

Zombie Apocalypse



In order to prepare for the apocalypse, survivors must gather good food and water, fend off zombies, and care for the wounded.

My mission

I have written the code for a program which calculates the amount of food and water that a group of survivors would need to survive for the duration of a Zombie Apocalypse.

My deliverables

- The program gets the following info from the user:
 - How long the apocalypse lasts.
 - How many characters are in the group.
 - What the is role of each character.
 - What the is age of each character.
- The program calculates the following using the info above:
 - How much water the group needs.
 - How much food the group needs.

My pseudo code

Reset all values from previous uses

Prompt for Apocalypse duration

 If duration less than zero, prompt again

Prompt for # of Characters

 If # of Characters less than zero, prompt again

Repeat following (# of characters) times

 Prompt for type of Character (Hunter, Caregiver, Gatherer)

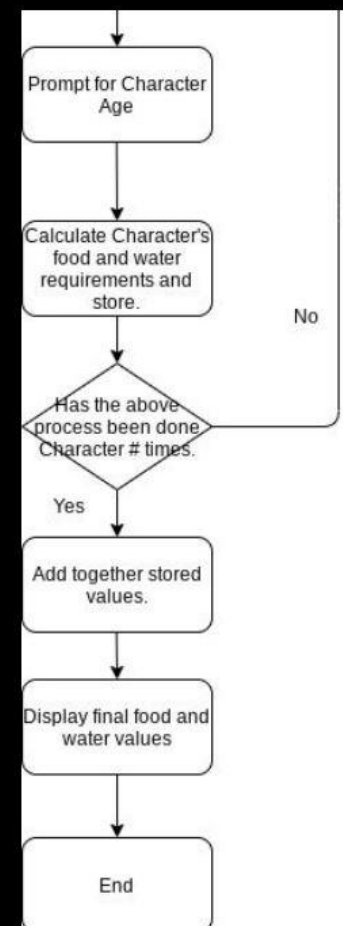
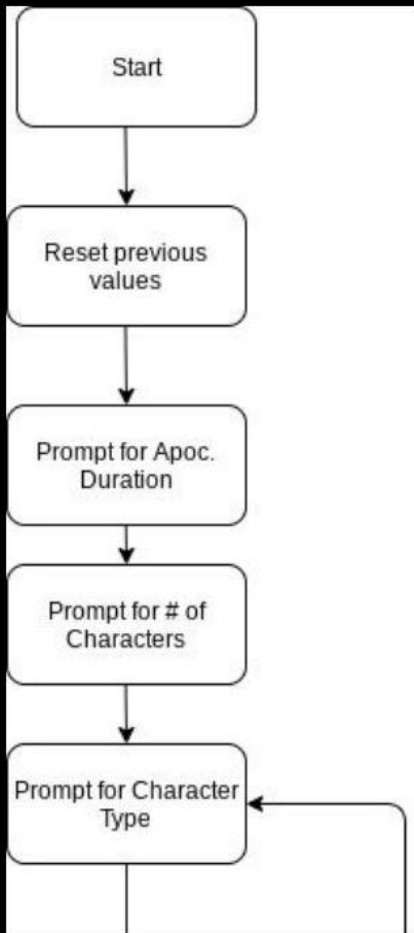
 Prompt for Character's Age

 Calculate Character's food and water needs with Age and Role

 Add values to final food and water total

Display final food and water values

Flow chart of code



Challenges I faced

One of the challenges I faced was getting the program to repeat the character information process for the number of characters in the group. If I hadn't been able to get that to work I would've had to add 3x the code that it currently has.

Another challenge I faced was with getting the food and water totals. Originally, I had the program store the food and water totals in a variable, then added them together at the end, however this meant every character needed a variable, forcing me to add a 5 character limit. I solved this problem by adding the food and water from each character to the total immediately after calculating them.

Powered by  trinket

Zombie Apocalypse Resource Calculator

Please enter general info to begin.

