

Snake in Python (with Curses)

Preparação:

1. Transferir a package do Curses para a versão do Python a utilizar (https://github.com/neeitad/ClubeProgramacao-2018_03_07/tree/master/CursesInstallers);
2. Instalar a package através da consola do Anaconda, com o seguinte comando na diretoria onde a package está localizada: `python -m pip install curses-2.2-cp##-win32/64.whl`;
3. Nas opções (*Tools/Preferences*) do Spyder, na categoria *Run*, seleccionar *Execute in an external system terminal* e *Interact with the Python console after execution*.

Construção do programa:

1. Fazer import do módulo curses: `import curses`. Curses é uma biblioteca que fornece funções para tratamento do teclado e desenho num terminal de texto;
2. Inicializar a biblioteca (`curses.initscr()`), criar uma janela com as proporções desejadas (`win = curses.newwin(altura, comprimento, startingpointx, startingpointy)`) e fechar a janela no final do código (`curses.endwin()`);
3. Criar o vetor **snake**, com o tamanho do corpo inicial da cobra (neste caso, 3) e com a coordenada de cada parte do corpo (respeitando o tamanho da janela criada):
`snake = [[4,10], [4,9], [4,8]]`;
4. Criar o vetor **food**, que têm as coordenadas de onde a comida irá estar no ecrã (respeitando o tamanho da janela criada): `food = [10,20]`;
5. Inicializar a pontuação: `score = 0`;
6. A cabeça da cobra está direccionada para a direita, por isso inicializamos a tecla ativa como sendo a seta da direita:
 - a. Ativar o reconhecimento das teclas de direcção (seta para cima, seta para baixo, etc) na janela criada: `win.keypad(1)`;
 - b. Inicializar tecla ativa: `key = curses.KEY_RIGHT`;
7. Desenhar a comida no ecrã com a função `addch(coorx, coory, char)` do objeto **win**:
`win.addch(food[0], food[1], '*')`;
8. O jogo irá correr enquanto o jogador não perder ou não pressionar o *Esc*, logo cria-se um `while key != 27`: que irá conter o resto do código todo (27 é o *Esc* em ASCII);
9. Criar interface gráfica:
 - a. Ativar a borda à volta da janela: `win.border(0)`;
 - b. Escrever a pontuação: `win.addstr(0, 2, 'Score : ' + str(score) + '')`;
 - c. Escrever o nome do jogo: `win.addstr(0, 27, ' SNAKE ')`;
10. Definir a velocidade a cobra (na verdade, é definido o tempo que a janela espera pelo input do utilizador): `win.timeout(int((150 - (len(snake)/5 + len(snake)/10)%120))`;
11. Registrar o input do utilizador:
 - a. Ler o input do utilizador (retorna -1 se não houver input): `event = win.getch()`;
 - b. Registrar nova tecla ativa e guardar a anterior:

```
if (event != -1):
    prevKey = key
    key = event
```
12. Caso o input tenha sido a tecla *Space* (if `key == ord(' ')`):
 - a. Elimina a tecla ativa: `key = -1`;

