CMSI 370-01

INTERACTION DESIGN

Fall 2015

Assignment 1211 Feedback—Direct Manipulation Application

Chris Franco *cf7 / spe861@gmail.com*

Notes while running (asterisks indicate major observations):

- Flicking velocity needs calibration. (*3b*, *4a*)
- Acceleration vector is not consistently applied (a "ghost push" sometimes appears to happen). (*4a*)
- Accelerometer or physics engine sometimes affects movement while a finger is on a box—there should be none at all. (*3b*, *4a*)

Code review (asterisks indicate major observations):

1. Gah, a single tab got past you in *boxes-touch.css* **:)** (*4c*)
2. This file is out-of-date and in fact not needed—it should be removed. (*4b*)
3. I'm not sure that this is the best way to implement friction—I'm wary of sign changes near zero. Why not just multiply by a value less than 1? (*3b*, *4a*)
4. Lines marked with note #4 are part of the "sticking" fix. Essentially, at fractional coordinates, the browser mind "round" your offset coordinates to a different value than what you originally set. This skews computations when the velocity is also fractional. The solution is to store the offset *independently* of the browser, so that the web browser cannot overwrite it. This keeps the computations self-consistent and thus fixes the "sticking" behavior. (*3a*, *3b*, *4a*)
5. This is the wrong event! You should be listening for `devicemotion`. We already did this in class, so I'm not sure why you decided to change it. A comment would be appropriate here so that you can explain why the change was made. See *https://developer.mozilla.org/en-US/docs/Web/Events/devicemotion* (*3b*, *4a*, *4d*)
6. This is *way* more work than you need to do, because you are listening for the wrong event. The `devicemotion` event already comes with an `accelerationIncludingGravity` vector that does all this for you. Not sure why you decided to go in this direction. (*3b*, *4a*, *4d*)
7. I don't have enough time right now to investigate the "ghost push," but there is definitely something up with the way the code is updating the velocity. This should be very clean, just update the position, then update the velocity with the acceleration, then update the acceleration whenever `devicemotion` is detected. There's some state management going on (`flicked`, `initial`, `gravity`) plus a lot of additional logic (note 6, `applyFriction`, `flick`) that I think gets in the way. If you like, we can go over this next semester and we can see what's up. (*4a*, *4b*)

2b — **+** …No regressions in design here; bounds are respected and gravity direction is right.

3a — **+** …No issues with interface structure.

3b — **|** …Misdirected device event is the main sticking point here. "Sticking" behavior is something else but that one is inherently tricky so I won't ding you for that.

4a — **|** …Even with the "sticking" behavior fixed, there is a glitch in the physics which comes into play only after you've touched a box. (you can tell because box movement is pretty natural if you just rotate the device before touching anything) This tells me that the flicking code leaves behind some undesirable state, such that after the finger lifts, some code still interferes with correct calculation of velocity and position.

CMSI 370-01

INTERACTION DESIGN

Fall 2015

Assignment 1211 Feedback—Direct Manipulation Application

4b — **|** …Structure is generally right, no glaringly bad code, but there is undue complexity in the gravity code (due to the misdirected event listener) and velocity updates after a finger touches down. Plus there is that extraneous file that should have been removed.

4c — **+** …No major gaffes, and no I won't hit you for that stray tab **:)**

4d — **+** …Overall decent use of course information, and then some (even though it was not totally necessary—that would be the gravity computation algorithm).

4e — **+**

4f — **|** …A few tweaks and cleanup after the due date.