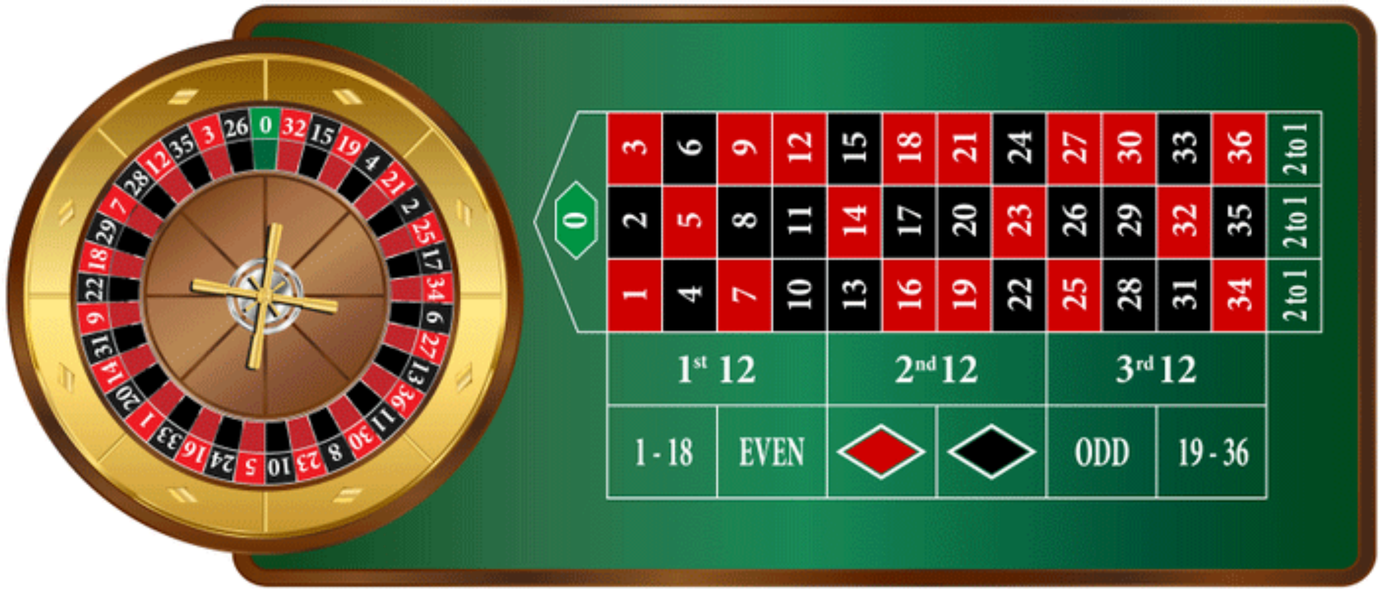
Roulette

T1A3 - Python Terminal Application

Johnson Wang 18/12/2022

Roulette

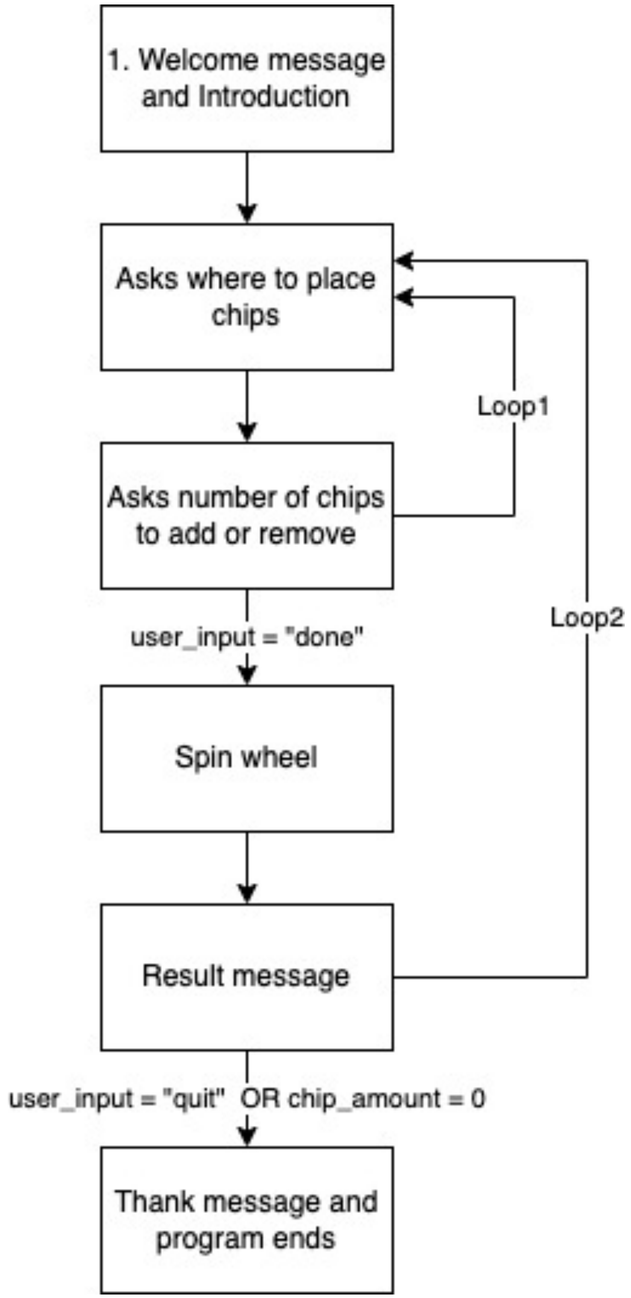
- Casino game
- Purpose of application: enjoy game without monetary input



