Game Project

By Plamen & Josh

Introduction

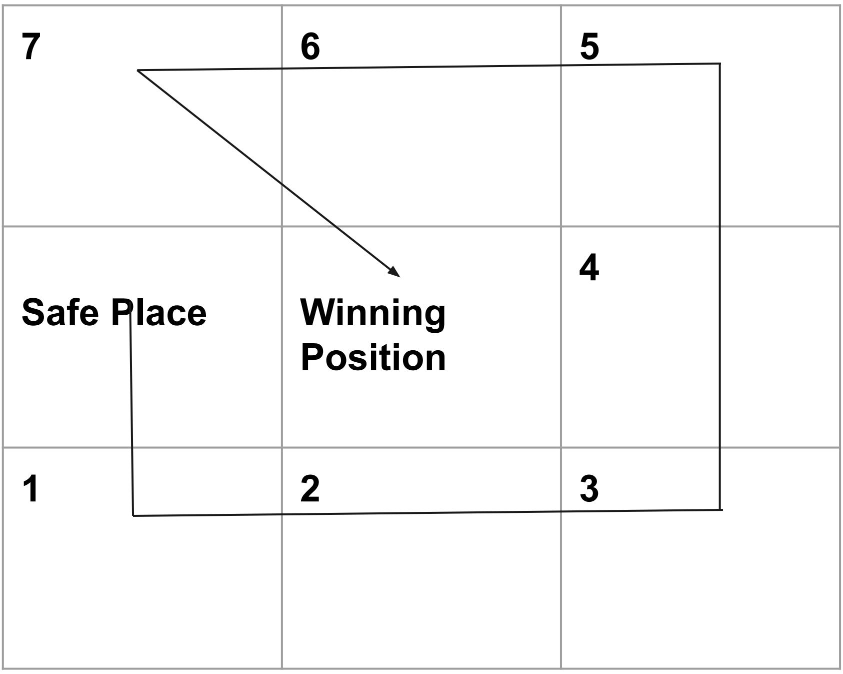
Objective and Rules:

- The objective of the game is to start from safe place and progress around the board until you reach the winning position spot.

- Players roll dice to decide moves, the dice consists of values 1-3.

