**Purpose** Compute a sustainable, year-by-year retirement spending plan that avoids running out of money, minimizes taxes and penalties through timing of withdrawals and Roth conversions, forecasts annual tax bills, and preserves ACA subsidies before Medicare.

**Basic functions**

1. **Inputs & Derived Basics**

* Load all user inputs.
* Derive years, ages, living status, and life phase per year.

1. **Spending & Income Engine**

* Set inflation-indexed base spend per phase.
* Add Social Security at chosen start ages with COLA.
* Fill remaining need via ordered withdrawals to meet spend.
* Plan Roth conversions within constraints to reduce lifetime tax and penalties.
* Compute annual taxes owed from all sources.

1. **Account Balance Engine**

* Apply withdrawals, conversions, RMDs, and growth per account.
* Track cash for taxes and spending.
* Update end-of-year balances for Brokerage, Roth, Traditional.

1. **Policy Coordination**

* Manage MAGI to target ACA subsidies until Medicare eligibility.
* Enforce tax rules to avoid penalties (e.g., early withdrawals, underpayment).

1. **Annual Projections Output**

* Produce a year-by-year table with: ages, phase, spend, taxes, SS, each account draw, conversions, RMDs, MAGI, ACA subsidy, and end balances.
* Enable charts and simple analytics from this table.

Below is the **minimal, unambiguous input set** with tags, defaults, and UI constraints.

### **Personal & Starting Balances**

* **Birth Year (You)** — integer [Required] • range 1900–2100 • default: last-used
* **Birth Year (Spouse)** — integer [Optional-defaulted] • range 1900–2100 • default: last-used
* **Final Age (You)** — integer [Required] • range 60–100 • default: last-used
* **Final Age (Spouse)** — integer [Optional-defaulted] • range 60–105 • default: last-used
* **Filing Status** — choice [Required] • options: MFJ, Single • default: **MFJ** (sticky)
* **Brokerage Balance** — dollars [Required] • ≥0 • default: last-used
* **Roth Balance** — dollars [Required] • ≥0 • default: last-used
* **IRA Balance** — dollars [Required] • ≥0 • default: last-used
* **Projection Start Year** — integer [Required] • range: current year…current+5 • default: derived from today, editable year only

### **Social Security**

* **SS Start Age (You)** — integer [Required] • 62–70 • default: last-used
* **SS Start Age (Spouse)** — integer [Optional-defaulted] • 62–70 • default: last-used
* **SS Annual Amount (You, at start age, today’s $)** — dollars [Required] • ≥0 • default: last-used
* **SS Annual Amount (Spouse, at start age, today’s $)** — dollars [Optional-defaulted] • ≥0 • default: last-used

### **Spending Goals (today’s $)**

* **GoGo Annual Spend** — dollars [Required] • ≥0 • default: last-used
* **Slow Annual Spend** — dollars [Required] • ≥0 • default: last-used
* **NoGo Annual Spend** — dollars [Required] • ≥0 • default: last-used
* **GoGo Years** — integer [Required] • 0–30 • default: last-used
* **Slow Years** — integer [Required] • 0–30 • default: last-used
* **Survivor Spend Percent** — percent [Required] • 50–100% • default: last-used

### **Growth & Inflation (preset but adjustable)**

* **Inflation Rate** — percent [Optional-defaulted] • 0–10% • default: preset
* **Brokerage Growth Rate** — percent [Optional-defaulted] • −10–15% • default: preset
* **Roth Growth Rate** — percent [Optional-defaulted] • −10–15% • default: preset
* **IRA Growth Rate** — percent [Optional-defaulted] • −10–15% • default: preset

### **Tax Rules (preset but adjustable)**

* **MAGI Target (base year, today’s $)** — dollars [Optional-defaulted] • ≥0 • default: preset
* **Standard Deduction (base year)** — dollars [Optional-defaulted] • ≥0 • default: preset
* **RMD Start Age** — integer [Optional-defaulted] • choices: 73, 75 (rule-set) • default: preset per rule year