Game Interface

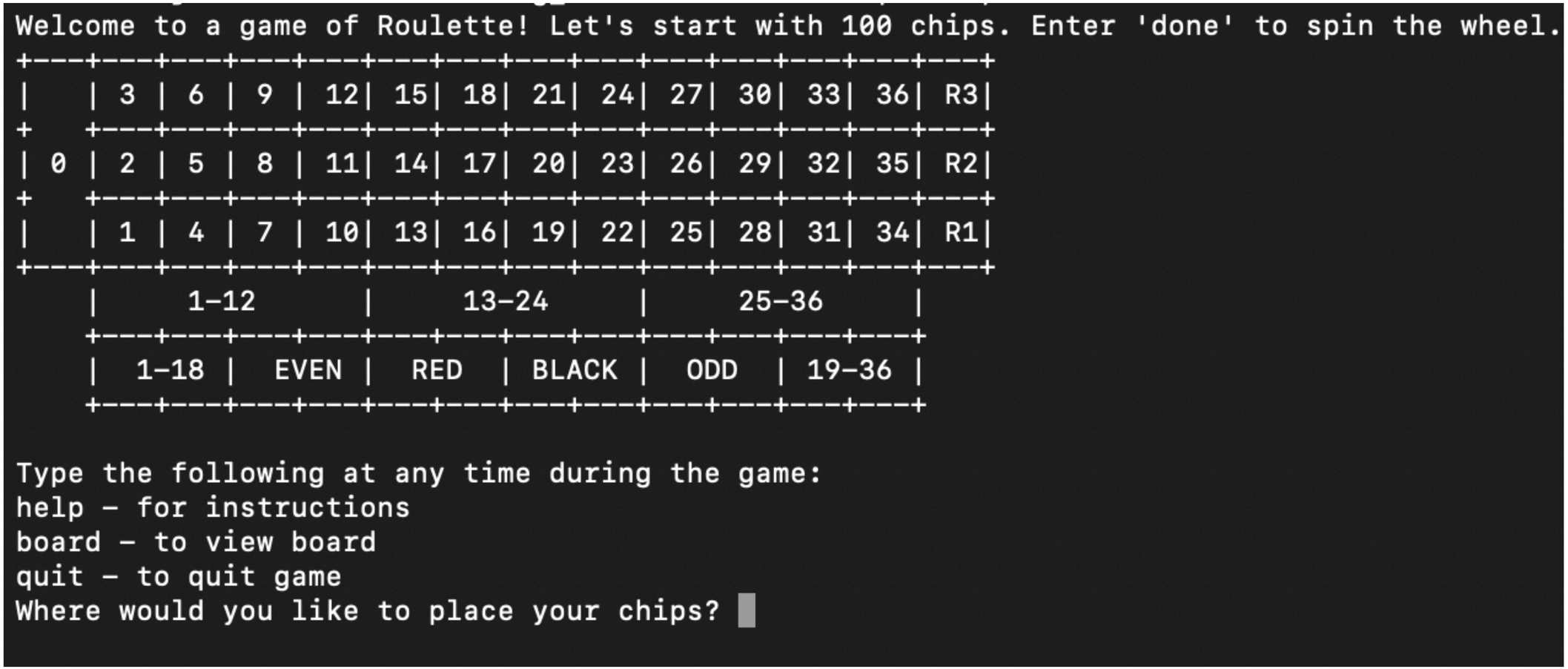
Structure

- Loop1: betting phase, program exits loop only when user inputs “done”.
- Loop2: game play, includes betting phase and wheel spin phase, program exits loop when player inputs “quit” or they run out of chips.



