

Understanding the Problem

The Zoo Tycoon has to pass seven fundamental steps. The first step is offering choices to the user. For instance, the program will ask the user to choose buying an animal, selling an animal, feeding an animal, or curing an animal in the zoo. Then the user will select one of the options and the program will reflect the result of the choice to the program. The third step is that when the user wants to end the turn, but don't want to quit the program, the program will go back to the first step with the changed settings from the second step, and add one month. The fourth step is that the program has to operate random event which is chosen among the default events, and randomly allocate the total attendance of the zoo. The fifth step is that the program has to add a revenue which is depending on the amount of total attendance of the zoo and the number of total animals. The sixth step is that the program has to add the babies of animals in the zoo. After the first operation of the third step, the program has to repeat fourth step to sixth step until the user quits the game.

To be specific, the program starts with no animal and \$50000. It offers choices to the user. It asks the user to buy an animal, sell an animal, or cure the animal in the zoo. However, the user cannot sell or cure the animal if there is no animal in the zoo. Thus, in the first time, the only thing the user can do is buying the animal. When the user choose to buy an animal, the user can select three species which are monkey, sea otter, and slater. Moreover, the user can buy one species at a time(month), and buy the animal up to two. When the user chooses to end the turn, the program will go to the next month, and the age of the animal in the zoo will be increased by one month. The program will spend the money for feeding the animal by each month. Moreover, the program will randomly allocate the events and the revenue which is depending on the number of animals. The revenue which the animal earn by month will be different by the species and the age of the animal. Finally, the program will iterate the game infinitely unless the user selects to quit the game.

<Assumption>

I assume that the program will regulate the user not to buy more than one specie per month.

I assume that the program will inform the user that if the user don't choose to cure the sick animal, the animal will die.

I assume that the program will end the game if the user has no money in the bank account.

I assume that the user can only buy one specie per month.

I assume that the program will inform the user where the money had spent.

<Class Assumption>

Zoo class

-member variable: total attendance of the zoo, bank account, money

Animal class

-member variable: specie of the animal, age of the animal

Monkey class

-member variable: the revenue that monkey earns, the cost of the monkey, the number of babies which the monkey can reproduce at once, the price of the food

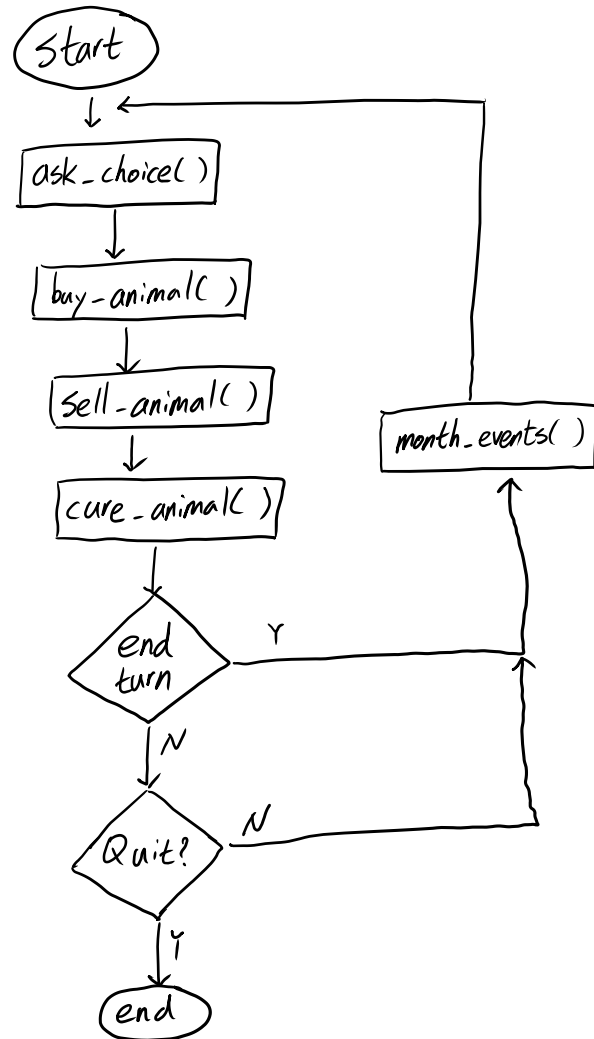
Sea otter class

-member variable: the revenue that sea otter earns, the cost of the sea otter, the number of babies which the sea otter can reproduce at once, the price of the food

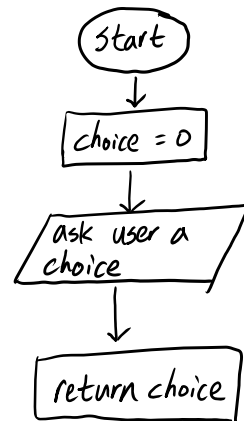
Slater class

-member variable: the revenue that slater earns, the cost of the slater, the number of babies which the slater can reproduce at once, the price of the food.

