CMPSC 121: Project 9

Jung-woo Sohn (jwsohn@ist.psu.edu)

November 9, 2017

1 Overview

This time we will add the following features to our game, extending implementations in Project 8 (counters for arrest, checking whether the detective is correctly tracking the criminal, assigning a random place for the criminal when moving to a new place, etc.)

- Checking the remaining hours for the detective.
  - Take hours off whenever the detective travels to a new place. (Assume 5 hours for an investigation and 8 hours for a travel)
  - When the detective runs out of the time before capturing the criminal, the game ends.
- Make source codes developer-friendly with abstraction: further separations of fheaders or cpp files using `*.h` headers or `*.cpp` files.

The following shows an example output from a working program:

```
$ ./game

***** Welcome to WHERE IN THE WORLD IS CARMEN SANDIEGO *****

Your name, please: Foobar
Welcome, detective Foobar

You have 7 days and 12 hours to arrest a criminal.

You are currently in New York.

Do you want to investigate around? [y] y
She told me that she had always wanted to see Proms festival.

You spent 5 hours.
Now you have a total of 7 days and 7 hours left.

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Please select your next destination:
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

1. New York
2. London
3. Cairo
4. Rio de Janeiro
```

```

Please enter your selection:
Your input is : 2
Your next destination is : London
You spent 8 hours.
Now you have a total of 6 days and 23 hours left.
8 hours passed while you were flying to London.

You are currently in London.

Do you want to investigate around? [y] y
She checked out books related to pyramids and ancient mysteries.

You spent 5 hours.
Now you have a total of 6 days and 18 hours left.

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Please select your next destination:
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

1. New York
2. London
3. Cairo
4. Rio de Janeiro

Please enter your selection:
Your input is : 4
Your next destination is : Rio de Janeriro
You spent 8 hours.
Now you have a total of 6 days and 10 hours left.
8 hours passed while you were flying to Rio de Janeriro.

You are currently in Rio de Janeriro.

Do you want to investigate around? [y] y
Well, I can contact you if I happen to see any person like him... or her.

You spent 5 hours.
Now you have a total of 6 days and 5 hours left.

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Please select your next destination:
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

1. New York
2. London
3. Cairo
4. Rio de Janeiro

Please enter your selection:
Your input is : 3
Your next destination is : Cairo
You spent 8 hours.
```

```
Now you have a total of 5 days and 21 hours left.
8 hours passed while you were flying to Cairo.

You are currently in Cairo.

Do you want to investigate around? [y] y
She exchanged her money into dollars.

You spent 5 hours.
Now you have a total of 5 days and 16 hours left.

=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Please select your next destination:
=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

1. New York
2. London
3. Cairo
4. Rio de Janeiro

Please enter your selection:
Your input is : 1
Your next destination is : New York
You spent 8 hours.
Now you have a total of 5 days and 8 hours left.
8 hours passed while you were flying to New York.

*******************************************
          There goes the criminal!
*******************************************

Congratulations, Detecive Foobar!
You have successfully arrested the criminal.
We greatly appreciate your effort.

You have 5 days and 8 hours left.
```

2 Instructions

2.1 Having a `data.h` file for game-wide variables

Suppose you are contacted by a developer interested newly participating in this gaming project. What should you do? First, you will need show him or her a summarizing overview on how your program works.

Of course, having a simple documentation is a good idea. But you should not forget that the best documentation is your source code. It should be readable, clear, and supplemented by comments. And it will be the best if your source code can serve as the documentation.

In your source code, which part can best serve as an "overview" section? Additionally, what kind of variables (and functions) do you want to show them first? When you want to create a new project-wide variable, where do you want to put it in — so that you and your collaborator can easily understand and communicate with?

In our project, most of such variables have been stored in the `functions.cpp` file. This time let us use `data.h` file particlularly as a *new* place for storing project-wide variables.

Follow the instructions below and creat `data.h` file:

- Create a new `data.h` file. Then copy and paste variable definitions from the beginning section of `functions.cpp` file. Listing 1 shows `data.h`.

Listing 1: `data.h` separated out from `functions.cpp`

```
#include <string>

using namespace std;

const int NUM_CITIES = 4;
string cities[] = {"New York", "London", "Cairo", "Rio de Janeriro"};

int hours = 0;

int currentLocation = 0;
int criminalLocation = 0;

int arrestCounter = 3;          // Criminal three turns away
```

- At the beginning of `functions.cpp`, insert

```
#include "data.h"
```

  so that the functions inside `functions.cpp` files can access the variable definitions separated out to `data.h`. Delete the duplicate variable definitions.

2.2 Checking the remaining hours for the detective: building block(s)

First of all, recall that the remaining hours for the detective is stored in `hours` variable (now defined in `data.h`).

Again, the primary learning objective in this project includes *function design*. Its importance can never be stressed much. Start from (i) determining what will be input parameters and return value and consider (ii) how to keep the features of the function simpliest as possible. Also consider how you can access `hours` variable by making use of `get...` and `set...` functions as the access interface; get familiar with this data abstraction technique.[1]

According to the game rule, note that the game comes to an over when the remaining hours reaches zero. Thus, the implementations have to be able to print out different messages dependent on the remaining hours being positive or negative. And they have to redirect the program execution flow either to continue the game (after taking some hours away) or to end the game.

