cpsc 543 Lab 5: Make Something Haptically Engaging

Version: 1.0

You now have experience working with the Twiddlerino and Processing. Next: make something ***haptically engaging*** by combining haptic control (i.e. Twiddler) with graphics, audio or both.

Note: do this lab on your own (not with a partner). Fine to use lab mates for feedback!

Learning Goals: By the end of this lab, you will be able to:

- **Define** a custom virtual environment
- **Render** a virtual environment with at least two modalities (haptic and either graphically or with audio).
- **Tweak** a program to be fun, engaging, or compelling – give it that extra something.

Setup:

Lab 5 uses skeleton code to give you a head start. An Arduino sketch controls the Twiddlerino; a Processing sketch provides graphics (and audio). These sketches implement "Pong": your Twiddler acts through a virtual environment on the client Arduino, to interact *bidirectionally* with an output displayed on the host computer via Processing. You will need:

Just as for Lab 4:

- Arduino; Processing; PIDLibrary for Arduino; Twiddlerino library for Arduino.

New to Lab 5:

- The Lab5 skeleton code (Arduino and Processing sketches) from: http://www.cs.ubc.ca/~cs543/current-term/del/lab5-engaging-codeLibraries.zip
- Minimum 2.2.2 for **Processing**. Download from http://code.compartmental.net/tools/minim/ and drag into your Processing libraries folder, or install via Processing's native library installer (Sketch > Import Library… > Add Library…).

To run, upload the Arduino sketch (.ino) to your Twiddler (note that it will be a Duemilanove Arduino). Once it's running, start up the Processing sketch (.pde) to connect to it. You will need to **change the serialPortName variable** in the Processing sketch to match your serial port. When you start, turn your sound on!

The Assignment:

Begin by modifying the supplied source code to familiarize yourself with the architecture; make it more interesting. Then, proceed to personalize it more extensively, or if you prefer, create your own code base. We suggest you follow these steps:

1. **Run the code and play some Pong.** Pay attention to the haptic rendering. Right now, the haptic environment is being updated on the Arduino (1msec updates), while the (slower-moving) game logic and graphics are implemented in Processing, with the two connected bidirectionally through a serial pipe. This is a classic client-server architecture. The Arduino prints the Twiddlerino's position to the serial port; this is read by Processing and moves the player's paddle. Processing occasionally sends a command byte over serial to the Arduino: "i" tells the Arduino code to add an impulse to the haptic display command, because the virtual ball has hit the virtual paddle.
2. **Modify the Arduino source control to accomplish the same output without using the provided library PID controller.** That is, manually set PWMOut = <some equation>, rather than calculating myPID.Compute(). Be careful about assuming a fixed time step! *Provide an excerpt or screenshot of your code on your blog to show how you accomplished this. It should look similar to the PID equations in Lab 4.*
3. **Make an engaging experience!** This does not need to be a game, but must use the Twiddlerino knob. You can use the PID controller. You can base this experience on the Pong source code, connect it to your previous labs, or relate it to your project. However, this experience must **include haptics and either graphics or audio**, and it must be **substantially different from the initial Pong game**: not Pong, and at minimum a different virtual environment at its core.

Parameters

I suggest you write down your problem-solving approach as you work, and get feedback from your classmates or friends when you are fine-tuning your experience.

The usual rules (mostly) apply:

1) Work **individually**.
2) Video/photo document your work and describe on your blog. Answer assignment questions; use screenshots to illustrate your point as appropriate). Post the blog link onto the course twiki: https://www.cs.ubc.ca/wiki/do/view/CS543/Labs
3) "Adequate documentation" for assignment element #3 includes code, as well as explaining and showing result. For this,
   (a) write pseudocode on the blog (if your code is quite simple, use the code itself; and
   (b) provide easy access to complete code with an appropriate common-sense mechanism that does not obfuscate the rest of the blog. E.g., by attaching or linking to another file, or (if brief) posting at bottom of blog.

CPSC 543 Lab 5 Mark Sheet (completed by instructor)

Name: _____

- ☐ (10%) Re-implemented Pong spring model without PID controller.
- ☐ (15%) Came up with an experience that is substantially different from the Pong game provided.
- ☐ (10%) The experience contains both haptics and graphics/audio rendering.
- ☐ (10%) Demonstrated appropriate problem-solving for an engaging experience (rapid iteration, gathering feedback).
- ☐ (10%) Final product is fun or engaging.

- ☐ (30%) Reflection on above items
- ☐ (15%) Documentation adequate (visual and words)

- ☐ Multiplier applied for late hand-ins, as described on course homepage.

OVERALL MARK:

☐	**Great (100%)**	Entirely satisfied and exceptionally well done
☐	**Good (85%)**	Entirely satisfied and well done
☐	**Fair (70%)**	Largely satisfied, few major issues
☐	**Poor (54%)**	Some worthwhile, comprehensible effort, with substantial issues
☐	**Zero (0%)**	Little to no real comprehensible effort; not handed in or otherwise unacceptable