**The Wrecked Village**

This is a toy model of an interest-based monetary system. It is the story of how a simple loan system can wreck not just a village but an entire economy

Imagine five friends migrating to a remote island and establishing a small village there. Each takes up a profession that suits them, complementing one another. After a short while, a sixth friend, who doesn't possess any particular skill but has a large amount of gold dinars, decides to join them for a while.

At that point, the friends realize that it would be beneficial for each of them to have some money to facilitate the exchange of their goods and services with one another. So, each borrows one hundred dinars from their new friend. However, the friend does not agree to lend them the money without interest. He stipulated a 4% interest rate on each loan, requiring each villager to repay their debt at the end of the year.

At first glance, nothing seems unusual. However, the lender's condition is impossible to fulfill. Why?

Because the total amount of dinars available in the village is only 500 dinars. Yet, each borrower must repay 104 dinars, totaling 520 dinars. This is impossible, as the total amount of dinars in the village is only 500. This is impossible since the village lacks the additional 20 dinars. The lender is demanding money that simply does not exist. What will happen then?

Some borrowers will find themselves unable to repay the principal along with the interest. For instance, if three members manage to repay the full amount of 104 dinars each (totaling 312 dinars), only 500 – 312 = 188 dinars will remain in the village. If the other two villagers generate equal incomes, each would repay half of this amount, which is 94 dinars. However, this leaves each of them short of the 104 dinars they owe. The shortage of principal repayment of the two (total = 12) is equal to the total interest paid by the other three borrowers. But the two are not only short on principal; they are short on the interest as well. Total unpaid debt by the end of the first year is 20 dinars.

After collecting the initial 500 dinars, the offers new loans—again with 4% interest. For those who failed to repay in full, their unpaid debt is added to the new loans. As a result, the village's total debt grows each year. In the first year, total debt was 520 dinars, with 500 repaid and 20 left unpaid. In the second year, the lender loans another 500 dinars, reschedules the 20 dinars and charges interest on the total. The village now owes 540.8 dinars. And so, the debt continues to grow year after year. In year 5, total debt = 608.33 dinars, of which 108.33 will be unpaid.

The problem the village is facing is clear: the lender is demanding dinars from the members that simply do not exist in the economy. Thus, they have no choice but to return to the lender to borrow to repay the money he is demanding from them. Naturally, these new loans will also carry interest. As a result, the debt grows exponentially.

The system of the Wrecked Village is such that no one can repay their debt unless someone else fails to repay theirs. Thus, the village becomes a “zero-sum economy.”

The code simulates the growth of debt in the village. It also simulates the number of winners and losers for each year. Winners are those who were able to repay their debt in full. Losers are those who fail to repay in full.

\*\* \*\*

Step-by-Step Explanation of Randomization Logic (from GPT-4.5-Preview):

**Step 1**: Total Debt Calculation

At the beginning of each year, each villager's outstanding debt (carried from previous years) accumulates interest. Additionally, each villager borrows again (the same initial amount) every year, which also accrues interest immediately. Thus, before repayment starts, each villager has a total debt that includes:

* Previous unpaid debts plus accrued interest.
* This year's newly borrowed amount, also with interest.

**Step 2**: Randomizing Order of Repayment

Here's where the randomization explicitly comes into play:

* All villagers are placed into a repayment queue.
* The order of villagers in this queue is randomly shuffled each year (using Python's random.shuffle). This random shuffle is critical, as it determines the repayment sequence each year.

**Step 3**: Repayment Process (Winners vs. Losers)

After shuffling:

* The first villager in line attempts to repay their total debt fully. If sufficient money is available, this villager successfully repays and becomes a "winner".
* Subsequent villagers then take turns attempting to repay. Each villager continues repaying fully as long as enough money remains available.
* Eventually, the available money (which is fixed) runs out before all villagers have repaid their debt.
* At this point, villagers remaining in the queue can no longer fully repay their debts. These villagers become "losers," carrying unpaid debt forward to the next year.

**Step 4**: Consequences of Randomization

Due to the randomized ordering each year:

* Different villagers experience being winners or losers from year to year. Because the queue order changes randomly, villagers who were able to repay fully in one year might fail to repay fully the next year, and vice versa.
* As debt accumulates over time due to interest, the total debt burden grows. This generally leads to fewer villagers successfully repaying each year, but the exact number of winners and losers fluctuates unpredictably due to random ordering each year.

**In Short:**

The randomization explicitly occurs every year when villagers are randomly reordered in the repayment queue. Those at the front of this random queue have a higher chance of becoming winners (fully repaying), while those at the back have a higher risk of becoming losers (unable to repay fully), since the available money is limited. This randomness produces dynamic, realistic, and varying outcomes each year.

\*\* \*\*

There are two code files for modeling the Wrecked Village:

1. wrecked\_village\_simulation.py. Clear, straightforward Python script.
2. wrecked\_village\_simulation.html. Simple interactive HTML/JavaScript tool with interactive charts. It can be run on any browser.

The code was developed with the assistance of GPT-4.5-Preview.