I/O in Lab 4

Version 1

Contents

I/O

In lab 4, all balls will be read from a file given on the command line. The flipper activity might be read directly off the keyboard for a bonus point.

The command line must have the right number of arguments

The initialization checks in lab 4 are more extensive than lab 3.

- If argc is less than 2, the lab should give an error and decline to run.
- If argc is 3 and bonus mode is not present in the code, the lab should give an error and decline to run.
- If argc is 3 and the lab is in TEXT mode, the lab should give an error and decline to run regardless of whether bonus code is on-board or not.
- If argc is 3, the lab is in GRAPHICS modes, and the bonus code is on-board, the lab must run in bonus mode if all other initializations pass.

The output below is for when no file name is given on the command line (the first bullet point above):

```
[kirby.249@cse-sl1 lab4]$ lab4
ERROR: Insufficient arguments (argc is 1).
Total run time is 0.000042677 seconds.
[kirby.249@cse-sl1 lab4]$
```

The output below is the same for the second and third cases given in the bullet points above:

```
[kirby.249@cse-sl1 lab4]$ lab4 xinf.pbd bonus
ERROR: Bonus code is not present.
Total run time is 0.000033617 seconds.
[kirby.249@cse-sl1 lab4]$
```

The file must open for reading

The input filename is given as the first argument to the lab4 command. If just the file is given, the code will attempt to run the lab in non-bonus mode. If the file cannot be opened for reading, the lab should decline to run. If the file opens, a DIAGNOSTIC will be printed:

```
[kirby.249@cse-sl1 lab4]$ lab4 xinf.pbd
DIAGNOSTIC: Successfully opened xinf.pbd for reading.
```

The input file should be opened before attempting to initialize graphics, so your code will not need to check for TEXT mode before printing errors or diagnostics. If a file that cannot be opened is given on the command line, an ERROR message must be printed.

```
[kirby.249@cse-sl1 lab4]$ lab4 this_file_does_not_exsist
ERROR: Unable to open this_file_does_not_exsist for reading.
Total run time is 0.000491619 seconds.
[kirby.249@cse-sl1 lab4]$
```

Use the **fopen** function to open the file. Google that function and read up on file I/O. Open it in read-only mode using "r" as the mode.

At the end, the file must be closed

Before the lab ends, close the input file pointer using **fclose** and print a DIAGNOSTIC message. This can happen late in the game, but it must happen before the final run time computation. It is OK to close it after graphics are torn down –doing so means that the code won't have to check for text mode.

```
DIAGNOSTIC: Input file closed.
```

Reading balls from the file

Use **fscanf** instead of **scanf** to read balls.

Reading flippers – bonus

You need the V4 pinball library for the bonus.

If the bonus flag is set in the sim structure, your code will loop performing non-sleeping reads one character at a time using **pb_getch()** which is similar to **getchar()**. This call always returns immediately regards of whether a typed-in character awaits on the input stream or not. If a character was waiting, it will be returned. If no character was waiting, it returns ERR, a macro defined in

<curses.h> as having a value of -1. It may be best to add the following line to input.c instead of including curses.h.

```
#define ERR      (-1)
```

 If a value other than ERR is returned, process that character and then read again. If ERR is returned, ignore it and the loop is done. ERR means that the user hasn't pressed any keys since we read that last one. We will read all waiting characters so that we can do both flippers at the same time.

The input.c file will need to include libpb.h to get access to **pb_getch**.

Testing

You should get the same numbers for text mode as lab 3. There will be formatting changes as demanded in the output document.

Bonus mode should be tested with xinf.pbd to have any prayer of being able to play the game.