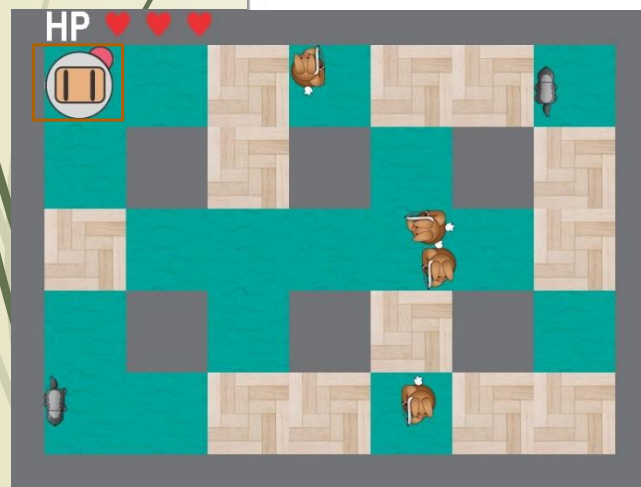
            if evento.key == pygame.K_a or evento.key == K_LEFT:
                teclas[1] = False
```

```
def MoverBomberman():
    if teclas[0] and person.posicao[1] > 50:
        person.posicao[1] -= velocidadeP
        if person.colidiu_bloco(blocos_indestrutíveis) or person.colidiu_bloco(blocos_madeira) :
            person.posicao[1] += velocidadeP

    if teclas[1] and person.posicao[0] > 50:
        person.posicao[0] -= velocidadeP
        if person.colidiu_bloco(blocos_indestrutíveis) or person.colidiu_bloco(blocos_madeira) :
            person.posicao[0] += velocidadeP

    if teclas[2] and person.posicao[1] + person_size[1] < 550:
        person.posicao[1] += velocidadeP
        if person.colidiu_bloco(blocos_indestrutíveis) or person.colidiu_bloco(blocos_madeira) :
            person.posicao[1] -= velocidadeP

    if teclas[3] and person.posicao[0] + person_size[0] < 750:
        person.posicao[0] += velocidadeP
        if person.colidiu_bloco(blocos_indestrutíveis) or person.colidiu_bloco(blocos_madeira) :
            person.posicao[0] -= velocidadeP
```





```
def desenharBomba():
    global cont_bomba_explosao_v, cont_bomba_explosao_h

    if ativar_bomba:
        tela.blit(bomba.imagem, bomba.posicao)
        cont_bomba_explosao_v += 1
        cont_bomba_explosao_h += 1

def desenharExplosao():
    global cont_bomba_explosao_v, cont_bomba_explosao_h, ativar_bomba, cont_explosao_desaparecer, cont_explosao

    if cont_bomba_explosao_v == 110 and cont_bomba_explosao_h == 110:
        ativar_bomba = False

        explosao_V.posicao[0], explosao_V.posicao[1] = bomba.posicao[0], bomba.posicao[1] - 105
        explosao_H.posicao[0], explosao_H.posicao[1] = bomba.posicao[0] - 105, bomba.posicao[1]

        tela.blit( explosao_V.imagem, explosao_V.posicao )
        tela.blit( explosao_H.imagem, explosao_H.posicao )

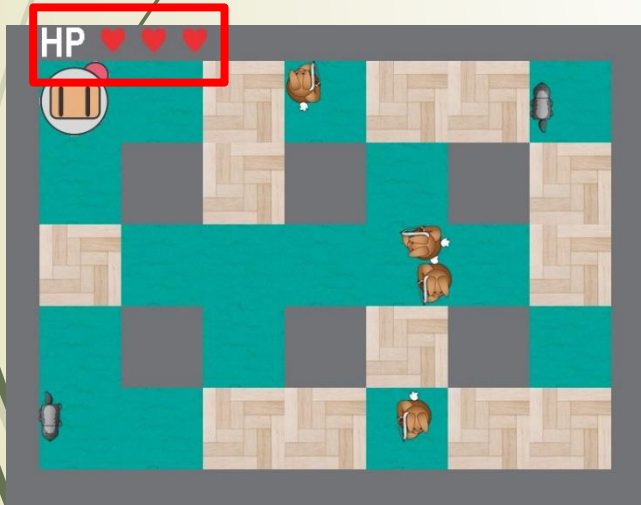
        if cont_explosao_desaparecer == 0:
            som_explosao.audio.play(0)