### **Healthcare (preset but adjustable)**

* **ACA Ends Age** — integer [Optional-defaulted] • 60–70 • default: 65
* **ACA Subsidy (annual, if fixed)** — dollars [Optional-defaulted] • ≥0 • default: blank (tool calculates unless user overrides)

**UI rules**

* All fields are single-source truths. No inputs that are outputs elsewhere.
* Defaults are **sticky**: last-used > preset.
* Widgets enforce type and range: integer spinners, percent sliders, dropdowns for statuses/ages with bounds.
* Today’s date sets the initial **Projection Start Year** suggestion; month is derived, not an input.

## **Purpose**

Provide a year-by-year retirement projection that:

1. Calculates safe annual spending without running out of money.
2. Minimizes taxes and penalties with optimized Roth conversions and withdrawal sequencing.
3. Forecasts annual tax liabilities so they can be anticipated.
4. Preserves ACA subsidies before Medicare eligibility.

## **Core Design Principles**

* **Single source of truth**: inputs are only real-world facts (ages, balances, spend targets). No inputs that are calculated elsewhere.
* **Strict validation**: integer ranges, percent ranges, dropdowns for enums, sticky defaults.
* **Modularity**: each financial logic block isolated in its own module.
* **Deterministic**: given inputs and policy, projections are reproducible.
* **Single window GUI**: PySimpleGUI with inputs at the top, projections at the bottom, selectable outputs.

## **Inputs**

All in today’s dollars. Defaults sticky from last run or preset policy.

**Required**

* Birth year (You, Spouse)
* Final age (You, Spouse)
* Filing status (default: MFJ)
* Starting balances: Brokerage, Roth, IRA
* Projection start year (from today)
* Social Security: start age + annual amount at that age (You, Spouse)
* Spending goals: GoGo / Slow / NoGo spend, years in each, survivor percent

**Optional (defaulted)**

* Inflation rate
* Growth rates (Brokerage, Roth, IRA)
* MAGI target (base year)
* Standard deduction (base year)
* RMD start age (policy default 73)
* ACA subsidy (blank = calculated, override allowed)
* ACA end age (default 65)

**Events (flexible)**

* Year, label, amount, tax treatment, repeat.  
  + Handles weddings, cars, capital gains, windfalls, big medical, etc.

## **Outputs (Projections Table)**

Canonical columns:  
 Year, Age\_You, Age\_Spouse, Phase, Spend\_Target, Taxes, Total\_Spend, SS\_Income, Draw\_Brokerage, Draw\_Roth, Draw\_IRA, Roth\_Conversion, RMD, MAGI, Std\_Deduction, End\_Bal\_Brokerage, End\_Bal\_Roth, End\_Bal\_IRA, Total\_Assets.

Selectable views:

* Summary
* Taxes
* Draws
* Balances
* ACA/MAGI
* Events impact

Charts: assets, taxes, draw mix, MAGI vs target.

## **File Layout**

retireplan/

retireplan/

inputs.py # schema, validation, defaults

policy.py # fixed rule tables (RMD, tax, SS COLA, ACA)

engine.py # year loop orchestration

accounts.py # Brokerage/Roth/IRA classes

taxes.py # taxable income, std deduction, est taxes

social\_security.py # start ages, COLA, taxable portion

spending.py # inflation, survivor, phases

projections.py # assemble DataFrame

io\_csv.py # export CSV

io\_xlsx.py # export XLSX

utils.py # helpers

gui.py # PySimpleGUI single window

examples/

sample\_inputs.yaml

sample\_events.yaml

tests/

test\_\*.py

## **Workflow**

1. Load inputs.yaml + optional events.yaml.
2. Validate against schema.
3. Run yearly projection loop.
4. Store results in Pandas DataFrame.
5. Display in PySimpleGUI table with selectable columns/views.
6. Export as needed.

This keeps the tool **lean, modular, and extensible** while solving the core problems: spending safety, tax minimization, ACA subsidy protection, and clarity of outputs.