# Bonus Mode in Lab 4 – Live Flippers

OK, it's really about character I/O.

## Contents

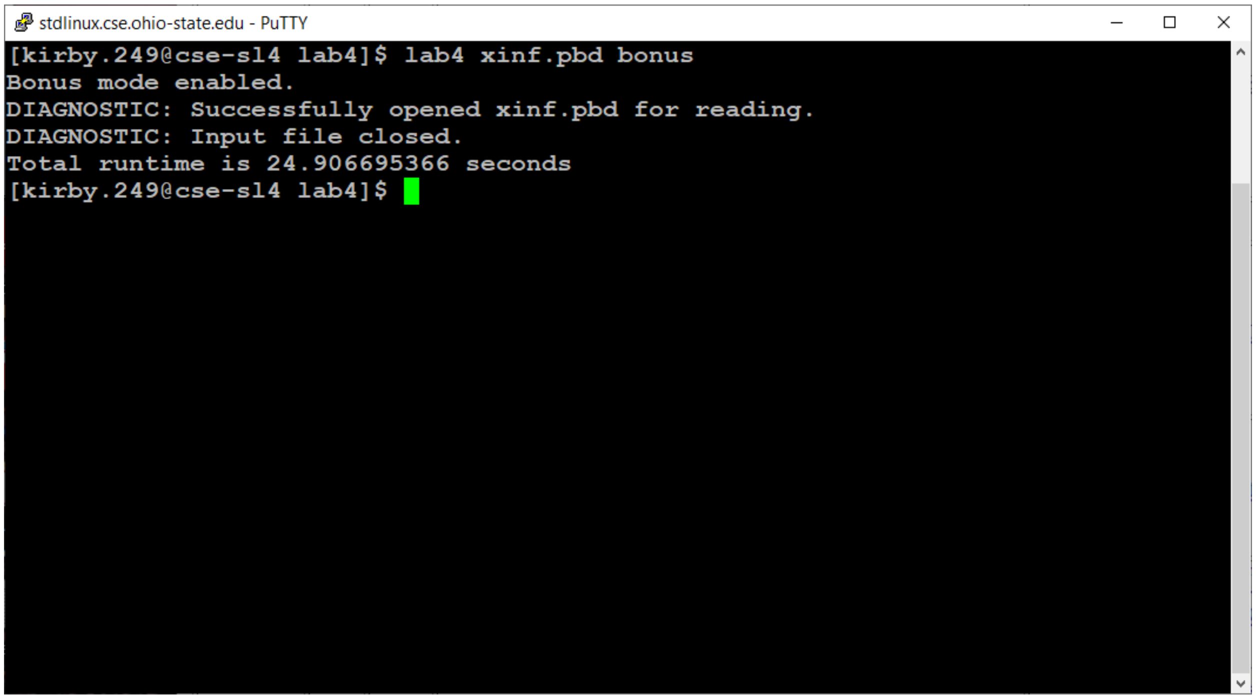
## Overall

See the section on I/O about the underlying I/O needs of bonus mode and the command line arguments needed to activate it. This document assumes that the code is in GRAPHICS mode and that you intend to have bonus code on-board.

The bonus mode uses character input in real-time to activate the flippers. Press a key and a flipper is live for 8 simulation steps (8/64 of a second, or 1/8th of a second). Note that there is lag in the system and the balls tend to be fast moving. This is not an easy game to play.

## Bragging Rights

After you complete the whole lab, including bonus mode, feel free to participate in the bragging rights competition. No points are awarded, but if you have a competitive streak, it's fun. Take a screenshot of your best run of xinf.pbd after it has finished and you are at a command prompt.



```
[kirby.249@cse-sl4 lab4]$ lab4 xinf.pbd bonus
Bonus mode enabled.
DIAGNOSTIC: Successfully opened xinf.pbd for reading.
DIAGNOSTIC: Input file closed.
Total runtime is 24.906695366 seconds
[kirby.249@cse-sl4 lab4]$
```

Note that the bonus flag on the command line must be visible. The goal is the longest total runtime. Longer times are better. Anything near 40 seconds is really good. Post this as a follow-up on Piazza to the bragging rights thread.

## Making it Happen

### Changes to the sim structure

You need a bonus flag so that code can tell if it is in bonus mode or not. You also need an integer "timer" for each of the two flippers. Initialize those timers to 1, regardless of mode. The initialization sequence will handle setting the bonus flag when it is checking command line arguments.

### Changes to the flipper constraint code

Each flipper constraint function should check its timer first thing. If the timer has time on it, the flipper is active and can interact with the current ball. If the timer is zero, the flipper is disabled and will do nothing to the current ball. In regular mode, no code will change the timer from the default value of 1, so in regular mode they will always be enabled.

## Additional code

### Simulation loop code

In the simulation loop, the new sequence of events will be

- The existing call to output the world
- The existing increase to the elapsed time by delta_t
- <u>An added call into a new input function that handles keystrokes</u> (Only call it in bonus mode)
- The existing update the world call that moves balls and deals with off-table balls

- If you didn't include dealing with balls that came off the table as part of updating the world, your existing code to do so remains at the bottom of the loop body.