        cont_explosao_desaparecer += 1
        if cont_explosao_desaparecer == 65:
            cont_bomba_explosao_v = 0
            cont_explosao_desaparecer = 0

            explosao_V.posicao[0], explosao_V.posicao[1] = -300, -300
            explosao_H.posicao[0], explosao_H.posicao[1] = -300, -300

            indestrutivel = False
            chefe_indestrutivel = False
```





```
def vitoriaDerrota():
    global nivel_fase, qnt_vida, salvar_jogo

    if len(lista_inimigos) == 0:
        som_lvl1.audio.stop()
        som_lvl2.audio.stop()
        som_tensao.audio.stop()
        som_vitoria.audio.play()

        tela.blit(ganhou.imagem, ganhou.posicao)
        nivel_fase += 1

        pygame.display.flip()
        pygame.time.wait(4000)

        pygame.mixer.fadeout(500)
        som_vitoria.audio.stop()

        salvar_jogo = True

    if nivel_fase == 1:
        carregarFuncionalidadesMapa1()
    elif nivel_fase == 2:
        carregarFuncionalidadesMapa2()
    elif nivel_fase == 3:
        carregarFuncionalidadesMapa3()

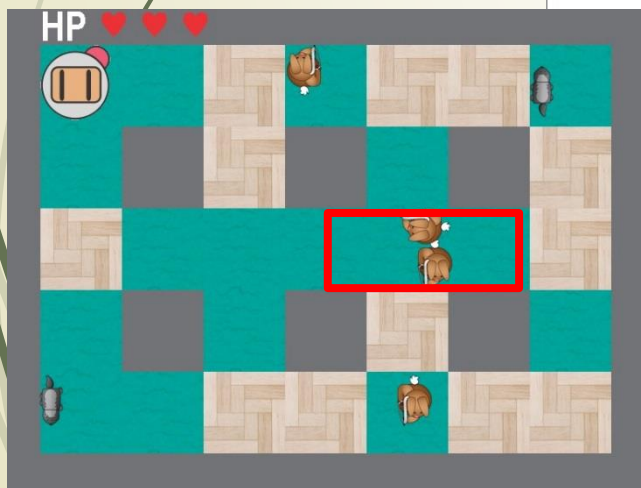
    elif qnt_vida <= 0:
        som_lvl1.audio.stop()
        som_tensao.audio.stop()
        som_derrota.audio.play()

        tela.blit(perdeu.imagem, perdeu.posicao)
        qnt_vida = 3
        nivel_fase = 1

        pygame.display.flip()
        pygame.time.wait(2000)

        som_derrota.audio.stop()

        carregarFuncionalidadesMapa1()
```



```
def mov_aleatorio(mob,direção):#principal
    global tempo_posição,tempo_posiçãoF
    tempo_posição += 1

    if tempo_posição > tempo_posiçãoF: #Onde muda
        tempo_posiçãoF = tempos_finais()
        direção = direção_aleatoria()
        tempo_posição = 0

    if direção == 1:#Para cima
        if not(mob.posicao[1]) < 50 and not(mob.posicao[0] < 150 and mob.posicao[1] < 150):#Parede/bloco invisivel
            mob.posicao[1] -= 1

