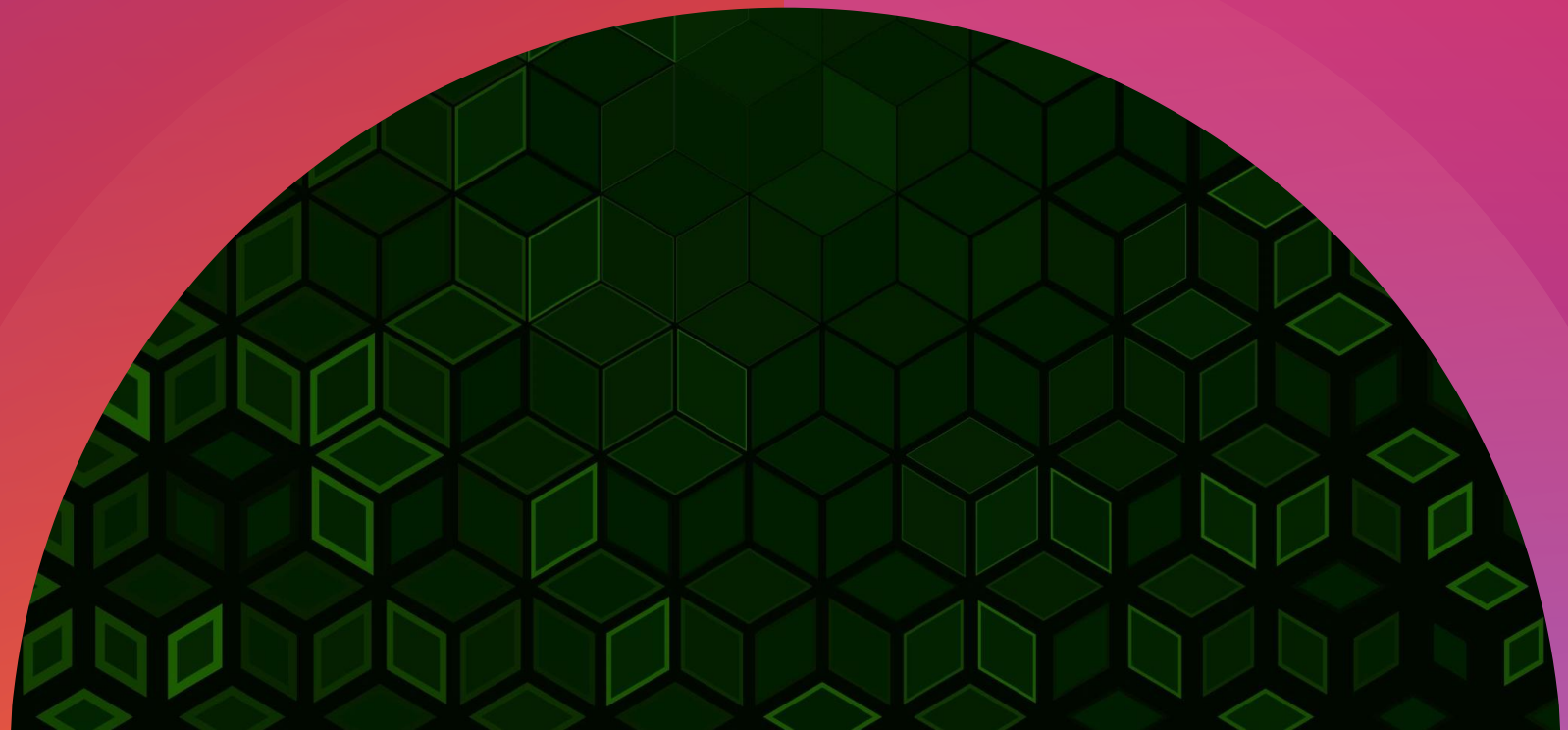
LADDER GAME, BUT IT'S A MATRIX

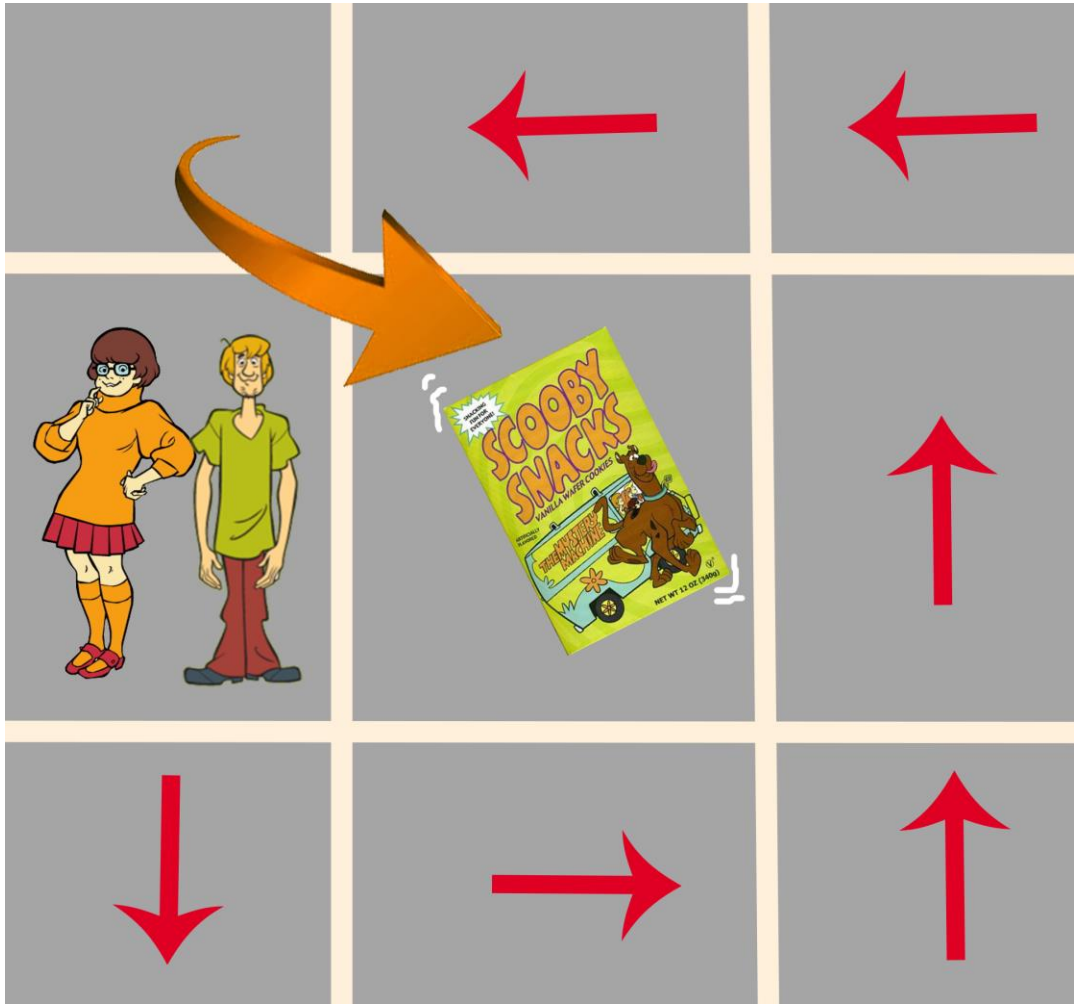
OLHA KACHAN

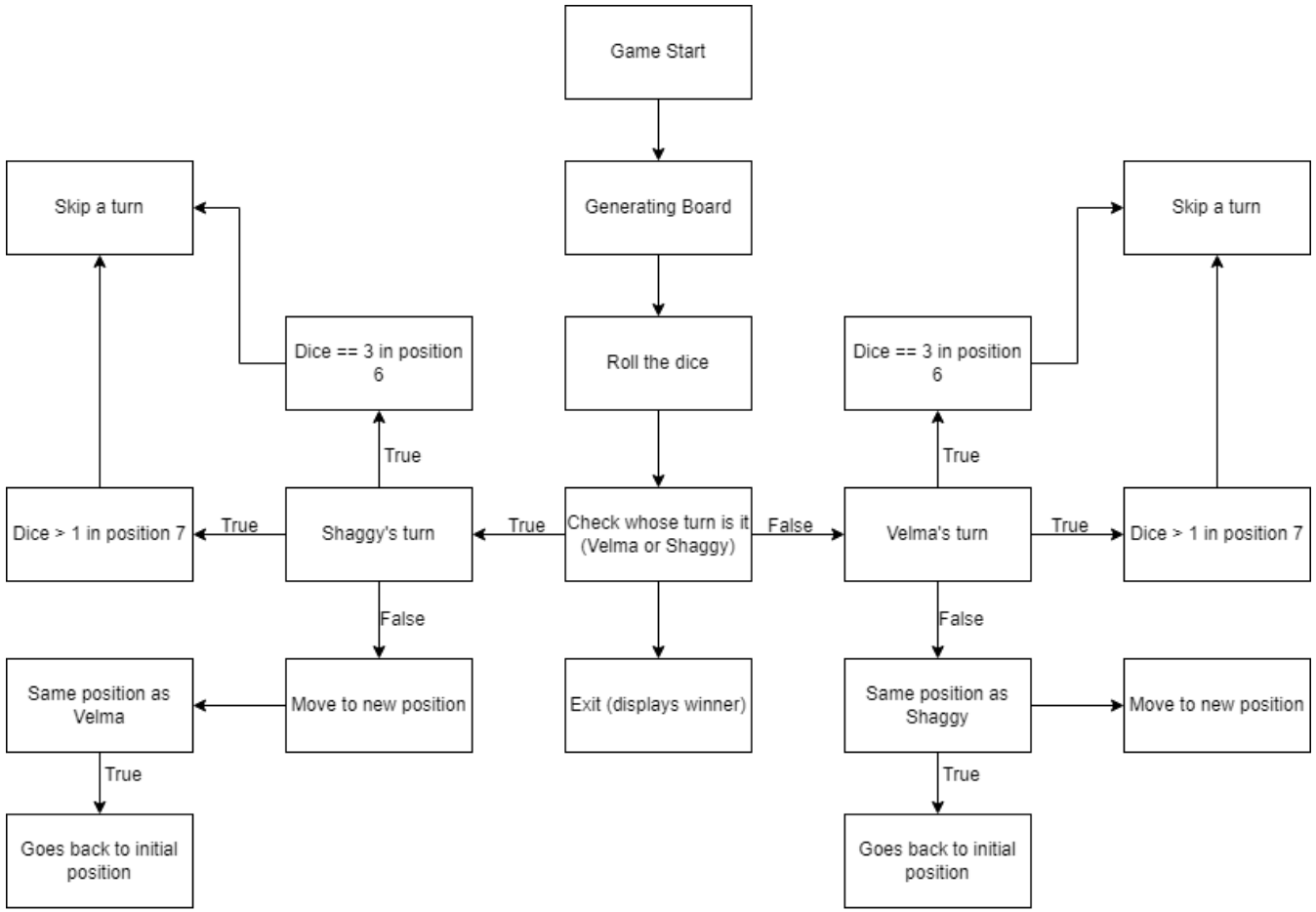
BEATRICE ANTONIU



INTRODUCTION

- We have two players: S and O (in our case Velma and Shaggy)
- Each player rolls the dice between 1 and 3 and then move the according number shown on the die.
- If player meet on the same position, the last one to get there, kills (not so viciously murders) the first one, who was already in the spot.
- If any of the players find themselves on the 6th position, they must roll either 1 or 2 or their turn is given to the next player.
- If any of the players find themselves on the 7th position, they must roll only 1 or their turn is given to the next player.
- The goal is to reach the 8 position and get home.