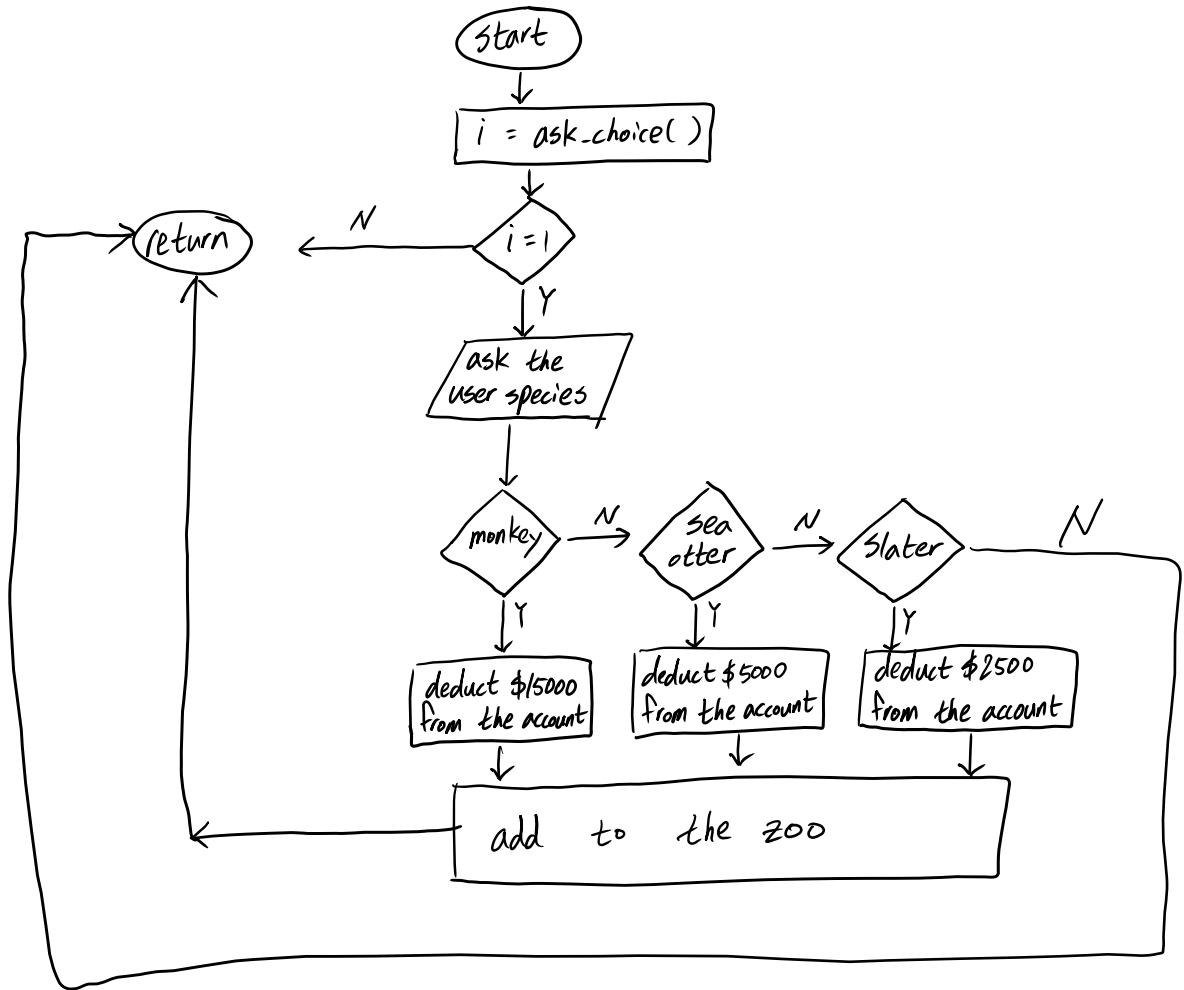
Program Design



<ask_choice()>



<buy_animal()>



<sell_animal()>

start

count = 0, CS = 0

total animal > 0

i = ask_choice()