            if mob.imagem == coelho1.imagem or mob.imagem == coelho2.imagem or mob.imagem == coelho3.imagem or mob.imagem == coelho4.imagem:
                mob.imagem = coelho2_cima.imagem #Olhar para cima
            if mob.imagem == rato1.imagem or mob.imagem == rato2.imagem:
                mob.imagem = rato_cima.imagem

            if ( mob.colidiu_bloco(blocos_indestrutíveis) or mob.colidiu_bloco(blocos_madeira) ) and mob.imagem != chefe_final.imagem:
                mob.posicao[1] += 1
                direção = direção_aleatoria()
        else: #para parede
            direção = direção_aleatoria()

    elif direção == 2:#Para esquerda
        if not(mob.posicao[0]) < 50 and not(mob.posicao[0] < 150 and mob.posicao[1] < 150):
            mob.posicao[0] -= 1

            if mob.imagem == coelho1.imagem or mob.imagem == coelho2.imagem or mob.imagem == coelho3.imagem or mob.imagem == coelho4.imagem:
                mob.imagem = coelho2_esquerda.imagem #Olhar para esquerda
            if mob.imagem == rato1.imagem or mob.imagem == rato2.imagem:
                mob.imagem = rato_esquerda.imagem

            if ( mob.colidiu_bloco(blocos_indestrutíveis) or mob.colidiu_bloco(blocos_madeira) ) and mob.imagem != chefe_final.imagem:
                mob.posicao[0] += 1
                direção = direção_aleatoria()
        else:
            direção = direção_aleatoria()
```

```

def carregarFuncionalidadesMapa1():
    global l_traco_posicao2, tempo_posicao, tempo_posicaoF, direcao1, direcao2, direcao3, direcao4, direcao5, direcao6, continuar, indestrutivel, tracoP, l_traco
    tempo_posicao = 0
    tempo_posicaoF = 120
    direcao1, direcao2, direcao3, direcao4, direcao5, direcao6 = 1, 1, 1, 1, 1, 1
    #Carregar Objetos
    person.posicao = [50, 50]
    bomba.posicao = [50, 50]
    coelho1.posicao = [270, 270]
    coelho2.posicao = [360, 65]
    coelho3.posicao = [560, 270]
    coelho4.posicao = [465, 475]
    rato1.posicao = [665, 85]
    rato2.posicao = [165, 485]
    explosao_V.posicao = [-300, -300]
    explosao_H.posicao = [-300, -300]
    ganhou.posicao = (200, 200)
    perdeu.posicao = (200, 200)
    fundo_menu.posicao = (0, 0)
    person_size = person.imagem.get_size()
    #Carregar variaveis
    continuar = True
    qnt_vida = 3
    qnt_habi = 0
    blocos_madeira = [ (250, 50), (450, 50), (550, 50), (250, 150), (650, 150), (50, 250), (650, 250), (450, 350), (250, 450), (350, 450), (550, 450), (650, 450) ]
    blocos_indestrutíveis = (150, 150), (350, 150), (550, 150), (150, 350), (350, 350), (550, 350)
    lista_inimigos = [coelho1, coelho2, coelho3, coelho4, rato1, rato2]
    teclas = [False, False, False, False] #w/a/s/d
    ativar_bomba = False
    velocidadeP = 2
    lista_habilidades = []
    l_traco_posicao = (526, 406), (526, 491), (526, 568)
    l_traco_posicao2 = (129, 432), (591, 442)
    tracoP = 0
    indestrutivel = False
    chefe.posicao = [450, 250]

    while qnt_habi != 3: #carregando habilidades
        hab = random.choice( blocos_madeira )
        if hab not in lista_habilidades:
            lista_habilidades.append( hab )
            qnt_habi += 1

    #contadores de tempo
    cont_bomba_explosao_v = 0
    cont_bomba_explosao_h = 0
    cont_explosao_desaparecer = 0

```







# Equipe:

- ▶ Adriano Alves dos Santos
  - ▶ Carlos Henrique de Macêdo
  - ▶ Pedro Bandeira Milfont
- 