- b. Escreve “PAUSE” no ecrã: `win.addstr(10, 25, ' PAUSED ');`
 - c. Espera até o jogador premir novamente o *Space*:
`while key != ord(' '):`
`key = win.getch()`
 - d. Assim que for premido novamente o *Space*, volta para a tecla ativa anterior:
`key = prevKey;`
 - e. Apagar o texto “PAUSE” do ecrã: `win.addstr(10, 25, ' ');`
 - f. Volta para o início do while, ignorando o resto do código: `continue;`
13. Se a tecla premida não for nenhuma das setas direcionais nem o *Esc*, voltar para a tecla ativa anterior:
`if key not in [curses.KEY_LEFT, curses.KEY_RIGHT, curses.KEY_UP, curses.KEY_DOWN, 27]:`
`key = prevKey`
14. Impedir que a cobra se mova diretamente para trás:
 - a. Se se tiver a mover para a direita, não pode ir para a esquerda:
`if (key == curses.KEY_LEFT and prevKey == curses.KEY_RIGHT):`
`key = prevKey`
 - b. Se se tiver a mover para a esquerda, não pode ir para a direita:
`elif (key == curses.KEY_RIGHT and prevKey == curses.KEY_LEFT):`
`key = prevKey`
 - c. Se se tiver a mover para baixo, não pode ir para cima:
`elif (key == curses.KEY_UP and prevKey == curses.KEY_DOWN):`
`key = prevKey`
 - d. Se se tiver a mover para cima, não pode ir para baixo:
`elif (key == curses.KEY_DOWN and prevKey == curses.KEY_UP):`
`key = prevKey`
15. Calcular as novas coordenadas da cabeça da cobra (ao ela mover-se):
 - a. Se a tecla ativa for seta para baixo, adiciona a cabeça uma coordenada abaixo:
`if(key == curses.KEY_DOWN):`
`snake.insert(0, [snake[0][0] + 1, snake[0][1]])`
 - b. Se a tecla ativa for seta para cima, adiciona a cabeça uma coordenada acima:
`elif(key == curses.KEY_UP):`
`snake.insert(0, [snake[0][0] - 1, snake[0][1]])`
 - c. Se a tecla ativa for seta para a esquerda, adiciona a cabeça uma coordenada à esquerda:
`elif(key == curses.KEY_LEFT):`
`snake.insert(0, [snake[0][0], snake[0][1] - 1])`
 - d. Se a tecla ativa for seta para a direita, adiciona a cabeça uma coordenada à direita:
`elif(key == curses.KEY_RIGHT):`
`snake.insert(0, [snake[0][0], snake[0][1] + 1])`
 - e. **NOTA:** a cauda não é retirada. Isso é feito mais tarde, consoante a cobra come a comida ou não;
16. Quando a cabeça da cobra tocar numa parede:
 - a. Terminar o jogo (19 e 59 são os limites verticais e horizontais da janela, respetivamente): `if snake[0][0] == 0 or snake[0][0] == 19 or snake[0][1] == 0 or snake[0][1] == 59: break;`

- b. Ou a cobra passa para o outro lado:

```
if snake[0][0] == 0: snake[0][0] = 18
if snake[0][1] == 0: snake[0][1] = 58
if snake[0][0] == 19: snake[0][0] = 1
if snake[0][1] == 59: snake[0][1] = 1
```

17. Se a cobra tocar nela própria, terminar o jogo: `if snake[0] in snake[1:]: break;`

18. Se a cabeça da cobra tocar na comida (`if snake[0] == food:`):

a. Aumentar pontuação: `score += 1;`

b. Esvaziar vetor **food**: `food = [];`

c. Calcular novas coordenadas para a comida:

- i. Se as coordenadas calculadas pertencerem à cobra, volta a recalcular:

```
while food == []:
    food = [randint(1, 18), randint(1, 58)]
    if food in snake: food = []
```

- ii. Adicionar a função `randint(first, last)` do módulo **random** (no início do código): `from random import randint;`

d. Redesenhar comida no ecrã (nas novas coordenadas): `win.addch(food[0], food[1], '*');`

19. No entanto, se a cabeça da cobra não tocar na comida (`else:`):

a. Eliminar a cauda do vetor **snake**: `last = snake.pop();`

b. Apagar a cauda da cobra do ecrã: `win.addch(last[0], last[1], ' ');`

20. Desenhar a cabeça da cobra: `win.addch(snake[0][0], snake[0][1], '#');`

21. Desativar o echo criado pelo input de caracteres (antes do while): `curses.noecho()`. Sem este comando, quando é premido o espaço e o escape, o input destes aparece no ecrã momentaneamente;

22. Esconder o cursor (antes do while): `curses.curs_set(0);`

23. Escrever a pontuação final depois de o jogo acabar (depois do `curses.endwin()`):
`print("\nScore - " + str(score))`