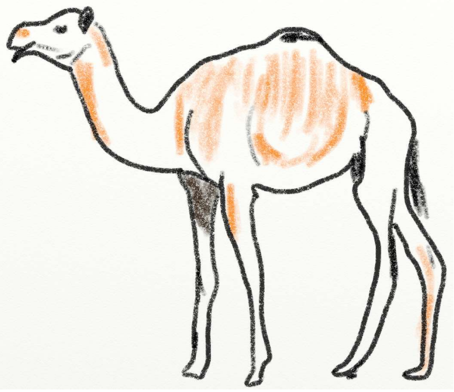
Program Arcade GamesWith Python And Pygame

Lab 4: Camel

4.1 Description of the Camel Game


(labs/camel/camel.png)

The idea for Camel originally came from the Heath Users Group and was published in More BASIC Computer Games (http://www.atariarchives.org/morebasicgames/showpage.php?page=24) in 1979.

The idea is to ride your camel across the desert while being chased. You need to manage your thirst, how tired the camel is, and how far ahead of the natives you are.

This is one of the first games I programmed on the Apple //e. The game is flexible. I've had students create Star Wars themed versions of this game where you need to ride a wampa across Hoth. It is easy to add sandstorms and other random events to the game to make it more interesting. Even though it is only a text-based game, I've found some students really get into playing it. Once I had a professor come and complain when my students were too loud while playing this game on the “big screen.”

4.2 Sample Run of Camel

Here is a sample run of the game:

```
Welcome to Camel!
You have stolen a camel to make your way across the great Mobi desert.
The natives want their camel back and are chasing you down! Survive your
desert trek and outrun the natives.

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
Q. Quit.
Your choice? C

You traveled 12 miles.

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
```

```
Q. Quit.
Your choice? C

You traveled 17 miles.

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
Q. Quit.
Your choice? e

Miles traveled:  29
Drinks in canteen:  3
The natives are 31 miles behind you.

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
Q. Quit.
Your choice? b

You traveled 6 miles.

...and so on until...

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
Q. Quit.
Your choice? C

You traveled 12 miles.
The natives are getting close!

A. Drink from your canteen.
B. Ahead moderate speed.
C. Ahead full speed.
D. Stop and rest.
E. Status check.
Q. Quit.
Your choice? C

You traveled 11 miles.
The natives are getting close!
You made it across the desert! You won!
```

4.3 Programming Guide

Here are the steps to complete this lab. Feel free to modify and add to the lab. Try the game with friends and family.

1. Create a new program and print the instructions to the screen. Do this with multiple **`print`** statements. Don't use one **`print`** statement and multiple **`\n`** characters to jam

everything on one line.

```
Welcome to Camel!
You have stolen a camel to make your way across the great Mobi desert.
The natives want their camel back and are chasing you down! Survive your
desert trek and out run the natives.
```

2. Create a boolean variable called `done` and set to `False`.
3. Create a `while` loop that will keep looping while `done` is False.
4. Inside the loop, print out the following:

   ```
   A. Drink from your canteen.
   B. Ahead moderate speed.
   C. Ahead full speed.
   D. Stop for the night.
   E. Status check.
   Q. Quit.
   ```

5. Ask the user for their choice. Make sure to add a space before the quote so the user input doesn't run into your text.
6. If the user's choice is Q, then set `done` to `True`. By doing something like `user_choice.upper()` instead of just `user_choice` in your `if` statement you can make it case insensitive.
7. Test and make sure that you can quit out of the game.
8. Before your main program loop, create variables for miles traveled, thirst, camel tiredness. Set these to zero.
9. Create a variable for the distance the natives have traveled and set it to -20. (Twenty miles back.)
10. Create and set an initial number of drinks in the canteen.
11. Add an `elif` in your main program loop and see if the user is asking for status. If so, print out something like this:

    ```
    Miles traveled:  0
    Drinks in canteen:  3
    The natives are 10 miles behind you.
    ```