The `hoursLeftMsg` function shown in Listing 2 will handle the requirements. Its role is:

- It uses `if-else` statement to determine what messages to print out.
- Only when the remaining hours are postitive, it takes away the hours spent on the travel and *update* the `hours` variable for the entire game.
- For redirecting the program execution flow, it simply returns true or false. Therefore, the function designer's intention here is that `hoursLeftMsg` will not do any *actions* on redirection of the program flow; however, it will take care of the work up to the scope of *making decisions*. It returns true or false dependent on whether the game can continue or not.
- Actually, the remaining tasks are completed in `main` function which calls `hoursLeftMsg` function.
- Note that the function is placed in `navigate.cpp` file, implying that the function can be considered as a high-level one, composed of low-level parts.

Listing 2: `hoursLeftMsg()` function in `navigate.cpp`

```
bool hoursLeftMsg(int decrement)
{
    int hrs = getHours() - decrement;

    if (hrs >= 0)
    {
        cout << "You spent " << decrement << " hours." << endl;
        cout << "Now you have a total of " << (hrs / 24) << " days"
             << " and " << (hrs % 24) << " hours left." << endl;

        setHours(hrs);    // update hours

        return true;
    }
    else
    {
        return false;
    }
}
```

2.3 Taking hours for investigation or travel: modifications in the game

Since all the "parts" are ready now, we can apply the remaining-hours rule to the actual gameplay.

When do you need to check remaining hours during gameplay? The detective are spending time (i) when investigating around the city or (ii) travelling to a new destination city. To be specific, you need to call `hoursLeftMsg` function when your code is processing either (i) or (ii).

[1]This technique later becomes very important with object-oriented programming.

Listing 3 shows the code template for the implementation of handling detective's remaining hours during gameplay. Note how `if - else` statements were effectively used and how the `while` loop can handle the gameflow with a proper condition. Finally, make it sure to complete the game by calling `alert` function correctly.

In the meantime, we assume that the detective spend 5 hours for an investigation and 8 hours for traveling by air for now. Check out the constant definitions at top of `welcome.cpp` template.

Listing 3: Template for `welcome.cpp`

```cpp
#include <iostream>
#include <string>
#include "fheaders.h"

using namespace std;


/* NOTE: provided for convenience. Also note the use of const and
   CAPITALIZATION naming style for constants */
const int CRIMINAL_COUNT = 3;
const int INVESTIGATION_HOURS = 5;
const int FLIGHT_HOURS = 8;

/* FILL IN THE BLANKS: list the prototypes of the functions that you will use
   here */

/* welcome() omitted for convenience
   .
   .
   .
   */

int main()
{
    string name = "";

    /* display welcome and get the name from the user */
    name = welcome();

    /* repeat until the counter reaches zero (or the hours left reaches zero */

    resetArrestCounter(CRIMINAL_COUNT);


    /* NOTE! NOTE! NOTE! on the use of getHours() > 0 condition. Now the game
       continues as long as (i) the detective keep hitting the correct destination
       up to the arrest counter times and (ii) the dective still has hours left
       for tracking. */

    while (getArrestCounter() > 0 && getHours() > 0)
    {
        int cityIndex = 0, hoursRemain = 0;
        char input;

        cityIndex = getCurrentLocation();
```

```
cout << endl;
cout << "You are currently in " << /* FILL IN THE BLANK */ << "." << endl;
cout << endl;

/* investigate on Carmen’s destination */
cout << "Do you want to investigate around? [y] ";
cin >> input;

if (input == ’y’ || input == ’Y’)
{
    cout << endl << /* FILL IN THE BLANK: investigation message? */
         << endl << endl;

    /* FILL IN THE BLANK: check the detective has hours left. Print
       out messages needed accordingly. How can you use hoursLeftMsg
       function here? */
}
else
{
    cout << endl << "Okay, skipping investigations." << endl << endl;;
}

/* select a destination and then update the location */

cityIndex = navigate();
hoursRemain = getHours();

/* NOTE: I am leaving most portion of hoursLeftMsg function call
   example for your information. There are two places you need to call
   hoursLeftMsg() function: (i) for investigation around the city (ii)
   for travelling to a new destination city */

if (hoursLeftMsg(FLIGHT_HOURS) > 0)
{
    setCurrentLocation(cityIndex);      // update with the new location

    cout << FLIGHT_HOURS << " hours passed while you were flying to "
         << /* FILL IN THE BLANK: how can you get the destination city name? */
         << "." << endl;

    if (cityIndex == getCriminalLocation())     // in case correctly tracking Carmen
    {
        decrementArrestCounter();       // getting closer to Carmen
    }
}
else
{
    cout << "Sorry but you cannot travel. The flight takes "
        << FLIGHT_HOURS << " hours " << "but you have only "
        << hoursRemain << " hours left." << endl;

    setHours(0);
```

```
        }
    }

    /* Check the condition for capturing Carmen. Result in arrest or lost */

    /* FILL IN THE BLANK: how can you call arrest() function here? */

    return 0;
}
```