i = 2

count > 2
CS > 1

monkey

add \$7500
to account

sea
otter

add \$2500
to account

slater

add \$1250
to account

deduct from the zoo

count++ , CS++

return

N

N

Y

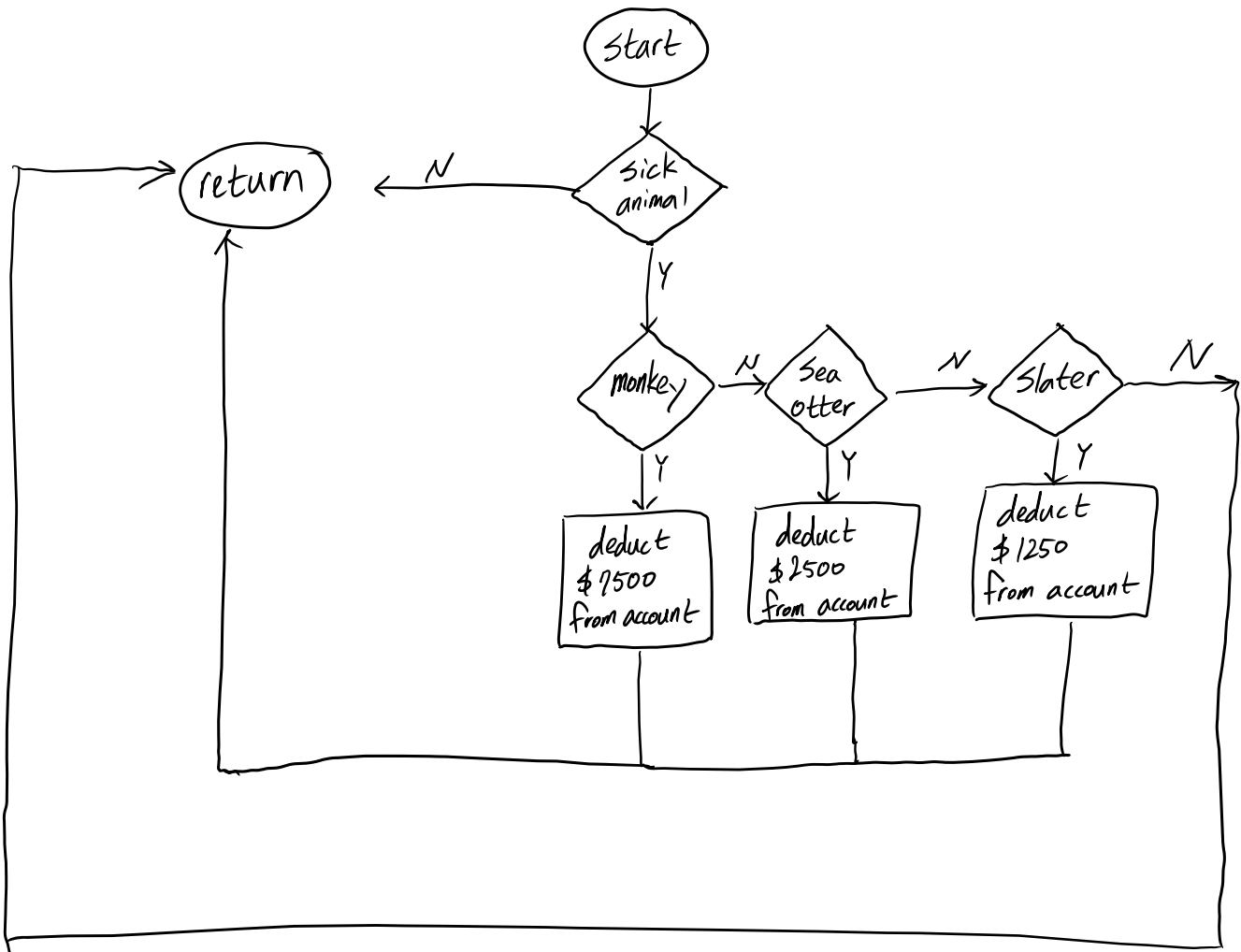
N

N

N

N

<cure_animal()>



<month_event()>

start

total attendance = total animal \times (rand() % 51 + 50)

month = 0

rand_event()