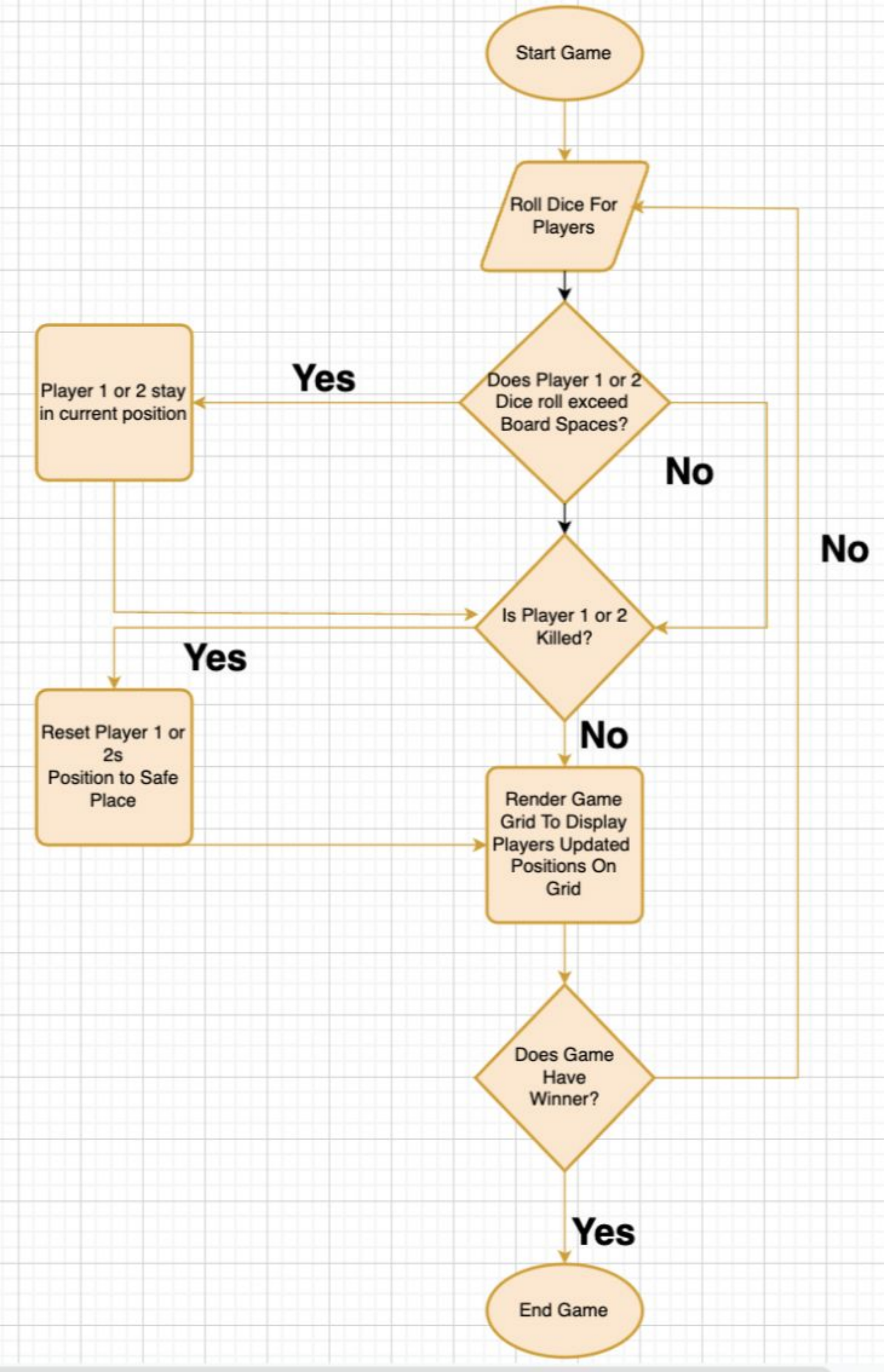
- When a player is in reach of the winning position, they require a perfect roll to move to winning position, any value greater than or less than will mean they maintain position.

- When a player lands on a square that is the same value as another player, the player that was already on the position is killed and reset back to the safe place.