Game Interface

Demo - Welcome message



Intro message

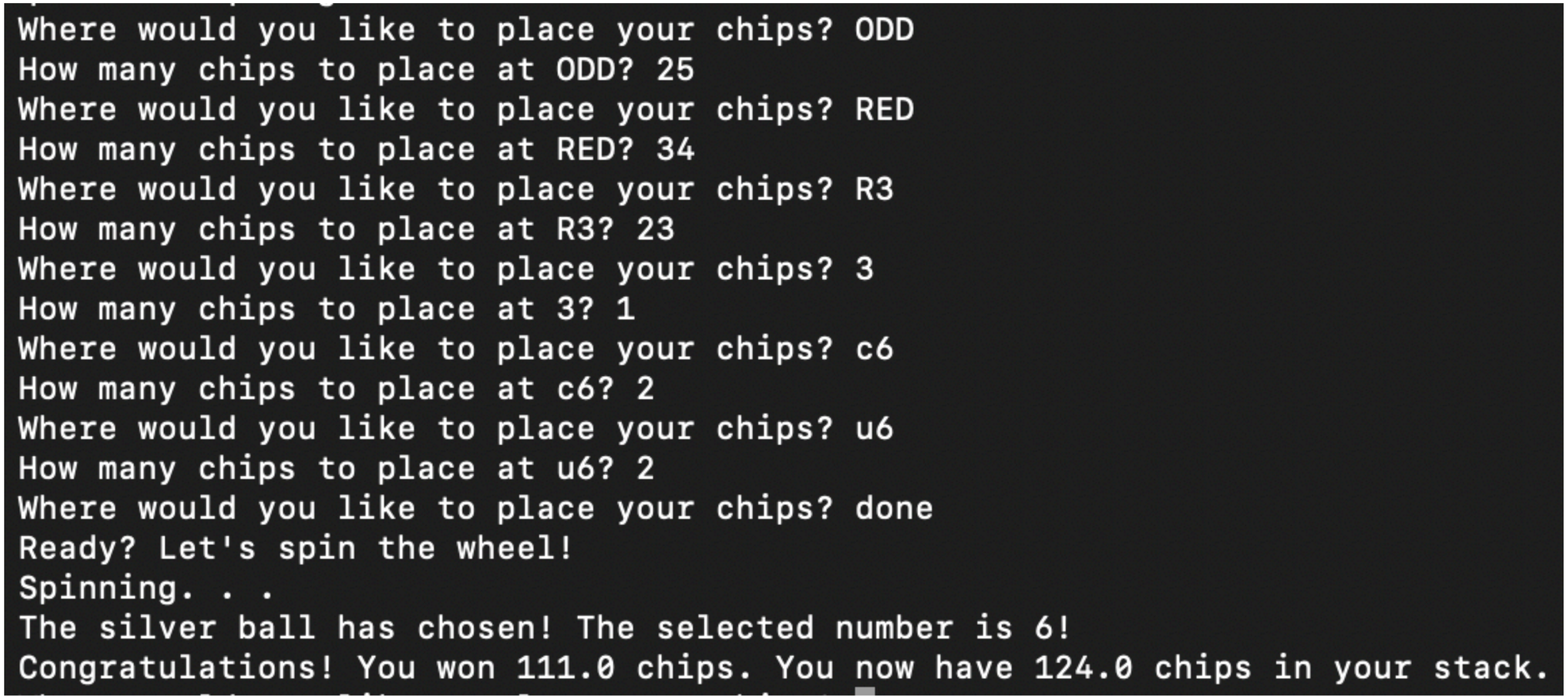
Roulette board

Quick instructions

Game play

Game Interface

Demo - Welcome message



Betting phase

Roulette wheel spin

Result message

Game Support

User commands during game play

```
Where would you like to place your chips? help
    u
  +---+
l | 1 | r
  +---+
    d
Use the above diagram to add chips to edges of a number (except 0). For example u1 to add chips on top edge of number 1.
To place chip at the corner between 3-4 numbers, use cN, where N is the largest of the numbers.
For the remainder, add chips as appeared on the board.
To remove chips, use a negative number (-x) when prompted.

Where would you like to place your chips? board
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   | 3 | 6 | 9 | 12| 15| 18| 21| 24| 27| 30| 33| 36| R3|
+   +---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 2 | 5 | 8 | 11| 14| 17| 20| 23| 26| 29| 32| 35| R2|
+   +---+---+---+---+---+---+---+---+---+---+---+---+---+
|   | 1 | 4 | 7 | 10| 13| 16| 19| 22| 25| 28| 31| 34| R1|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |     1-12      |     13-24     |     25-36     |
    +---+---+---+---+---+---+---+---+---+---+---+---+
    |  1-18 |  EVEN |  RED  | BLACK |  ODD  | 19-36 |
    +---+---+---+---+---+---+---+---+---+---+---+---+

Where would you like to place your chips?
```

