

Projeto de Programação Orientada a Objetos 1

Julien Hervot de Mattos Vaz

Novembro 2019

Pokérún

Matrícula: 19202276

Turma: 01208B

Professor: Maicon Zatelli

1 Resumo

O intuito desse projeto foi de, com os conhecimentos adquiridos em POO1 durante o semestre e com o auxílio de conteúdos externos às aulas, criar um jogo com interface gráfica utilizando a biblioteca Pygame. O tipo de jogo escolhido foi um side scroller por ser um tipo popular e por oferecer certo desafio na construção. O tema do jogo foi uma mistura de temas que me remetem à infância, sendo eles a série de jogos Pokémon e o MMORPG Ragnarok Online. O meu apreço por pixel art também se fez presente no trabalho.

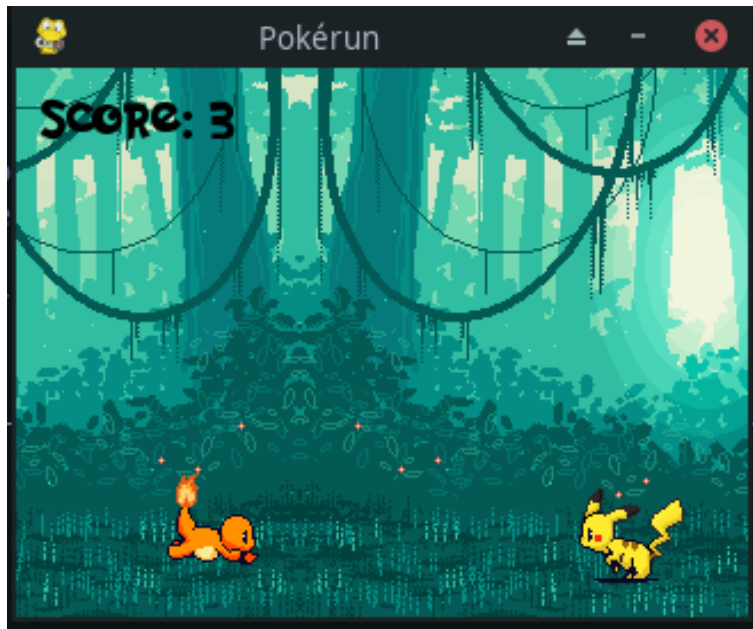


Figure 1: Screenshot do jogo.

2 Materiais utilizados

2.1 Código fonte

Em virtude do raso conhecimento e da falta de experiência em programação, foi necessário recorrer a tutoriais e vídeo aulas sobre o Pygame. O canal acessado foi Tech With Tim e a playlist foi Side Scroller. Também foram retiradas informações da documentação oficial do Pygame.

O fundo móvel foi implementado usando variáveis posicionais relativas, como explicado nos comentários do código. Quando a imagem ultrapassa o limite da tela, ela reseta sua posição.

A pontuação é referente a velocidade atual do loop de jogo.

As colisões são detectadas a partir da sobreposição dos sprites dos personagens. Verifica-se duas possíveis situações: colisão dianteira e colisão traseira.

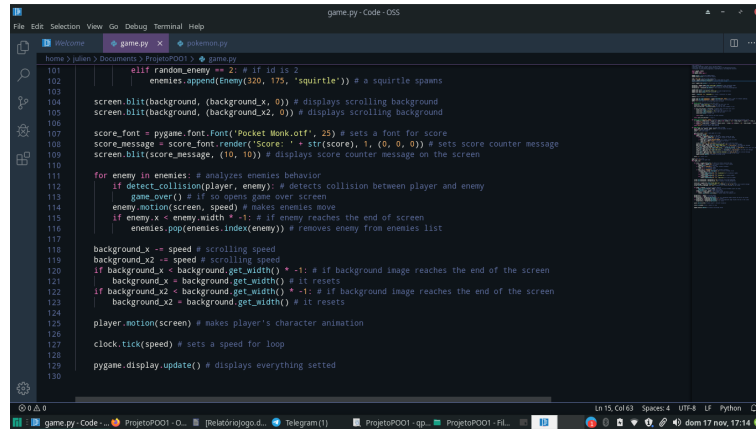


Figure 2: Fundo móvel, pontuação, colisões e animação.

