

Project Description:

For this programming project, you are tasked to make a simple investment based game. At the start of the game, ask the player how much money they would like to start with (from 100,000 to 10,000,000 inclusive), the maintaining amount (from 100,000 to the value of the starting money, inclusive), and the target amount (a value from the starting amount to any positive number, inclusive).

The starting amount will be stored in a bank. The player may freely spend any amount from the bank to perform actions enumerated below. At the start of each day (before periodic events, see the section called Periodic Events for more details), if the money in the bank is less than or equal to the maintaining amount, the player loses. If the money is greater than or equal to the target amount instead, the player wins.

The updated bank balance, maintaining amount, and target amount should be displayed at all times. The updated bank balance should also be displayed every time money is added or subtracted from the bank account (e.g. due to actions or events). The day number should also be displayed at the beginning of each day.

As mentioned, at any given day, the player may perform any number of the following actions (unless otherwise specified):

1. **Buy real estate** – players may only select this option if they have at least 4,000,000 in their bank. The player then spends 4,000,000 to buy one unit of real estate. The effect of this investment will be discussed later (see the Day End Events section for more details). This can only be done once per day.
2. **Loan real estate** – players may only select this option if they have at least 2,000,000 in their bank and if they currently aren't paying off any other loaned real estate. The player spends 2,000,000 now, and for the next three days, is charged 1,000,000 per day. If all payments are completed (2,000,000 down payment and another 3,000,000 distributed equally across the next 3 days), the player gets 1 unit of real estate (they only get this unit at the day of the final payment). If the player cannot finish the payments (for example, because they do not have 1,000,000 in the bank on the second, third or fourth days), then the player does not get 1 unit of real estate and 2,000,000 will be refunded back to the bank of the player. For the sake of ordering, the 1,000,000 payment is made immediately after all periodic events (see the section on Periodic Events for more details).
3. **Sell real estate** – players may only select this option if they have at least one unit of real estate. Selecting this option will sell one unit of real estate, giving the player 2,000,000.
4. **Invest in gold** – players may only select this option if they have at least 100,000 in their bank. The player is then asked how many gold pieces he or she would like to buy. Each gold bar is worth 100,000. A player may not be allowed to buy more gold than they can pay for. The effect of this investment will be discussed later (see the Day End Events section for more details).

5. **Sell gold** – players may only select this option if they have at least one gold bar. When a player selects this option, they are asked how many bars they would like to sell. A player cannot sell more gold bars than they currently own. The player then loses a number of gold bars equal to how many they sold, and they would gain 75,000 for every gold bar sold (for example, if I had 10 gold bars and I sold 3, I would have 7 gold bars left and would earn 225,000 into their bank).
6. **End the day** – ending the day moves the game to the following day. When the day ends, certain events happen (see the section on Day End Events for more details).
7. **End the day x10** – this action ends the day 10 times, one after the other, essentially skipping to the ten days later. All events that could happen in the skipped days still happen, players simply aren't given any other option to do anything on the skipped days.

The actions should be presented to the user as a menu (a sample run is shown below), where the user will only select from the values 1-7

```
How much money would you like to start with (100000 - 10000000)? 50000000
Oops! Sorry, that's too much!
```

```
How much money would you like to start with (100000 - 10000000)? 10000000
How much is the maintaining balance (100000 - starting amount)? 1000000
How much is your target amount (>=starting amount)? 1000000000
```

```
Day 1
```

```
Bank Savings:10000000
Maintaining Balance: 1000000
Target Amount: 1000000000
```

```
What would you like to do?
```

1. Buy real estate
2. Loan real estate
3. Sell real estate
4. Invest in gold
5. Sell gold
6. End the day
7. End the day x10

```
Enter the number of your choice: 1
```

```
...
```

End of the Day Events:

At the end of each day, the following events happen **in order**:

1. If the money of the player in the bank is below 50,000, they earn 5,000.
2. The player earns an amount of money equal to 1% of their total money in the bank, rounded up to the nearest tens place.
3. The player then earns an amount of money equal to n% of their total money in the bank, rounded up the nearest tens place, where n is their total money divided by 500,000 rounded down to the nearest ones place.
4. Every 30th day from the first day (i.e. day 30, day 60, day 90,etc), the player earns 100,000 for every unit of real estate they own.

5. Every 10th day from the first day (i.e. day 10, day 20, day 30, etc.) the player earns 10,000 for every gold bar they own.
6. If the player owns at least 1 unit of real estate, they pay 1,000. (Flat Real Estate Tax)
7. For every unit real estate a player owns, they pay 2,000 (Real Estate Tax per unit, this is a different fee from number 6).
8. For every gold bar a player owns, they pay 500. (Gold Bar Tax per unit)
9. Every 100th day from the first day (e.g. day 100, day 200 etc.), the player loses half their money rounded up to the nearest tens. (Catastrophic event)

Remember, these day end events are checked at the end of each day in the order they were listed. This essentially means players earn money first before they have to pay.

Start of the Day Events:

These events happen at the start of each day:

1. End of Season Earning: Every 60th day, a player earn a 1,000.
2. Homeowner's Bonus: Every 40th day, a player with at least one real estate earns 55,000.
3. Lucky Bonus: Every 200th day from the first day, the money of the player doubles in value.
4. Prime Bonus: If every digit of the day number is prime (the day number itself may not be prime, for example every digit of 222 is prime, but 222 itself is not prime), the player earns 5,869 (which happens to be the 773rd prime).
5. Earthquake: Every 50th day, if the player owns more than one unit of real estate, they will lose one unit of real estate.
6. Petty Thievery: Every 13th day, if the player has more than 50,000, the player loses 100.
7. Gold Value Drop: Every 30th day, instead of earning 10,000 for every gold bar they own on the fifth day end event, players earn only 5,000.
8. Real Estate Oversupply: Every 90th day, instead of earning 100,000 for each unit of real estate the player owns on the fourth day end event, players earn only 40,000.

Events that happen periodically (i.e. occur every fixed number of days), may overlap, if this is the case, 2 or more periodic events can happen at the same time. If more than one periodic event is to happen on a day, they are performed in the order the periodic events are above:

Approaching the Machine Problem

With regards to how you are to start, I suggest attacking the problem conceptually first. Do not forget to use your computational thinking skills to first break the big programming project apart into individual scenarios, list them all down, then visualize how they all connect to each other. This breakdown and visualization is not required and would not be submitted, but it is highly recommended that you plan your solution out first before starting any coding.

Requirements & Documentation

This project is to be done in **pairs**. You are not allowed to share any code between you and your friends.

For this project, you are required to submit a Python file and a test case file in AnimoSpace. No external libraries are allowed for now, as the project is easy enough to accomplish without them. It is recommended that you submit at least **a week before the deadline**, and just submit again **before the deadline** should you have any updates.

Note the following requirements:

1. You are required to implement the project using Python Language.
2. You are free to look for resources online pertaining to programming constructs that are not discussed in class (e.g. lists, for loops, functions, etc.). The only catch is that you cite

everything. The use of loop controls - such as **break**, **continue**, **exit** or **pass** - is **NOT** allowed. You are also **NOT** allowed to use delay functions, such as the **sleep** function under time. Usage of these will result in corresponding deductions.

3. The implementation will require you to:
 - a. Appropriately use conditional statements, loops and other constructs discussed in class
 - b. Follow coding standards - descriptive variable names, proper tabs etc.
 - c. Include internal documentation (i.e. comments) to explain certain portions of your code.
4. You are to submit via AnimoSpace the following requirement before **Sept 28 6:00pm**.
 - a. **Python code (.py)**
 - b. **Test Cases (.pdf)**