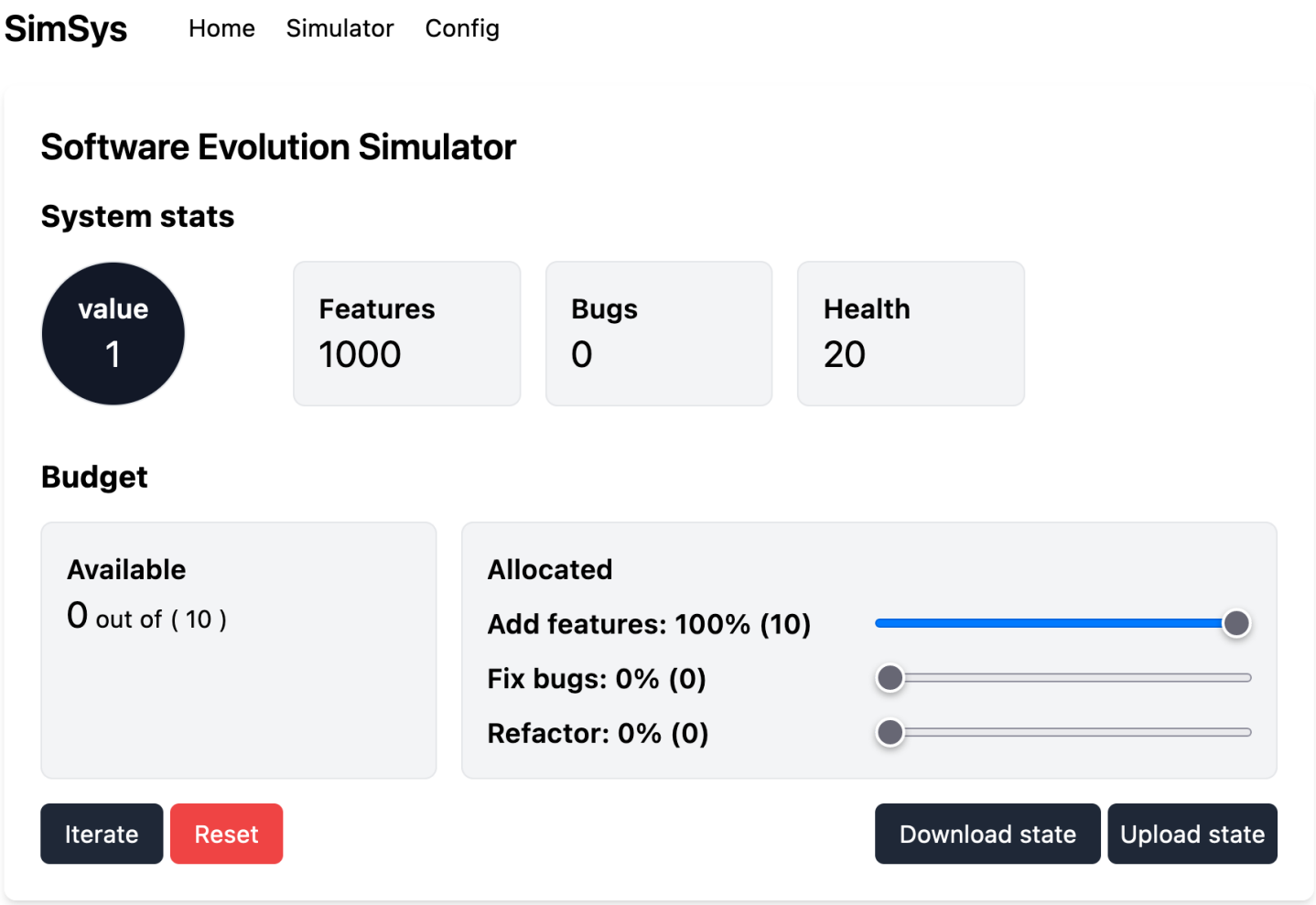
Software Evolution Simulator

`SimSys` is a simulation of the impacts of technical debt accumulation over time. The simulation considers in particular the system's value, health, number of bugs, and number of features.



Scenario

The simulation is based on the scenario of a team that develops a software system. The team has a budget that it can allocate to three tasks: adding features, fixing bugs, and maintaining/improving the system's health. The team's goal is to maximize the system's value over time.

Gameplay

The simulator runs in iterations, where each iteration represents a development period (e.g., a sprint).

Before each iteration, the user must allocate the budget for each of three tasks:

- adding features,
- fixing bugs, and
- maintaining/improving the system's health.

The entire budget must be allocated. The user can allocate the entire budget to a single task, or distribute it across the three tasks. Then, each iteration comprises the following steps:

1. Update the system's statuses (health, number of bugs, and number of features) based on the budget allocations and various factors.
2. Calculate the system's value.
3. Calculate the available budget for the next iteration.

The simulation ends when the system's value reaches zero.

Formulae

The system's statuses are updated based on the following formulae:

- $\text{Health}_{t+1} = \text{Health}_t + \text{Budget}_{\text{refactor}} \times \text{Factor}_{\text{RE}} - \text{Factor}_{\text{L}}$
- $\text{Features}_{t+1} = \text{Features}_t + \text{Budget}_{\text{feature}} \times \text{Health}_t \times \text{Factor}_{\text{FAE}}$
- $\text{Bugs}_{t+1} = \text{Bugs}_t - \text{FixedBugs} + \text{NewBugs}$
  - $\text{FixedBugs} = \text{Budget}_{\text{bugfix}} \times \text{Health}_t \times \text{Factor}_{\text{BFE}}$
  - $\text{NewBugs} = \left(\frac{\text{Budget}_{\text{feature}}}{\text{Health}_t}\right) \times \text{Factor}_{\text{NBP}}$

The system's value is calculated based on new statuses according to the following formula:

- $\text{Value} = \text{FeatureMerit} - \text{BugThreat}$
  - $\text{FeatureMerit} = \text{Features}_{t+1} \times \text{Factor}_{\text{VF}}$
  - $\text{BugThreat} = \text{Bugs}_{t+1} \times \text{Factor}_{\text{VB}}$

The available budget for the next iteration is calculated based on the system's new value according to the following formula:

- $\text{Budget}_{t+1} = \text{Value}_t \times \text{Factor}_{\text{BM}}$

Factors

The formulae described rely on a set of factors that can be adjusted by the user. These factors are:

- **L**: Inspired by Lehman's Law, it represents the inherent degradation of system health over time.
- **RE**: Refactoring effectiveness.
- **FAE**: Feature addition effectiveness.
- **BFE**: Bug-fix effectiveness.
- **NBP**: New bug proneness.
- **VF**: Value generated per feature.
- **VB**: Value lost per bug.
- **BM**: Budget multiplier.

Saving and Restoring

The simulator uses browser cookies to serialize and restore its state, allowing for continuity across sessions. The state is saved automatically after each iteration. When restarting a simulation, the configured initial values will be set. You can also download the current state as a JSON file, which can be used to restore the simulation later.