How long will the apocalypse last? (In Months) 12

How many characters are in our group? (Maximum of 5 characters per group) 3

Thank you, now please enter the information for character number 1.

What is this character's role? (Hunter, Gatherer, or Caregiver) Hunter

What is this character's age? 27

Thank you, now please enter the information for character number 2.

What is this character's role? (Hunter, Gatherer, or Caregiver) Gatherer

What is this character's age? 19

Thank you, now please enter the information for character number 3.

What is this character's role? (Hunter, Gatherer, or Caregiver) Caregiver

What is this character's age? 23

Thank you for your time. Here are the food and water requirements for your survivors to outlast the Apocalypse.

Calories Required:

2430000.0

Water Required (In Gallons):

31860.0

GUI and User interaction

Black text is text given by the program. This text prompts the user to respond to questions, such as what a character's age is, as well as give them information, such as their final resource requirements.

Blue text is text imputed by the user. These are the answers to prompts given by the program, such as a character's age being 23, and are used to calculate the final resource requirements.

Code

```
? set Final Gallons Needed to 0
set Final Calories Needed to 0
set Current Character to 1
set Duration (Months) to 0
set Number of Characters to 0
set Type of Character to 0

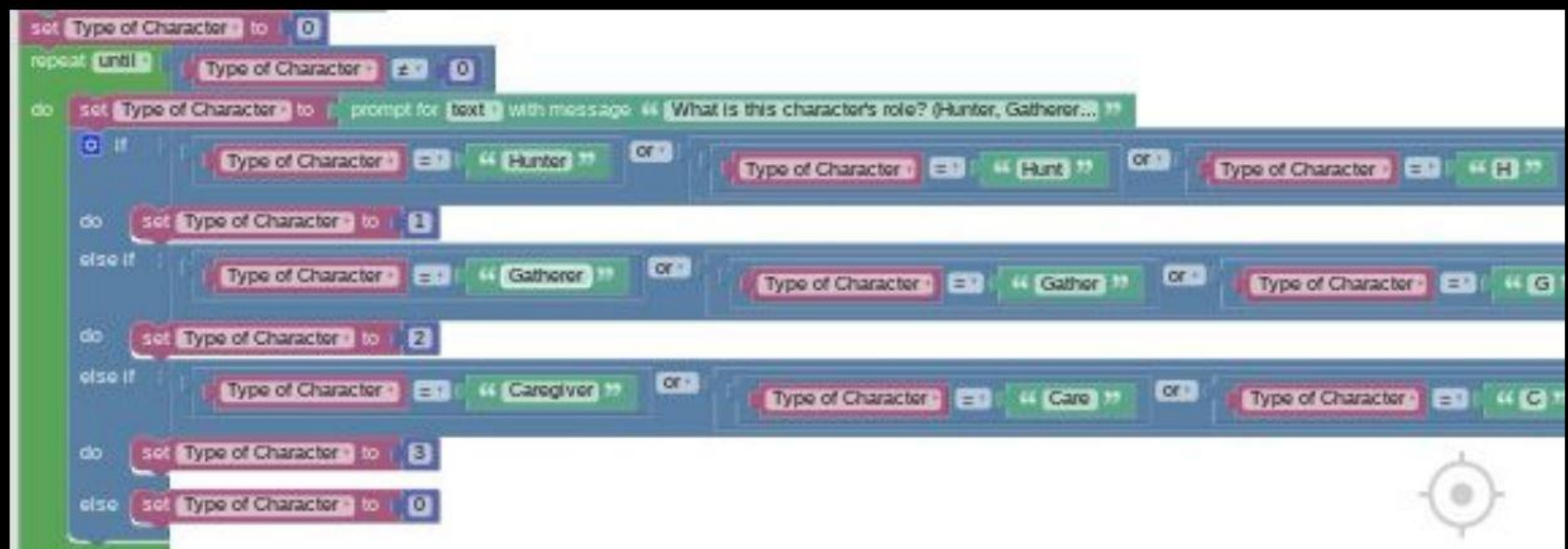
? print "Zombie Apocalypse Resource Calculator"
print "Please enter general info to begin."

? set Duration (Months) to prompt for number with message "How long will the apocalypse last? (In Months)"
if Duration (Months) ≤ 0
do
  repeat until Duration (Months) > 0
  do
    print "Error: Duration of Apocalypse must be greater than..."
    set Duration (Months) to prompt for number with message "How long will the apocalypse last? (In Months)"

? set Number of Characters to prompt for number with message "How many characters are in our group?"
if Number of Characters ≤ 0
do
  repeat until Number of Characters ≥ 0
  do
    print "Error: Number of Characters must be greater than..."
    set Number of Characters to prompt for number with message "How many characters are in our group?"

? repeat Number of Characters times
do
  print create text with "Thank you, now please enter the information for ..."
  Current Character
  " "
  set Type of Character to 0
```