Game Start
Generating Board
Roll the dice
Check whose turn is it (Velma or Shaggy)
True
False
Shaggy's turn
Dice > 1 in position 7
True
Skip a turn
Dice == 3 in position 6
True
Move to new position
False
Same position as Velma
True
Goes back to initial position
Velma's turn
Dice > 1 in position 7
True
Skip a turn
Dice == 3 in position 6
True
Same position as Shaggy
False
Move to new position
True
Goes back to initial position
Exit (displays winner)

FLOWCHART

PSEUDOCODE

- while Velma or Shaggy's position isn't 8
- generate board
- roll the dice randomly

```
41    while not (S == (2, 2) or O == (2, 2)):
42        gen_board()
43        dice_val = random.randrange(1, 4)  # rolling the dice
```

CODE SNIPPET

PSEUDOCODE

- if next_turn is false
- if get_pos of Velma is 6 and dive_val is 3
- print skip turn
- elif get_pos of Velma is 7 and dice_val is greater than 1
- print skip turn
- else
- calculate next position
- print Velma's movement
- Velma's position is equal to the next position
- if Velma and Shaggy's positions are the same and the game hasn't just started
- print Velma kills Shaggy
- Shaggy's position is equal to starting position

```
if not next_turn:  # checking if it's Velma's turn
    if get_pos(S) == 6 and dice_val == 3:  # if Velma in position 6 and dice_val is 3 - skip the turn
        print("Velma skips the turn.")
    elif get_pos(S) == 7 and dice_val > 1:  # if Velma in position 7 and dice_val > 1 - skip the turn
        print("Velma skips the turn.")
```

CODE SNIPPET

PSEUDOCODE

- else
- if get_pos of Shaggy is 6 and dive_val is 3
- print skip turn
- elif get_pos of Shaggy is 7 and dice_val is greater than 1
- print skip turn
- else
- calculate next position
- print Shaggy's movement
- Shaggy's position is equal to the next position
- if Shaggy and Velma's positions are the same and the game hasn't just started
- print Shaggy kills Velma
- Velma's position is equal to starting position

```
    else:
        next_position = get_board_pos((get_pos(S) + dice_val) % len(board))  # calculate new position
        print("Velma moves from {}".format(get_pos(S)), "to {}".format(get_pos(next_position)))
        S = next_position  # assign new position to Velma
        if S == O != board[0]:  # if Velma's position is the same as Shaggy's - press F for Shaggy
            print("JEENKIES!1!Velma killed Shaggy. Brute!")
            O = board[0]  # Shaggy is returning to safe place
```

CODE SNIPPET

PSEUDOCODE

- else
- if get_pos of Shaggy is 6 and dive_val is 3
- print skip turn
- elif get_pos of Shaggy is 7 and dice_val is greater than 1
- print skip turn
- else
- calculate next position
- print Shaggy's movement
- Shaggy's position is equal to the next position
- if Shaggy and Velma's positions are the same and the game hasn't just started
- print Shaggy kills Velma
- Velma's position is equal to starting position

```
else:  # executes if it's Shaggy's turn
    if get_pos(O) == 6 and dice_val == 3:  # if Shaggy in position 6 and dice_val is 3 - skip the turn
        print("Shaggy skips the turn.")
    elif get_pos(O) == 7 and dice_val > 1:  # if Shaggy in position 7 and dice_val > 1 - skip the turn
        print("Shaggy skips the turn.")
    else:
        next_position = get_board_pos((get_pos(O) + dice_val) % len(board))  # calculate new position
        print("Shaggy moves from {}".format(get_pos(O)), "to {}".format(get_pos(next_position)))
        O = next_position  # assign new position to Shaggy
        if S == O != board[0]:  # if Shaggy's position is the same as Velma's - press F for Shaggy
            print("JEENKIES!1!Shaggy killed Velma. Brute!")
            S = board[0]  # Velma is returning to safe place
```