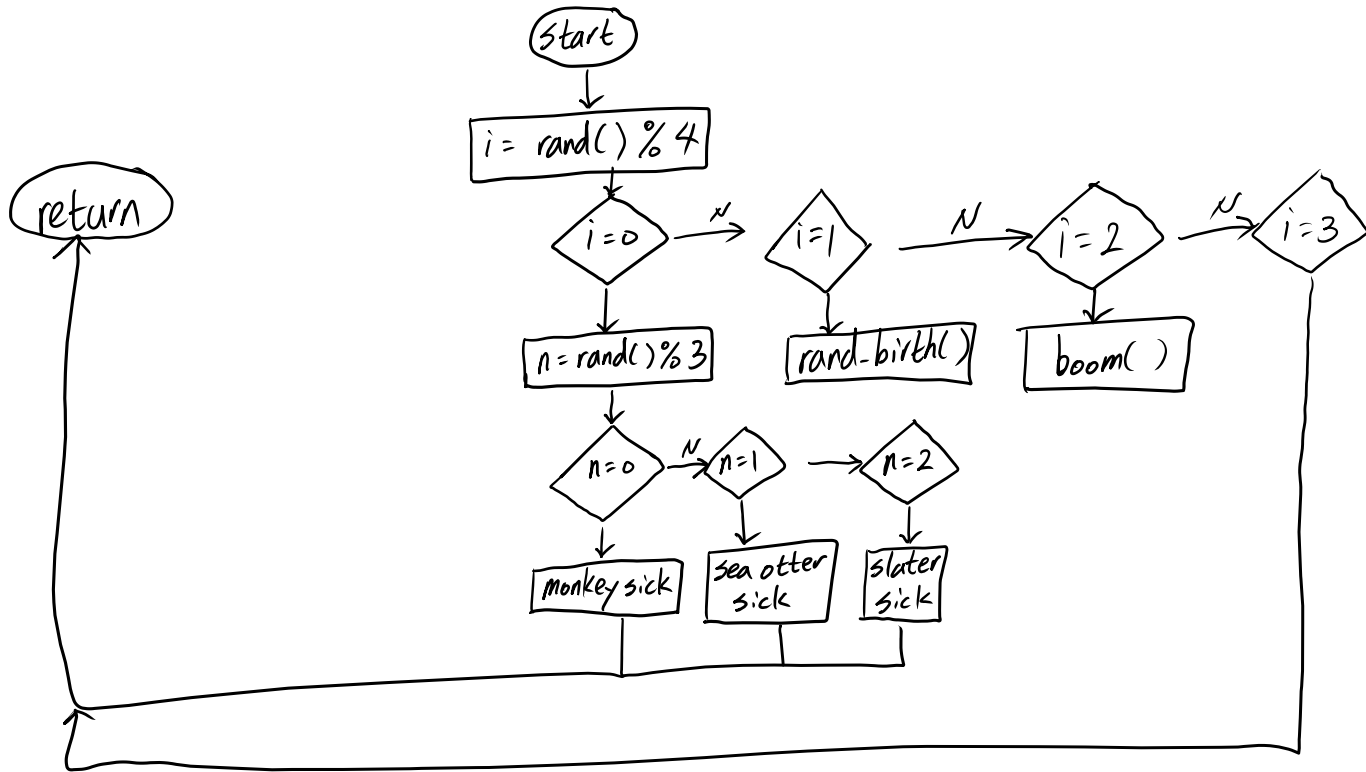
feed_animal()

revenue()