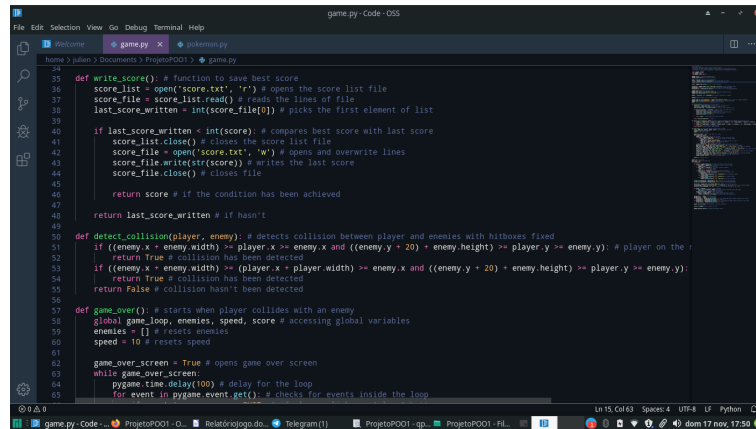


Figure 3: Pontuação máxima e colisões.

As classes Player e Enemy herdam o método construtor da classe Pokemon, assim como a função de animação dos frames.

Para facilitar o controle das animações foi criada a função motion em ambas as classes, porém com comportamentos diferentes (polimorfismo). A do jogador permite a animação de corrida e a de salto. Enquanto que a dos inimigos permite a animação de corrida e a de mudança de posição.

A animação do personagem é descrita dentro da classe respectiva. Existe um contador de frames que reseta quando chega em 3 e incrementa enquanto não alcança tal valor. O personagem recebe o frame atual durante a animação.

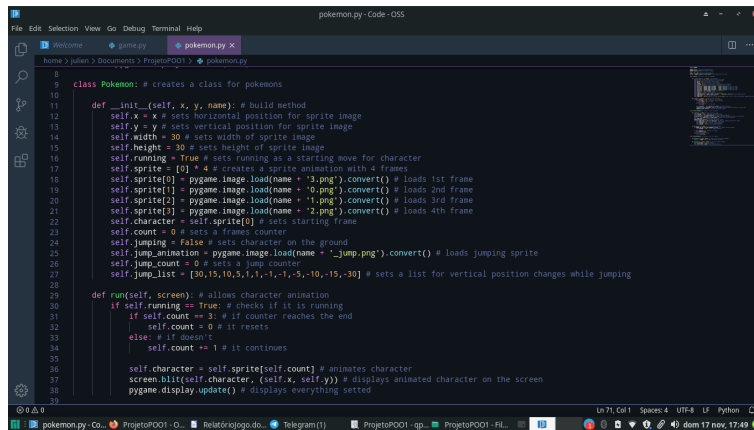


Figure 4: Método construtor da classe Pokemon, animação de frames, de salto e movimento.

A animação de salto foi implementada com um contador para o tempo de vôo e uma lista com as mudanças em pixel da posição vertical do personagem. Verificando se o jogador está pulando, quando a barra de espaço é pressionada, a animação de corrida é interrompida e a de salto inicia. A posição do personagem é modificada seguindo a lista de mudanças em pixel, as quais o contador acessa. Apenas o jogador consegue pular, os inimigos não. Os inimigos possuem, além da animação de frames, uma função que move sua posição horizontal.

