

Trabalho de Linguagem de Programação

Space game

Autor: Heitor Vinícius da Costa Silva.

Introdução ao jogo

Primeiramente eu havia recriado o jogo “flappy bird” porém devido orientação do monitor durante a primeira apresentação resolvi mudar a estética do jogo mantendo alguns elementos da jogabilidade do flappy bird.

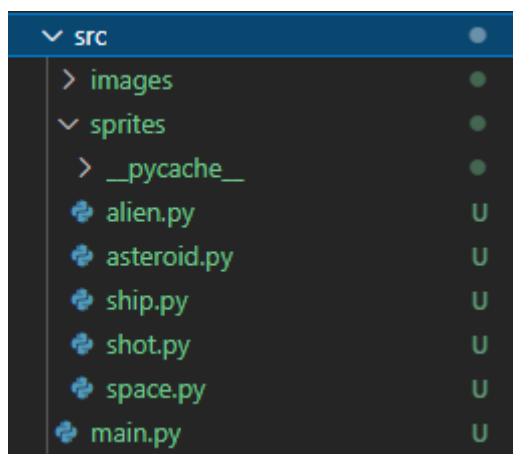
Dessa vez o projeto consiste em um jogo no espaço onde uma nave espacial precisa desviar de asteroides e matar alienígenas para continuar sua viagem, o jogo tem uma jogabilidade simples onde você pode somente controlar através das setas do teclado e atirar utilizando a tecla espaço.

Embora à primeira vista o jogo pareça simples, com ele pude implementar diversos conceitos estudados na disciplina de Linguagem de programação como: Laços, classes, funções e loops.

Organização do código

Durante o processo de produção do código, antes de organizá-lo em vários arquivos e pastas vinha encontrando diversos problemas para ler o código e localizar os elementos, porém, utilizando os métodos de organização indicados pelo professor Sergio e pelos monitores dividi o código tanto em duas pastas (uma chamada “images” para os arquivos dos sprites e outra chamada “sprites” para os códigos das classes) deixando também o arquivo “main.py”, responsável por rodar o jogo, separado fora dessas duas pastas.

Além disso, dentro da pasta “sprites” criei um arquivo .py individual para cada classe usada no código e os importei posteriormente para o arquivo “main.py”, é possível ver um pouco da organização do código na imagem abaixo:



Funções presentes no jogo

Para a entrega do projeto fiz diversas funções para o jogo, desde funções que não interagem com o usuário, sendo alguns exemplos delas as funções `pos_asteroide()` e a função `fora_da_tela()` que são responsáveis respectivamente por gerar a posição dos asteroides na tela e por verificar se os sprites estão fora da tela evitando uso de memória desnecessário

Além disso, temos também as funções para realização de ações dentro do jogo como as funções que movem a nave e a função responsável por gerar os tiros, ademais o código conta com diversas classes, servindo para representar coisas como a nave, os asteroides, os aliens e os projéteis lançados por eles, algumas dessas principais funções e classes estão representadas nas imagens abaixo:

```
class Ship(pygame.sprite.Sprite):

    def __init__(self):
        pygame.sprite.Sprite.__init__(self)

        self.image = pygame.image.load('src/images/Ship.png').convert_alpha()
        self.image = pygame.transform.scale(self.image, (45,40))
        self.mask = pygame.mask.from_surface(self.image)

        self.rect = self.image.get_rect()
        self.rect[0] = 400 / 2
        self.rect[1] = 800 / 2

    def down(self):
        self.rect[1] += 40

    def up(self):
        self.rect[1] -= 40

    def left(self):
        self.rect[0] -= 40

    def right(self):
        self.rect[0] += 40
```

```
class Alien(pygame.sprite.Sprite):

    def __init__(self, xpos, ypos):
        pygame.sprite.Sprite.__init__(self)

        self.image = pygame.image.load('src/images/alien.png').convert_alpha()
        self.image = pygame.transform.scale(self.image, (65,60))

        self.mask = pygame.mask.from_surface(self.image)

        self.ypos = ypos

        self.rect = self.image.get_rect()
        self.rect[0] = xpos
        self.rect[1] = ypos
        self.operator = '-'

        self.shot_timer = pygame.time.get_ticks()
        self.shot_delay = 1000
```

Funções que não estão presentes no jogo

A única funcionalidade que eu pretendia adicionar no jogo e que não está presente ainda era a presença de um menu antes do jogo iniciar, que infelizmente não foi implementada por falta de tempo, além disso uma outra implementação interessante no jogo seria adicionar outros tipos de inimigos e alguns “power-ups” para a nave, como um escudo, algo que aumentasse a velocidade dos tiros entre outros.

Observações e dificuldades sobre o processo de produção do jogo

Inicialmente tive uma certa dificuldade para compreender o modo como o pygame posicionava os elementos e atualizava a tela, porém depois de me familiarizar com as funções `get_rect()`, `update()` e `draw()` o processo de produção do jogo se tornou muito mais fluido

Além disso, tive somente alguns pequenos problemas de lógica na hora de implementar algumas funções como fazer o tiro dos aliens perseguir a nave e fazer os aliens se movimentarem para cima e para baixo.

Como executar o jogo

Para executar o jogo basta baixar o projeto presente em [spacegame](#) e executar o arquivo “main.py” que se encontra na pasta “src” e então jogar a vontade.