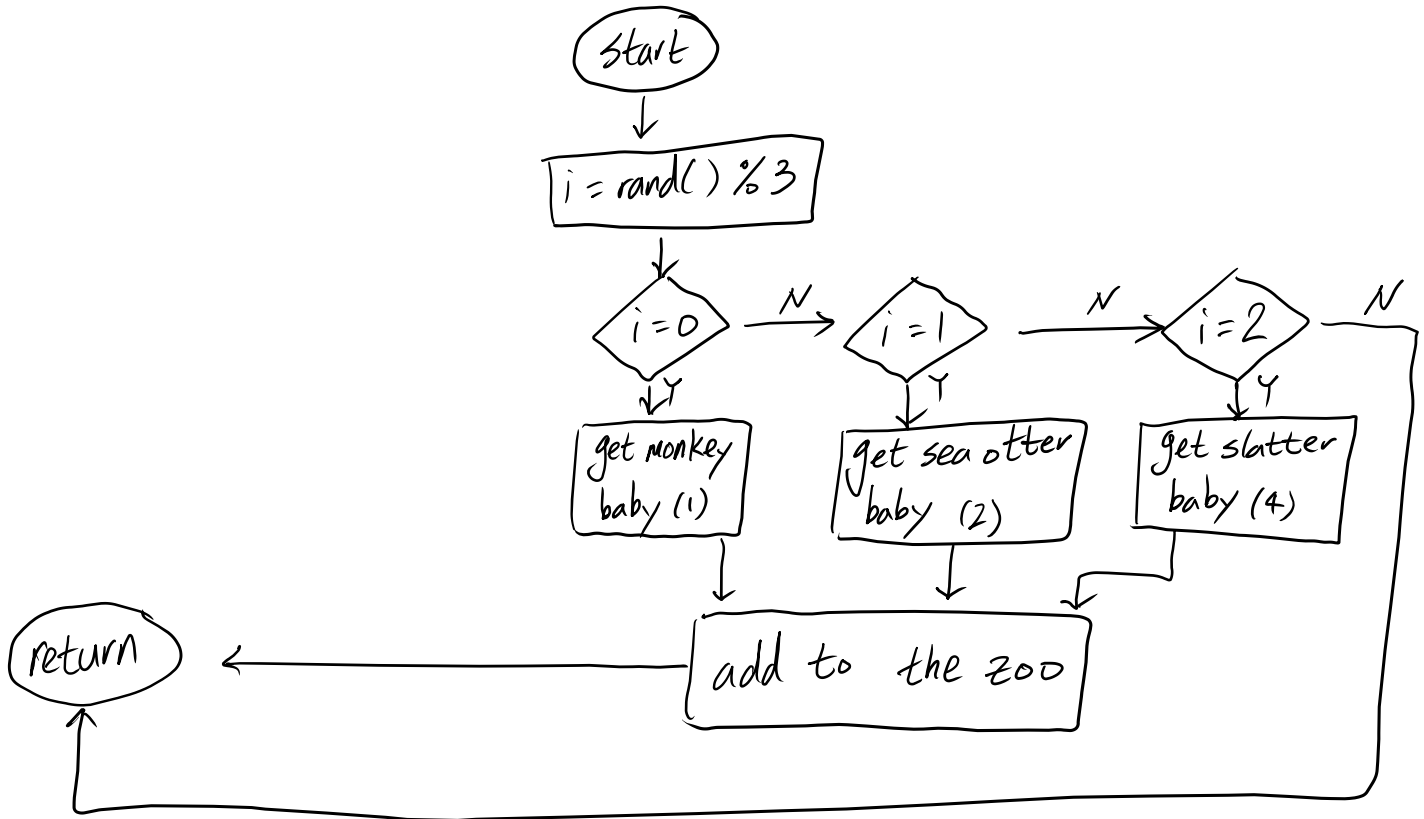
month++

return

<rand_event()>



<rand_birth()>



<boom()>

start



2x total attendance



2x base food cost



return

<feed_animal()>

start

deduct
 $4 \times \text{total monkey} \times \text{base food cost}$
from account

deduct
 $2 \times \text{total sea otter} \times \text{base food cost}$
from account

deduct
 $\text{total slater} \times \text{base food cost}$
from account

return

<revenue()>

Start

add
 $(\text{total adult monkey} + 2 \times \text{total baby monkey}) \times \text{total attendance} \times 2$
to account

add $(\text{total adult sea otter} + 2 \times \text{total baby sea otter}) \times \text{total attendance} \times \frac{1}{2}$ to account

add $(\text{total adult slater} + 2 \times \text{total baby slater}) \times \text{total attendance} \times \frac{1}{2}$ to account

return

Program Testing

Setting	Input	Expected Result
<case 1>	2	Print “You cannot sell animal!”
<case 1>	1	<case 2>
<case 2>	1	Print “You paid \$15000 for buying a monkey” Go to the options.
<case 1>	4	<case 3>
<case 3>	5	The program ends the game.

<case 1>

Month: 0

Animals: 0, Bank account: \$50000

Please choose the options:

1. buy an animal
2. sell an animal
3. cure an animal
4. end the turn
5. quit

<case 2>

Please choose the species:

1. Monkey (\$15000)
2. Sea Otter (\$5000)
3. Slater (\$2500)

<case 3>

Month: 1

Animals: 1, Bank account: \$35000

Please choose the options:

1. buy an animal
2. sell an animal
3. cure an animal

4. quit