Code



The image shows a Scratch script for classifying a character's role based on user input. The script uses a 'repeat until' loop to ensure the user provides a valid response. Inside the loop, a 'prompt for text' block asks the user for the character's role. An 'if' block checks for 'Hunter', 'Hunt', or 'H' and sets the 'Type of Character' to 1. An 'else if' block checks for 'Gatherer', 'Gather', or 'G' and sets it to 2. Another 'else if' block checks for 'Caregiver', 'Care', or 'C' and sets it to 3. A final 'else' block sets it back to 0. A small target icon is visible in the bottom right corner of the code area.

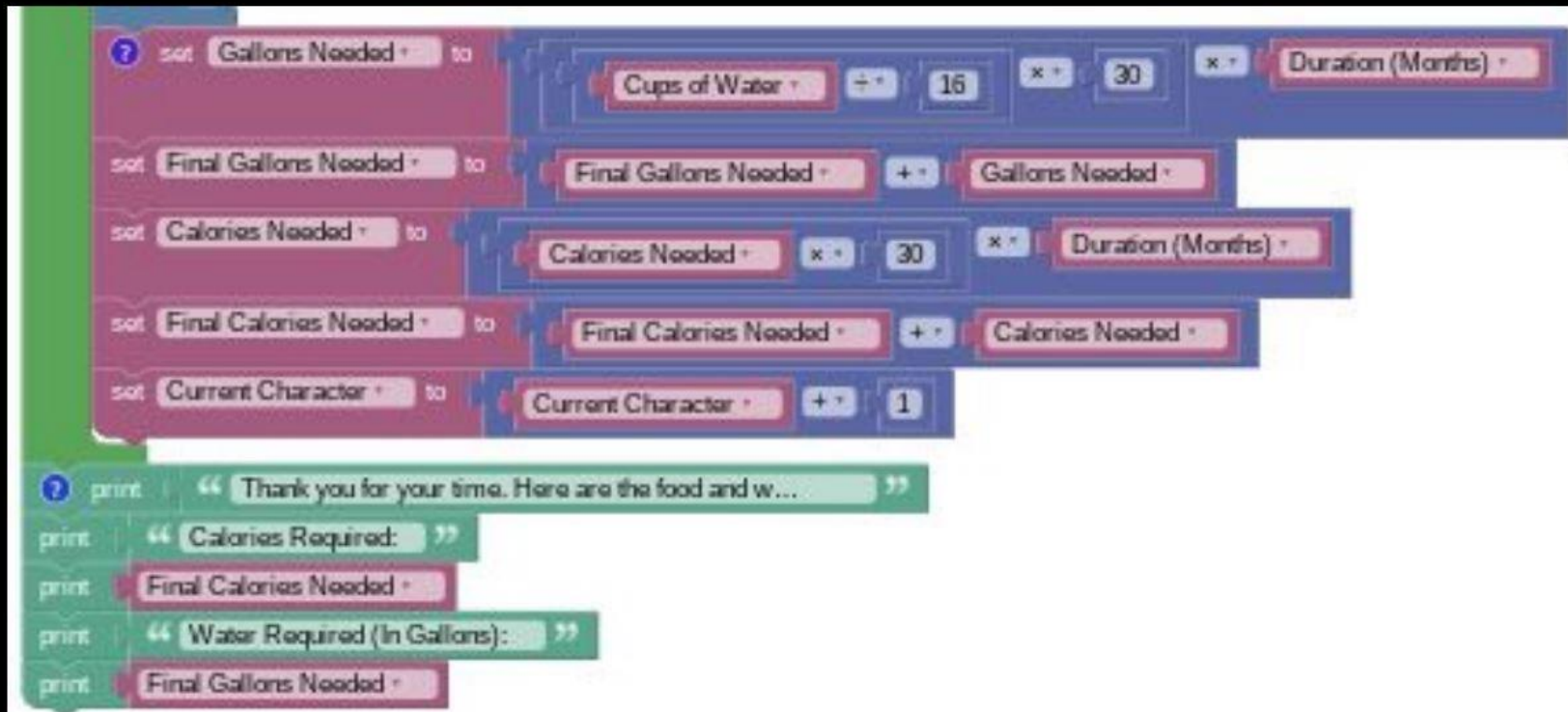
```
set Type of Character to 0
repeat until Type of Character  $\neq$  0
do
  set Type of Character to prompt for text with message "What is this character's role? (Hunter, Gatherer...)"
  if
    Type of Character = "Hunter" or
    Type of Character = "Hunt" or
    Type of Character = "H"
  do
    set Type of Character to 1
  else if
    Type of Character = "Gatherer" or
    Type of Character = "Gather" or
    Type of Character = "G"
  do
    set Type of Character to 2
  else if
    Type of Character = "Caregiver" or
    Type of Character = "Care" or
    Type of Character = "C"
  do
    set Type of Character to 3
  else
    set Type of Character to 0
```