7	6	5
Safe Place	Winning Position	4
1	2	3

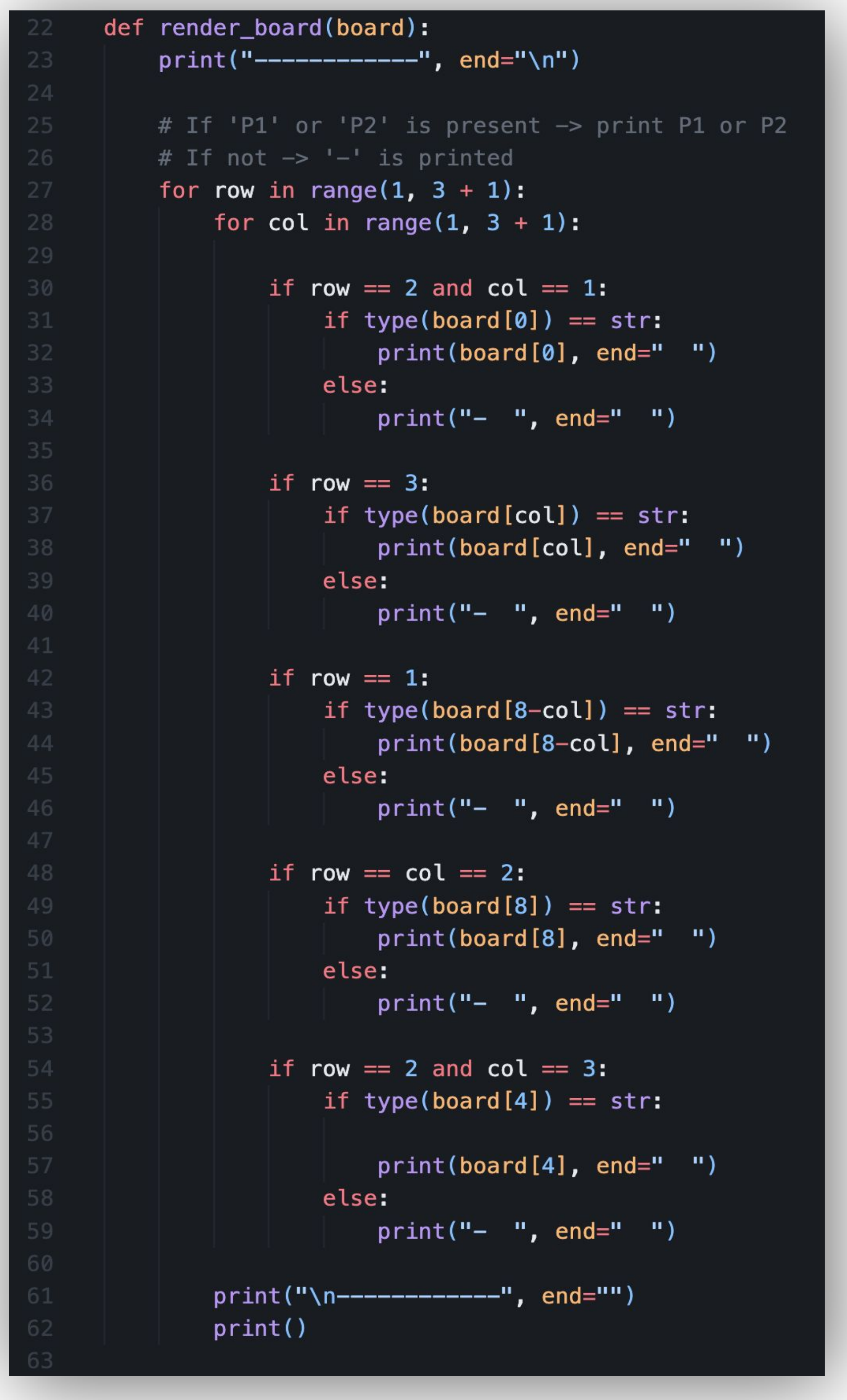
FlowChart


Start Game
Roll Dice For Players
Does Player 1 or 2 Dice roll exceed Board Spaces?
Yes
Player 1 or 2 stay in current position
No
No
Is Player 1 or 2 Killed?
Yes
Reset Player 1 or 2s Position to Safe Place
No
Render Game Grid To Display Players Updated Positions On Grid
Does Game Have Winner?
Yes
End Game