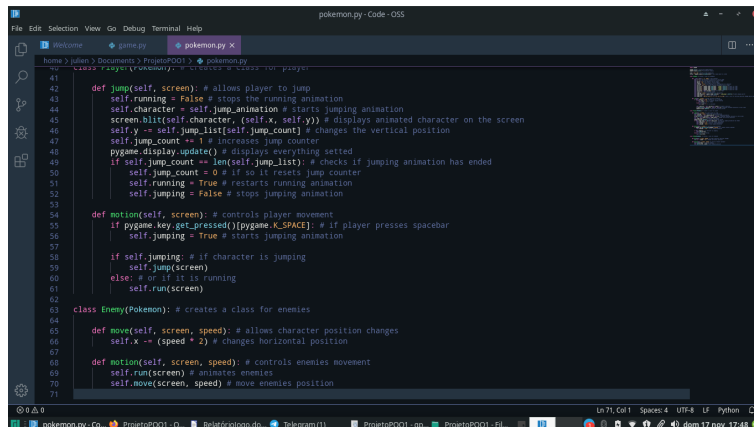


Figure 5: Funções de salto e movimento.

Também foram implementados timers, para aumentar a velocidade do jogo e para gerar inimigos aleatoriamente. Não menos importante, foi implementado uma detecção de evento, caso o jogador queira fechar o jogo.

```

79 press_space = game_over_font.render('Click to continue!', 1, (255,255,0)) # sets continue message
80 screen.blit(press_space, (60, 180)) # displays continue message on the screen
81 pygame.display.update() # displays everything setted
82
83 score = 0 # resets score
84
85 score = 0 # sets score
86 game_loop = True # sets game loop
87 while game_loop:
88     for event in pygame.event.get(): # checks for events inside the loop
89         if event.type == pygame.QUIT: # if player clicks on 'close' button
90             pygame.quit() # quits the game
91         if event.type == USEREVENT1: # if timer reaches threshold
92             speed += 1 # it increases speed
93             score += 1 # it increases score
94         if event.type == USEREVENT2: # if timer reaches threshold
95             random_enemy = random.randrange(0, 3) # it sorts an enemy id out
96             if random_enemy == 0: # if it is 0
97                 enemies.append(enemy(320, 175, 'bulbassaur')) # a bulbassaur spawns
98             elif random_enemy == 1: # if it is 1
99                 enemies.append(enemy(320, 175, 'pikachu')) # a pikachu spawns
100             elif random_enemy == 2: # if it is 2
101                 enemies.append(enemy(320, 175, 'squirtle')) # a squirtle spawns
102
103     screen.blit(background, (background_x, 0)) # displays scrolling background
104     screen.blit(background, (background_x2, 0)) # displays scrolling background
105
106     score_font = pygame.font.Font('Pocket Monk.ttf', 25) # sets a font for score
107     score_message = score_font.render('Score: ' + str(score), 1, (0, 0, 0)) # sets score counter message
108     screen.blit(score_message, (10, 10)) # displays score counter message on the screen
109
110

```

Figure 6: Eventos de fechar janela de jogo, aumento da velocidade e geração de inimigos aleatórios.

A pontuação máxima é gravada em um arquivo de texto e é acessada pela função respectiva. Verifica-se a pontuação obtida e compara-se com a máxima registrada no arquivo de texto. A maior pontuação é armazenada.

```

56 def game_over(): # starts when player collides with an enemy
57     global game_loop, enemies, speed, score # accessing global variables
58     enemies = [] # resets enemies
59     speed = 10 # resets speed
60
61     game_over_screen = True # opens game over screen
62     while game_over_screen:
63         pygame.time.delay(100) # delay for the loop
64         for event in pygame.event.get(): # checks for events inside the loop
65             if event.type == pygame.QUIT: # if player clicks on 'close' button
66                 pygame.quit() # quits the game
67             if event.type == pygame.MOUSEBUTTONDOWN: # if player clicks on screen
68                 game_over_screen = False # closes game over screen
69                 game_loop = True # restarts the game
70
71         screen.fill((0,0,0), (0,0,320,240)) # fills the game over screen with black
72         game_over_font = pygame.font.Font('Pocket Monk.ttf', 30) # sets a font for messages
73         game_over_message = game_over_font.render('Game over!', 1, (255,255,0)) # sets game over message
74         screen.blit(game_over_message, (100, 30)) # displays game over message on the screen
75         best_score = game_over_font.render('Best score: ' + str(write_score(1, 1, (255,255,0))), 1, (255,255,0)) # sets best score message
76         screen.blit(best_score, (35, 80)) # displays best score message on the screen
77         new_score = game_over_font.render('New score: ' + str(score), 1, (255,255,0)) # sets new score message
78         screen.blit(new_score, (35, 120)) # displays new score message on the screen
79         press_space = game_over_font.render('Click to continue!', 1, (255,255,0)) # sets continue message
80         screen.blit(press_space, (60, 180)) # displays continue message on the screen
81         pygame.display.update() # displays everything setted
82
83     score = 0 # resets score
84
85     score = 0 # sets score
86     game_loop = True # sets game loop