CODE SNIPPET

PSEUDOCODE

- next_turn is negated
- if Velma's position is equal to 8
- print Velma wins
- else
- print Shaggy wins
- print game over

```
    next_turn = not next_turn  # change the turn


if S == (2, 2):  # if Velma is in (2,2) position
    print("Velma wins!")
else:  # Shaggy is in (2,2) position
    print("Shaggy wins!")
print("GAME OVER. Here's your Scooby Snack")
```

CODE SNIPPET

MORE CODE SNIPPETS

- get_pos function handles the position of each player on the board.

```
# returns current position of player
def get_pos(player):
    for board_i in range(1, len(board)):
        if player[0] == board[board_i][0] and \
                player[1] == board[board_i][1]:
            return board_i
    return 0
```

MORE CODE SNIPPETS

- get_board_pos function handles the matrix position that corresponds to the indexes of the board list.

```
# returns matrix position which corresponds to index of list
def get_board_pos(pos):
    return board[pos]
```

MORE CODE SNIPPETS

- gen_board function is responsible for generating the game board and the player movements on it. It will print the board in the shape of a 3x3 matrix and on each position will display the corresponding numbers and the players.

```
# illustrates the board and position of players on it
def gen_board():
    for i in range(1, len(board) // 3 + 1):
        for j in range(1, len(board) // 3 + 1):
            tuple = (i, j)
            if get_pos(S) == get_pos(O) == board.index(tuple) == 0:
                print("V, S", end='\t')
            elif get_pos(S) == get_pos(tuple):
                print("V", end='\t')
            elif get_pos(O) == get_pos(tuple):
                print("S", end='\t')
            else:
                print(board.index(tuple), end='\t')
        print()
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Python - main.py
Python main.py
main.py
Run: main
D:\Anaconda\python.exe D:/Work/Python/main.py
THE GAME BEGINS!1!1
7 6 5
V, S 8 4
1 2 3
Velma moves from 0 to 2
7 6 5
S 8 4
1 V 3
Shaggy moves from 0 to 1
7 6 5
0 8 4
S V 3
Velma moves from 2 to 5
7 6 V
0 8 4
S 2 3
Shaggy moves from 1 to 2
7 6 V
0 8 4
1 S 3
Velma moves from 5 to 7
V 6 5
0 8 4
1 S 3
Shaggy moves from 2 to 5
V 6 S
0 8 4
1 2 3
Velma skips the turn.
V 6 S
0 8 4
1 2 3
Shaggy moves from 5 to 7
OUTPUT
Version Control
Run
Debug
TODO
Problems
Terminal
Python Packages
Python Console
Endpoints
Breakpoint reached (today 15:58)
72:49 CRLF UTF-8 4 spaces Python 3.8


File Edit View Navigate Code Refactor Run Tools VCS Window Help
Python - main.py
Python main.py
main.py
Run: main
V 6 5
0 8 4
1 S 3
Shaggy moves from 2 to 5
V 6 S
0 8 4
1 2 3
Velma skips the turn.
V 6 S
0 8 4
1 2 3
Shaggy moves from 5 to 7
JEENKIES!1!Shaggy killed Velma. Brute!
S 6 5
V 8 4
1 2 3
Velma moves from 0 to 3
S 6 5
0 8 4
1 2 V
Shaggy skips the turn.
S 6 5
0 8 4
1 2 V
Velma moves from 3 to 4
S 6 5
0 8 V
1 2 3
Shaggy moves from 7 to 8
Shaggy wins!
GAME OVER. Here's your Scooby Snack
Process finished with exit code 0
OUTPUT
Version Control
Run
Debug
TODO
Problems
Terminal
Python Packages
Python Console
Endpoints
Breakpoint reached (today 15:58)
72:49 CRLF UTF-8 4 spaces Python 3.8

THANK YOU