

**CMSI 370-01**  
**INTERACTION DESIGN**  
Fall 2015

## Assignment 1211 Feedback—Direct Manipulation Application

Mondo Yamaguchi

*myamagu1 / mondo.yamaguchi@gmail.com*

*Notes while running (asterisks indicate major observations):*

- Nice styling, plus a little bit of HUD! (+2*b*)
- Bounds checking uses the body, however, and not the drawing area. (3*a*, 3*b*, 4*a*)
- Frame rate seems lower than it has to be. (3*b*, 4*a*)
- Animation updates still affect boxes when fingers are on them. (3*b*, 4*a*)
- Device motion appears to work, although frame rate keeps the motion from feeling natural. (3*b*, 4*a*)
- Flicking does not seem to be implemented; hard to tell with the frame rate. (3*a*, 3*b*, 4*a*)

*Code review (asterisks indicate major observations):*

1. Yay, no tabs :) (+4*c*)
2. Hmmmm, to this point no major changes in the code. That explains why flicking does not work, and partially bounds checking (one way to implement bounds checking is to limit the position while a box is being dragged). (3*a*, 3*b*, 4*a*)
3. OK, simple enough; this is based on my classroom code so changing this fixes the frame rate issue. I'm a little surprised that you didn't speed it up. (3*b*, 4*a*)
4. And, pretty clearly, here is where the bounds checking needs to be fixed. You should check against the drawing area's location and size, not the hardcoded origin nor the window's dimensions. (3*a*, 3*b*, 4*a*)
5. Further, you have an inappropriate hardcode here. You have the box element on-hand—just use its width instead of the hardcoded value! It is the right number anyway. (3*a*, 4*a*, 4*b*)
6. Another optimization: note that both the horizontal and vertical boundary conditions do the same thing: negate the corresponding velocity component. Thus, you can combine the conditions. The result is long, but at least there is no redundancy. (4*b*)
7. This is where the “jittery box” issue comes from—the animation code is *unconditionally repositioning the boxes* even if a finger is on them. Some guard code needs to be written so that physics is applied only when a finger is *off* a box. If a finger is on a box, there shouldn't be any physics because technically the user is “holding” that box at that moment. (3*b*, 4*a*)
8. Commented-out code should be deleted. If you don't want to lose it, commit to version control first, then delete. (4*c*, 4*e*)
9. \*\*\* Magic numbers—what do they mean? What do they intend to do? Hardcoding them like this does not help communicate any of this. Give them names. (4*b*, 4*c*)

2*b* — +

3*a* — / ...The bounds-checking issue affects this, because it signals some missing knowledge about how to access the location and dimensions of both the boxes and the drawing area. The lack of flicking also hits here, because it shows some difficulty with accessing the available information about box position and movement.

3*b* — / ...This one also is affected by flicking, but also the misplaced animation updates and to some degree the unnatural-feeling device motion as well.

**CMSI 370-01**  
**INTERACTION DESIGN**  
Fall 2015

**Assignment 1211 Feedback—Direct Manipulation Application**

*4a* — / ...The missing flick functionality alone is a major gap here. But the other bugs also combine to hold this proficiency down.

*4b* — | ...The magic numbers are definitely a habit to break (or is it better to say that naming your magic numbers is definitely a habit to *form*?).

*4c* — | ...Code is decently presented...the main drags are the relatively large swaths of commented-out code, which disrupts things enough to go down a notch.

*4d* — / ...Missing a major requested functionality indicates some issues with finding/seeking the necessary information to implement it.

*4e* — +

*4f* — +