```

Figure 7: Função de fim de jogo.

Quando o personagem colide com algum inimigo, a função de fim de jogo é chamada e o jogador depara-se com a tela de fim de jogo. Essa contém a pontuação máxima e a atual, além de instruções para continuar a jogar.

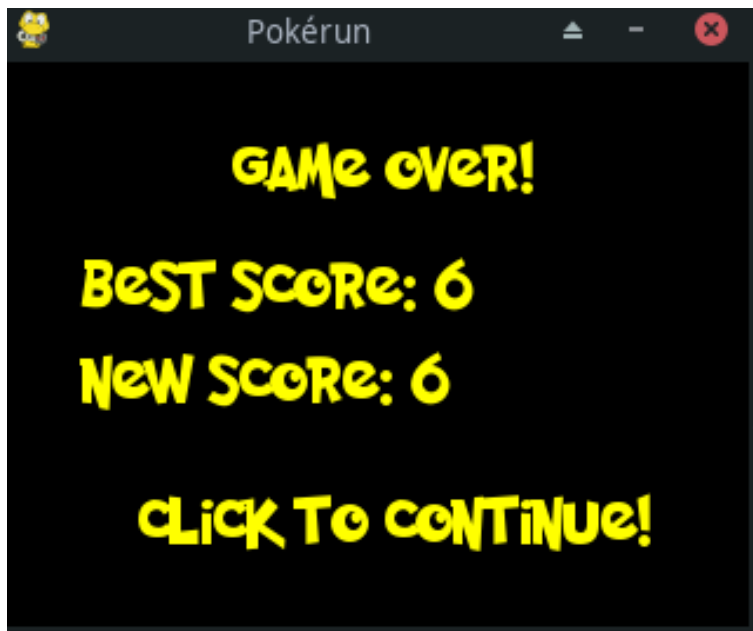


Figure 8: Tela de fim de jogo.

2.2 Imagens

Tanto a imagem de fundo, quanto os sprites dos personagens foram encontrados na galeria digital DeviantArt, sendo os artistas Forheksed (imagem de fundo) e chi-u (sprites).

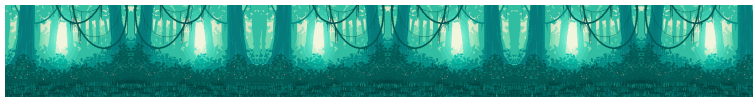


Figure 9: Imagem de fundo



Figure 10: Alguns sprites

2.3 Música

A trilha sonora foi retirada do MMORPG Ragnarok Online e foi escolhida por ambientar bem a imagem de fundo.

2.4 Fonte

A fonte escolhida foi Pocket Monk, criada por Chequered Ink, por ser uma fonte parecida com a oficial da série Pokémon.

3 Regras e comandos

As regras do jogo são simples. Utilize a barra de espaço para pular sobre os inimigos que vêm de encontro a você e tente chegar o mais longe possível. Clique na tela para recomeçar o jogo quando perder. A pontuação é definida pela velocidade atual alcançada.