PseudoCode

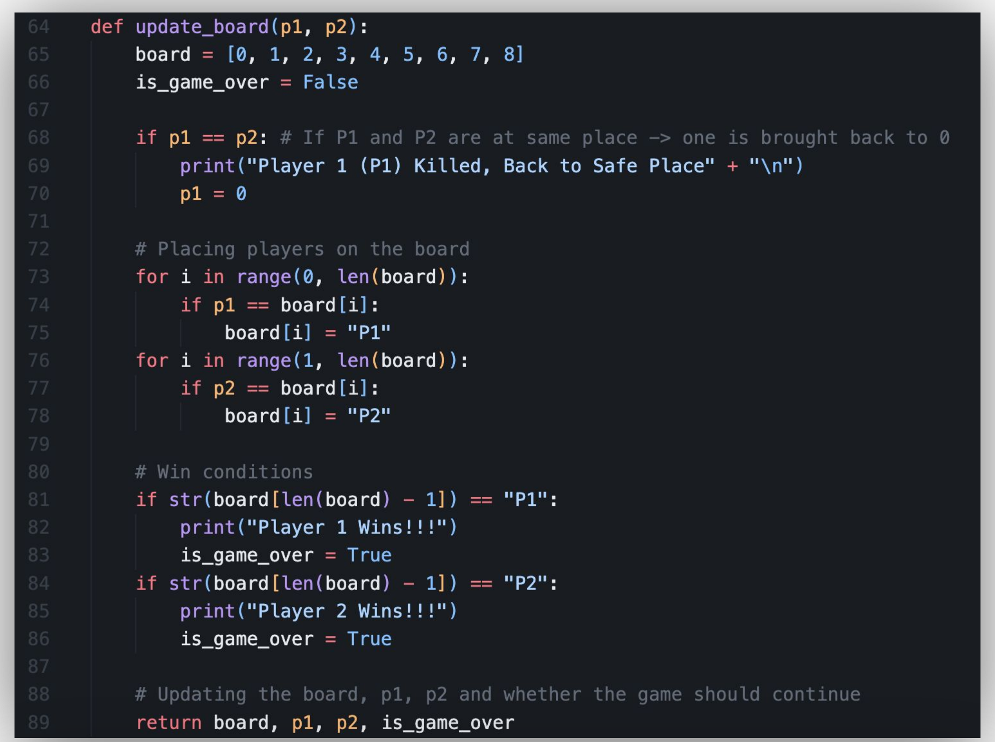
```
Program Start

Initialise variable player_1_pos, player_2_pos to zero
Initialise game to true

While game is true

    player_1_pos, player_2_pos equal to dice_roll()

    call render_board(player_1_pos, player_2_pos)
        print board

    call update_board(player_1_pos, player_2_pos)

        initialise game_over to false

        if player killed
            player_x_pos is equal to zero

        update player position

        if player wins
            print result output
            game_over equal to true

        return player_1_pos, player_2_pos, game_over

    assign player_1_pos, player_2_pos, game from returned values of update_board

    if game
        continue loop
    else
        end loop

Program ends
```

Code Snippets

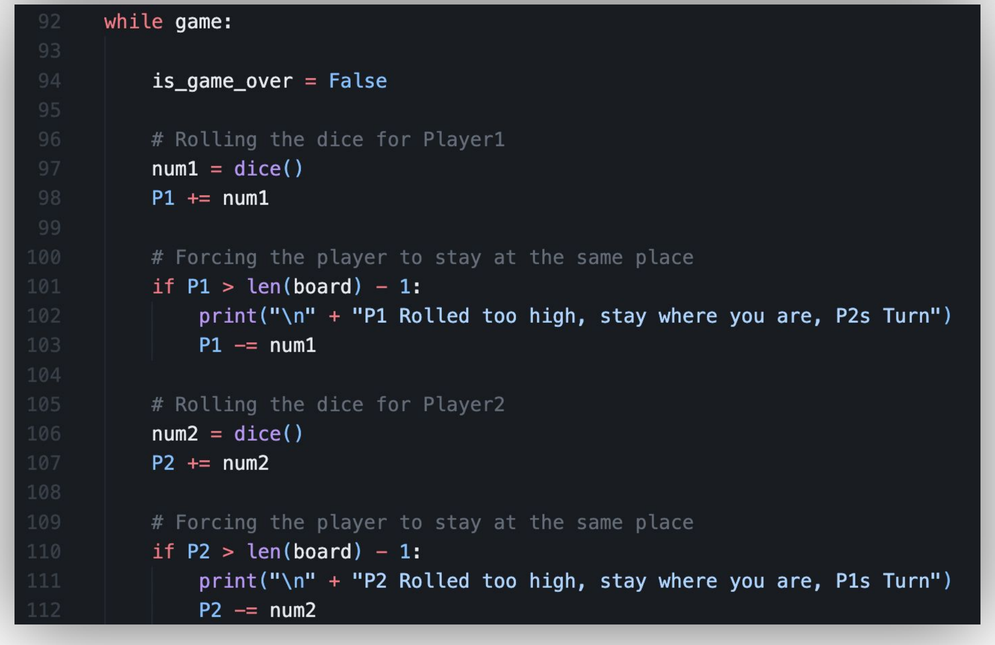


```python
def render_board(board):
    print("------------", end="\n")

    # If 'P1' or 'P2' is present -> print P1 or P2
    # If not -> '-' is printed
    for row in range(1, 3 + 1):
        for col in range(1, 3 + 1):

            if row == 2 and col == 1:
                if type(board[0]) == str:
                    print(board[0], end="  ")
                else:
                    print("-  ", end="  ")

            if row == 3:
                if type(board[col]) == str:
                    print(board[col], end="  ")
                else:
                    print("-  ", end="  ")

            if row == 1:
                if type(board[8-col]) == str:
                    print(board[8-col], end="  ")
                else:
                    print("-  ", end="  ")

            if row == col == 2:
                if type(board[8]) == str:
                    print(board[8], end="  ")
                else:
                    print("-  ", end="  ")

            if row == 2 and col == 3:
                if type(board[4]) == str:

                    print(board[4], end="  ")
                else:
                    print("-  ", end="  ")

        print("\n------------", end="")
        print()
```

