```
# Name: Trien Bang Huynh
# Assignment 1: UI class

from states import States

class UI:
    '''
    A UI class which interacts with the user based on their input choices.
    '''
    def __init__(self):
        while True:
            try:
                filename = input("Enter the input file name: ")
                self.states = States(filename)
                break
            except FileNotFoundError:
                print("File not found. Please try again.")


    '''
    A method that prints the states and their population for a particular year.
    '''
    def print_states_by_year(self):
        max_states = self.states.max_num_states
        while True:
            try:
                year = int(input("Enter year: "))
                year_index = (year - 1990) // 10 + 1
                assert year_index in [1,2,3,4]
                num_states = int(input(f"Enter number of states: "))
                num_states = min(num_states, max_states)
                states_gen = self.states.byPop(year)
                break
            except ValueError:
                print("Must enter a number")
            except AssertionError:
                print(f"{year} is not a valid census year")
                return


        for i in range(num_states):
            state, pop = next(states_gen)
            print(f"{state:<25}{pop:15,d}")

    '''
    A method that prints the states and their positive or negative growth.
    '''
    def print_states_by_growth_rate(self):
        while True:
            choice = input("p. positive \nn. negative\nYour choice: ").lower()
            if choice == 'p':
                is_positive = True
                break
            elif choice == 'n':
                is_positive = False
                break
            else:
                print("Invalid input. Please try again.")
        states_gen = self.states.growth(is_positive)
        for state, rate in states_gen:
            print(f"{state:<25}{rate:>15.2%}")


    '''
    A method that prints whether there is at least 1 state that dropped in
population between 2 given years.
    '''
    def print_population_drop(self):
        while True:
            try:
                year = int(input("Enter start year (1990, 2000, 2010, or 2020): "))
                assert (year - 1990) % 10 == 0
                year_index = (year - 1990) // 10 + 1
                assert year_index in [1,2,3,4]
                start_year = year

                year = int(input("Enter end year (1990, 2000, 2010, or 2020): "))
                assert (year - 1990) % 10 == 0
                year_index = (year - 1990) // 10 + 1
                assert year_index in [1,2,3,4]
                end_year = year

                if self.states.drop(start_year, end_year):
                    print(f"Population drop in at least one state between
{start_year} and {end_year}")
                else:
                    print(f"No population drop across all states between
{start_year} and {end_year}")
                break
            except ValueError:
                print("Must enter a number. Start over again!")
            except AssertionError:
                print(f"Invalid year {year}")
                return


    '''
    A method that prints a menu and keeps prompting the user until there’s a valid
choice.
    '''
    def print_menu(self):
        print("\nMenu:\n1. View most populous states\n2. View growth in 2021\n3.
Check population drop\n4. Quit\n")


    '''
    A method that loops to print the menu and process the user choice until the
user chooses to quit.
    '''
    def run(self):
        while True:
            try:
                self.print_menu()
                choice = int(input("Enter your choice: "))
                func_list = [0, self.print_states_by_year,
self.print_states_by_growth_rate, self.print_population_drop]
                if choice == 4:
                    break
                func_list[choice]()
            except IndexError:
                print("Invalid choice")
            except ValueError:
                print("Choose 1-4")


UI().run()
```