Code

```
set Age of Character to prompt for number with message "What is this character's age?"

if Age of Character >= 18
do
  if Type of Character = 1
  do
    set Cups of Water to 15.5
    set Calories Needed to 2500
  else if Type of Character = 2
  do
    set Cups of Water to 13.5
    set Calories Needed to 2250
  else if Type of Character = 3
  do
    set Cups of Water to 11.5
    set Calories Needed to 2000
  end if
else
  if Age of Character >= 13
  do
    set Cups of Water to 9
    set Calories Needed to 2000
  else if Age of Character >= 9
  do
    set Cups of Water to 7
    set Calories Needed to 1800
  else if Age of Character >= 6
  do
    set Cups of Water to 5
    set Calories Needed to 1400
  else if Age of Character >= 4
  do
    set Cups of Water to 4
    set Calories Needed to 1200
  else if Age of Character <= 3
  do
    set Cups of Water to 2
    set Calories Needed to 1000
  end if
end if
end if
```

Code



My experience debugging

One of the bugs that came up was with the final calculations. I typed in a wrong number, and it caused the calculations to be off. I was able to fix this, and have the program give the correct final resource numbers.

Another bug I encountered was that if the user imputed an unusable number, such as a negative number, the program would still use it. I solved this by adding conditional loops which were triggered by the presence of an unusable number. These made the user input their answer again until it was usable.

How I could enhance my code in the future.

I would like to add more variables to the equation, such as a rate of survival for characters or the possibility of characters that join the group later into the Apocalypse. These would allow for more accurate info depending on what happens in the user's story.

Another thing I would've liked to do is transform the program into a text based game, similar to Oregon Trail, where players would have to make decisions for their characters to make sure they survive the Apocalypse.

If someone was to inherit your code...

What advice would I give?

I would tell them to make sure they remember the numbers which correspond to the roles of survivors as getting those mixed up would ruin the calculations at the end of the program.

Another piece of advice I would give would be to always double check the final resource totals to ensure they were properly calculated as that is probably the best way to find out if there is a bug in the code.