Code Snippets

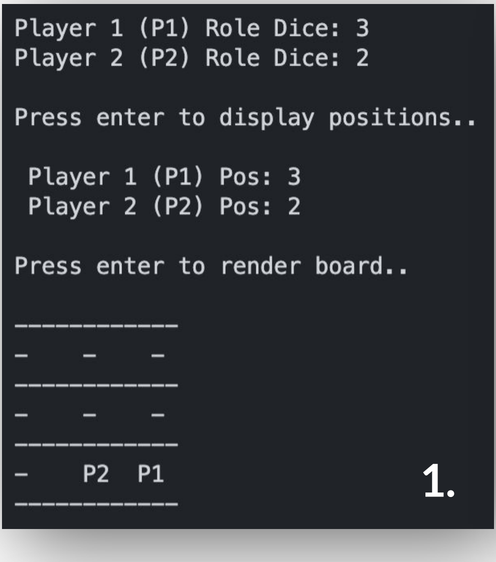


```python
def update_board(p1, p2):
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    is_game_over = False

    if p1 == p2: # If P1 and P2 are at same place -> one is brought back to 0
        print("Player 1 (P1) Killed, Back to Safe Place" + "\n")
        p1 = 0

    # Placing players on the board
    for i in range(0, len(board)):
        if p1 == board[i]:
            board[i] = "P1"
    for i in range(1, len(board)):
        if p2 == board[i]:
            board[i] = "P2"

    # Win conditions
    if str(board[len(board) - 1]) == "P1":
        print("Player 1 Wins!!!")
        is_game_over = True
    if str(board[len(board) - 1]) == "P2":
        print("Player 2 Wins!!!")
        is_game_over = True

    # Updating the board, p1, p2 and whether the game should continue
    return board, p1, p2, is_game_over
```



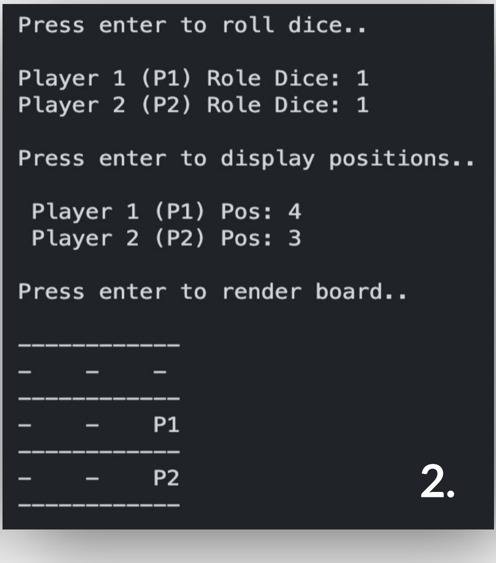
```python
while game:

    is_game_over = False

    # Rolling the dice for Player1
    num1 = dice()
    P1 += num1

    # Forcing the player to stay at the same place
    if P1 > len(board) - 1:
        print("\n" + "P1 Rolled too high, stay where you are, P2s Turn")
        P1 -= num1

    # Rolling the dice for Player2
    num2 = dice()
    P2 += num2

    # Forcing the player to stay at the same place
    if P2 > len(board) - 1:
        print("\n" + "P2 Rolled too high, stay where you are, P1s Turn")
        P2 -= num2
```