In bonus mode, this additional code will turn the flippers on and off just prior to the updates that eventually result in calls to the flipper constraint functions.

### Input code

This code lives in input.c (the same file that owns the scanf call). The function called from the simulation loop will decrement the two timers. Then it will have a loop that reads input characters until there are no more input characters from the keyboard to read. (See the IO document). The pb_getch() function will return ERR (-1) when there are no characters waiting for it. It will hand read characters off to be parsed and will terminate the loop and return when pb_getch() returns ERR.

Interesting characters:

| Left Flipper Keys | Right Flipper Keys |
|---|---|
| 'a' | 'f' |
| 'j' | ';' |
| '4' | '6' |

Your code must handle all of the keys int the above table. The parsing function will check the given character to see if it is interesting. If it is, it will activate the correct flipper. A switch-case construct might be best for this. Regarding the ten-line limit, do not count **case:** lines.

Activating a flipper first checks the timer for that flipper. If the timer is at 0, the code calls either pb_left() or pb_right(). Regardless of the timer value, the appropriate timer is set to 8.

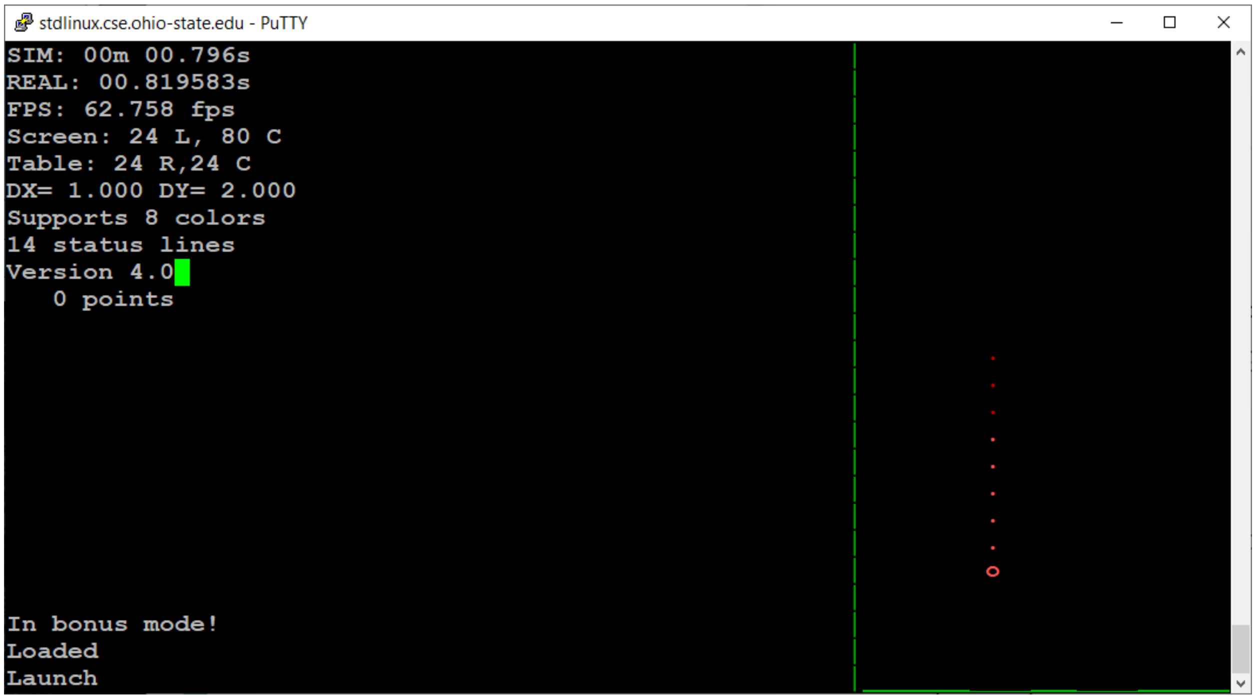
**Note:** While you can test with other numbers than 8, do not post the bragging rights on Piazza with any value other than 8.

## Messages when in Bonus mode

Before you initialize graphics, do the other initializations. Bonus mode code owes a bonus mode message when it processes the command line arguments.

```
[kirby.249@cse-sl2 lab4]$ lab4 xinf.pbd bonus
Bonus mode enabled.
DIAGNOSTIC: Successfully opened xinf.pbd for reading.
DIAGNOSTIC: Input file closed.
Total runtime is 5.128400564 seconds
[kirby.249@cse-sl2 lab4]$
```

After graphics initializes make a pb_status call that says bonus mode as seen here on the lower left.



```
SIM: 00m 00.796s
REAL: 00.819583s
FPS: 62.758 fps
Screen: 24 L, 80 C
Table: 24 R,24 C
DX= 1.000 DY= 2.000
Supports 8 colors
14 status lines
Version 4.0
   0 points






In bonus mode!
Loaded
Launch
```