“help” - shows instructions

“board” - shows roulette board

```
Where would you like to place your chips? quit
Thanks for playing! You finished with 124.0 chips in your stack and you lasted 1 spins.
```

“quit” - thank you message and program ends

Game Internals

Error handling - is_location

- Handles user input error when player enters location to place their chips
- When placed in main_loop the program responds with error message

```
#Check if user input for chip placement is valid
def is_location(user_input):
    location_number = 0

    if user_input in valid_locations:
        return True

    # For user input beginning with u, d, l, r, c
    elif user_input[0] in location_syntax:
        try:
            location_number = int(user_input[1:])
        except ValueError:
            return False
```

```
location = user_input

if is_location(location):
    break
elif user_input in user_commands:
    continue
else:
    print("Error: Please enter a valid location on the board")
    continue
```

Game Internals

Game state control

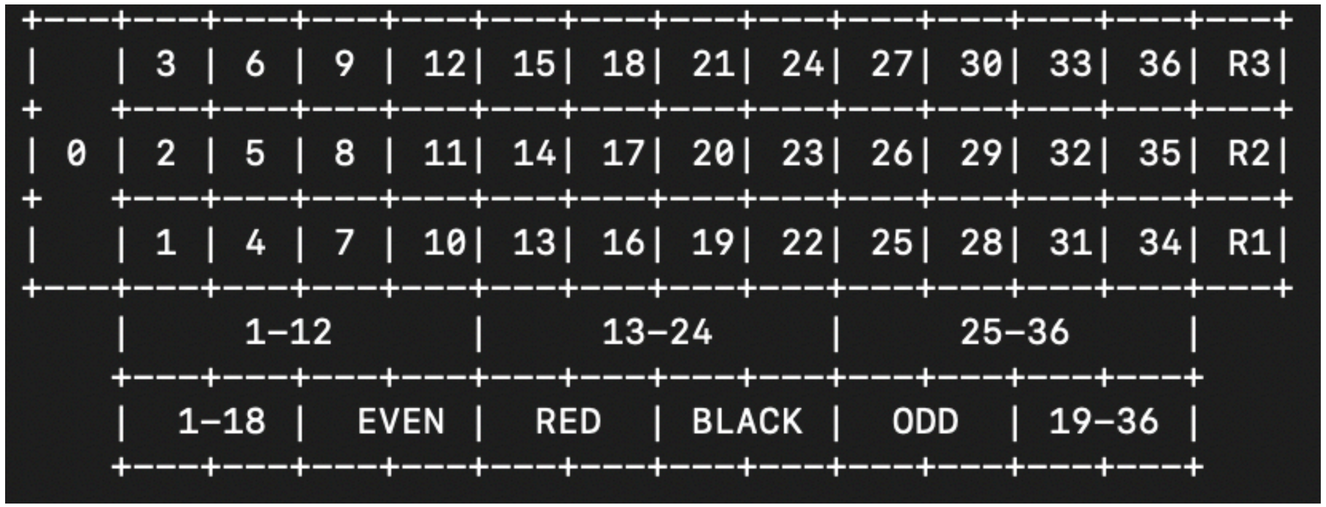
- chip_placement (dict): key=location, value=num_chips
- calculate_winnings function: returns list of payout rates for each number
- winnings[selected_number] is the chip payout to the player for the round

```python
def calculate_winnings(chip_placement):
    winnings = [0] * 37

    for location, num_chips in chip_placement.items():
        betted_numbers = location_to_number(location)
        for number in betted_numbers:
            winnings[number] += 36 * num_chips / len(betted_numbers)

    return winnings
```

Challenge

Coding

Choosing to write more concisely
vs. with program logic




```python
def draw_board():
    row_top1 = "+---" * 14 + "+"
    row_top2 = "+   " + "+---" * 13 + "+"
    row_top3 = "    " + "+---" * 12 + "+"

    for row in range(3, -1, -1):

        if row in [1, 2]:
            print(row_top2)
        else:
            print(row_top1)
            if row == 0:
                break

        if row == 2:
            print("| 0", end=" ")
        else:
            print("|  ", end=" ")

        for col in range(1, 13):
            boardnum = row + 3 * (col - 1)

            if boardnum / 10 < 1:
                print(f"| {boardnum}", end=" ")
            else:
                print(f"| {boardnum}", end="")

        print(f"| R{row}|")

    print("    |     1-12      |     13-24     |     25-36     |")
    print(row_top3)
    print("    |  1-18 |  EVEN |  RED  | BLACK |  ODD  | 19-36 |")
    print(row_top3, end="\n\n")
```

Thank you