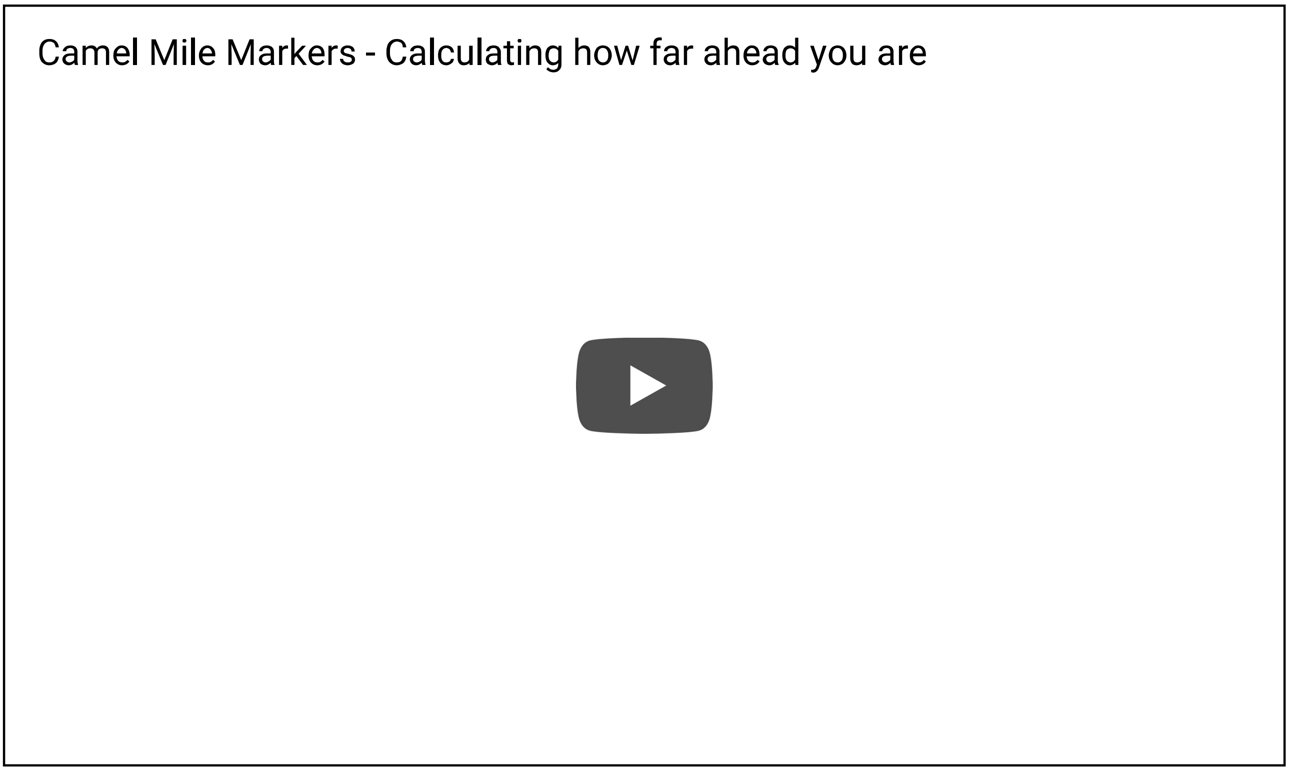
12. Add an `elif` in your main program loop and handle if the user wants to stop for the night. If the user does, reset the camel's tiredness to zero. Print that the camel is happy, and move the natives up a random amount from 7 to 14 or so.
13. Add an `elif` in your main program loop and handle if the user wants to go ahead full speed. If the user does, go forward a random amount between 10 and 20 inclusive. Print how many miles the user traveled. Add 1 to thirst. Add a random 1 to 3 to camel tiredness. Move the natives up 7 to 14 miles.
14. Add an `elif` in your main program loop and handle if the user wants to go ahead moderate speed. If the user does, go forward a random amount between 5 and 12 inclusive. Print how many miles the user traveled. Add 1 to thirst. Add 1 to camel tiredness. Move the natives up 7 to 14 miles.
15. Add an `elif` in your main program loop and handle if the user wants to go ahead drink from the canteen. If the user does, make sure there are drinks in the canteen. If there are,

subtract one drink and set the player's thirst to zero. Otherwise print an error.

16. In the loop, print “You are thirsty.” if the user's thirst is above 4.
17. Print “You died of thirst!” if the user's thirst is above 6. Set `done` to true. Make sure you create your code so that the program doesn't print *both* “Your are thirsty” and “You died of thirst!” Use `elif` as appropriate.
18. Print “Your camel is getting tired.” if the camel's tiredness is above 5.
19. Print “Your camel is dead.” if the camel's tiredness is above 8. Like the prior steps, print one or the other. It is a good idea to include a check with the `done` variable so t hat you don't print that your camel is getting tired after you died of thirst.
20. If the natives have caught up, print that they caught the player and end the game.
21. Else if the natives are less than 15 miles behind, print “The natives are getting close!”
22. If the user has traveled 200 miles across the desert, print that they won and end the game. Make sure they aren't dead before declaring them a winner.
23. Add a one-in-twenty chance of finding an oasis. Print that the user found it, refill the canteen, reset player thirst, and rest the camel.
24. Play the game and tune the numbers so it is challenging but not impossible. Fix any bugs you find.

4.4 Hints

- Remember that it is good idea to put blank lines between logical groupings of code in your program. For example, but a blank line after the instructions, and between each user command.
- It is considered better style to use `while not done:` instead of `while done == False:`
- To prevent bad message combinations, such as printing “You died of thirst.” and “You found an oasis!” on the same turn, use the `and` operator. Such as, `if not done and thirst > 4:`



Video: Camel Mile Markers - Calculating how far ahead you are.

Video: Camel End Game - Figuring out when user is caught or crosses the desert.

You are not logged in. Log in here (user_progress_report.php) and track your progress.

English version (http://programarcadegames.com/index.php?lang=en) by Paul Vincent Craven (http://simpson.edu/author/pcraven-2/)
Spanish version (http://programarcadegames.com/index.php?lang=es) by Antonio Rodríguez Verdugo
Russian version (http://programarcadegames.com/index.php?lang=ru) by Vladimir Slav
Turkish version (http://programarcadegames.com/index.php?lang=tr) by Güray Yildirim
Portuguese version (http://programarcadegames.com/index.php?lang=pt-br) by Armando Marques Sobrinho (mailto:marquessbr@gmail.com) and Tati Carvalho
Dutch version (http://programarcadegames.com/index.php?lang=nl) by Frank Waegeman
Hungarian version (http://programarcadegames.com/index.php?lang=hu) by Nagy Attila
Finnish version (http://programarcadegames.com/index.php?lang=fi) by Jouko Järvenpää
French version (http://programarcadegames.com/index.php?lang=fr) by Franco Rossi
Korean version (http://programarcadegames.com/index.php?lang=ko) by Kim Zeung-Il
Chinese version (http://programarcadegames.com/index.php?lang=cn) by Kai Lin