Output



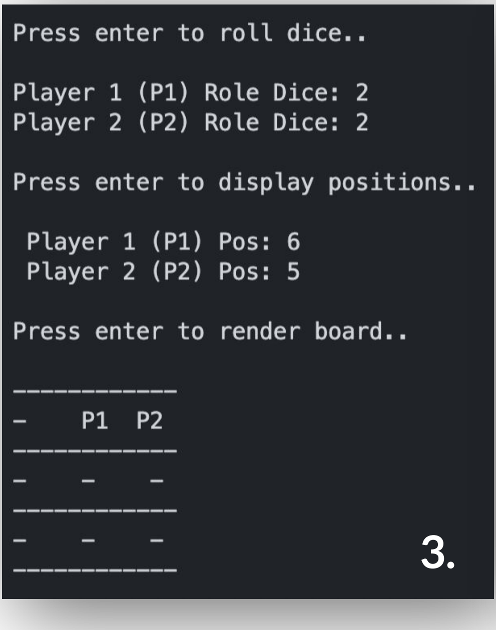
```
Player 1 (P1) Role Dice: 3
Player 2 (P2) Role Dice: 2

Press enter to display positions..

 Player 1 (P1) Pos: 3
 Player 2 (P2) Pos: 2

Press enter to render board..

------------
-    -    -
------------
-    -    -
------------
-    P2  P1
------------
```

1.



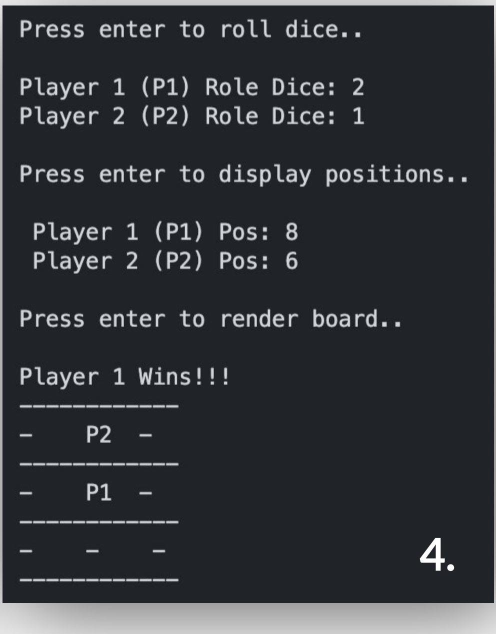
```
Press enter to roll dice..

Player 1 (P1) Role Dice: 1
Player 2 (P2) Role Dice: 1

Press enter to display positions..

 Player 1 (P1) Pos: 4
 Player 2 (P2) Pos: 3

Press enter to render board..

------------
-    -    -
------------
-    -    P1
------------
-    -    P2
------------
```

2.



```
Press enter to roll dice..

Player 1 (P1) Role Dice: 2
Player 2 (P2) Role Dice: 2

Press enter to display positions..

 Player 1 (P1) Pos: 6
 Player 2 (P2) Pos: 5

Press enter to render board..

------------
-    P1  P2
------------
-    -    -
------------
-    -    -
------------
```

3.



```
Press enter to roll dice..

Player 1 (P1) Role Dice: 2
Player 2 (P2) Role Dice: 1

Press enter to display positions..

 Player 1 (P1) Pos: 8
 Player 2 (P2) Pos: 6

Press enter to render board..

Player 1 Wins!!!
------------
-    P2  -
------------
-    P1  -
------------
-    -    -
------------
```

4.